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. 645049-234049/DBM52-M10302010-JDBA
Publication Date: 2010-10-30

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 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 to use interface. You can perform database functions without having to learn SQL. JDBA Tool's monitoring functions also provides statistical data and information on who is using your database.

This manual gives systematic instructions for using the JDBA Tool to manage database objects in DBMaker. JDBA Tool features are first briefly described followed by procedures that guide you through proper use of the feature.

This book is for users who will design and administer DBMaker databases using JDBA Tool. It teaches people unfamiliar with using DBMaker, but who are familiar with the operation of relational databases. We assumed the user has some knowledge of their database platform. Experienced users may also find the information in this manual to be a helpful reference.

To perform JDBA Tool tasks, please refer to specific sections within the chapters. You should read any descriptions included before the procedures and follow each of the steps in the procedures. We 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 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 is included 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 charge).

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

CASEMaker support contact information, by post mail, phone, or email, for your area () is at: www.casemaker.com/support. We recommend searching the most current database of FAQ's before contacting CASEMaker support staff.

Please have the following information available when phoning support for a troubleshooting enquiry or include this information in your correspondence:

- 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. This chapter also describes how to connect and disconnect from a database.

JDBA Tool, like all the other *DBMaker* Java tools (e.g., JServer Manager and Java Configuration Tool), operates in the Java Run Time Environment. To start these programs the Java Development Kit (JDK) or the Java Run Time Environment (JRE) should be installed on the system. DBMaker release 5.1, when installed from disk, automatically installs JRE 1.5 on Windows systems.

2.1 Summary of Features

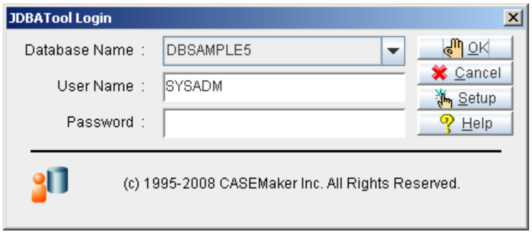
JDBA Tool has a wide variety of features for effectively managing your databases.

- 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 (e.g., data type, default value, and value constraint) so explicit specification is not required.
- After successfully creating a table, it is easy to create an index, primary key, or foreign key in a table. Foreign key establish relationships between different tables in a database.
- DBMaker provides easily querying 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. Please consider the mutual relationships between objects before dropping some objects. For example, drop foreign keys before dropping the referenced primary keys.
- Any authorized user can create database links to access database objects on remote databases.
- Use the Table Replication command to transfer table data between local and remote table objects.
- Frequently used SQL commands are easily managed by creating stored commands that can be reused whenever needed.
- Stored procedures can help make performing frequently repeated tasks even easily. Thus, eliminating the need for repeated SQL compilation and optimization.

- Triggers can handle complex transactions in the database.Triggers can automatically execute when the user inserts, deletes, or updates data in the database.
- A trigger actions can call stored procedures.
- Text indices can be created for columns with CHAR, VARCHAR, LONGVARCHAR, and FILE data types.
- System administrator can add users or groups to a database. After adding users and groups to a database, privileges can be assigned to users and groups. Groups provide an efficient method for assigning identical privileges to many users (i.e., all of the users in the group).
- 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. We recommend configuring all database settings before logging into 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 database's advanced settings.



JDBATool Login

Database Name : DBSAMPLE5

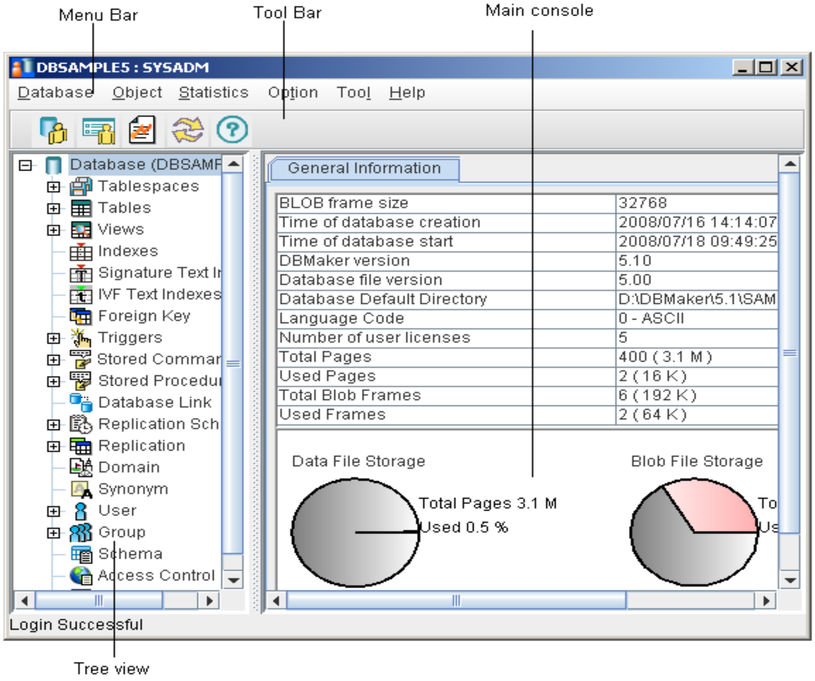
User Name : SYSADM

Password :

(c) 1995-2008 CASEMaker Inc. All Rights Reserved.

Figure 2-1 JDBA Tool Login window

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



BLOB frame size	32768
Time of database creation	2008/07/16 14:14:07
Time of database start	2008/07/18 09:49:25
DBMaker version	5.10
Database file version	5.00
Database Default Directory	D:\DBMaker\5.1\SAM
Language Code	0 - ASCII
Number of user licenses	5
Total Pages	400 (3.1 M)
Used Pages	2 (16 K)
Total Blob Frames	6 (192 K)
Used Frames	2 (64 K)

Figure 2-2 Elements of the JDBA Tool Graphical User Interface

Menu Bar

The menu bar displays JDBA Tool’s pull-down menus. Each menu performs various functions.

The Database in Tree View

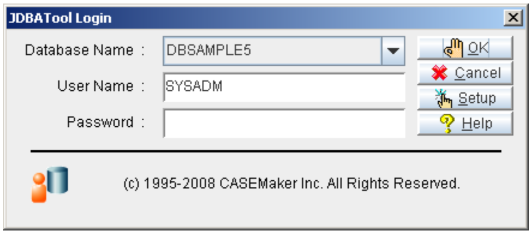
The database tree displays all the objects in the database. The root of the tree is the name of the database, the rest of the objects follow down the tree. The name of each object is also displayed. With the help of the database monitor in the tree, the user can monitor the database and keep track of users and their transactions.

2.3 Connecting to a Database

To manipulate data in a database, you must first connect (i.e., login) to the database. JDBA Tool allows you to connect to a single database at a time. 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. Once started, proceed with the following steps to connect to the database.

➲ Connecting to a database:

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



JDBATool Login

Database Name : DBSAMPLE5

User Name : SYSADM

Password :

(c) 1995-2008 CASEMaker Inc. All Rights Reserved.

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.*

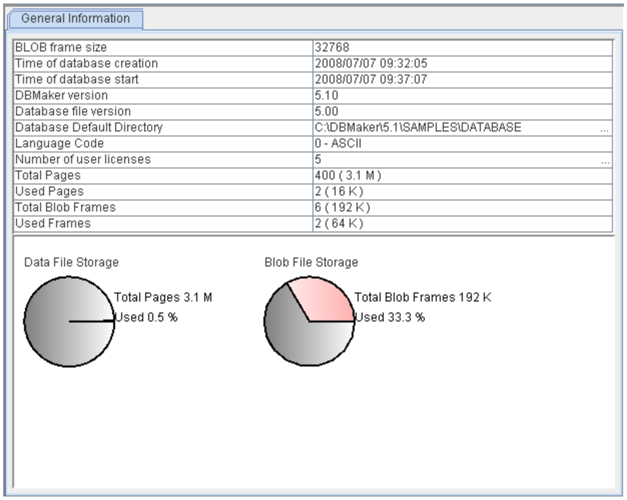
3. If you have been assigned a password, enter it in the **Password** field.
4. Click **Setup** to open the Connect Database Advanced Setting window.
5. Set options in the following categories:

- Connection
- DmServer
- Data Manipulation
- SQL Attributes

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

6. Click **OK** after you are finished setting the options. The **General Information** window appears next and displays the following information:

   - DBMaker version (5.1)
   - File version
   - Date and time that the database was created
   - Date and time when the database was started
   - BLOB frame size
   - Database default directory
   - Language code
   - Number of users who have permission to access the database
   - Number of data pages reserved by the database
   - Number of reserved data pages that are in use
   - Number of BLOB frames reserved by the database
   - Number of BLOB frames in use



BLOB frame size	32768
Time of database creation	2008/07/07 09:32:05
Time of database start	2008/07/07 09:37:07
DBMaker version	5.10
Database file version	5.00
Database Default Directory	C:\DBMaker\5.1\SAMPLES\DATABASE ...
Language Code	0 - ASCII
Number of user licenses	5 ...
Total Pages	400 (3.1 M)
Used Pages	2 (16 K)
Total Blob Frames	6 (192 K)
Used Frames	2 (64 K)

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

7. If you wish to connect to another database, click the **Database** pull-down menu and select **Connect Database**. Repeat steps 1 through 4.

NOTE *Databases must be started by the server program before you can connect to them. Connecting to a database automatically disconnects the currently connected database. In other words, only a single database connection can exist.*

2.4 Disconnecting from a Database

Once you have finished, 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.*

➲ Disconnecting from a database:

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

3 Working with Schema

A schema is essentially a namespace. A schema contains schema objects, also known as named objects. Tables, views, indexes, synonyms, domains, commands, and procedures are all examples of schema objects. Synonyms and domains must have unique names in the database, but other schema object names may duplicate those of other objects existing in other schemas. Schema objects are accessed by qualifying their names. Qualifying a schema object means tagging the object with the schema's name as a prefix.

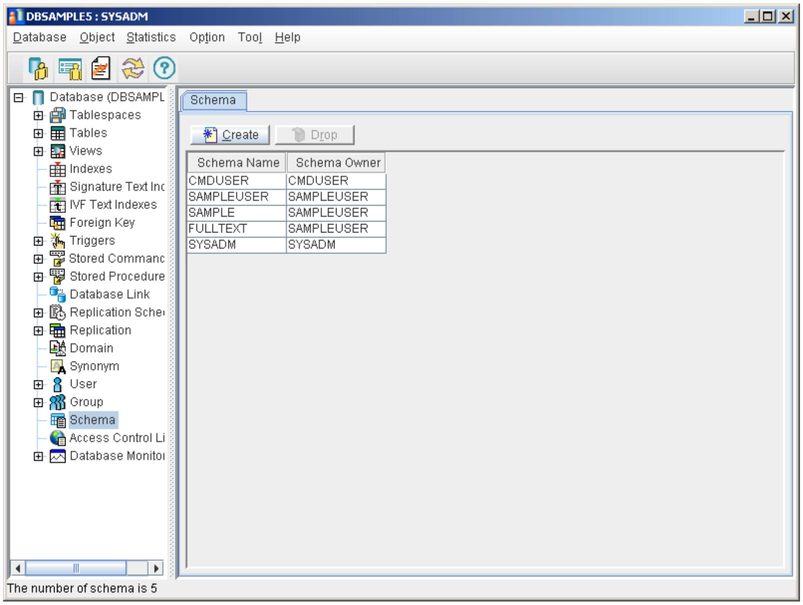
You can easilyadd and remove schema for the current database using the JTool interface. The schema name must be unique within the current database.

3.1 Browsing Schema

You can browse all schema contained in a database by selecting the **Schema** icon from the data structure tree. The schema’s name and owner are displayed in the panel to the right of the data structure tree.

➲ Browsing schema:

1. Click the **Schema** icon from the tree. A list of all schemas in the database is displayed.



Schema Name	Schema Owner
CMDUSER	CMDUSER
SAMPLEUSER	SAMPLEUSER
SAMPLE	SAMPLEUSER
FULLTEXT	SAMPLEUSER
SYSADM	SYSADM

The number of schema is 5

2. Browse the schema listed in the right panel. It may be necessary to use the scroll bar to see all of the information.

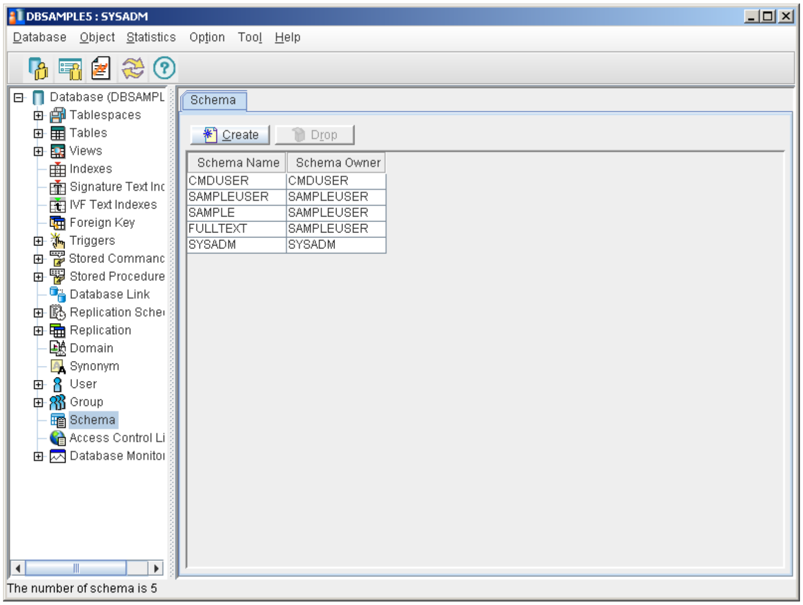
3.2 Creating a Schema

Creating a schema using JDBATool is a very intuitive process. You are guided through the process by a user-friendly user interface. After the schema is created, you can create objects within the schema. The schema owner has full control of the privileges granted on the schema including those created by the owner and others assigned by the database administrator (DBA).

NOTE *Schema must have a unique name within the database.*

➲ Creating a schema:

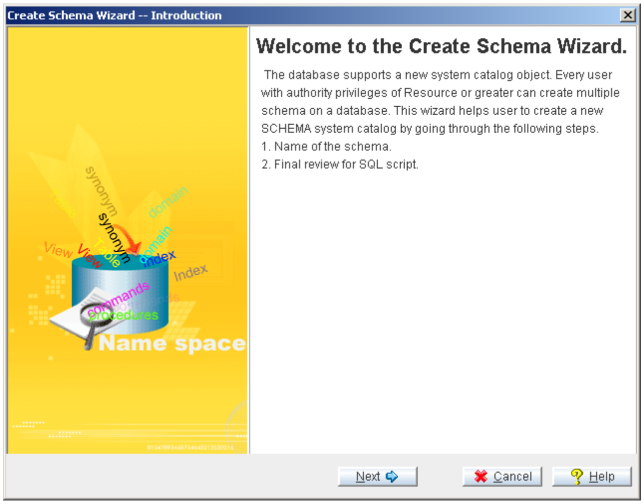
1. Click the **Schema** icon located in the tree. A list of all schemas in the database appears in the panel to the right.



Schema Name	Schema Owner
CMDUSER	CMDUSER
SAMPLEUSER	SAMPLEUSER
SAMPLE	SAMPLEUSER
FULLTEXT	SAMPLEUSER
SYSADM	SYSADM

The number of schema is 5

2. Click the **Create** button in the **Schema** window. The **Create Schema Wizard Introduction** window appears.

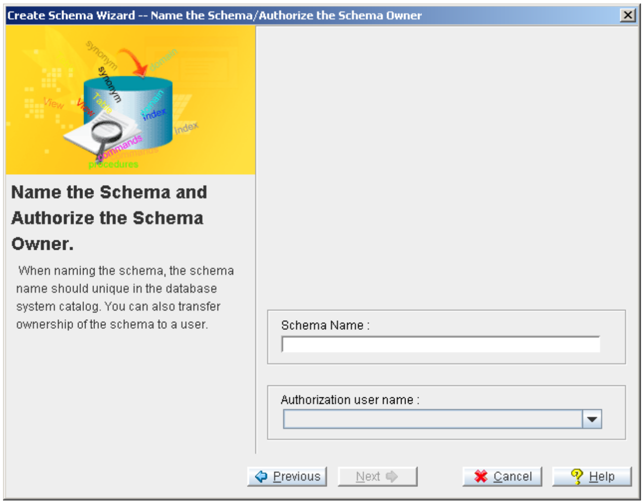


Create Schema Wizard -- Introduction

Welcome to the Create Schema Wizard.

The database supports a new system catalog object. Every user with authority privileges of Resource or greater can create multiple schema on a database. This wizard helps user to create a new SCHEMA system catalog by going through the following steps.
1. Name of the schema.
2. Final review for SQL script.

3. Click the **Next** button in the **Create Schema Wizard Introduction** window. The **Create Schema Wizard Name Schema/Authorize the Schema** window appears.



Create Schema Wizard -- Name the Schema/Authorize the Schema Owner

Name the Schema and Authorize the Schema Owner.

When naming the schema, the schema name should unique in the database system catalog. You can also transfer ownership of the schema to a user.

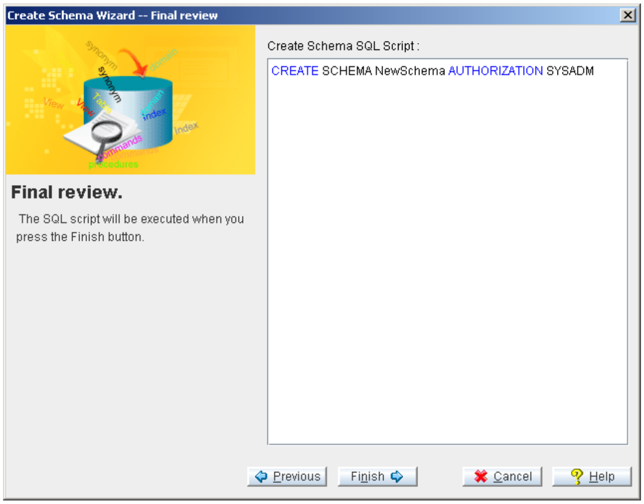
Schema Name :

Authorization user name :

4. In the **Schema name** field, enter a name for the schema. Schema names may not contain spaces and every schema name must be unique within the current database.

5. Select the schema owner from the **Authorization user name** drop down list. If no name is selected from the list, the **Create Schema Wizard** assigns ownership to the schema creator.

6. Click the **Next** button. The **Create Schema Wizard Final Review** window appears. The SQL code in the **Create Schema SQL Script** field can be edited.



Create Schema Wizard -- Final review

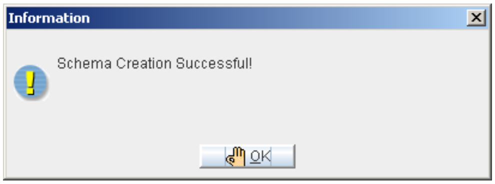
Final review.

The SQL script will be executed when you press the Finish button.

Create Schema SQL Script :

```
CREATE SCHEMA NewSchema AUTHORIZATION SYSADM
```

7. Click the **Finish** button when you are satisfied with the SQL code that will be used for creating the schema. A message appears to confirm a successful schema creation. The new schema now appears in the **Schema** window to the right.



Information

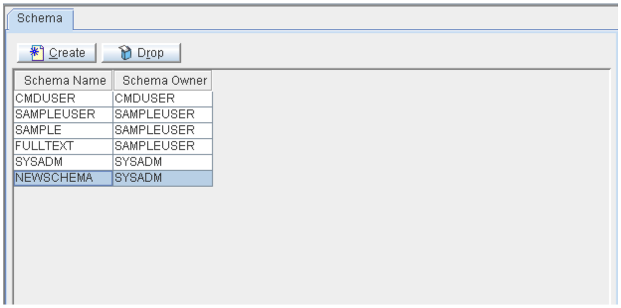
Schema Creation Successful!

3.3 Dropping a Schema

When a schema is no longer required, it can be removed.

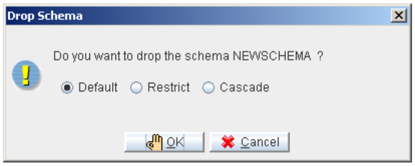
➲ Removing (i.e., dropping) a schema:

1. Click the **Schema** icon from the tree. A list of all schemas in the database is displayed in the panel to the right.



Schema Name	Schema Owner
CMDUSER	CMDUSER
SAMPLEUSER	SAMPLEUSER
SAMPLE	SAMPLEUSER
FULLTEXT	SAMPLEUSER
SYSADM	SYSADM
NEWSCHEMA	SYSADM

2. Click the schema you want to drop then click the **Drop** button in the **Schema** window. The **Drop Schema** confirmation dialog box appears.



Do you want to drop the schema NEWSCHEMA ?

● Default ○ Restrict ○ Cascade

3. There are three options for dropping schema. The **Default** option deletes the shema without considering dependent object within the schema. Dependent objects become invalid after the schema is dropped. The **Restrict** option considers table, domain, and procedure dependencies. This option does not permit a schema drop when one or more objects depend on the schema. The **Cascade** option drops the schema and all of its dependent objects. Click a radio button to select a schema dropping option.

4. Click the **OK** button. The schema is removed from the database or a warning dialog box appears if the schema cannot be dropped.

4 Working with Tablespaces

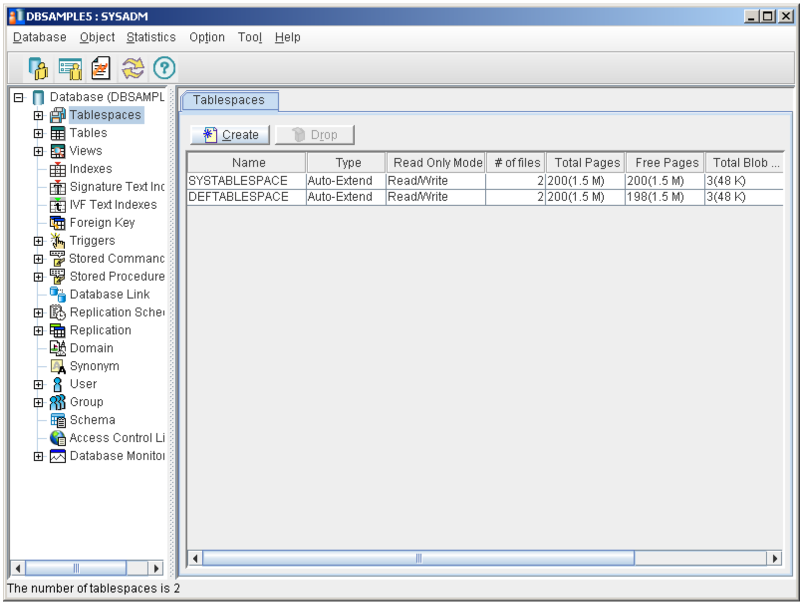
DBMaker databases are composed of one or more logical divisions known as *tablespaces*. Tablespaces help divide the database into manageable areas. In the logical view, a tablespace contains one or more tables and indexes. In the physical view, a tablespace is the physical storage (i.e., directory path) 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.

4.1 Browsing Tablespaces

All DBMaker databases have at least two tablespaces. The first is 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) stores 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.

➲ Browsing a tablespace:

1. Click the **Tablespace** object in the tree to display a list of all tablespaces.



Name	Type	Read Only Mode	# of files	Total Pages	Free Pages	Total Blob ...
SYSTABLESPACE	Auto-Extend	Read/Write	2	200(1.5 M)	200(1.5 M)	3(48 K)
DEFTABLESPACE	Auto-Extend	Read/Write	2	200(1.5 M)	198(1.5 M)	3(48 K)

The number of tablespaces is 2

2. Browse the tablespaces listed in the right panel using the scroll bar. All of the tablespace information appears in following fields.

- **Name** – tablespace name
- **Type** – tablespaces can be regular or autoextend
- **Read Only Mode** – specifies whether a tablespace is Read-Only or Read/Write. Read-only tablespaces do not allow adding new objects or editing existing ones.
- **# of files** – total number of data files and BLOB files in the tablespace
- **Total Pages** – number of pages in data files
- **Free Pages** – number of free pages in data files
- **Total Frames** – number of frames in BLOB files
- **Free Frames** – number of free frames in BLOB files
- **Creation Time** – tablespace creation time

4.2 Creating a Tablespace

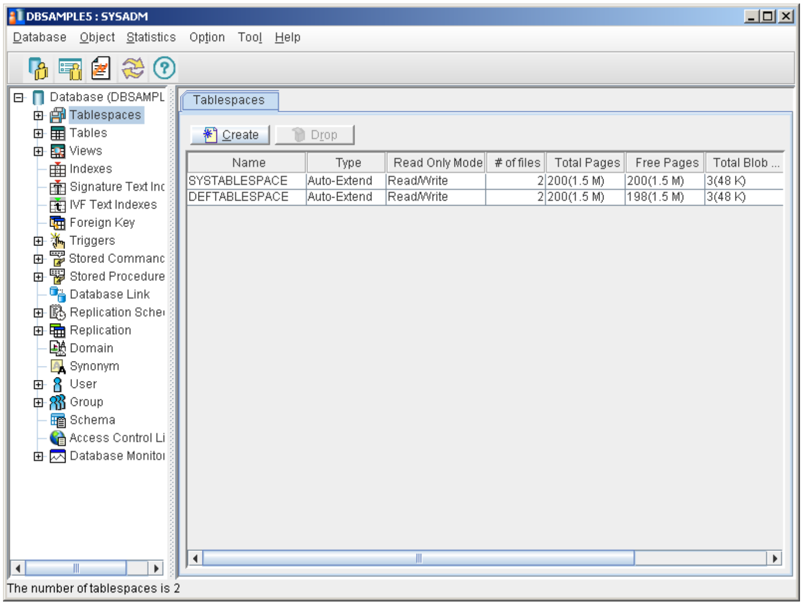
Smaller tables that contain data of a similar nature can be grouped in a single tablespace. Alternatively, very large tables should be placed in their own tablespaces. A regular tablespace has a fixed size and contains one or more data files. If the physical file of a regular tablespace becomes insuffcient, you must enlarge it manually. An autoextend tablespace, given this same situation, automatically grows as needed.

Consider the following when creating a tablespace:

- The logical name must be unique
- Files all have a file type of either DATA or BLOB
- The size of DATA files is measured and reported in pages. The size of BLOB files is measured and reported in frames.
- Tablespaces must contain at least one BLOB file and at least one DATA file. Tablespace creation fails when at least one DATA file is not included.

➲ Creating a tablespace:

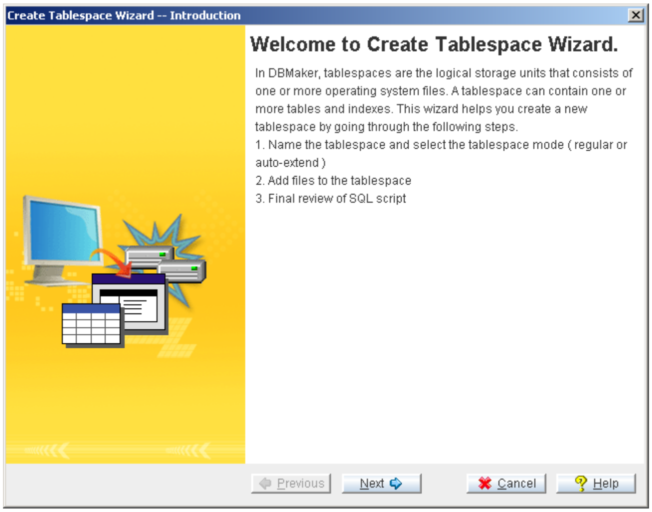
1. Click the **Tablespaces** object in the tree to display the **Tablespaces** page.



Name	Type	Read Only Mode	# of files	Total Pages	Free Pages	Total Blob ...
SYSTABLESPACE	Auto-Extend	Read/Write	2	200(1.5 M)	200(1.5 M)	3(48 K)
DEFTABLESPACE	Auto-Extend	Read/Write	2	200(1.5 M)	198(1.5 M)	3(48 K)

The number of tablespaces is 2

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

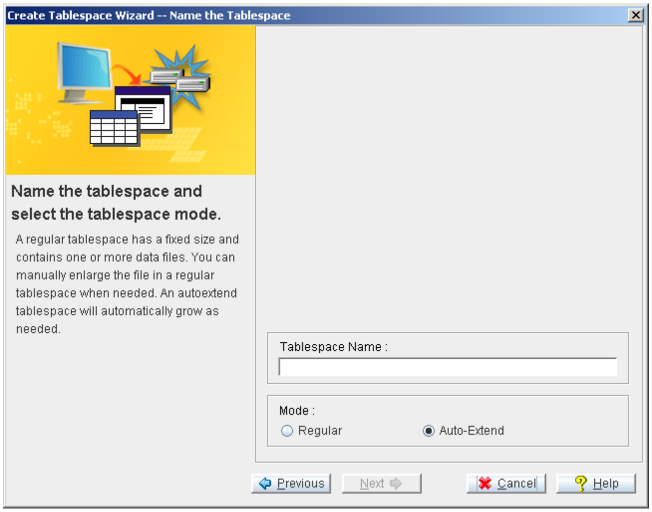


Welcome to Create Tablespace Wizard.

In DBMaker, tablespaces are the logical storage units that consists of one or more operating system files. A tablespace can contain one or more tables and indexes. This wizard helps you create a new tablespace by going through the following steps.

1. Name the tablespace and select the tablespace mode (regular or auto-extend)
2. Add files to the tablespace
3. Final review of SQL script

3. Click the **Next** button. The **Name the Tablespace** window appears.



Create Tablespace Wizard -- Name the Tablespace

Name the tablespace and select the tablespace mode.

A regular tablespace has a fixed size and contains one or more data files. You can manually enlarge the file in a regular tablespace when needed. An autoextend tablespace will automatically grow as needed.

Tablespace Name :

Mode :

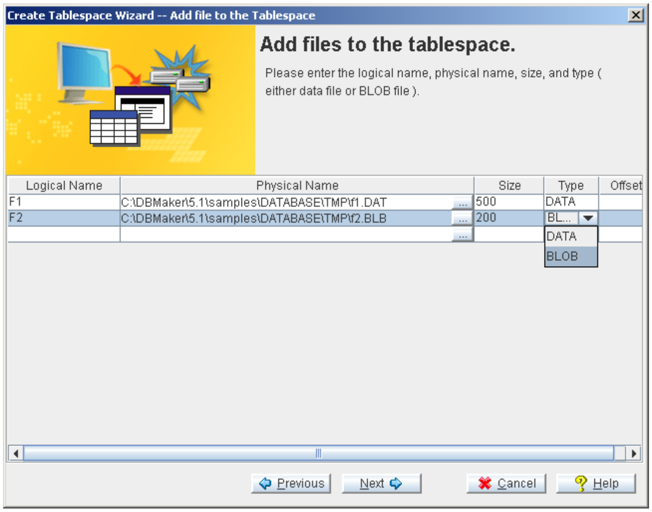
○ Regular ● Auto-Extend

4. Enter the name for 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 appears.



Create Tablespace Wizard -- Add file to the Tablespace

Add files to the tablespace.

Please enter the logical name, physical name, size, and type (either data file or BLOB file).

Logical Name	Physical Name	Size	Type	Offset
F1	C:\DBMaker\5.1\samples\DATABASE\TMP\f1.DAT	500	DATA	
F2	C:\DBMaker\5.1\samples\DATABASE\TMP\f2.BLB	200	BL...	

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

8. Enter a physical name for 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.*

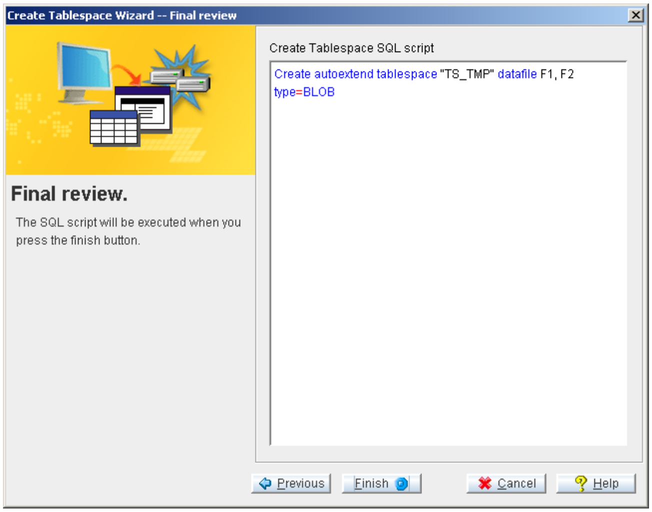
9. Enter a size for the tablespace in the **Size** column.

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

11. Set a file offset for raw devices in the **Offset** column.

NOTE *This option is only for databases running in a UNIX environment.*

12. Click the **Next** button to proceed. Next, the SQL command for creating the new tablespace appears. Users can edit this command.



Create Tablespace Wizard -- Final review

Final review.

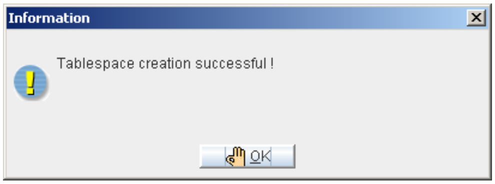
The SQL script will be executed when you press the finish button.

Create Tablespace SQL script

```
Create autoextend tablespace "TS_TMP" datafile F1, F2
type=BLOB
```

13. Edit the command by entering or deleting command syntax in the command line area.

14. Click the **Finish** button after you have finished making an optional edits to the SQL command. The **Information** window of the **Create Tablespace Wizard** appears.



Information

Tablespace creation successful !

4.3 Altering a Tablespace

It may be necessary to alter a tablespace after its creation. Tablespaces can be alterted in the following ways:

- Add DATA or BLOB files
- Compression
- Add pages to files (in the tablespace)
- Change from autoextend to regular or vice versa
- Change from read only to read/write or vice versa

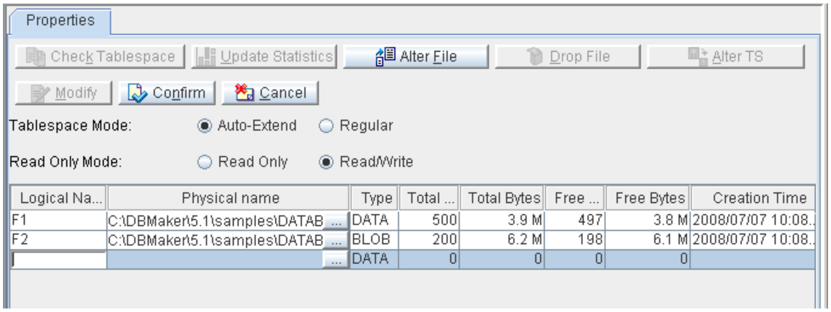
Adding Files to a Tablespace

Once you have created a tablespace, you can later add DATA or BLOB files to the tablespace. Please consider the following when adding files to an existing tablespace:

- The logical name must be unique name
- There are two types of files available: DATA or BLOB
- DATA file sizes are measured and reported in pages. BLOB file sizes are measured and reported in frames.

➲ Adding files to a tablespace:

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



Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08..
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08..
	...	DATA	0	0	0	0	

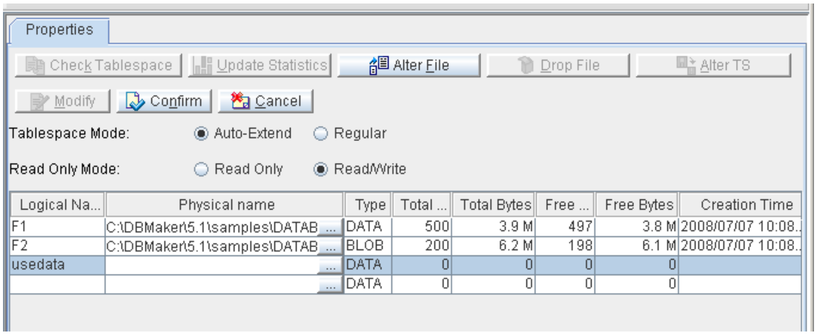
3. To change the **Tablespace mode** from Auto-Extend to Regular, select the **Regular** option button above the list.

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

5. To change the **Read Only Mode** from Read Only to Read/Write, select the **Read/Write** option button above the list.

6. To change the **Read Only Mode** from Read/Write to Read-Only, select the **Read Only** option button above the list.

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

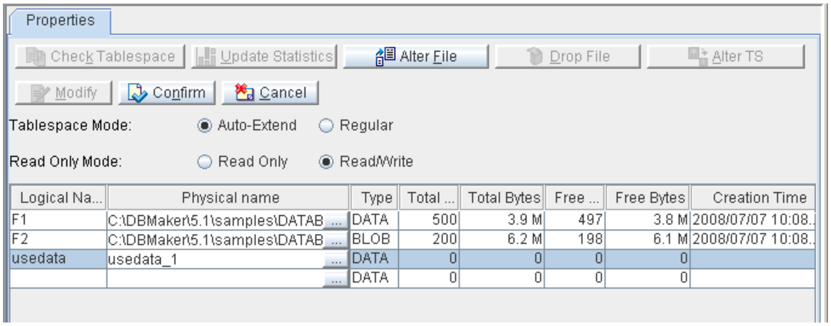


Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08...
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08...
usedata	...	DATA	0	0	0	0	
	...	DATA	0	0	0	0	

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

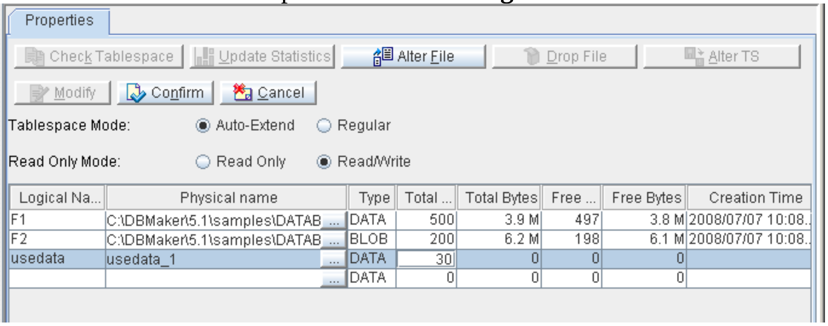


Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08...
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08...
usedata	usedata_1 ...	DATA	0	0	0	0	
	...	DATA	0	0	0	0	

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

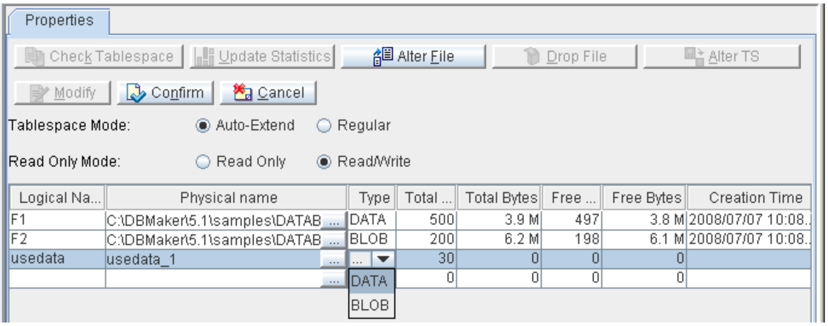


Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08...
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08...
usedata	usedata_1 ...	DATA	30	0	0	0	
	...	DATA	0	0	0	0	

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



Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08...
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08...
usedata	usedata_1 ...	...	30	0	0	0	
	...	DATA BLOB	0	0	0	0	

11. Click the **Confirm** button. The list of files in the tablespace now includes the newly added file.

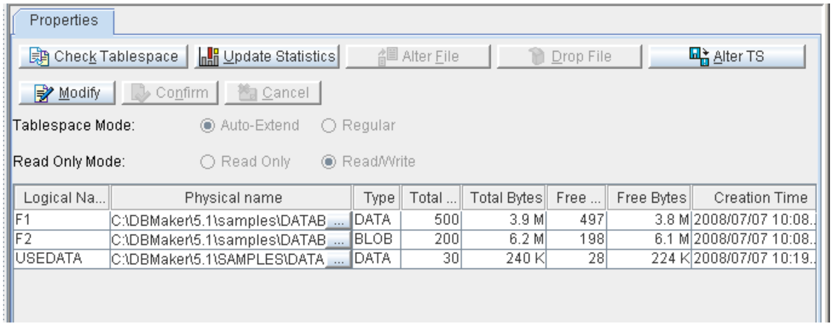
Shrinking and Compressing a Tablespace

Two ways to optimize storage space are shrinking and compressing tablespaces. Compressing the tablespace moves the DATA pages and BLOB frames to the beginning of the file. When shrinking, the tablespace reduces by the number of free pages remaining in each of the files in the tablespace.

The desired number of tailing free pages contained in the tablespace should be less than the number of free pages being replaced. Replacing the number of tailing free pages in a tablespace altered the total number of free pages in each of the files in the tablespace.

➲ Shrinking and/or compressing a tablespace:

1. Select the tablespace that you want to compress or shrink. Next, the **Properties** page appears.

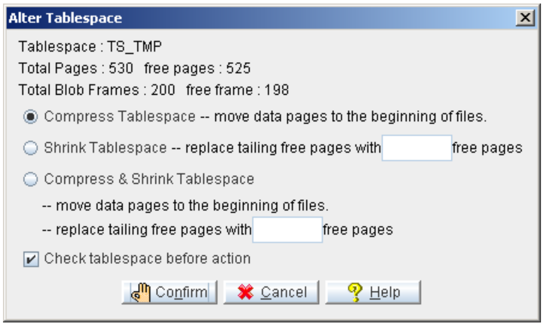


Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08..
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08..
USEDATA	C:\DBMaker\5.1\SAMPLES\DATA ...	DATA	30	240 K	28	224 K	2008/07/07 10:19..

2. Click the **Alter TS** button located toward the top of the window. The **Alter Tablespace** dialog box appears.



Tablespace : TS_TMP
Total Pages : 530 free pages : 525
Total Blob Frames : 200 free frame : 198

● Compress Tablespace -- move data pages to the beginning of files.

○ Shrink Tablespace -- replace tailing free pages with free pages

○ Compress & Shrink Tablespace
-- move data pages to the beginning of files.
-- replace tailing free pages with free pages

☑ Check tablespace before action

3. To compress the tablespace, click the **Compress Tablespace** radio button.

4. To shrink the tablespace, click the **Shrink Tablespace** radio button. This option reduces the tailing or remaining free pages in tablespace files.

5. If you selected Shrink Tablespace in step 4, then it is necessary to enter the desired number of tailing free pages into the **free pages** text field thereby reducing the tailing or remaining free pages in the files of the tablespace.

6. To compress **and** shrink the tablespace, click the **Compress and Shrink Tablespace** radio button. Next, enter the number of tailing free pages you want the tablespace to contain in the **free pages** text field.

7. Click the **Check tablespace before action** checkbox to execute a consistency check of the database. Do this before applying changes to the tablespace.

8. Click the **Confirm** button. The **Alter Tablespace** confirmation dialog box appears to confirm the successful modifications.

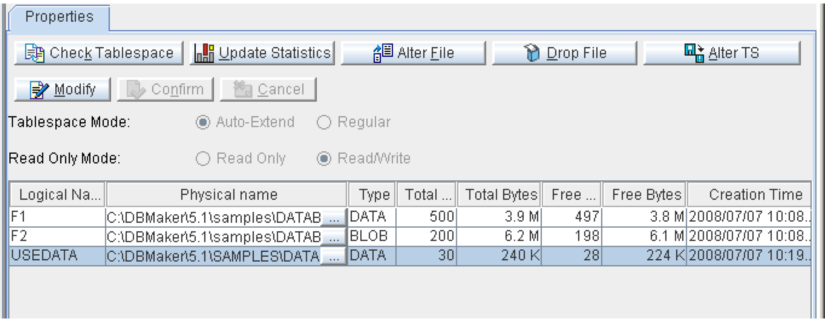
9. Click the **OK** button.

Modifying a File in a Tablespace

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

➲ Modifying 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 appear in the main window.

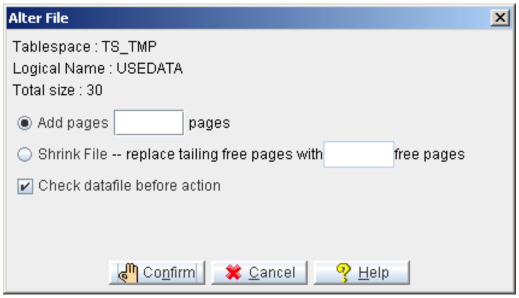


Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08...
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08...
USEDATA	C:\DBMaker\5.1\SAMPLES\DATA ...	DATA	30	240 K	28	224 K	2008/07/07 10:19...

2. Click the **Alter File** button. The **Alter File** dialog box appears.



Tablespace : TS_TMP
Logical Name : USEDATA
Total size : 30

● Add pages [ ] pages

○ Shrink File -- replace tailing free pages with [ ] free pages

☑ Check datafile before action

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

3. Enter into the **pages** text field 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** radio button and enter the desired number of tailing free pages into the **free pages** text field.

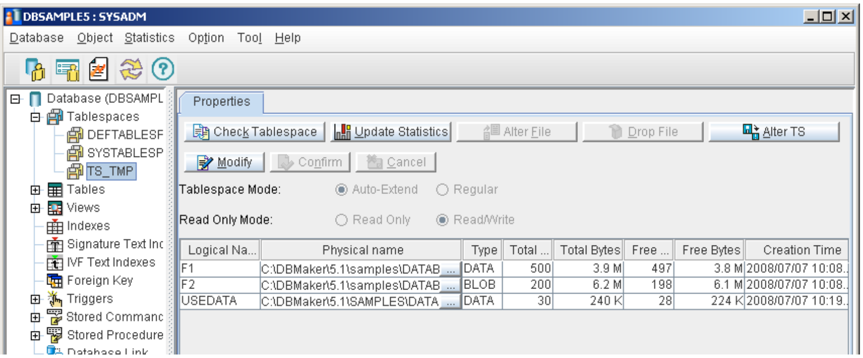
5. Click the **Check datafile before action** check box if you want to check the consistency of the data file before altering the file.
6. Click the **Confirm** button. A confirmation message appears verifying the file modification.
7. Click the **OK** button to dismiss the message dialog window.

Changing Tablespace Modes

DBMaker supports changing the mode of a tablespace from autoextend to regular or from regular to autoextend. 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.

➲ Changing the tablespace's mode:

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

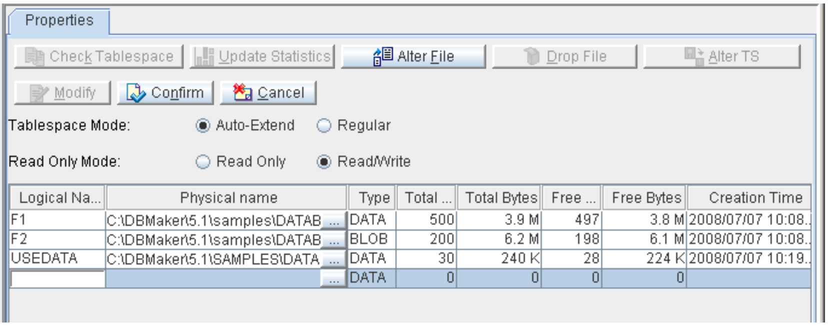


Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08..
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08..
USEDATA	C:\DBMaker\5.1\SAMPLES\DATA ...	DATA	30	240 K	28	224 K	2008/07/07 10:19..

2. Click the **Modify** button.



Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08.
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08.
USEDATA	C:\DBMaker\5.1\SAMPLES\DATA ...	DATA	30	240 K	28	224 K	2008/07/07 10:19.
	...	DATA	0	0	0	0	

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

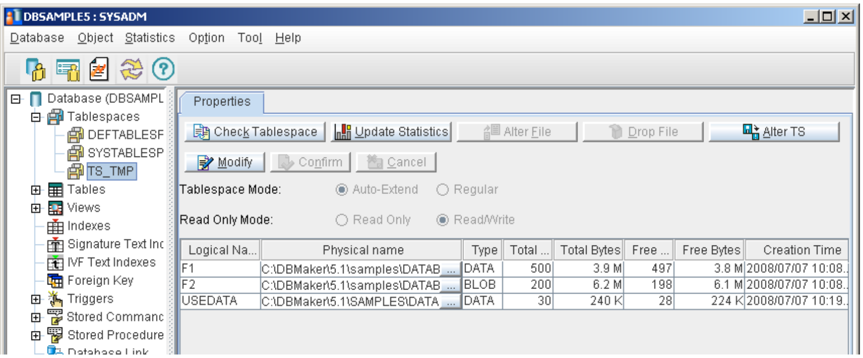
4. Click the **Confirm** button.

Changing the Read Only Mode

Change the **Read Only Mode** from Read Only to Read/Write or from Read/Write to Read-Only is supported in DBMaker. The Read Only tablespace does not allow any updates or creations of new objects in the tablespace. The **SYSTABLESPACE** cannot be set to **Read Only** mode.

➲ Changing the Read Only mode:

1. Select the tablespace to which you want change. The properties for the tablespace appears in the main console.

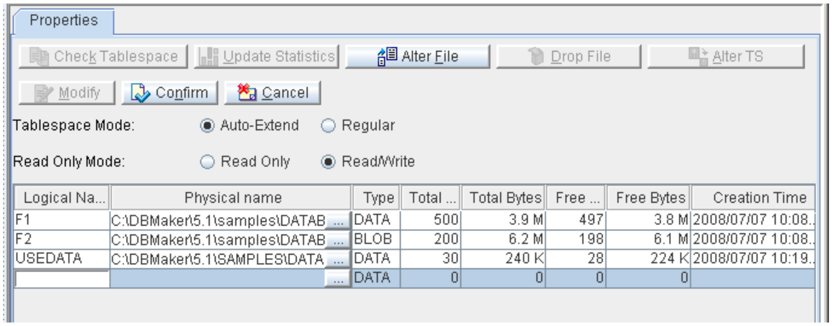


Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08.
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08.
USEDATA	C:\DBMaker\5.1\SAMPLES\DATA ...	DATA	30	240 K	28	224 K	2008/07/07 10:19.

2. Click the **Modify** button.



Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08..
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08..
USEDATA	C:\DBMaker\5.1\SAMPLES\DATA ...	DATA	30	240 K	28	224 K	2008/07/07 10:19..
	...	DATA	0	0	0	0	

3. Select the desired Read Only Mode: **Read Only** or **Read/Write**

4. Click the **Confirm** button**.**

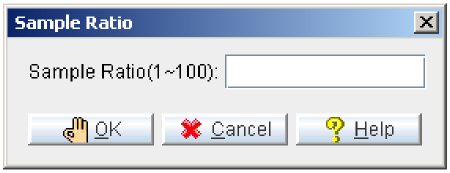
Updating Statistics in a Tablespace

Updating a tablespace's statistics is a significant part of an optimize strategy for tablespace performance.The Sample Ratio value specifies the percentage of data in the object that is 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 the time needed for updating the tablespace's statistics.

➲ Updating 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 appears.

2. Click the **Update Statistics** button. The **Sample Ratio** dialog box appears.



Sample Ratio(1~100):

3. Enter a sample ratio in the range of 1 to 100.

4. Click the **OK** button. The statistics of the tablespace are updated.

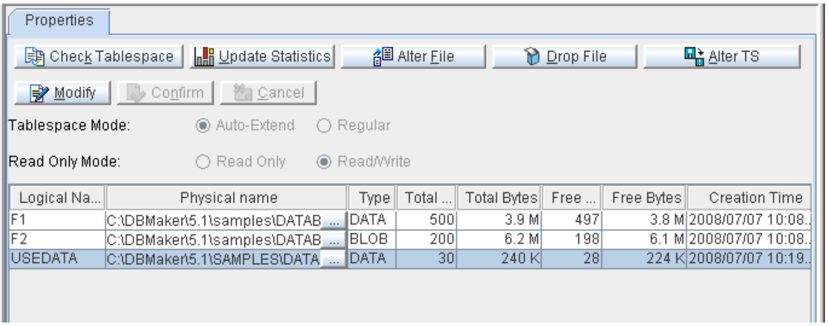
4.4 Dropping a Datafile from a Tablespace

You can drop a datafile file from a tablespace when it is no longer required. To drop a datafile successfully from a tablespace the datafile must be empty and cannot be the single remaining dataspace in the tablespace. The datafiles in the system and default tablespaces cannot be dropped.

➲ Dropping a datafile from a tablespace:

1. Select the tablespace you want to modify from the tree or the tablespaces page. The **Properties** page appears.
2. Select the datafile that you want to remove from the tablespace.

NOTE *You cannot remove a datafile from a tablespace if it is the only datafile remaining in the tablespace or if it is not empty.*



Tablespace Mode: ● Auto-Extend ○ Regular

Read Only Mode: ○ Read Only ● Read/Write

Logical Na...	Physical name	Type	Total ...	Total Bytes	Free ...	Free Bytes	Creation Time
F1	C:\DBMaker\5.1\samples\DATAB ...	DATA	500	3.9 M	497	3.8 M	2008/07/07 10:08..
F2	C:\DBMaker\5.1\samples\DATAB ...	BLOB	200	6.2 M	198	6.1 M	2008/07/07 10:08..
USEDATA	C:\DBMaker\5.1\SAMPLES\DATA ...	DATA	30	240 K	28	224 K	2008/07/07 10:19..

3. Click the **Drop File** button.

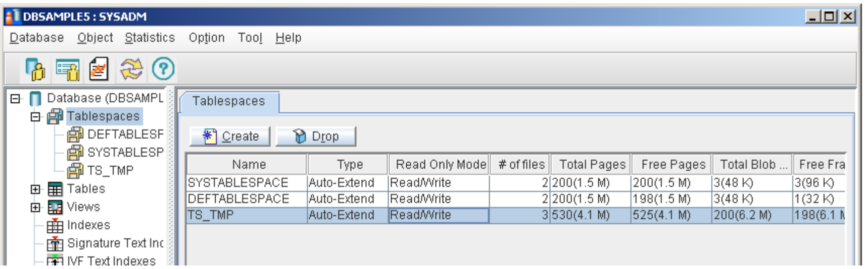
NOTE *You cannot drop the system or default tablespaces.*

4.5 Dropping a Tablespace

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

➲ Dropping a tablespace:

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

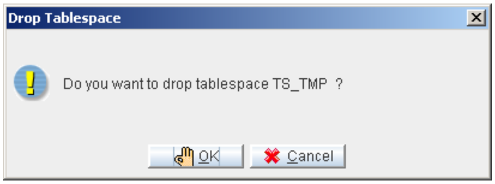


Name	Type	Read Only Mode	# of files	Total Pages	Free Pages	Total Blob ...	Free Fra
SYSTABLESPACE	Auto-Extend	Read/Write	2	200(1.5 M)	200(1.5 M)	3(48 K)	3(96 K)
DEFTABLESPACE	Auto-Extend	Read/Write	2	200(1.5 M)	198(1.5 M)	3(48 K)	1(32 K)
TS_TMP	Auto-Extend	Read/Write	3	530(4.1 M)	525(4.1 M)	200(6.2 M)	198(6.1 M

2. Select the tablespace you want to delete from the tree.

3. Click the **Drop** button at the top of the **Tablespace** page. Alternatively, right click the tree node of the tablespace and select **Drop Tablespace** from the popup menu. A window appears to confirm the drop of the selected table.

4. The Drop Tablespace confirmation dialog box appears.



Do you want to drop tablespace TS_TMP ?

5. Click the **OK** button to drop the tablespace.

5 Working with Tables

Tables are the logical unit of storage used by DBMaker to store data. A table consists of several columns and rows. Columns are also refered to as fields or attributes. Rows are also refered to as records or tuples.

DBMaker identifies each table 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 *Select **Preferences** from the **Option** menu to view the system tables. Click the **Show System Tables** radio button on the **Table** page in the **Preferences** window. Please refer to Chapter 21 for more information.*

5.1 Creating a Table

Tables provide the essential framework for designing databases. 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:

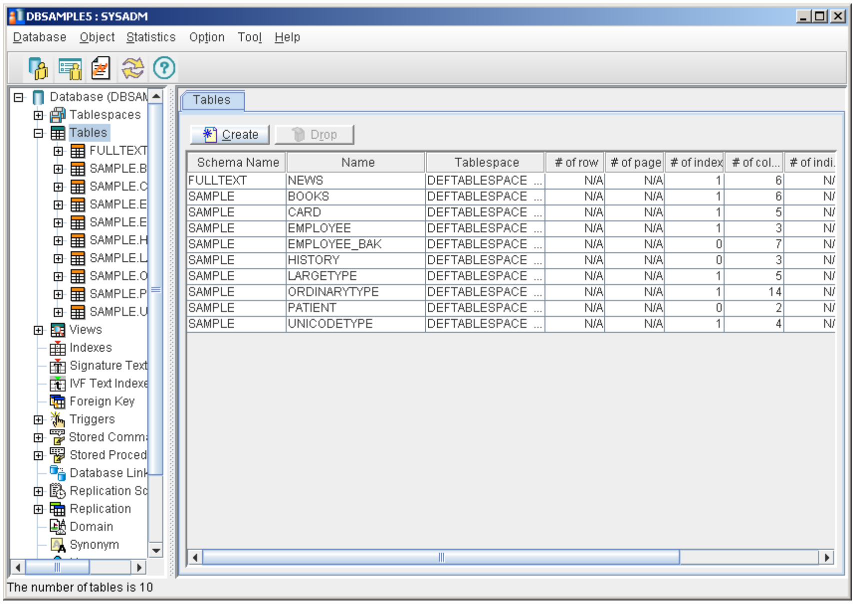
- Set table, tablespace, and schema options
- Set column options
- Set column and table constraints
- Create a unique constraint
- Create foreign keys
- Set advanced table options

Starting the Create Table Wizard

The **Create Table Wizard** provides an intuitive interface for creating tables. All table creation options are available within the wizard. 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.

➲ Using the Create Table Wizard:

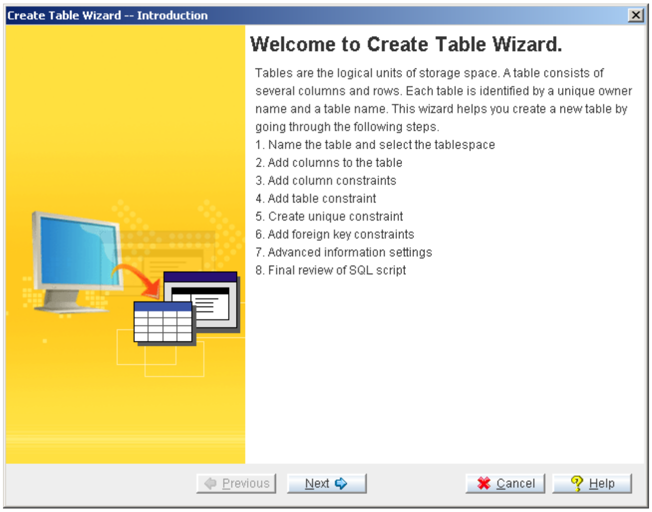
1. Select the **Table** object from the tree. The **Table** page appears.



Schema Name	Name	Tablespace	# of row	# of page	# of index	# of col...	# of indi.
FULLTEXT	NEWS	DEFTABLESPACE ...	N/A	N/A	1	6	N/
SAMPLE	BOOKS	DEFTABLESPACE ...	N/A	N/A	1	6	N/
SAMPLE	CARD	DEFTABLESPACE ...	N/A	N/A	1	5	N/
SAMPLE	EMPLOYEE	DEFTABLESPACE ...	N/A	N/A	1	3	N/
SAMPLE	EMPLOYEE_BAK	DEFTABLESPACE ...	N/A	N/A	0	7	N/
SAMPLE	HISTORY	DEFTABLESPACE ...	N/A	N/A	0	3	N/
SAMPLE	LARGETYPE	DEFTABLESPACE ...	N/A	N/A	1	5	N/
SAMPLE	ORDINARYTYPE	DEFTABLESPACE ...	N/A	N/A	1	14	N/
SAMPLE	PATIENT	DEFTABLESPACE ...	N/A	N/A	0	2	N/
SAMPLE	UNICODETYPE	DEFTABLESPACE ...	N/A	N/A	1	4	N/

The number of tables is 10

2. Click the **Create** button. The **Welcome to Create Table Wizard** window appears.



Create Table Wizard -- Introduction

Welcome to Create Table Wizard.

Tables are the logical units of storage space. A table consists of several columns and rows. Each table is identified by a unique owner name and a table name. This wizard helps you create a new table by going through the following steps.

1. Name the table and select the tablespace
2. Add columns to the table
3. Add column constraints
4. Add table constraint
5. Create unique constraint
6. Add foreign key constraints
7. Advanced information settings
8. Final review of SQL script

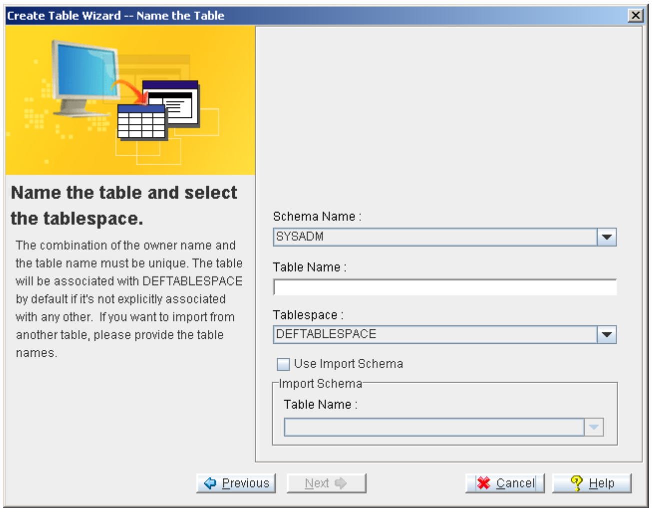
3. Click the **Next** button. The **Name the Table** window appears.

Setting Table, Tablespace and Schema Options

Once you have started the **Create Table Wizard**, you can set table, tablespace, and schema options. Users may import table schema from another table. Imported table schema, all columns (name, length, data type) initially have the same structure as found in the the original table. The table schema may be modified after being imported.

➲ Setting table, tablespace, and schema options:

1. Follow the steps in the **Create Table Wizard** until the **Name the Table** window appears.



Create Table Wizard -- Name the Table

Name the table and select the tablespace.

The combination of the owner name and the table name must be unique. The table will be associated with DEFTABLESPACE by default if it's not explicitly associated with any other. If you want to import from another table, please provide the table names.

Schema Name : SYSADM

Table Name :

Tablespace : DEFTABLESPACE

☐ Use Import Schema

Import Schema

Table Name :

2. Select the **Schema Name** from the drop down list. Your user login name appears by default.
3. Enter a table name in the **Table Name** field.

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

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

NOTE *By default, tables are created in the default user tablespace. The default tablespace appears in the* ***Tablespace*** *menu as* ***DEFTABLESPACE****).*

5. Importing 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 the **Next** button. The **Add columns to the table** window appears.

Setting Column Options

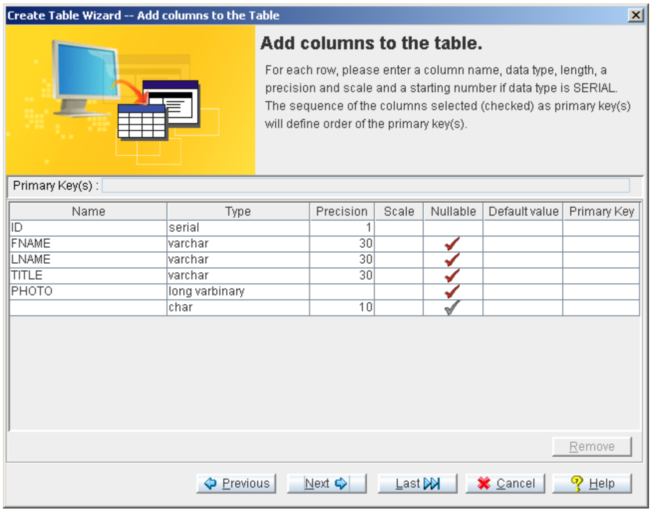
A variety of column options can be set to meet your data requirements. The settings follow:

- Name: specifies a column name
- Type: specifies the following data types: BIGINT, BIGSERIAL, 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, please refer to the *SQL Command and Function Reference*.
- Precision: specifies the length of the columns. This is necessary for the following data types that have variable length: VARCHAR, NVARCHAR, CHAR, NCHAR, DECIMAL, SERIAL, BIGSERIAL and BINARY.
- Scale: specifies the scale of the data type. This is required for the DECIMAL data type.
- Nullable: restricts or permits null values on columns. By specifying NOT NULL, null values cannot be added to the column. The default value is ***NULL***. NOT NULL is usually specified for primary key columns.
- Primary Key: specifies a column as a primary key in a table
- Default Value: specifies a default value for the column data

➲ Setting column options:

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

2. Enter column information under the appropriate headings.



Create Table Wizard -- Add columns to the Table

Add columns to the table.

For each row, please enter a column name, data type, length, a precision and scale and a starting number if data type is SERIAL. The sequence of the columns selected (checked) as primary key(s) will define order of the primary key(s).

Primary Key(s) :

Name	Type	Precision	Scale	Nullable	Default value	Primary Key
ID	serial	1				
FNAME	varchar	30		✓		
LNAME	varchar	30		✓		
TITLE	varchar	30		✓		
PHOTO	long varbinary			✓		
	char	10		✓		

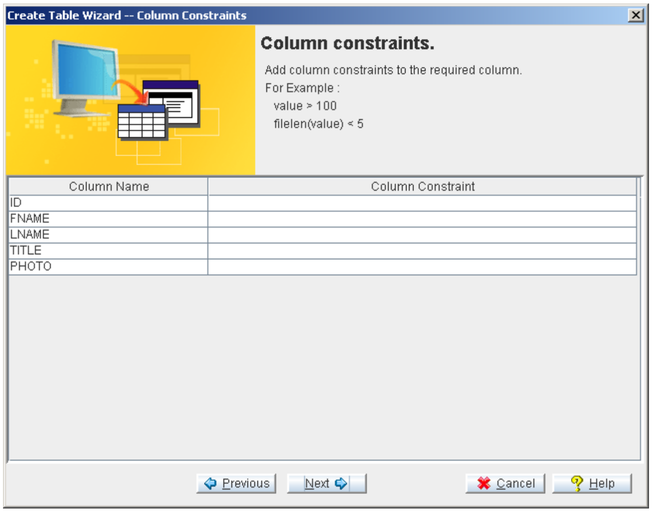
3. Click the **Next** button. The **Column constraints** window appears. If you do not need to add constraints, a foreign key, or set advanced options for the table you may click the **Last** button. The **Final Review** window appears.

Setting Column and Table Constraints

You can further customize your data requirements by setting column and table constraints. Column and table constraints must satisfy SQL syntax rules. Please see *Constraint Syntax* at the end of this chapter for additional information. Table constraints cannot be altered after the table has been created.

➲ Setting column and table constraints:

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



Create Table Wizard -- Column Constraints

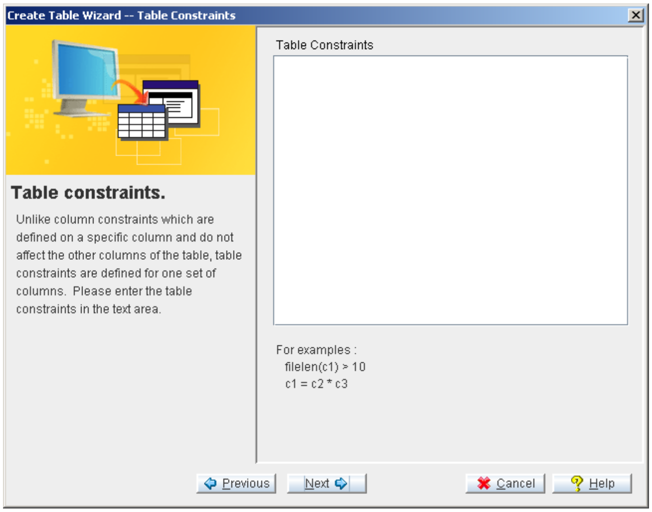
Column constraints.

Add column constraints to the required column.
For Example :
value > 100
filelen(value) < 5

Column Name	Column Constraint
ID	
FNAME	
LNAME	
TITLE	
PHOTO	

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

3. Click the **Next** button. The **Table Constraint** window appears.



Create Table Wizard -- Table Constraints

Table constraints.

Unlike column constraints which are defined on a specific column and do not affect the other columns of the table, table constraints are defined for one set of columns. Please enter the table constraints in the text area.

Table Constraints

For examples :
filelen(c1) > 10
c1 = c2 * c3

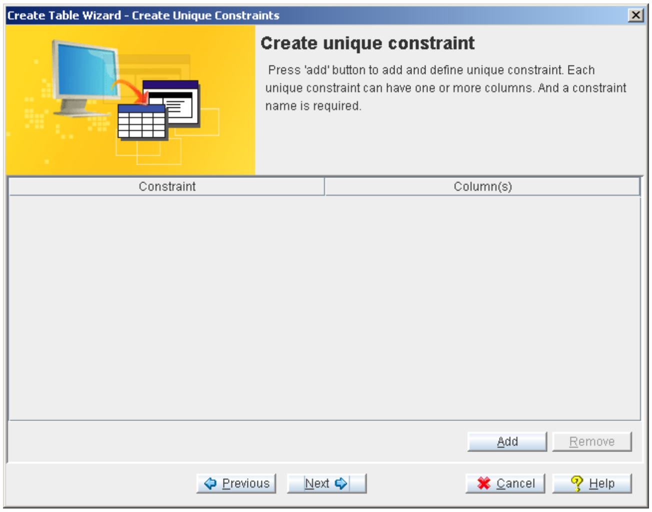
4. Enter table constraints in the **Table Constraints** field.
5. Click the **Next** button. The **Create Unique Constraint** window appears.

Adding a UNIQUE Constraint

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

➲ Adding a unique constraint

1. Follow the steps in the **Create Table Wizard** until the **Create Unique Constraint** window appears.



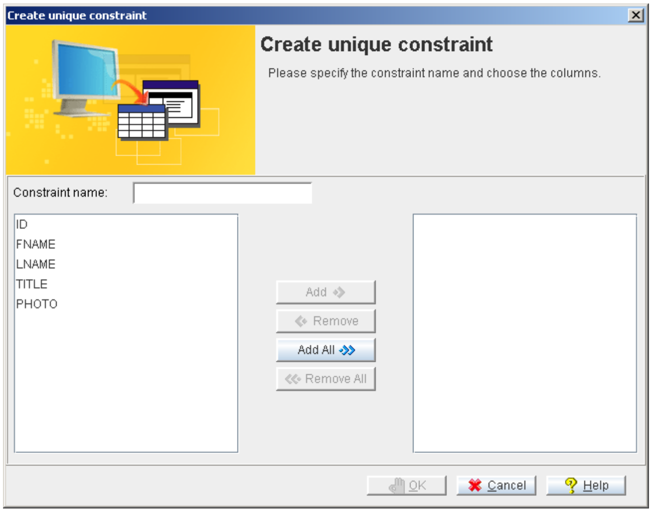
Create Table Wizard - Create Unique Constraints

Create unique constraint

Press 'add' button to add and define unique constraint. Each unique constraint can have one or more columns. And a constraint name is required.

Constraint	Column(s)

2. Click the **Add** button. The **Create Unique Constraint** window appears.



Create unique constraint

Create unique constraint

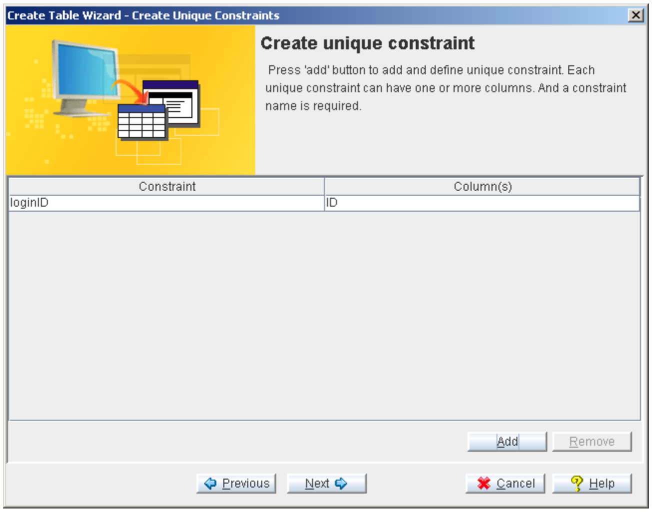
Please specify the constraint name and choose the columns.

Constraint name:

ID
FNAME
LNAME
TITLE
PHOTO

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

4. Select columns to apply the UNIQUE constraint to by double clicking the column names appearing on the left side of the window. Alternatively, use the **Add**, **Add All**, **Remove**, and **Remove All** buttons located in the center area of the window to add or remove columns. Selected columns appear on the right side of the window.
5. Click the **OK** button. The **Create unique constraint** window appears.



Create Table Wizard - Create Unique Constraints

Create unique constraint

Press 'add' button to add and define unique constraint. Each unique constraint can have one or more columns. And a constraint name is required.

Constraint	Column(s)
loginID	ID

6. Click the **Next** button. The **Create Foreign Key to identify Column(s)** window appears.

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 identical in both the referencing and referenced tables.

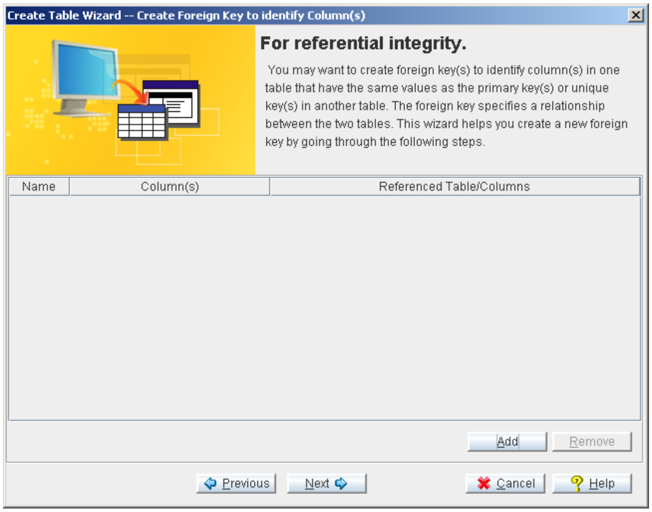
A foreign key ensures referential integrity between two tables. The foreign key accomplishes this task according to the action chosen for the foreign key. When a record in the referenced column(s) is deleted or updated, the foreign key performs the action on the referencing column(s). The possible actions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

- Set Null: Records in the foreign key column are set to null if the user updates or 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 or deletes the corresponding records in the primary key or unique index column
- No Action: No action is taken on the foreign key column if the user updates or deletes the corresponding records in the primary key or unique index column

Please note that the delete action of a foreign key may be different from the update action.

➲ Adding 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 appears.



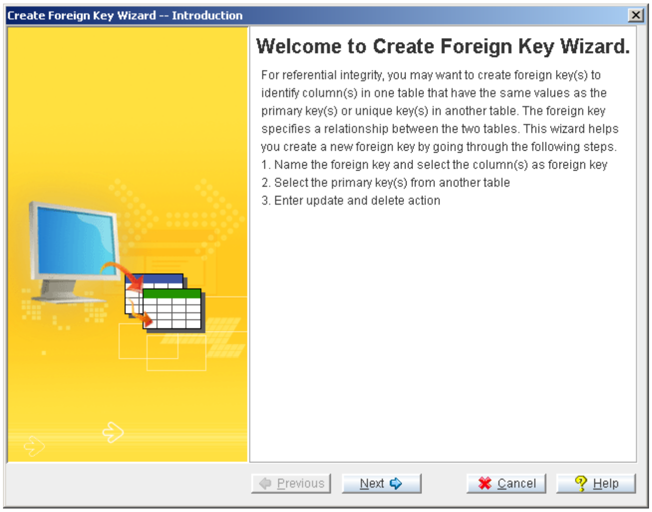
Create Table Wizard -- Create Foreign Key to identify Column(s)

For referential integrity.

You may want to create foreign key(s) to identify column(s) in one table that have the same values as the primary key(s) or unique key(s) in another table. The foreign key specifies a relationship between the two tables. This wizard helps you create a new foreign key by going through the following steps.

Name	Column(s)	Referenced Table/Columns

2. Click the **Add** button. The **Create Foreign Key – Introduction** window appears.

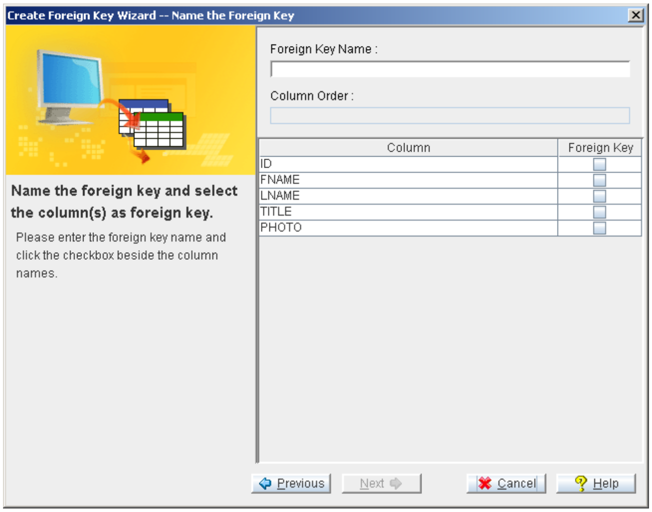


Create Foreign Key Wizard -- Introduction

Welcome to Create Foreign Key Wizard.

For referential integrity, you may want to create foreign key(s) to identify column(s) in one table that have the same values as the primary key(s) or unique key(s) in another table. The foreign key specifies a relationship between the two tables. This wizard helps you create a new foreign key by going through the following steps.
1. Name the foreign key and select the column(s) as foreign key
2. Select the primary key(s) from another table
3. Enter update and delete action

3. Click the **Next** button. The **Name the Foreign Key** window appears.



Create Foreign Key Wizard -- Name the Foreign Key

Name the foreign key and select the column(s) as foreign key.

Please enter the foreign key name and click the checkbox beside the column names.

Foreign Key Name :

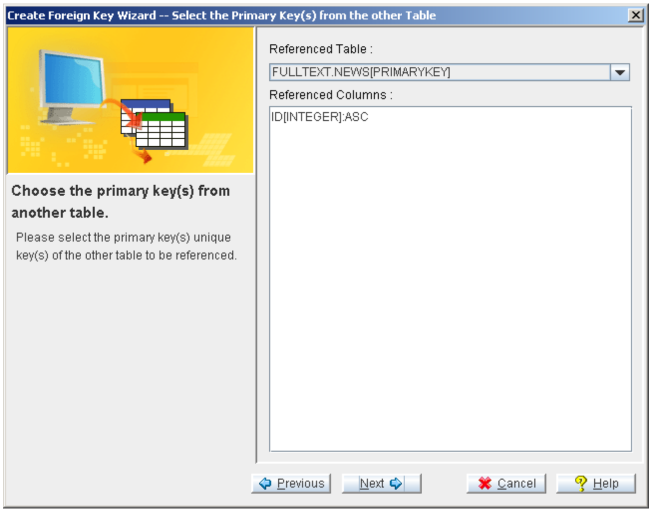
Column Order :

Column	Foreign Key
ID	☐
FNAME	☐
LNAME	☐
TITLE	☐
PHOTO	☐

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

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

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



Create Foreign Key Wizard -- Select the Primary Key(s) from the other Table

Choose the primary key(s) from another table.

Please select the primary key(s) unique key(s) of the other table to be referenced.

Referenced Table : FULLTEXT.NEWS[PRIMARYKEY]

Referenced Columns :

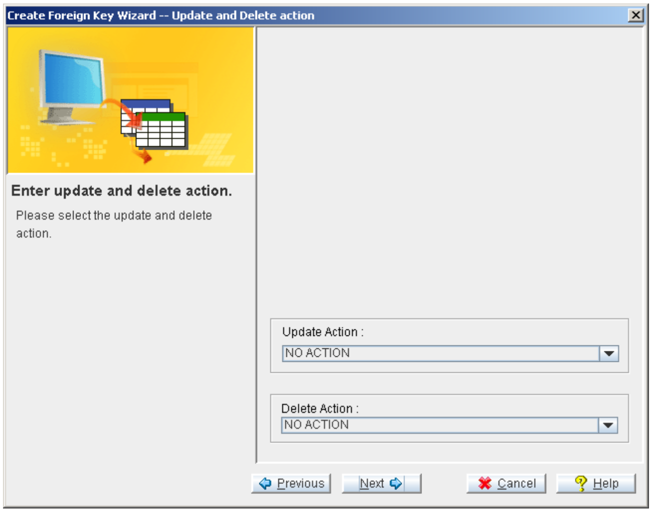
ID[INTEGER]:ASC

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 are displayed.

9. Click the **Next** button. The **Update action and Delete action** window appears.



Create Foreign Key Wizard -- Update and Delete action

Enter update and delete action.

Please select the update and delete action.

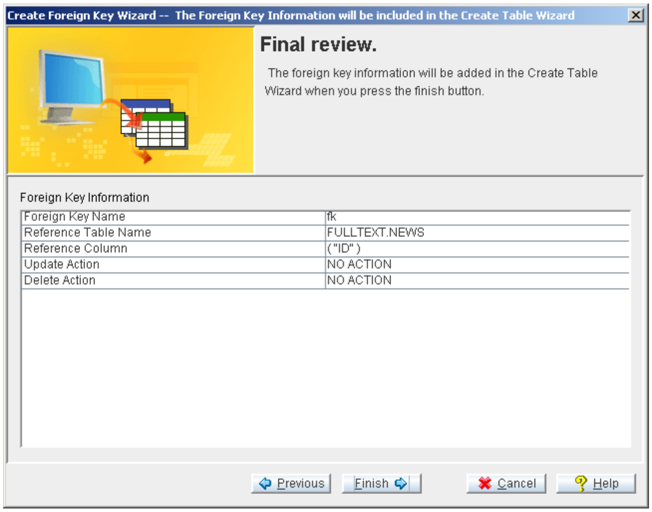
Update Action : NO ACTION

Delete Action : NO ACTION

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

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

12. Click the **Next** button. The **Final Review** window for foreign key creation appears.



Create Foreign Key Wizard -- The Foreign Key Information will be included in the Create Table Wizard

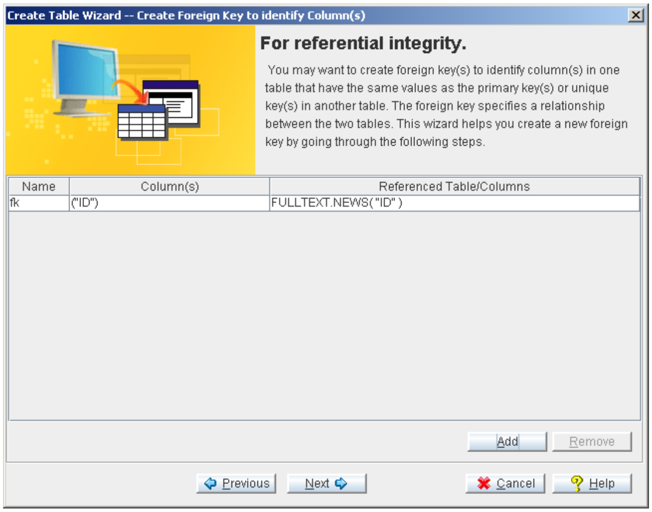
Final review.

The foreign key information will be added in the Create Table Wizard when you press the finish button.

Foreign Key Information

Foreign Key Name	fk
Reference Table Name	FULLTEXT.NEWS
Reference Column	("ID")
Update Action	NO ACTION
Delete Action	NO ACTION

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



Create Table Wizard -- Create Foreign Key to identify Column(s)

For referential integrity.

You may want to create foreign key(s) to identify column(s) in one table that have the same values as the primary key(s) or unique key(s) in another table. The foreign key specifies a relationship between the two tables. This wizard helps you create a new foreign key by going through the following steps.

Name	Column(s)	Referenced Table/Columns
fk	("ID")	FULLTEXT.NEWS("ID")

14. To delete a foreign key:

a) Select a foreign key.

b) Click the **Remove** button.

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

Setting Advanced Table Options

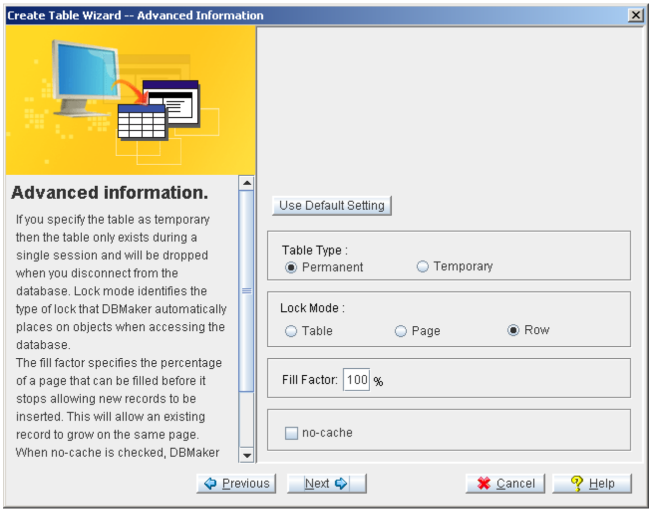
Many performance options are available when creating a table. These options can have a significant impact on the table's data handling performance. The options are:

- Temporary / permanent tables: Temporary tables support fast data operations but are only accessible by the creator. Temporary tables exist during a single session, and DBMaker automatically drops temporary tables when the user (i.e., creator) disconnects from the database. Permanent tables exist in the database until they are dropped by a user having access and delete privileges. The default table type is ***Permanent***.
- Lock mode: DBMaker locks objects in a table according to the table's lock mode. DBMaker supports three lock modes: TABLE, PAGE, and ROW. The default lock mode is **ROW**. When the lock mode is set to the higher level, TABLE, the number of concurrent database accesses decreases, but the required lock resources (i.e., shared memory) also decreases. When the lock mode is set to the lower level, **ROW**, the number of concurrent database accesses increases, but the required lock resources (i.e., shared memory) also increases.
- Fill Factor: specifies the threshold percent when a page no longer allows new records. 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. For a table created with the no-cache option, DBMaker only uses one page buffer to cache data retrieved from the table during table

scans. This prevents a single large table scan from exhausting the page buffers.

➲ Setting advanced table options:

1. Follow the steps in the **Create Table Wizard** until the **Advanced Information** window appears.



Create Table Wizard -- Advanced Information

Advanced information.

If you specify the table as temporary then the table only exists during a single session and will be dropped when you disconnect from the database. Lock mode identifies the type of lock that DBMaker automatically places on objects when accessing the database.
The fill factor specifies the percentage of a page that can be filled before it stops allowing new records to be inserted. This will allow an existing record to grow on the same page.
When no-cache is checked, DBMaker

Table Type : ● Permanent ○ Temporary

Lock Mode : ○ Table ○ Page ● Row

Fill Factor: 100 %

□ no-cache

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

3. If you wish to change the lock mode, select a different lock mode radio button. Please note, the default is **Row**. To use table lock mode, click the **Table** radio button. To use row lock mode, click the **Row** radio button.

4. To change the fill factor from its 100% default value, enter a new value 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.

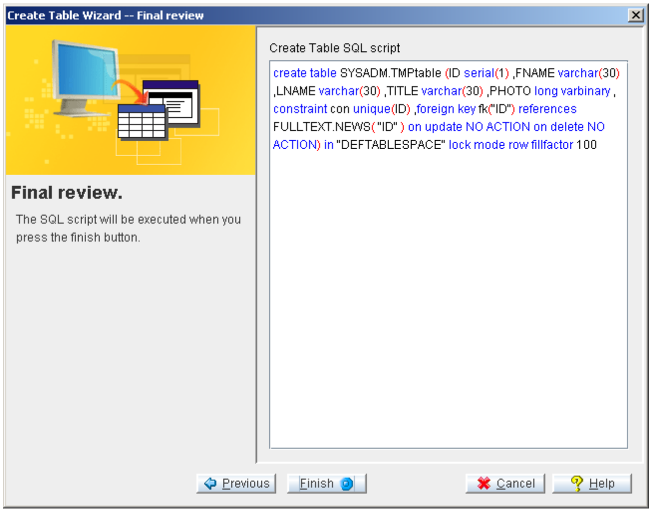
6. Click the **Next** button. Next, the **Final review** window appears.

Completing Table Creation

After selecting table options using the **Create Table Wizard**, you are ready to complete the process of creating the table. A full explanation of the CREATE TABLE statement and syntax is in the *SQL Command and Function Reference*.

➲ Completing table creation:

1. Follow the steps in the **Create Table Wizard** until the **Final Review** window appears.



Create Table Wizard -- Final review
Final review.
The SQL script will be executed when you press the finish button.
Create Table SQL script
Previous
Finish
Cancel
Help

```
create table SYSADM.TMPtable (ID serial(1) ,FNAME varchar(30)
,LNAME varchar(30) ,TITLE varchar(30) ,PHOTO long varbinary ,
constraint con unique(ID) ,foreign key fk("ID") references
FULLTEXT.NEWS( "ID" ) on update NO ACTION on delete NO
ACTION) in "DEFTABLESPACE" lock mode row fillfactor 100
```

2. The SQL statement generated by the **Create Table Wizard** can be modified by hand at this point. Simply use the mouse to place the cursor and insert new text. Edit the statement as needed but maintain SQL99 compliance.
3. Review the final SQL statement and click the **Finish** button. A confirmation message appears.

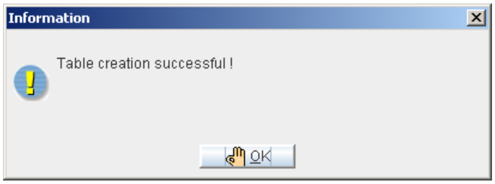


Table creation successful !

4. Click the **OK** button. The newly created table appears in the tree under the **table** node.

5.2 Modifying Table Schema

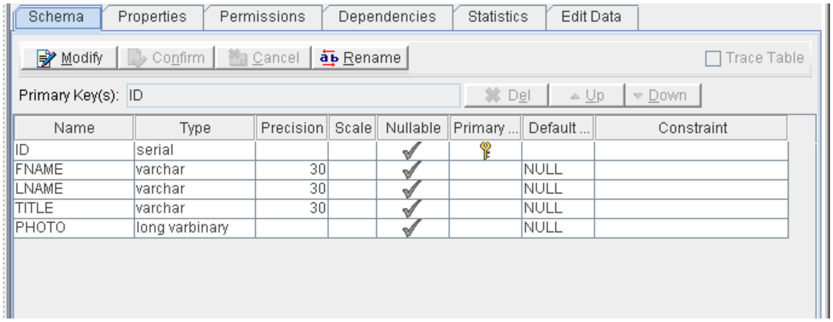
Once a table is created, you can modify its structure by adding or deleting columns, changing the column order, or modifying schema objects. Some restrictions exist to modifying schema. The following schema objects may be altered:

- Column Order
- Name
- Type
- Null / Not Null
- Primary Key(s)
- Default Value
- Column Constraints
- Adding columns
- Deleting columns

➲ Modifying 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 appears.



Primary Key(s): ID

Name	Type	Precision	Scale	Nullable	Primary ...	Default ...	Constraint
ID	serial			✓	🔑		
FNAME	varchar	30		✓		NULL	
LNAME	varchar	30		✓		NULL	
TITLE	varchar	30		✓		NULL	
PHOTO	long varbinary			✓		NULL	

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 the **Modify** button. 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.

➲ Changing 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 the **Confirm** button.

Changing Column Names

New or changed column names must conform to DBMaker's naming conventions. Database configuration determines whether column names are case-sensitive.

➲ Changing a column name:

1. Select a column to rename.

2. Select a field under the **Name** heading.
3. Enter a new name into the field.
4. Click the **Confirm** button.

Changing Column Type

Some data types cannot be converted. For example, CHAR type data cannot be converted to INTEGER type data and 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.

➲ Changing a column’s data type:

1. Select a column to retype.
2. Click a field under the **Type** heading, a drop down menu of available data types appears.
3. Select a new data type from the drop down menu.
4. Click the **Confirm** button.

Changing Precision and Scale

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

➲ Changing a column’s precision or scale:

1. Select the column to retype.
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 the **Confirm** button.

Changing Column Null / Not Null

This setting specifies 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.

➲ Making a column nullable/not nullable:

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

Changing the Primary Key

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

➲ Altering primary keys:

1. Select the column you want to modify.
2. Select the field under the **Primary Key** heading. A key appears in the field to indicate that it is a primary key. The column name also appears at the end of the list in the **Primary Key(s)** text field at the top of the **Schema** page.
3. Clicking 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 the **Confirm** button.

Changing the Column Default Value

Default values are inserted into columns where the user has not input a value for that column in a record. Default values must be valid for the column data type.

➲ Changing or adding a default value for the column:

1. Select a column to change.
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 the **Confirm** button.

Adding Column Constraints

Column constraints must conform to SQL syntax. Please see the section *Constraint Syntax* at the end of this chapter for more information.

➲ To change or add column constraints:

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

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. Foreign keys will be broken if the schema of the referenced table is changed. Please refer to section 4.5, *Checking Dependencies,* for directions on checking for referencing tables.

➲ Adding 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,or scale, or both 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.

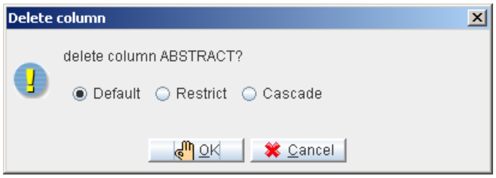
5. Clicking on the blank field under the **Primary key** heading toggles if the column is a primary key or not. A key icon indicates that the column is a primary key.
6. Enter a default value into the blank field under the **Default Value** heading at the bottom of the list (optional).
7. Enter column constraints into the blank field under the **Constraints** heading at the bottom of the list.
8. Click the **Confirm** button.

Deleting a Column

Columns may be deleted from a table if they are no longer needed.

➲ Deleting a column:

1. Select the column you want to delete.
2. Click the **Delete** button.
3. Click the **Confirm** button. The **Delete column** confirmation dialog box appears.



Delete column

delete column ABSTRACT?

● Default ○ Restrict ○ Cascade

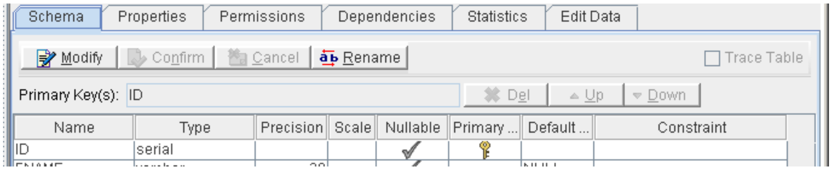
4. DBMaker provides three options for deleting columns. The **Default** option drops columns without checking for dependent object. Any dependent objects become invalid after the column drop. The **Restrict** option considers index, text index, and view dependencies. This option does not permit a column drop when one or more dependencies exist. The **Cascade** option drops the column and all of its dependent objects.
5. Click the **OK** button.

Tracing a Table

Whlie Keywords DB_LGSVR is equal or greater then 4, user can adds trace on a single table to log the detail OLD/NEW data.

- Tracing a table:

1. Select the table you want to add trace on.
2. Select the Trace Table checkbox.



Primary Key(s): ID

Name	Type	Precision	Scale	Nullable	Primary ...	Default ...	Constraint
ID	serial			✓	🔑		

5.3 Setting Table Properties

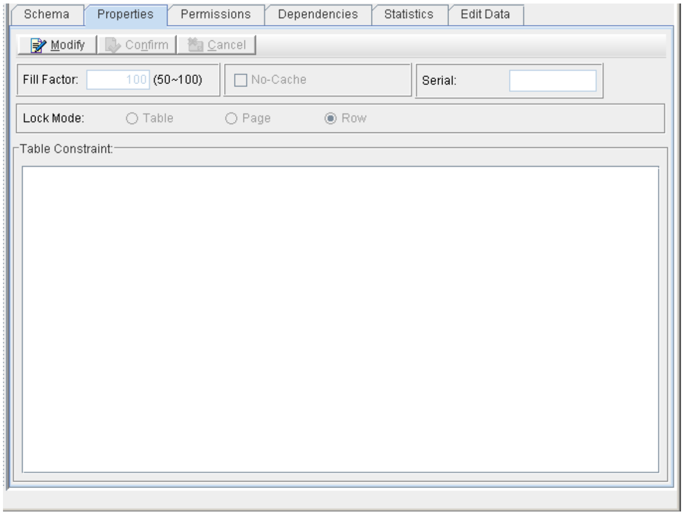
You can modify the settings you added when you created a table, and view the table constraints. The following properties may be altered:

- Lock Mode: DBMaker locks objects in a table according to the table's lock mode. DBMaker supports three lock modes: TABLE, PAGE, and ROW. The default lock mode is **ROW**. When the lock mode is set to the higher level, **TABLE**, the number of concurrent database accesses decreases, but the required lock resources (i.e., shared memory) also decreases. When the lock mode is set to the lower level, **ROW**, the number of concurrent database accesses increases, but the required lock resources (i.e., shared memory) also increases.
- Fill Factor: specifies the threshold percent when a page no longer allows new records. Lowering the fill factor value facilitates the growth of existing records in the page.
- 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. 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. For a table created with the no-cache option, DBMaker only uses one page buffer to cache the data retrieved from the table during table scans. This prevents a single large table scan from exhausting the page buffers.
- Serial: Resets the serial number for a serial type column in the table.

➲ Setting table properties:

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

3. Select the **Properties** tab. The **Properties** page appears.



Fill Factor: 100 (50~100)

□ No-Cache

Serial:

Lock Mode: ○ Table ○ Page ● Row

Table Constraint:

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

   - Click the **Table** radio button to choose table lock mode

   - Click the **Row** radio button to choose row lock mode

5. 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. Enter the starting number of the column in the **Serial** box to reset the serial number for a serial type column in the table.

8. Click the **Confirm** button.

5.4 Granting and 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 users access to tables. Privileges are granted to specific functions such as Insert, Update, Delete, Index, and Reference, or to specific columns of the table.

Users with resource or DBA privilege cannot have their privileges revoked on objects they own. These privileges are only revoked by changing the user's authority.

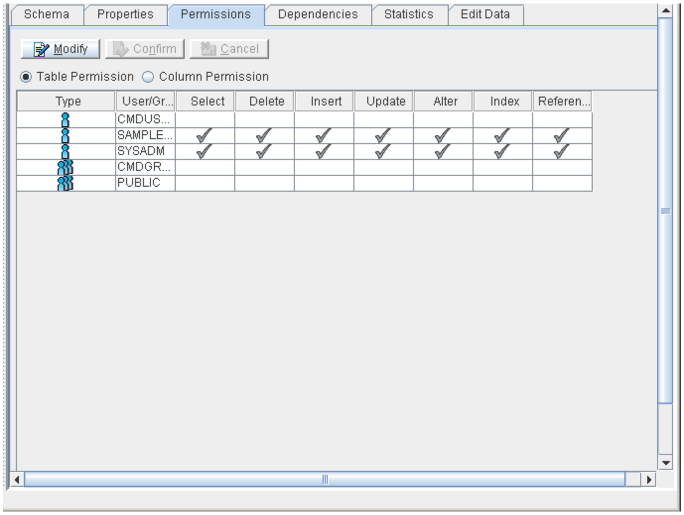
The following list summarizes table privileges:

- **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

➲ Granting and revoking table privileges:

1. Select a table. The **Schema** page appears.

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



● Table Permission ○ Column Permission

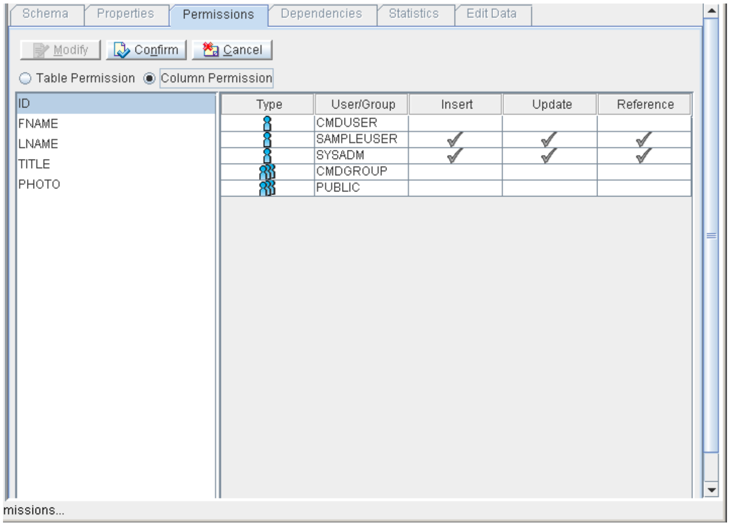
Type	User/Gr...	Select	Delete	Insert	Update	Alter	Index	Referen...
👤	CMDUS...							
👤	SAMPLE...	✓	✓	✓	✓	✓	✓	✓
👤	SYSADM	✓	✓	✓	✓	✓	✓	✓
👥	CMDGR...							
👥	PUBLIC							

3. Click the **Modify** button. The **Confirm** and **Cancel** buttons are now active.

4. Select one of the permissions in the cells for the user or group.

5. Click any of the checked columns to revoke that privilege.

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



○ Table Permission ● Column Permission

ID
FNAME
LNAME
TITLE
PHOTO

Type	User/Group	Insert	Update	Reference
👤	CMDUSER			
👤	SAMPLEUSER	✓	✓	✓
👤	SYSADM	✓	✓	✓
👥	CMDGROUP			
👥	PUBLIC			

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. Click any of the checked columns to revoke a privilege.

9. Click the **Confirm** button.

5.5 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.

➲ Checking dependencies:

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

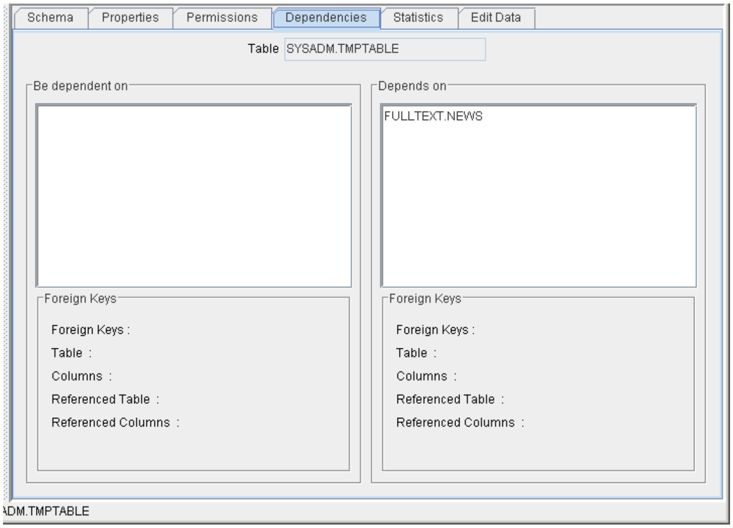


Table SYSADM.TMPTABLE

Depends on: FULLTEXT.NEWS

ADM.TMPTABLE

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

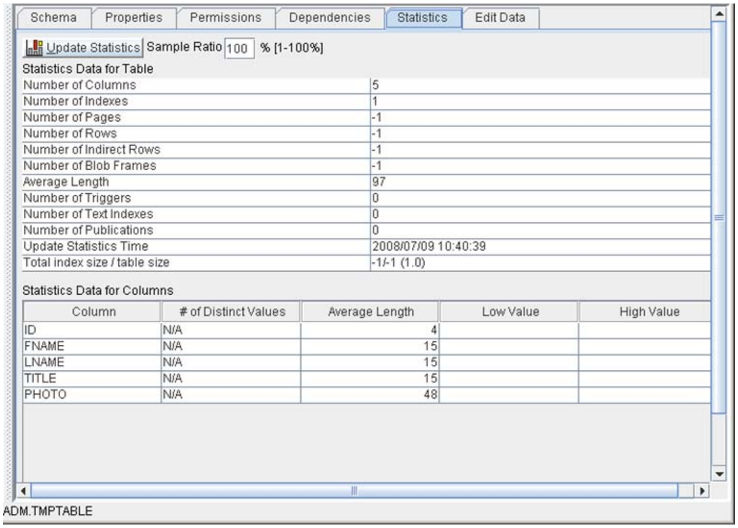
4. Selecting a table from the **Depends on** list displays foreign key information on the bottom.
5. Double-click the table name on either list to change the selected table in the **Table** field at the top of the page.

Checking and Updating Table Statistics

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

➲ Checking and updating table statistics:

1. Select a table. The **Schema** page appears.
2. Select the **Statistics** tab at the top of the window. The statistics appear as shown below.



Sample Ratio 100 % [1-100%]

Statistics Data for Table

Number of Columns	5
Number of Indexes	1
Number of Pages	-1
Number of Rows	-1
Number of Indirect Rows	-1
Number of Blob Frames	-1
Average Length	97
Number of Triggers	0
Number of Text Indexes	0
Number of Publications	0
Update Statistics Time	2008/07/09 10:40:39
Total index size / table size	-1/-1 (1.0)

Statistics Data for Columns

Column	# of Distinct Values	Average Length	Low Value	High Value
ID	N/A	4		
FNAME	N/A	15		
LNAME	N/A	15		
TITLE	N/A	15		
PHOTO	N/A	48		

ADM.TMPTABLE

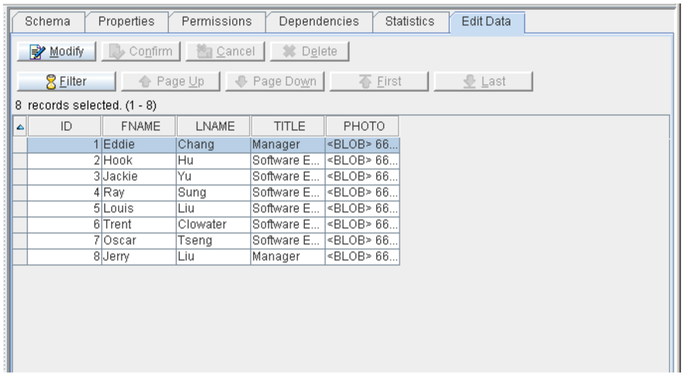
3. Select the **Sample Ratio** box and enter a sample ratio percentage to change the percentage of the data used to calculate updated statistics.
4. Click the **Update Statistics** button. The updated statistics appear.

5.6 Editing Data in a Table

Once you add records to a table, you may also update existing records in that table. Object privilege is required for table editing. Please refer to *Granting and Revoking Table Privileges* and *Granting and Removing Object Privileges* for information about managing table and column user privileges.

➲ Editing data in a table:

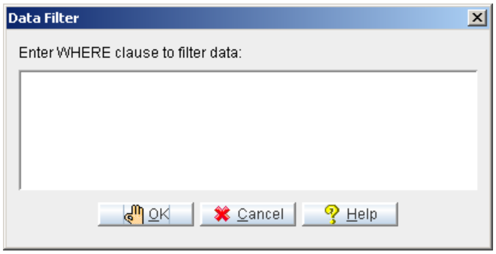
1. Double-click the **Table** node in the tree.
2. Double-click the table you want to modify. The **Schema** page appears.
3. Click the **Edit Data** tab on top of the window. The **Edit Data** page appears.



8 records selected. (1 - 8)

ID	FNAME	LNAME	TITLE	PHOTO
1	Eddie	Chang	Manager	<BLOB> 66...
2	Hook	Hu	Software E...	<BLOB> 66...
3	Jackie	Yu	Software E...	<BLOB> 66...
4	Ray	Sung	Software E...	<BLOB> 66...
5	Louis	Liu	Software E...	<BLOB> 66...
6	Trent	Clowater	Software E...	<BLOB> 66...
7	Oscar	Tseng	Software E...	<BLOB> 66...
8	Jerry	Liu	Manager	<BLOB> 66...

4. To filter the data, click **Filter**. The **Data Filter** window appears as shown below.



5. Enter a **WHERE** condition to filter the data and click the **OK** button. The main window for the table appears.

NOTE *WHERE conditions must conform to proper SQL syntax. Please refer to the section on Constraint Syntax at the end of this chapter for more information.*

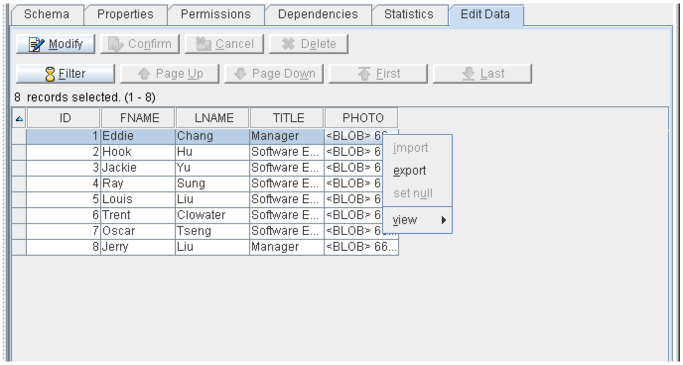
6. Updating data:

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

   b) Click the **Modify** button.

   c) Select the data on the screen.

   d) Edit the data. If the column contains BLOB data, a pop-up menu appears 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 to store the file name by 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 according to the selected view.



8 records selected. (1 - 8)

ID	FNAME	LNAME	TITLE	PHOTO
1	Eddie	Chang	Manager	<BLOB> 6
2	Hook	Hu	Software E...	<BLOB> 6
3	Jackie	Yu	Software E...	<BLOB> 6
4	Ray	Sung	Software E...	<BLOB> 6
5	Louis	Liu	Software E...	<BLOB> 6
6	Trent	Clowater	Software E...	<BLOB> 6
7	Oscar	Tseng	Software E...	<BLOB> 6
8	Jerry	Liu	Manager	<BLOB> 66...

   e) Click the **Confirm** button.

7. Deleting data:

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

   b) Click the **Modify** button.

   c) Select the data on the screen.

   d) Click the **Delete** button.

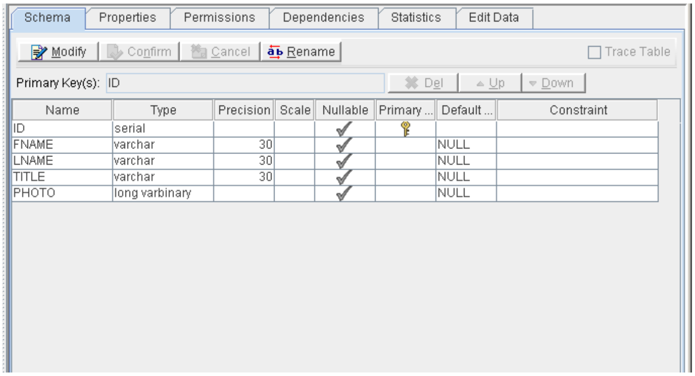
   e) Click the **Confirm** button.

5.7 Renaming a Table

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

➲ To rename a table:

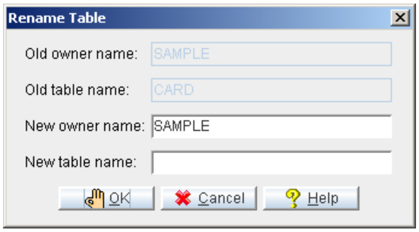
1. Expand the table node of the tree.
2. Select a table to change the name of from the tree. The table Schema page appears.



Primary Key(s): ID

Name	Type	Precision	Scale	Nullable	Primary ...	Default ...	Constraint
ID	serial			✓	🔑		
FNAME	varchar	30		✓		NULL	
LNAME	varchar	30		✓		NULL	
TITLE	varchar	30		✓		NULL	
PHOTO	long varbinary			✓		NULL	

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



Old owner name: SAMPLE

Old table name: CARD

New owner name: SAMPLE

New table name:

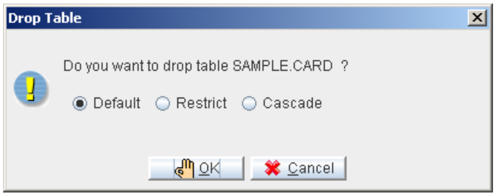
4. Enter a new owner name and table name into the appropriate fields and click **OK**.

5.8 Dropping a Table

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

➲ Dropping a table:

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



Do you want to drop table SAMPLE.CARD ?

● Default ○ Restrict ○ Cascade

3. There are three options for dropping tables. The **Default** option deletes the table without considering dependent object within the table. Dependent objects become invalid after the table is dropped. The **Restrict** option considers dependencies. This option does not permit a table drop when one or more objects depend on the table. The **Cascade** option drops the table and all of its dependent objects. Click a radio button to select a table dropping option.
4. Click the **OK** button.

5.9 Constraint Syntax

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

You can combine simple conditions with logical operators like 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, but not 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 5-1 Constraint Syntax Comparators

.

6 Working with Views

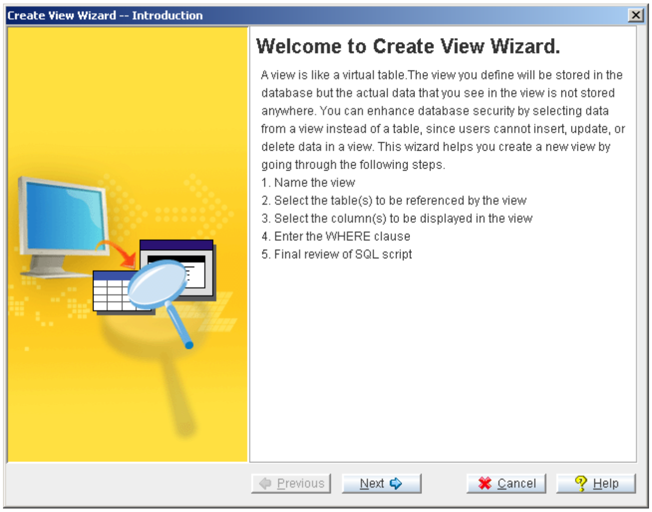
DBMaker provides the ability to define a virtual table, called a *view*. The view definition is a logical construct that is computed or collated (i.e., derived) from one or more tables and functions like a table but it is not stored in the database as a table itself. You can create views for flexible data queries on tables. You can use views to browse data in multiple tables. JDBA Tool allows you to create, browse, update, and drop views.

6.1 Creating a View

After you enter data into the database tables, you can use the tables to create views.

➲ Creating a view:

1. Select the **View** object from the tree. The owner name, view name and time of creation for all views on the database appear on a table in the **View** page.
2. Click **Create**. The **Create View Wizard – Introduction** window appears.



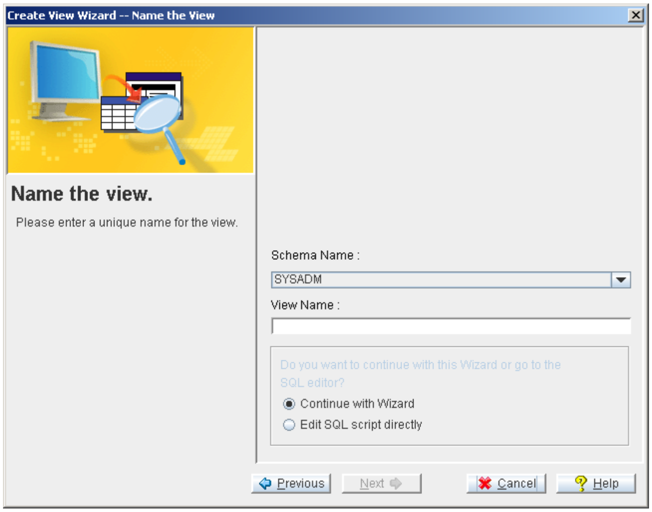
Create View Wizard -- Introduction

Welcome to Create View Wizard.

A view is like a virtual table.The view you define will be stored in the database but the actual data that you see in the view is not stored anywhere. You can enhance database security by selecting data from a view instead of a table, since users cannot insert, update, or delete data in a view. This wizard helps you create a new view by going through the following steps.

1. Name the view
2. Select the table(s) to be referenced by the view
3. Select the column(s) to be displayed in the view
4. Enter the WHERE clause
5. Final review of SQL script

3. Click **Next**. The **Name the View** window appears.



Create View Wizard -- Name the View

Name the view.

Please enter a unique name for the view.

Schema Name : SYSADM

View Name :

Do you want to continue with this Wizard or go to the SQL editor?

● Continue with Wizard

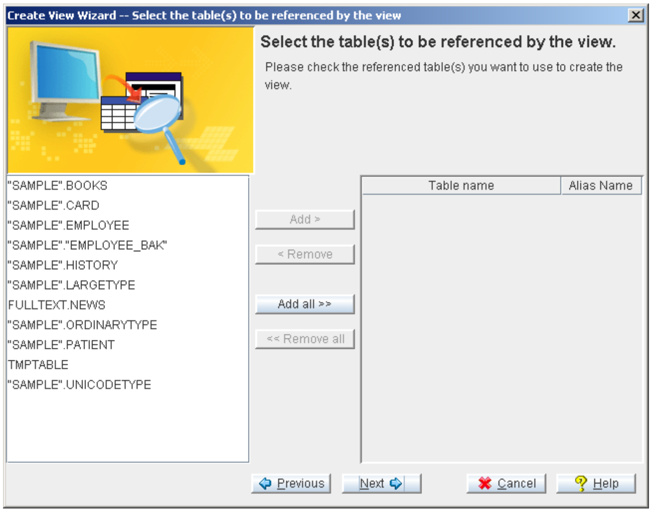
○ Edit SQL script directly

4. Select the schema name from the **Schema Name** drop down list.

5. Enter the name of the view in the **View Name** field.

6. Select the **Continue with Wizard** radio button orselect the **Edit SQL script directly** radio button to jump directly to the last step of the wizard (see step 14).

7. Click the **Next** button. The **Select the table(s) to be referenced by the view** dialog box appears.



Create View Wizard -- Select the table(s) to be referenced by the view

Select the table(s) to be referenced by the view.

Please check the referenced table(s) you want to use to create the view.

"SAMPLE".BOOKS
"SAMPLE".CARD
"SAMPLE".EMPLOYEE
"SAMPLE"."EMPLOYEE_BAK"
"SAMPLE".HISTORY
"SAMPLE".LARGETYPE
FULLTEXT.NEWS
"SAMPLE".ORDINARYTYPE
"SAMPLE".PATIENT
TMPTABLE
"SAMPLE".UNICODETYPE

Table name	Alias Name

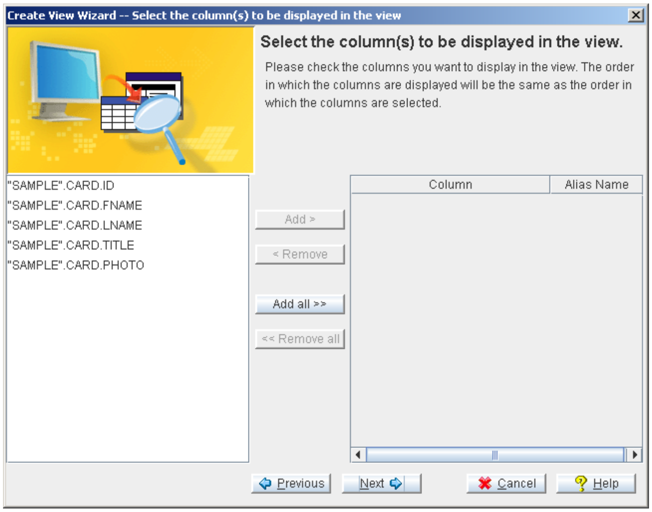
8. Select tables to use in the view by double-clicking the table names on the left. Alternatively, use the buttons in the middle to add or remove columns. Selected tables appear on the right.

9. You can add a table to a view more than once, however, an alias for the table is required. The alias is a new name for the table that allows referencing columns in the table more than once. To add an alias:

   a) Select a table already added to the view.

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

10. Click **Next**. The **Select the column(s) to be displayed in the view** window appears.



Create View Wizard -- Select the column(s) to be displayed in the view

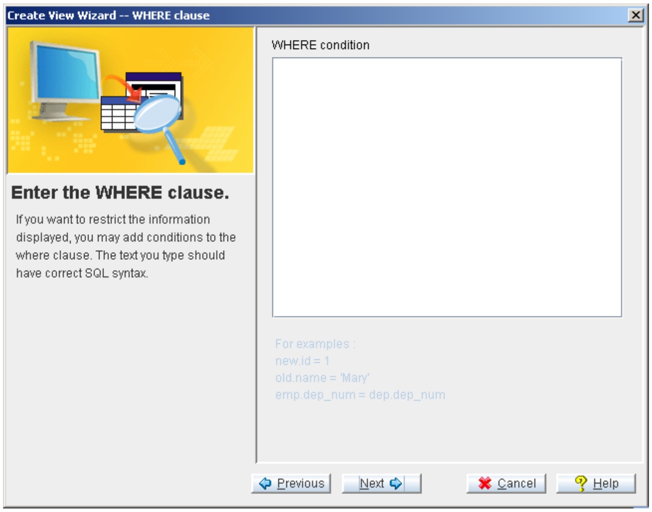
Select the column(s) to be displayed in the view.

Please check the columns you want to display in the view. The order in which the columns are displayed will be the same as the order in which the columns are selected.

"SAMPLE".CARD.ID
"SAMPLE".CARD.FNAME
"SAMPLE".CARD.LNAME
"SAMPLE".CARD.TITLE
"SAMPLE".CARD.PHOTO

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

12. Click **Next**. The WHERE clause window appears.



Create View Wizard -- WHERE clause

Enter the WHERE clause.

If you want to restrict the information displayed, you may add conditions to the where clause. The text you type should have correct SQL syntax.

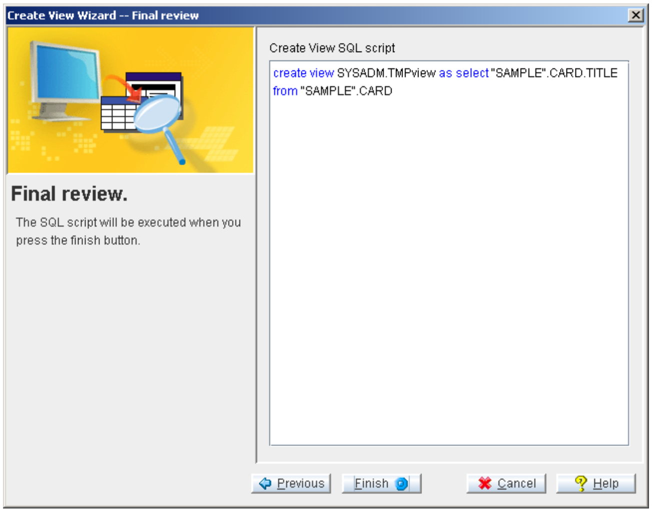
WHERE condition

For examples :
new.id = 1
old.name = 'Mary'
emp.dep_num = dep.dep_num

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

NOTE *WHERE statements must conform to SQL syntax rules. Please see the section on Constraint Syntax near the end of Chapter 4.*

14. Click **Next**. The **Final review** window appears.



Final review.

The SQL script will be executed when you press the finish button.

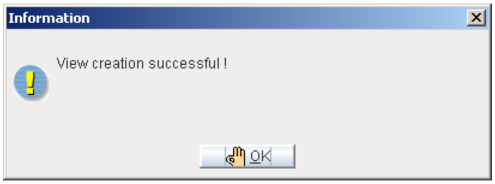
Create View SQL script

```
create view SYSADM.TMPview as select "SAMPLE".CARD.TITLE
from "SAMPLE".CARD
```

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

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

17. Click **OK**.



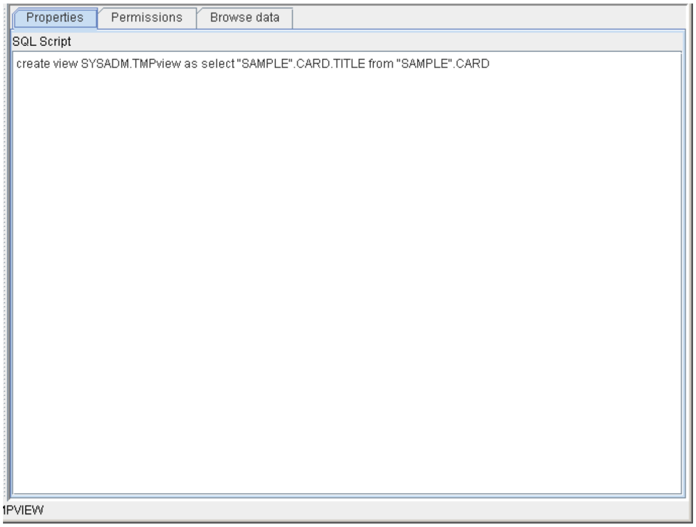
View creation successful !

6.2 Checking View Properties

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

➲ Checking view properties:

1. Click the **View** object in the tree. All the views in the database appear.
2. Select a view. The **Properties** page appears.



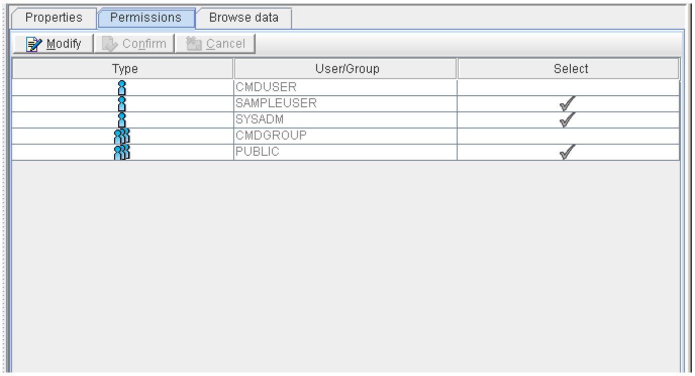
```
create view SYSADM.TMPview as select "SAMPLE".CARD.TITLE from "SAMPLE".CARD
```

6.3 Granting View Privileges

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

➲ Granting view privileges:

1. Select the **View** object in the tree.
2. Select a view. The **Properties** page appears.
3. Click the **Permissions** tab on the top of the window. The permission details appear.



Type	User/Group	Select
👤	CMDUSER	
👤	SAMPLEUSER	✓
👤	SYSADM	✓
👥	CMDGROUP	
👥	PUBLIC	✓

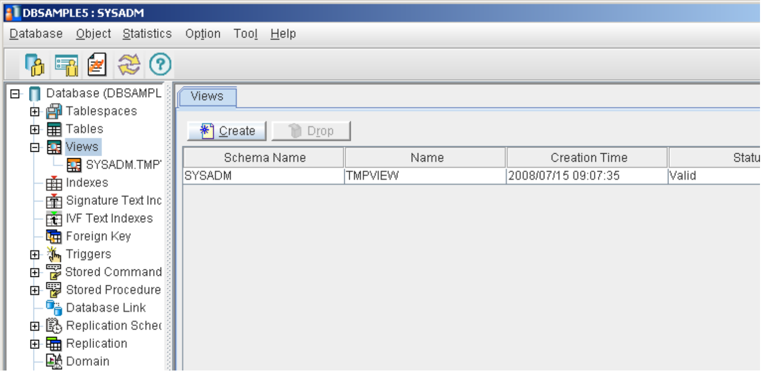
4. Click the **Modify** button.
5. Select the user/group that you want to grant or revoke select privilege to the view.
6. Click the **Select** column next to the selected User/Group. A check mark indicates that the user or group has permission on the view.
7. Click the **Confirm** button.

6.4 Browsing Views

You can quickly and efficiently browse the data contained in a view. You can sort this data using the filter data function in JDBATool. For optimal efficiency, break exceptional large views into several parts. This will increase the ease and speed that you will be able to browse the view's data. The maximum number of records that can be browsed in a single instance is 5,000.

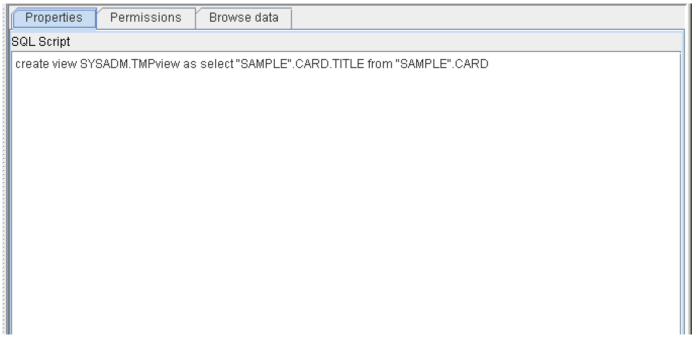
➲ Browsing a view:

1. Click the **Views** icon from the tree. A list of all the views in the database appears beneath the **Views** icon.



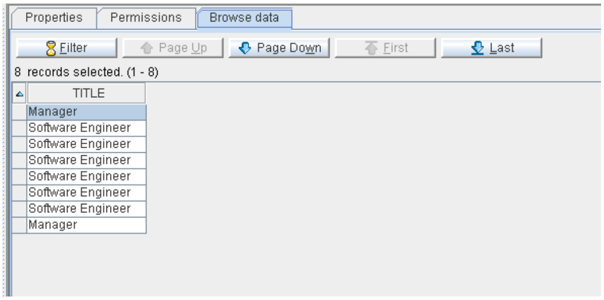
Schema Name	Name	Creation Time	Statu
SYSADM	TMPVIEW	2008/07/15 09:07:35	Valid

2. Select the view you want to browse from the list of views. The view appears in the panel to the right.



```
create view SYSADM.TMPview as select "SAMPLE".CARD.TITLE from "SAMPLE".CARD
```

3. Click the **Browse data** tab in the panel on the right. The **Browse data** window opens in the panel on the right.



8 records selected. (1 - 8)

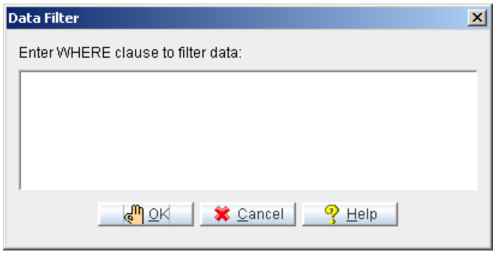
TITLE
Manager
Software Engineer
Software Engineer
Software Engineer
Software Engineer
Software Engineer
Software Engineer
Manager

a) To quickly browse the view by page, use the **Page Up** and **Page Down** buttons.

b) Click the **First** button to jump to the first entry in the view.

c) Click the **Last** button to jump to the last entry in the view.

4. Click the **Filter** button to set sort rules for browsing the view. The **Data Filter** dialog box appears.



a) Enter a WHERE clause or an ORDER BY clause to filter the data when browsing a view.

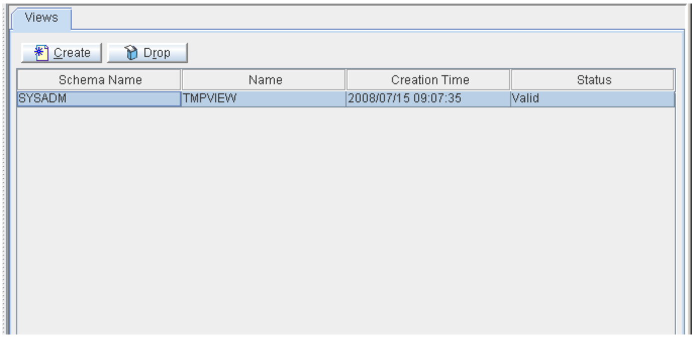
NOTE *WHERE statements must comply with SQL syntax rules. Please see the section on Constraint Syntax near the end of Chapter 4 for information.*

6.5 Dropping a View

Delete unwanted views using the following procedure.

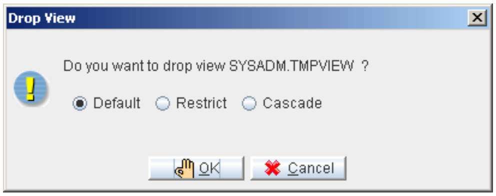
➲ Dropping a view:

1. Click the **View** object in the tree. All views in the database are displayed.
2. Select a view



Schema Name	Name	Creation Time	Status
SYSADM	TMPVIEW	2008/07/15 09:07:35	Valid

3. Click the **Drop** button at the top of the window. The **Drop View** dialog box appears. Alternatively, right click the view to drop from the tree, a pop-up menu appears. Select **Drop** from the pop-up menu.



Drop View

Do you want to drop view SYSADM.TMPVIEW ?

● Default ○ Restrict ○ Cascade

4. There are three options for dropping views. The **Default** option deletes the views without considering dependent object within the views. Dependent objects become invalid after the view is dropped. The **Restrict** option considers dependencies. This option does not permit a views drop when one or more objects depend on the views. The **Cascade** option drops the views and all of its dependent objects. Click a radio button to select a views dropping option.
5. Click **OK**. The remaining views in the database are displayed.

7 Working with Indexes

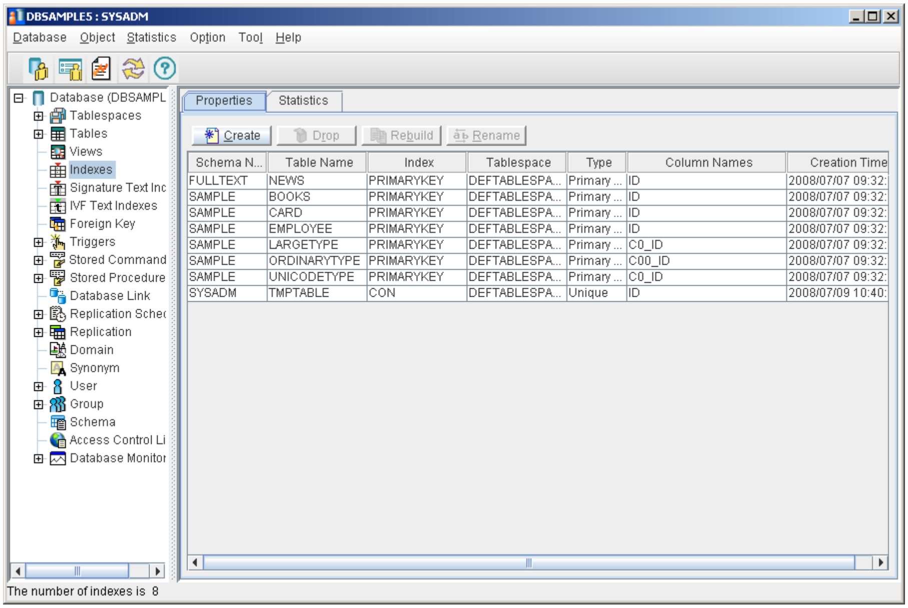
An index facilitates random access of table rows. An index is a sub-table of the table containing the accessing rows. The index table contains the key columns of its parent table. Indexes are particularly useful for tables that hold large amounts of data, or tables that are frequently accessed. An index 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.

7.1 Creating an Index

Create an index 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.

➲ Creating an index:

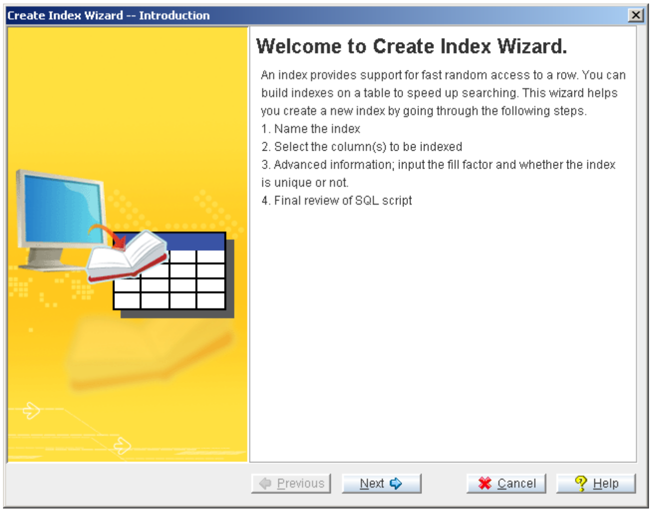
1. Click the **Indexes** object in the tree. DBMaker lists the database's indexes.



Schema N...	Table Name	Index	Tablespace	Type	Column Names	Creation Time
FULLTEXT	NEWS	PRIMARYKEY	DEFTABLESPA...	Primary ...	ID	2008/07/07 09:32:
SAMPLE	BOOKS	PRIMARYKEY	DEFTABLESPA...	Primary ...	ID	2008/07/07 09:32:
SAMPLE	CARD	PRIMARYKEY	DEFTABLESPA...	Primary ...	ID	2008/07/07 09:32:
SAMPLE	EMPLOYEE	PRIMARYKEY	DEFTABLESPA...	Primary ...	ID	2008/07/07 09:32:
SAMPLE	LARGETYPE	PRIMARYKEY	DEFTABLESPA...	Primary ...	C0_ID	2008/07/07 09:32:
SAMPLE	ORDINARYTYPE	PRIMARYKEY	DEFTABLESPA...	Primary ...	C00_ID	2008/07/07 09:32:
SAMPLE	UNICODETYPE	PRIMARYKEY	DEFTABLESPA...	Primary ...	C0_ID	2008/07/07 09:32:
SYSADM	TMPTABLE	CON	DEFTABLESPA...	Unique	ID	2008/07/09 10:40:

The number of indexes is 8

2. Click the **Create** button. The **Create Index Wizard** appears.



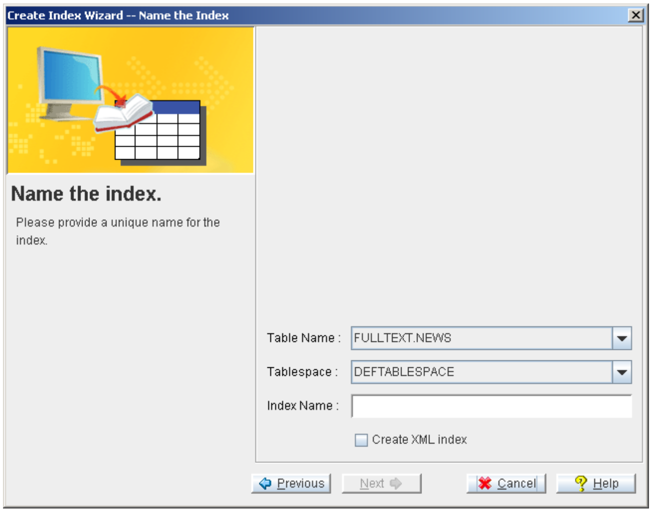
Create Index Wizard -- Introduction

Welcome to Create Index Wizard.

An index provides support for fast random access to a row. You can build indexes on a table to speed up searching. This wizard helps you create a new index by going through the following steps.

1. Name the index
2. Select the column(s) to be indexed
3. Advanced information; input the fill factor and whether the index is unique or not.
4. Final review of SQL script

3. Click the **Next** button. The **Name the Index** window appears.



Create Index Wizard -- Name the Index

Name the index.

Please provide a unique name for the index.

Table Name : FULLTEXT.NEWS

Tablespace : DEFTABLESPACE

Index Name :

□ Create XML index

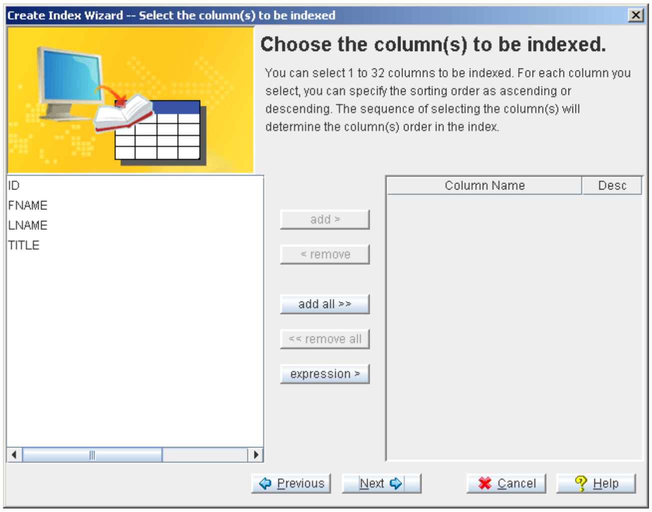
4. The index will refer to the table name that you select from the **Table Name** menu.

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. To improve the performance of XML queries, select the **Create XML Index** radio button. This creates a special XML index on the XML columns.

8. Click the **Next** button. The **Select the column(s) to be indexed** window appears.



Create Index Wizard -- Select the column(s) to be indexed

Choose the column(s) to be indexed.

You can select 1 to 32 columns to be indexed. For each column you select, you can specify the sorting order as ascending or descending. The sequence of selecting the column(s) will determine the column(s) order in the index.

ID
FNAME
LNAME
TITLE

Column Name	Desc

9. 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 adds all columns to the index. All the columns selected for the creation of the index are displayed in the list box to the right. Columns in the index can be removed by double-clicking the column name or selecting the column and clicking the **Remove** button.

10. The **expression** buttion allows you to input an express for the index.

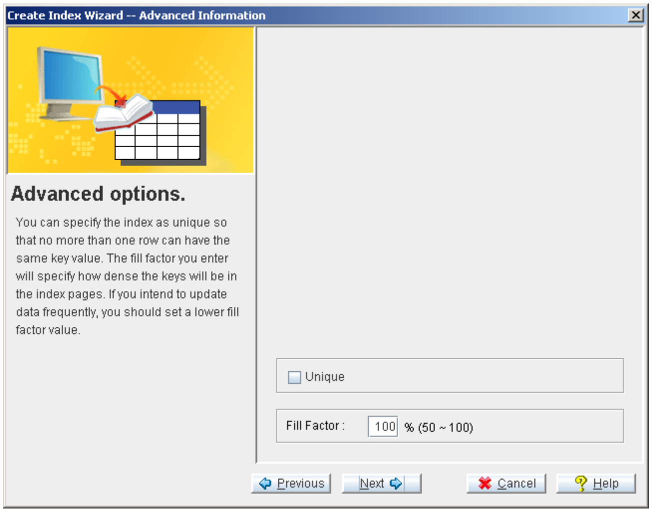
11. To change the sorting order to descending for a column:

    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.*

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



Create Index Wizard -- Advanced Information

Advanced options.

You can specify the index as unique so that no more than one row can have the same key value. The fill factor you enter will specify how dense the keys will be in the index pages. If you intend to update data frequently, you should set a lower fill factor value.

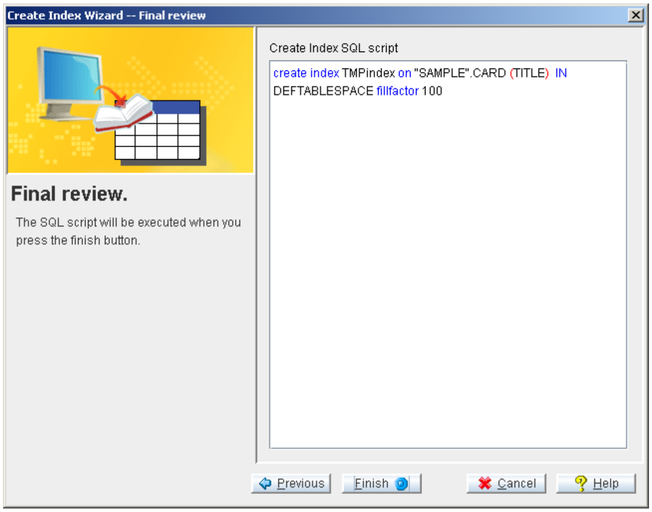
□ Unique

Fill Factor : 100 % (50 ~ 100)

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

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

15. Click the **Next** button. The **Final review** dialog box appears.

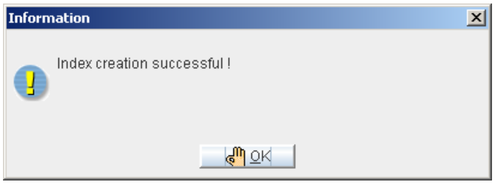


The SQL script will be executed when you press the finish button.

Create Index SQL script

```
create index TMPindex on "SAMPLE".CARD (TITLE) IN
DEFTABLESPACE fillfactor 100
```

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



Index creation successful !

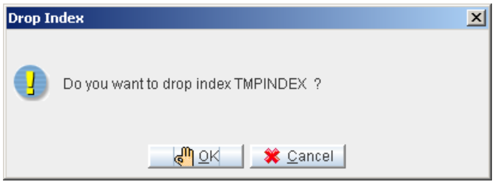
17. Click the **OK** button.

7.2 Dropping an Index

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

➲ Dropping 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 appears.



Drop Index

Do you want to drop index TMPINDEX ?

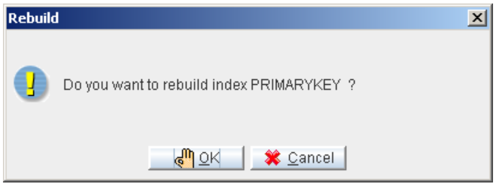
3. Click **OK**. All the remaining indexes in the database are displayed on the **Properties** page.

7.3 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 replaced with a new one.

➲ Rebuilding an index:

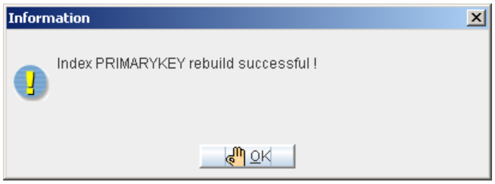
1. Click the **Index** object in the tree. DBMaker lists of all indexes in the database on the **Properties** page.
2. Select an Index to rebuild.
3. Click **Rebuild**. The **Rebuid** message box appears.



Rebuild

Do you want to rebuild index PRIMARYKEY ?

4. Click **OK**.The **Information** message box appears.



Information

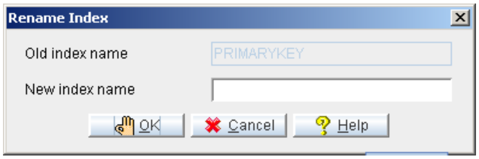
Index PRIMARYKEY rebuild successful !

7.4 Renaming an Index

Any index may be renamed. The renaming only affects the index name in the system catalog. Renaming does not rebuild the index in the database.

➲ Renaming 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 an index to be renamed.
3. Click the **Rename** button. The Rename Index message box appears.



Rename Index

Old index name: PRIMARYKEY

New index name:

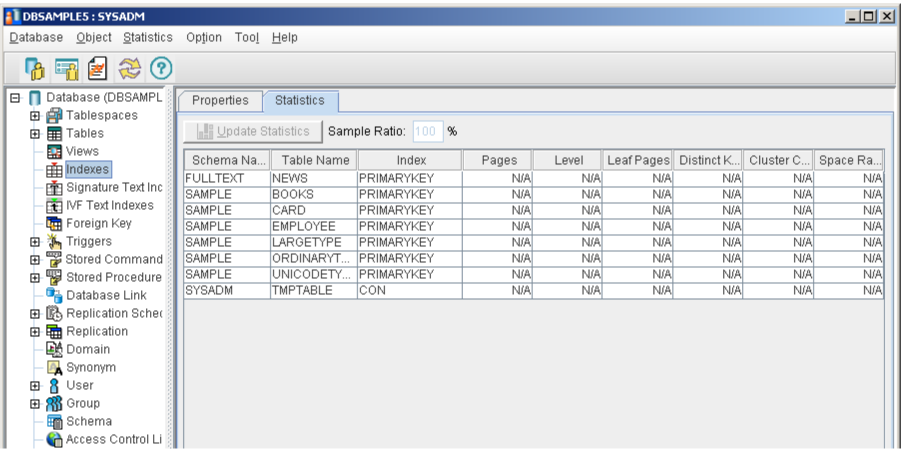
4. Input the new index name in the proper filed.
5. Click **OK**.

7.5 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.

➲ Updating 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 appears.

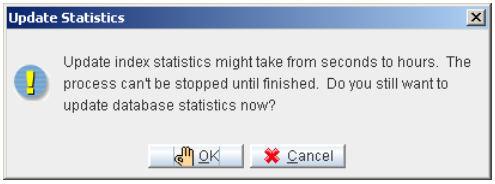


Sample Ratio: 100 %

Schema Na...	Table Name	Index	Pages	Level	Leaf Pages	Distinct K...	Cluster C...	Space Ra...
FULLTEXT	NEWS	PRIMARYKEY	N/A	N/A	N/A	N/A	N/A	N/A
SAMPLE	BOOKS	PRIMARYKEY	N/A	N/A	N/A	N/A	N/A	N/A
SAMPLE	CARD	PRIMARYKEY	N/A	N/A	N/A	N/A	N/A	N/A
SAMPLE	EMPLOYEE	PRIMARYKEY	N/A	N/A	N/A	N/A	N/A	N/A
SAMPLE	LARGETYPE	PRIMARYKEY	N/A	N/A	N/A	N/A	N/A	N/A
SAMPLE	ORDINARYT...	PRIMARYKEY	N/A	N/A	N/A	N/A	N/A	N/A
SAMPLE	UNICODETY...	PRIMARYKEY	N/A	N/A	N/A	N/A	N/A	N/A
SYSADM	TMPTABLE	CON	N/A	N/A	N/A	N/A	N/A	N/A

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

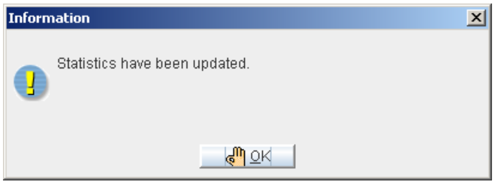
4. Click the **Update Statistics** button. The following window appears.



Update Statistics

Update index statistics might take from seconds to hours. The process can't be stopped until finished. Do you still want to update database statistics now?

5. Click **OK**.



Information

Statistics have been updated.

6. Click **OK**.

8 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 would be otherwise possible when handled 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 exist within the same tablespace as the columns that are indexed. Signature text indexes are well suited for building on columns that contain a combined total less than 200 MB. For larger text indexes, consider building an inverted file (IVF) text index. For detailed information on IVF text indexes, please refer to *Working with IVF Text Indexes*.

8.1 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.

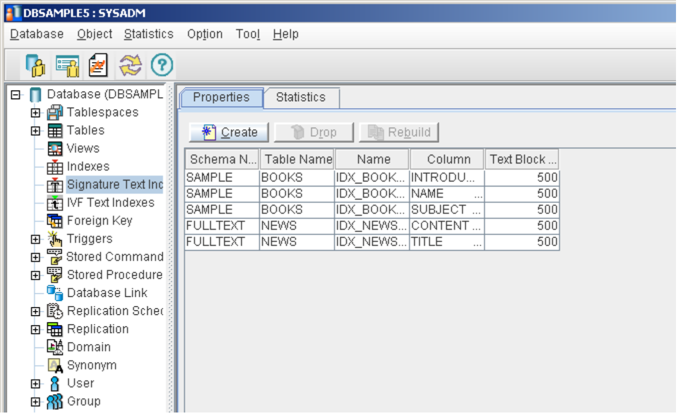
To ensure optimal performance for a signature text index, specify the total text size and scale. The total text size approximates the total number of megabytes of storage that used by the columns to be indexed. The total text size can range from 10 to 200 MB, and the default total text size is 32 MB.

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

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

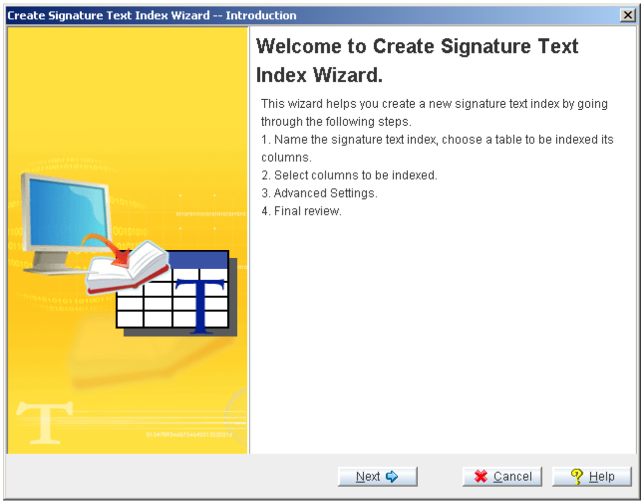
➲ Creating a Signature Text Index:

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



Schema N...	Table Name	Name	Column	Text Block ...
SAMPLE	BOOKS	IDX_BOOK...	INTRODU...	500
SAMPLE	BOOKS	IDX_BOOK...	NAME ...	500
SAMPLE	BOOKS	IDX_BOOK...	SUBJECT ...	500
FULLTEXT	NEWS	IDX_NEWS...	CONTENT ...	500
FULLTEXT	NEWS	IDX_NEWS...	TITLE ...	500

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



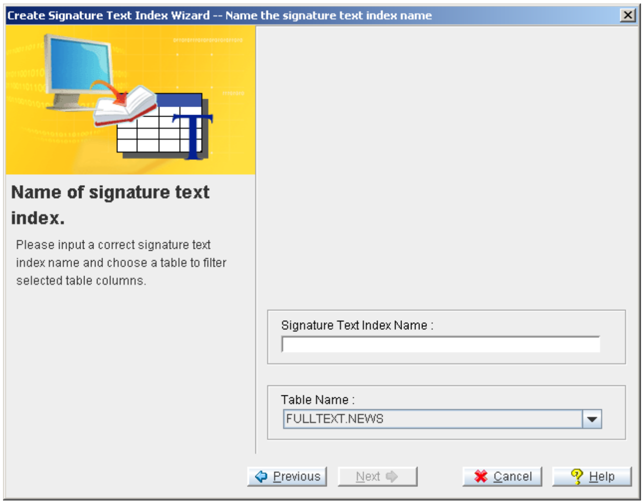
Create Signature Text Index Wizard -- Introduction

Welcome to Create Signature Text Index Wizard.

This wizard helps you create a new signature text index by going through the following steps.

1. Name the signature text index, choose a table to be indexed its columns.
2. Select columns to be indexed.
3. Advanced Settings.
4. Final review.

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



Create Signature Text Index Wizard -- Name the signature text index name

Name of signature text index.

Please input a correct signature text index name and choose a table to filter selected table columns.

Signature Text Index Name :

Table Name : FULLTEXT.NEWS

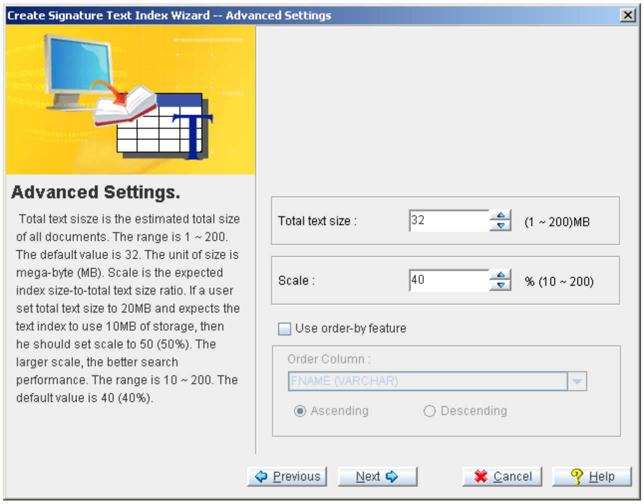
4. Select the name of the table from which the text index is to be created from the **Table Name** menu.

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 the column name or selecting the column and clicking the

Add button. The **Add all** button adds all columns to the index. All the columns selected for the creation of the text index are displayed in the list box to the right. Columns in the text index can be likewise removed by double-clicking the column name or selecting the column and clicking the **Remove** button.

7. Click **Next**. The **Advanced Settings** window appears.



Create Signature Text Index Wizard -- Advanced Settings

Advanced Settings.

Total text sisze is the estimated total size of all documents. The range is 1 ~ 200. The default value is 32. The unit of size is mega-byte (MB). Scale is the expected index size-to-total text size ratio. If a user set total text size to 20MB and expects the text index to use 10MB of storage, then he should set scale to 50 (50%). The larger scale, the better search performance. The range is 10 ~ 200. The default value is 40 (40%).

Total text size : 32 (1 ~ 200)MB

Scale : 40 % (10 ~ 200)

□ Use order-by feature

Order Column : FNAME (VARCHAR)

● Ascending ○ Descending

8. Changing the signature text index settings:

- To change the total text size, enter a value for **Total Text Size** between 1 and 200.
- To change the scale, enter a value for **Scale** between 10 and 200.

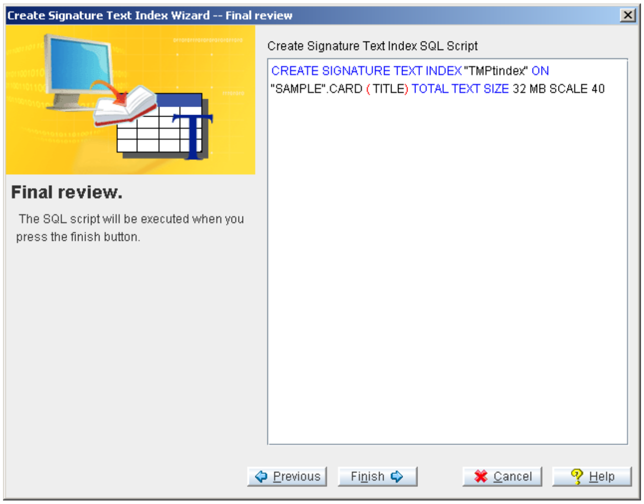
9. To display data as it relates to another column in the table, select the **Use order-by feature** check box.

   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.

10. Click **Next**. The **Final review** window appears.



Final review.

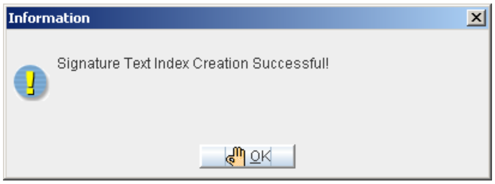
The SQL script will be executed when you press the finish button.

Create Signature Text Index SQL Script

```
CREATE SIGNATURE TEXT INDEX "TMPtindex" ON
"SAMPLE".CARD ( TITLE) TOTAL TEXT SIZE 32 MB SCALE 40
```

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

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



Signature Text Index Creation Successful!

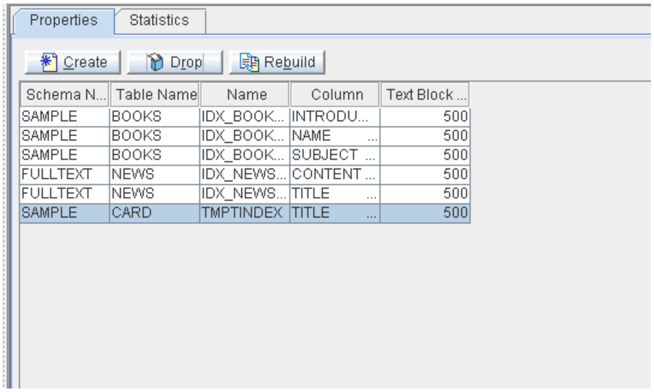
13. Click **OK**.

8.2 Dropping a Signature Text Index

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

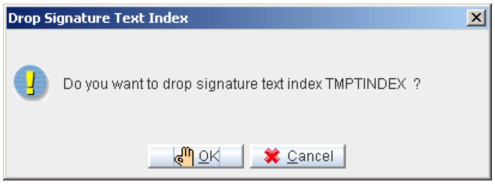
➲ Dropping a signature text index:

1. Click the **Signature Text Indexes** object in the tree. All the text indexes in the database are displayed in the **Properties** page.
2. Select a **Signature Text Index** to be dropped.



Schema N...	Table Name	Name	Column	Text Block ...
SAMPLE	BOOKS	IDX_BOOK...	INTRODU...	500
SAMPLE	BOOKS	IDX_BOOK...	NAME ...	500
SAMPLE	BOOKS	IDX_BOOK...	SUBJECT ...	500
FULLTEXT	NEWS	IDX_NEWS...	CONTENT ...	500
FULLTEXT	NEWS	IDX_NEWS...	TITLE ...	500
SAMPLE	CARD	TMPTINDEX	TITLE ...	500

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


Drop Signature Text Index

Do you want to drop signature text index TMPTINDEX ?

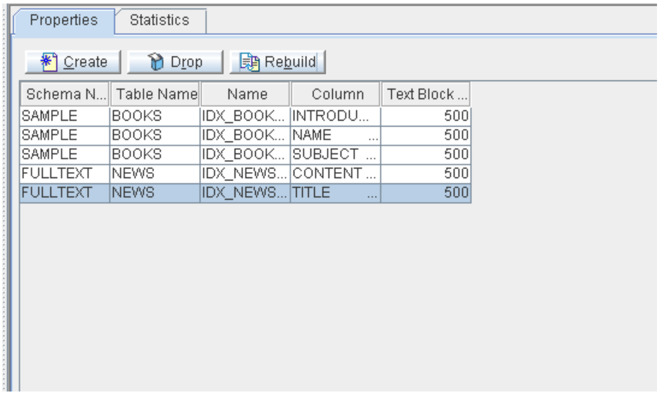
4. Click **OK**. The remaining signature text indexes in the database are displayed.

8.3 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.

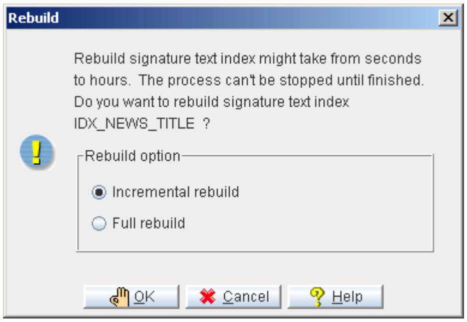
➲ Rebuilding a signature text index:

1. Click the **Signature Text Indexes** object in the tree. The **Properties** page appears.
2. Select a signature text index to rebuild.



Schema N...	Table Name	Name	Column	Text Block ...
SAMPLE	BOOKS	IDX_BOOK...	INTRODU...	500
SAMPLE	BOOKS	IDX_BOOK...	NAME ...	500
SAMPLE	BOOKS	IDX_BOOK...	SUBJECT ...	500
FULLTEXT	NEWS	IDX_NEWS...	CONTENT ...	500
FULLTEXT	NEWS	IDX_NEWS...	TITLE ...	500

3. Click **Rebuild**. The **Rebuild** dialog appears.



Rebuild signature text index might take from seconds to hours. The process can't be stopped until finished. Do you want to rebuild signature text index IDX_NEWS_TITLE ?

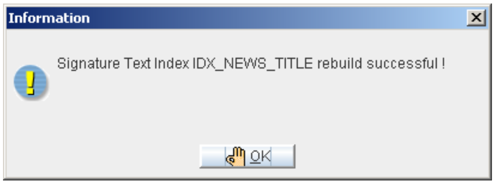
Rebuild option

● Incremental rebuild

○ Full rebuild

4. Select either the **Incremental rebuild** or **Full rebuild** radio button.

5. Click the **OK** button. A confirmation window appears as shown below.



Signature Text Index IDX_NEWS_TITLE rebuild successful !

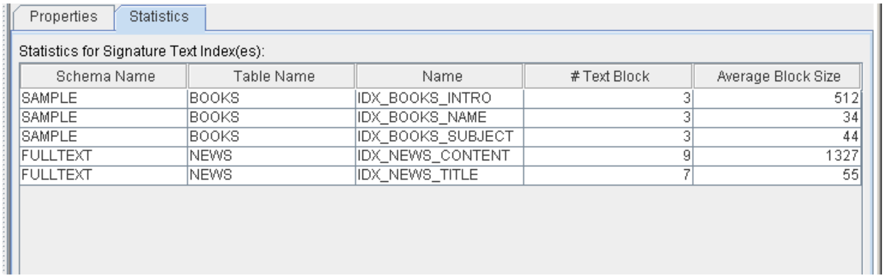
6. Click **OK**.

8.4 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.

➲ Viewing text index statistics:

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



Statistics for Signature Text Index(es):

Schema Name	Table Name	Name	# Text Block	Average Block Size
SAMPLE	BOOKS	IDX_BOOKS_INTRO	3	512
SAMPLE	BOOKS	IDX_BOOKS_NAME	3	34
SAMPLE	BOOKS	IDX_BOOKS_SUBJECT	3	44
FULLTEXT	NEWS	IDX_NEWS_CONTENT	9	1327
FULLTEXT	NEWS	IDX_NEWS_TITLE	7	55

9 Working with IVF Text Indexes

DBMaker builds inverted file (IVF) text indexes on files separate from the tablespace where the columns are stored. As a result, query performance is significantly better for larger text indexes. You need to create a logical path for storing the IVF text index files before creating an IVF text index. Specify the logical paths with the JConfiguration Tool or by modifying the DMConfig.ini file. First specify a path, and then 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 susceptible to fragmentation. Be sure to specify a logical path on a reserved partition and periodically defragment that 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.

9.1 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.

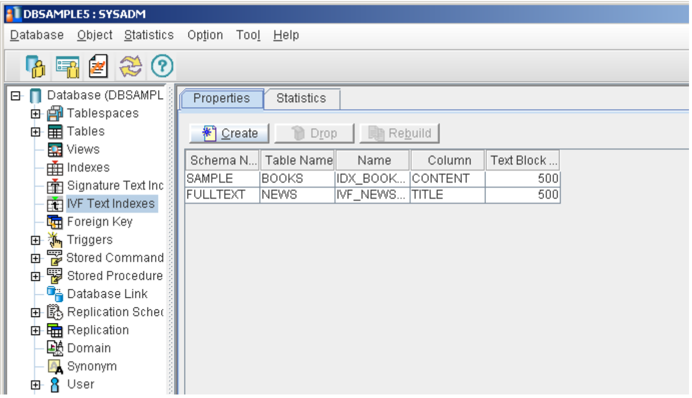
The three parameters used for IVF text indexes include the storage path, total text size, and scale. The storage path is the location of the IVF text index in the file system. You must specify a path in the configuration file before using an IVF text index. For more information on the storage path for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

The total text size approximates the total number of megabytes of storage that used by the columns on which the text index is to be built. The total text size ranges from 10 to 10,000 MB. The default total text size is ***32*** MB.

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

➲ Creating an IVF text index:

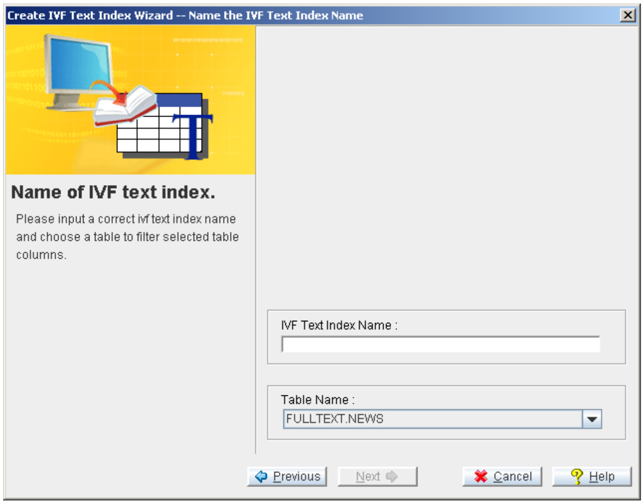
1. Click the **IVF Text Index** object in the tree. The **Properties** page appears.



Schema N...	Table Name	Name	Column	Text Block ...
SAMPLE	BOOKS	IDX_BOOK...	CONTENT	500
FULLTEXT	NEWS	IVF_NEWS...	TITLE	500

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

3. Click **Next**. The **Name the Text Index** window appears.


Create IVF Text Index Wizard -- Name the IVF Text Index Name

Name of IVF text index.

Please input a correct ivf text index name and choose a table to filter selected table columns.

IVF Text Index Name :

Table Name : FULLTEXT.NEWS

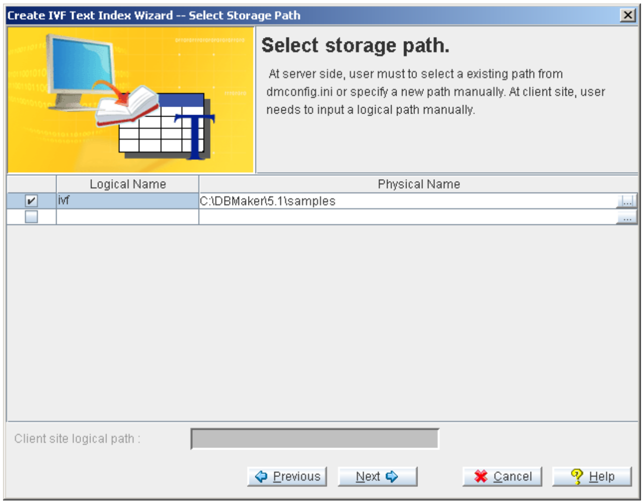
4. Select the name of the table from which the text index is to be created from the **Table Name** menu.

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 list box. Select the columns to use in the IVF text index by double-clicking the column name or selecting the column and clicking the **Add** button. The **Add all** button adds all columns to the index. All the columns selected for

the creation of the text index are displayed in the list box to the right. Columns in the text index can likewise be 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 appears.



Select storage path.

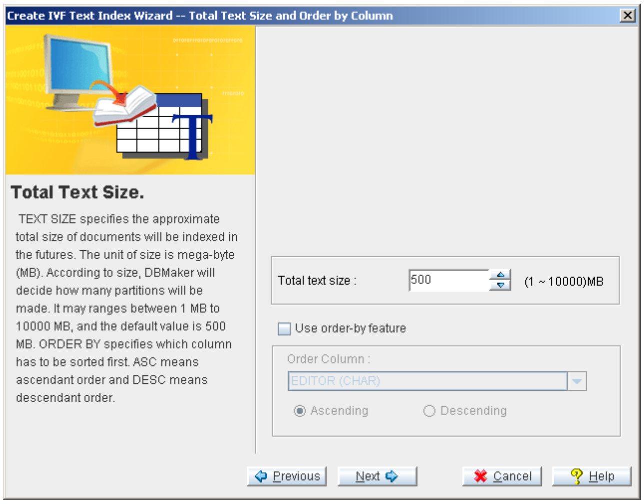
At server side, user must to select a existing path from dmconfig.ini or specify a new path manually. At client site, user needs to input a logical path manually.

	Logical Name	Physical Name
☑	ivf	C:\DBMaker\5.1\samples
□		

Client site logical path :

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 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 using the JConfiguration Tool, please refer to the *JConfiguration Tool Reference*.

9. Click the **Next** button. The **Total Text Size and Order by Column** window appears.



Create IVF Text Index Wizard -- Total Text Size and Order by Column

Total Text Size.

TEXT SIZE specifies the approximate total size of documents will be indexed in the futures. The unit of size is mega-byte (MB). According to size, DBMaker will decide how many partitions will be made. It may ranges between 1 MB to 10000 MB, and the default value is 500 MB. ORDER BY specifies which column has to be sorted first. ASC means ascendant order and DESC means descendant order.

Total text size : 500 (1 ~ 10000)MB

□ Use order-by feature

Order Column : EDITOR (CHAR)

● Ascending ○ Descending

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.

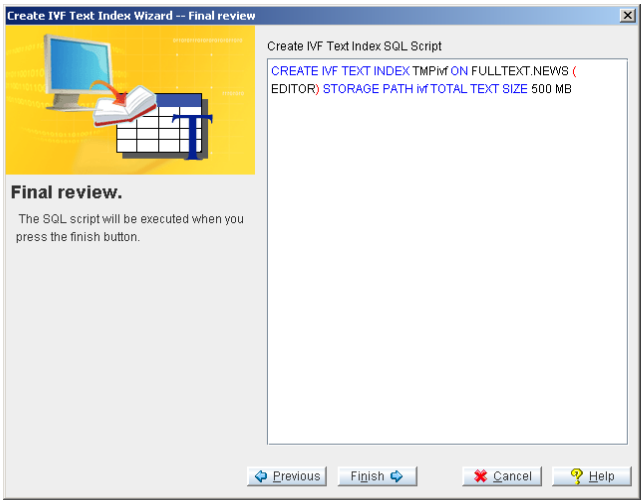
11. To present data as it relates to another column in the table select the **Use order-by feature** check box.

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 appears.



Final review.

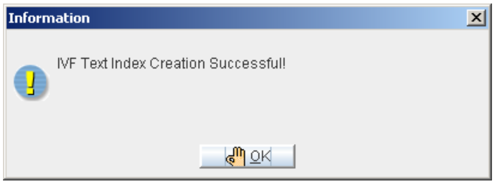
The SQL script will be executed when you press the finish button.

Create IVF Text Index SQL Script

```
CREATE IVF TEXT INDEX TMPivf ON FULLTEXT.NEWS (
EDITOR) STORAGE PATH ivf TOTAL TEXT SIZE 500 MB
```

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

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



IVF Text Index Creation Successful!

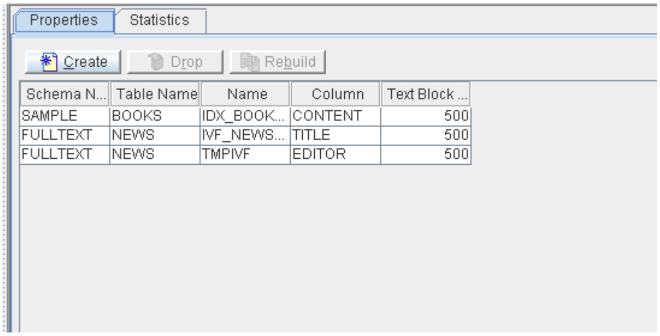
15. Click the **OK** button.

9.2 Dropping an IVF Text Index

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

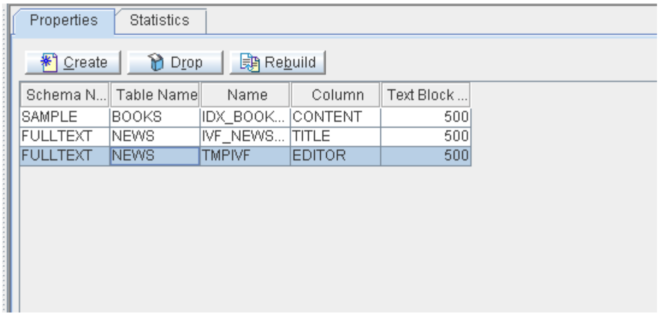
➲ Dropping an IVF text index:

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



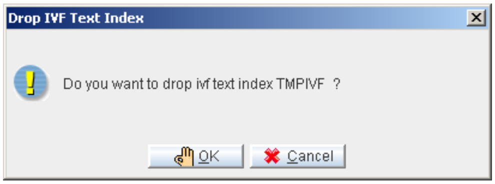
Schema N...	Table Name	Name	Column	Text Block ...
SAMPLE	BOOKS	IDX_BOOK...	CONTENT	500
FULLTEXT	NEWS	IVF_NEWS...	TITLE	500
FULLTEXT	NEWS	TMPIVF	EDITOR	500

2. Select the **IVF Text Index** to be dropped.



Schema N...	Table Name	Name	Column	Text Block ...
SAMPLE	BOOKS	IDX_BOOK...	CONTENT	500
FULLTEXT	NEWS	IVF_NEWS...	TITLE	500
FULLTEXT	NEWS	TMPIVF	EDITOR	500

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


Drop IVF Text Index

Do you want to drop ivf text index TMPIVF ?

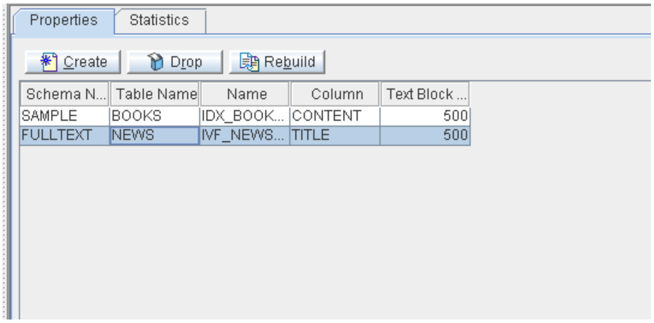
4. Click **OK**. The remaining IVF text indexes in the database are displayed.

9.3 Rebuilding an IVF Text Index

When 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.

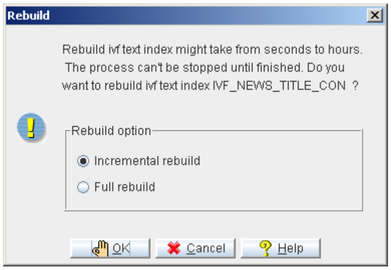
➲ Rebuilding an IVF text index:

1. Click the **Text Indexes** object in the tree. The **Properties** page appears.
2. Select an IVF text index to rebuild.



Schema N...	Table Name	Name	Column	Text Block ...
SAMPLE	BOOKS	IDX_BOOK...	CONTENT	500
FULLTEXT	NEWS	IVF_NEWS...	TITLE	500

3. Click **Rebuild**. The **Rebuild** dialog appears.



Rebuild ivf text index might take from seconds to hours. The process can't be stopped until finished. Do you want to rebuild ivf text index IVF_NEWS_TITLE_CON ?

Rebuild option

● Incremental rebuild

○ Full rebuild

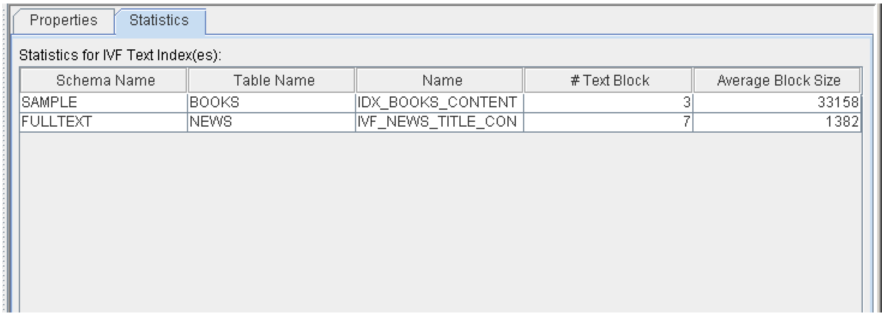
4. Click either the **Incremental** rebuild or **Full rebuild** radio button.
5. Click the **OK** button. A confirmation message appears.
6. Click **OK**.

9.4 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.

➲ Viewing text index statistics:

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



Statistics for IVF Text Index(es):

Schema Name	Table Name	Name	# Text Block	Average Block Size
SAMPLE	BOOKS	IDX_BOOKS_CONTENT	3	33158
FULLTEXT	NEWS	IVF_NEWS_TITLE_CON	7	1382

10 Using Foreign Keys

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

This section illustrates creating foreign keys for tables and then dropping foreign keys that are no longer required.

10.1 Creating a Foreign Key

A foreign key refers to another table by specifying the referencing and referenced columns. Both the referencing and referenced columns should map 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 primary key or unique index.

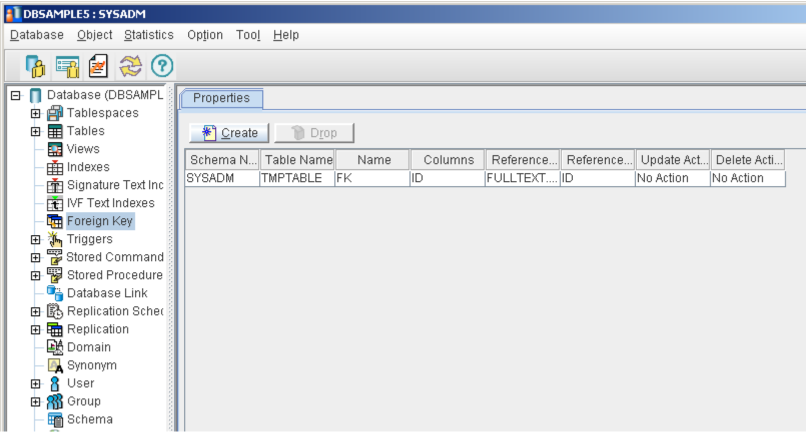
Adding a Foreign Key

Creating a foreign key comprises several phases. First, includes selecting a table that contains referencing columns, selecting columns, and naming a foreign key.

The 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.

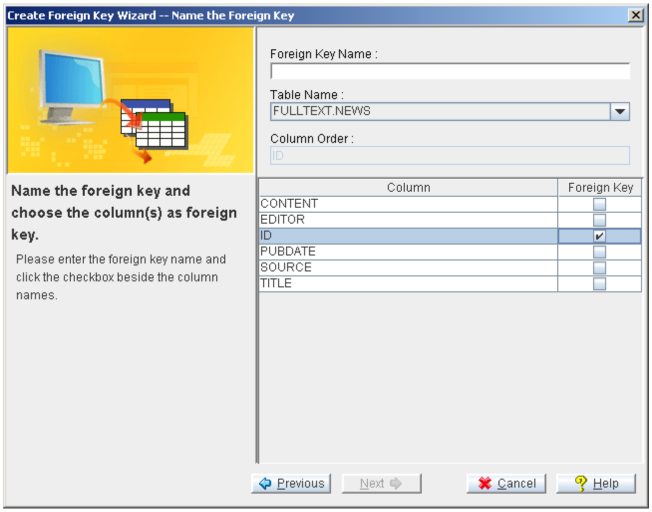
➲ Adding a foreign key:

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



Schema N...	Table Name	Name	Columns	Reference...	Reference...	Update Act...	Delete Acti...
SYSADM	TMPTABLE	FK	ID	FULLTEXT....	ID	No Action	No Action

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



Name the foreign key and choose the column(s) as foreign key.

Please enter the foreign key name and click the checkbox beside the column names.

Foreign Key Name :

Table Name : FULLTEXT.NEWS

Column Order : ID

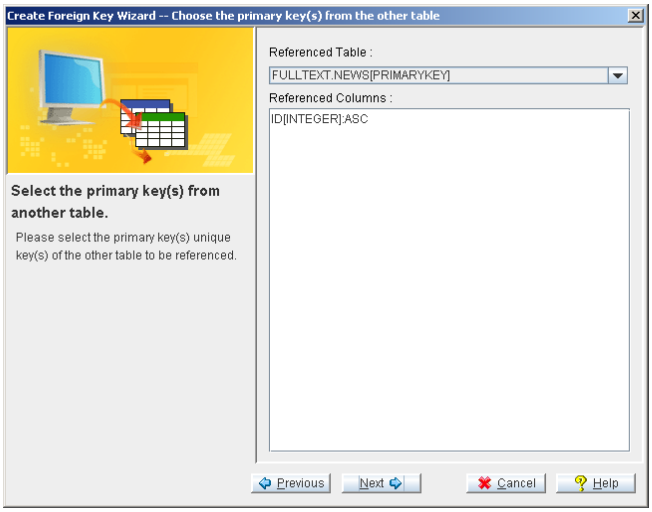
Column	Foreign Key
CONTENT	☐
EDITOR	☐
ID	☑
PUBDATE	☐
SOURCE	☐
TITLE	☐

3. Enter a name for the foreign key in the **Foreign Key Name** text field.

4. Select the referencing table name.

5. Click checkboxes beside the column names to specify the columns included in the foreign key.

6. Click the **Next** button. The **Choose the primary key(s) from the other table** window of the **Create Foreign Key Wizard** appears.



Create Foreign Key Wizard -- Choose the primary key(s) from the other table

Select the primary key(s) from another table.

Please select the primary key(s) unique key(s) of the other table to be referenced.

Referenced Table : FULLTEXT.NEWS[PRIMARYKEY]

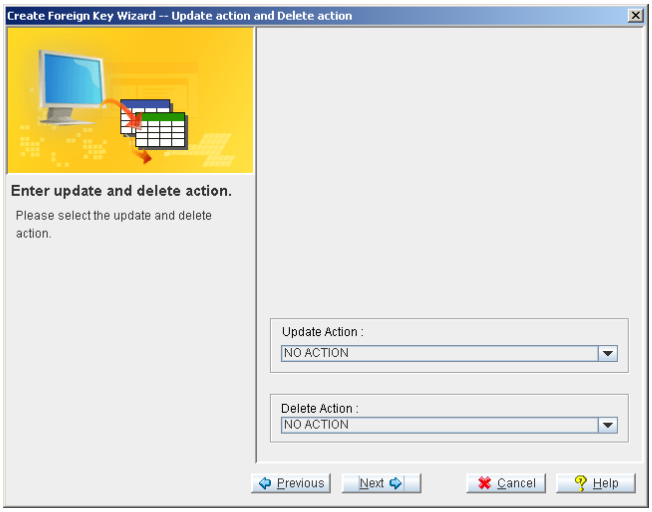
Referenced Columns : ID[INTEGER]:ASC

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

NOTE *Only tables containing primary keys or unique indexes matching the schema of the foreign key are displayed.*

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

9. Click the **Next** button. The **Update action and Delete action** window appears.



Create Foreign Key Wizard -- Update action and Delete action

Enter update and delete action.

Please select the update and delete action.

Update Action : NO ACTION

Delete Action : NO ACTION

Setting Foreign Key Options

Foreign key and their options can be specified. 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 corresponding records in the primary key or unique index column are updated or deleted.
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 corresponding records in the primary key or unique index column are updated or deleted.

FOREIGN KEY OPTION	DESCRIPTION
No Action	There is no action on the foreign key column if the corresponding records in the primary key or unique index column are updated or deleted.

Table 10-1Foreign Key Options

➲ Setting 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.

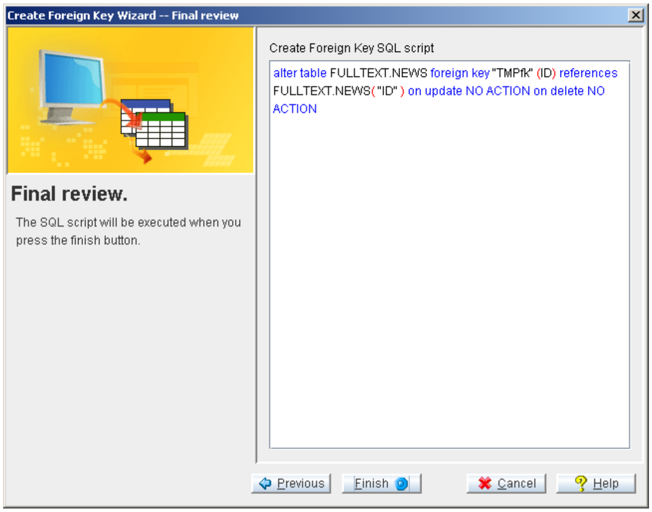
2. Choose one of the following foreign key settings from the **Delete Action** menu:

   - **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 the **Next** button. The window shown below appears. The SQL command generated by the wizard can be edited as needed.



The SQL script will be executed when you press the finish button.

Create Foreign Key SQL script

```
alter table FULLTEXT.NEWS foreign key "TMPfk" (ID) references
FULLTEXT.NEWS( "ID" ) on update NO ACTION on delete NO
ACTION
```

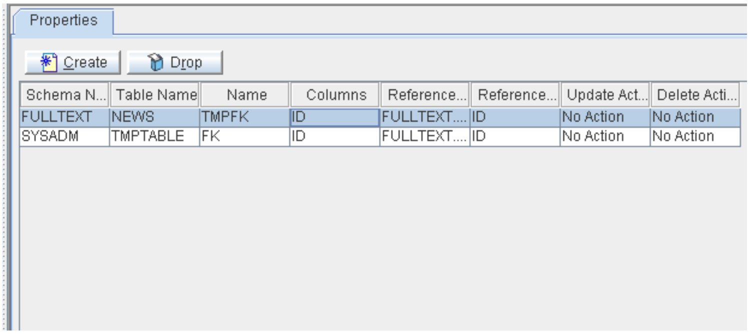
4. Click the **Finish** button. The **Properties** page reappeard with the new foreign key in the list.

10.2 Dropping a Foreign key

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

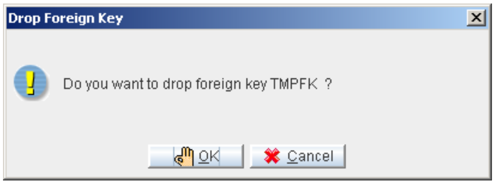
➲ Dropping a foreign key:

1. Select the **Foreign Key** object from the tree. The **Properties** page appears.
2. Click the foreign key to drop.



Schema N...	Table Name	Name	Columns	Reference...	Reference...	Update Act...	Delete Acti...
FULLTEXT	NEWS	TMPFK	ID	FULLTEXT....	ID	No Action	No Action
SYSADM	TMPTABLE	FK	ID	FULLTEXT....	ID	No Action	No Action

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



Do you want to drop foreign key TMPFK ?

4. Click the **OK button**. The remaining foreign keys in the database are displayed.

11 Using Triggers

Triggers are predefined commands that automatically execute in response to specific events, regardless of which user or application program causes the events. Triggers provide dtabase customization beyond what is possible with standard SQL commands. Triggers can:

- Implement business rules
- Create database activity audit trails
- Generate additional value from existing data
- Replicate data across multiple tables
- Perform security authorization procedures
- Control data integrity
- Define unconventional integrity constraints

This section illustrates creating triggers with various trigger options and then dropping triggers from a database.

11.1 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 by the trigger definition. Creating a trigger involves the following set of actions:

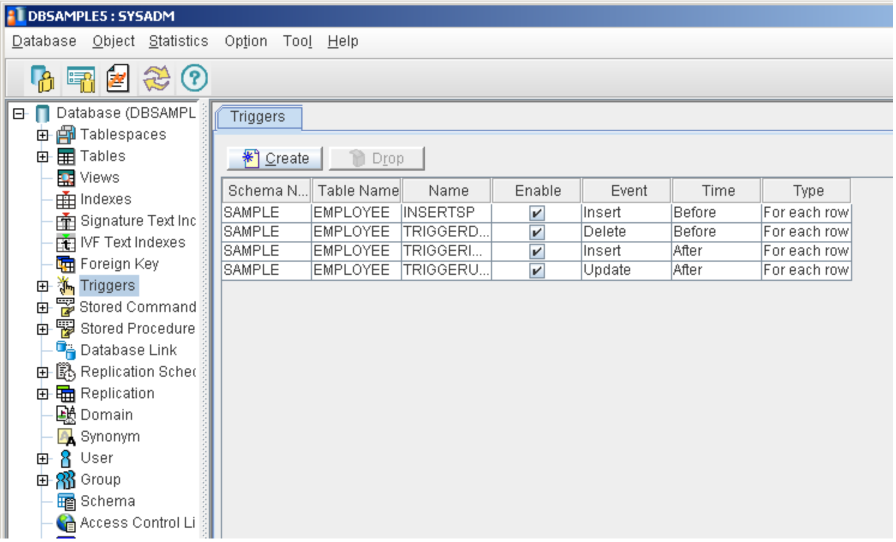
- Assigning a trigger name and table
- Specifying a trigger action time
- Defining a referencing clause
- Entering a WHEN condition clause
- Entering SQL statements for the trigger action

Assigning a Trigger Name and Table

A trigger name must be assigned and a table selected when creating a new trigger.

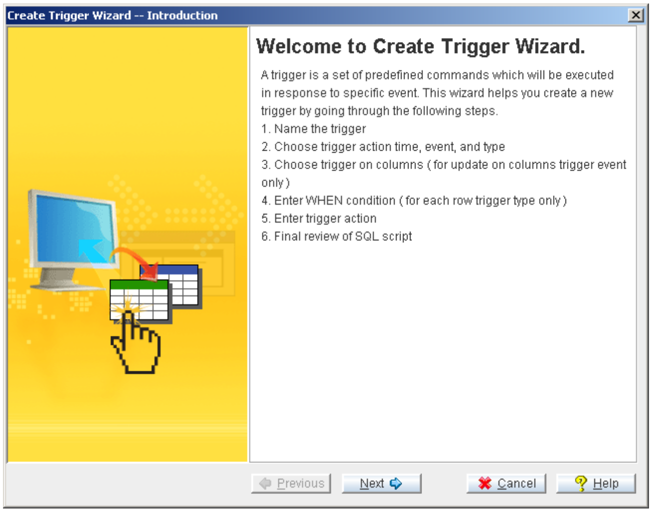
➲ Specifying a trigger name and table:

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



Schema N...	Table Name	Name	Enable	Event	Time	Type
SAMPLE	EMPLOYEE	INSERTSP	☑	Insert	Before	For each row
SAMPLE	EMPLOYEE	TRIGGERD...	☑	Delete	Before	For each row
SAMPLE	EMPLOYEE	TRIGGERI...	☑	Insert	After	For each row
SAMPLE	EMPLOYEE	TRIGGERU...	☑	Update	After	For each row

2. Click the **Create** button. The **Introduction** window of the **Create Trigger Wizard** appear.



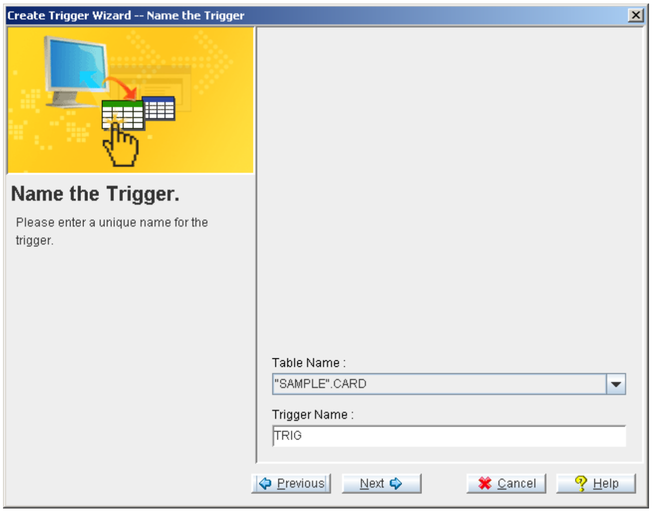
Welcome to Create Trigger Wizard.

A trigger is a set of predefined commands which will be executed in response to specific event. This wizard helps you create a new trigger by going through the following steps.

1. Name the trigger
2. Choose trigger action time, event, and type
3. Choose trigger on columns (for update on columns trigger event only)
4. Enter WHEN condition (for each row trigger type only)
5. Enter trigger action
6. Final review of SQL script

3. Click the **Next** button. The **Name the Trigger** window appears.

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.



Create Trigger Wizard -- Name the Trigger

Name the Trigger.

Please enter a unique name for the trigger.

Table Name : "SAMPLE".CARD

Trigger Name : TRIG

6. Click the **Next** button. The **Choose the Trigger Action Time** window appears.

Specifying Trigger Action Settings

After identified the trigger's name and table, specify the type of actions the trigger will perform. The settings are as follows:

Action Time: The trigger action time specifies whether a trigger should fire (i.e., begin executing) before or after the SQL statement that activated the trigger. The BEFORE and AFTER keywords specify the trigger's action time. The BEFORE keyword indicates the trigger action fires before the trigger statement. The AFTER keyword indicates the trigger action fires after the trigger statement. A trigger cannot have more than a single trigger time.

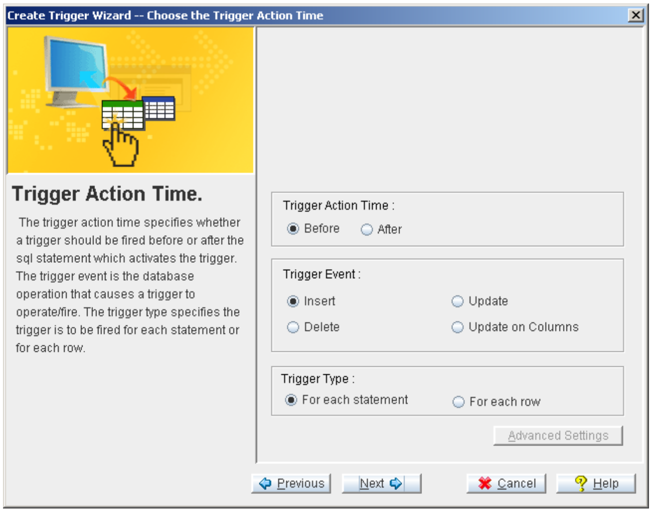
Trigger Event: The trigger event is the database operation that causes a trigger to execute (i.e., fire). The trigger event may be an INSERT, UPDATE, or DELETE statement that operates on the trigger's table. Only a single trigger event can be specified for each trigger.

Trigger Type: Specifies how many times the trigger fires for each trigger event. There are two types of triggers: row triggers and statement triggers. The FOR EACH ROW keyword specifies a row trigger. Row triggers fire a trigger action once for each row modified by the trigger event. The FOR EACH STATEMENT keyword specifies a statement trigger. Statement triggers firea trigger action once for each trigger event.

Referencing Clause: The advanced setting tab is highlighted when the selected Trigger type is FOR EACH ROW. 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.

➲ pecifying trigger action settings:

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



Create Trigger Wizard -- Choose the Trigger Action Time

Trigger Action Time.

The trigger action time specifies whether a trigger should be fired before or after the sql statement which activates the trigger. The trigger event is the database operation that causes a trigger to operate/fire. The trigger type specifies the trigger is to be fired for each statement or for each row.

Trigger Action Time :
● Before ○ After

Trigger Event :
● Insert ○ Update
○ Delete ○ Update on Columns

Trigger Type :
● For each statement ○ For each row

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

- Click the **After** radio button to fire the trigger after the SQL statement.
- Click the **Before** radio button to fire the trigger before the SQL statement.

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.

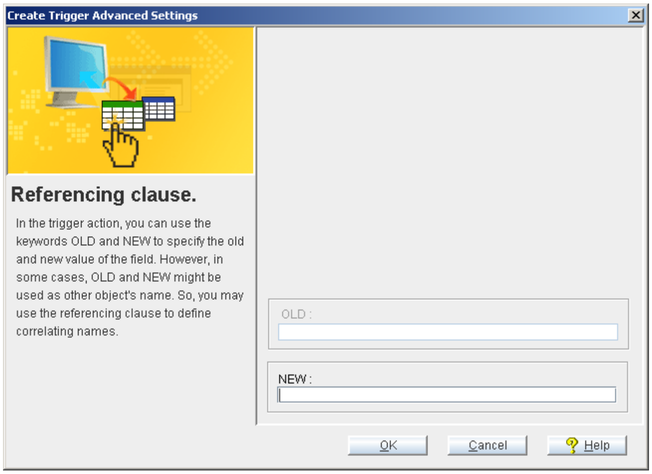
5. If the **For each statement** option button is selected, clicking **Next** opens the **Specify a WHEN condition** clause window. If the **For each row** option button is selected, clicking **Next** opens the **Create Trigger Advanced Settings** window.

Indicating the Referencing Clause

After defining the trigger options, specify the referencing clause of the trigger. These options are required when FOR EACH ROW is the trigger type. The REFERENCING clause defines correlation names for the old and new values of a column. Use this primarily when you cannot use the default OLD and NEW names because a column has the same name.

➲ Defining the referencing clause:

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



Create Trigger Advanced Settings

Referencing clause.

In the trigger action, you can use the keywords OLD and NEW to specify the old and new value of the field. However, in some cases, OLD and NEW might be used as other object's name. So, you may use the referencing clause to define correlating names.

OLD :

NEW :

2. When you create a **For each row** type trigger, enter a substitute name to refer to the old value in the **OLD** field and enter a substitute name to refer to the new value in the **NEW** field.

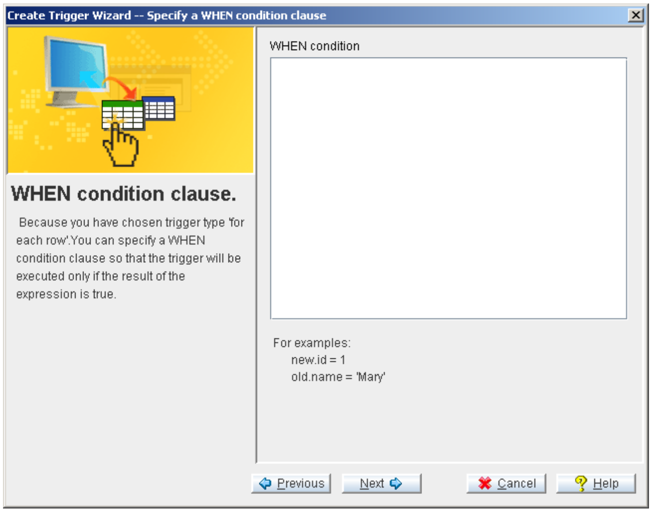
3. Click the **Next** button. The **Specify a WHEN condition clause** window appears.

Entering the WHEN Condition Clause

Once the settings for the trigger action are complete, specify a WHEN clause to place constraints on the actions that cause the trigger to fire. The WHEN statement must follow proper SQL syntax rules. Please see *Constraint Syntax*, near the end of Chapter 4 for more information.

- Specifying 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.



Create Trigger Wizard -- Specify a WHEN condition clause

WHEN condition clause.

Because you have chosen trigger type 'for each row'.You can specify a WHEN condition clause so that the trigger will be executed only if the result of the expression is true.

WHEN condition

For examples:
new.id = 1
old.name = 'Mary'

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

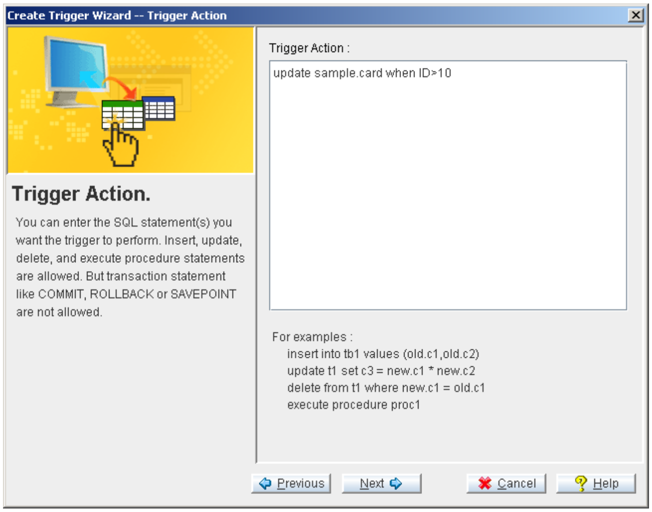
Entering SQL statements for the trigger action

After defining the trigger event and its constraints, 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 fires.

- Specifying SQL statements for the trigger action:

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

2. Enter the SQL statements that the trigger will execute.



Create Trigger Wizard -- Trigger Action

Trigger Action.

You can enter the SQL statement(s) you want the trigger to perform. Insert, update, delete, and execute procedure statements are allowed. But transaction statement like COMMIT, ROLLBACK or SAVEPOINT are not allowed.

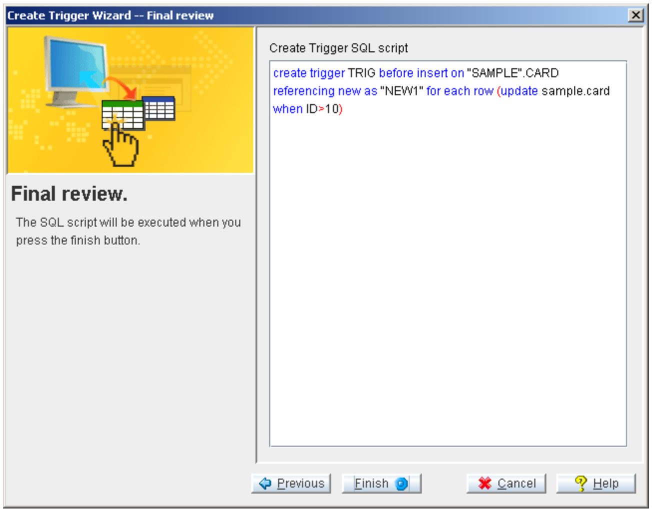
Trigger Action :

```
update sample.card when ID>10
```

For examples :

```
    insert into tb1 values (old.c1,old.c2)
    update t1 set c3 = new.c1 * new.c2
    delete from t1 where new.c1 = old.c1
    execute procedure proc1
```

3. Click the **Next** button. The **Final review** window appears.



Create Trigger Wizard -- Final review

Final review.

The SQL script will be executed when you press the finish button.

Create Trigger SQL script

```
create trigger TRIG before insert on "SAMPLE".CARD
referencing new as "NEW1" for each row (update sample.card
when ID>10)
```

4. Review the final SQL script and make edits as needed. Click the **Finish** button when you are satisfied with the script. A confirmation window appears.

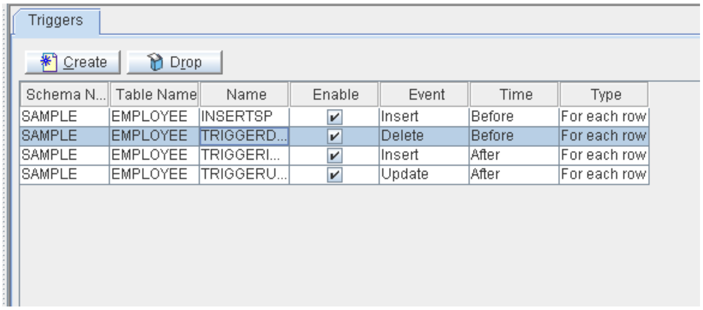
5. Click the **OK** button.

11.2 Dropping a Trigger

When a table's schema is altered and then a trigger on that table is fired, DBMaker executes the trigger according to the new table definition. However, if the specified column in a triggering event or action was dropped, the trigger execution fails, as does the triggering statement. 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 is also dropped.

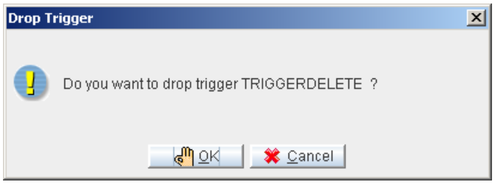
➲ Deleting a trigger:

1. Click the object **Trigger** in the tree. All the triggers in the database are displayed as shown.
2. Select a trigger.



Schema N...	Table Name	Name	Enable	Event	Time	Type
SAMPLE	EMPLOYEE	INSERTSP	☑	Insert	Before	For each row
SAMPLE	EMPLOYEE	TRIGGERD...	☑	Delete	Before	For each row
SAMPLE	EMPLOYEE	TRIGGERI...	☑	Insert	After	For each row
SAMPLE	EMPLOYEE	TRIGGERU...	☑	Update	After	For each row

3. Click the **Drop** button. A confirmation window appears.



Do you want to drop trigger TRIGGERDELETE ?

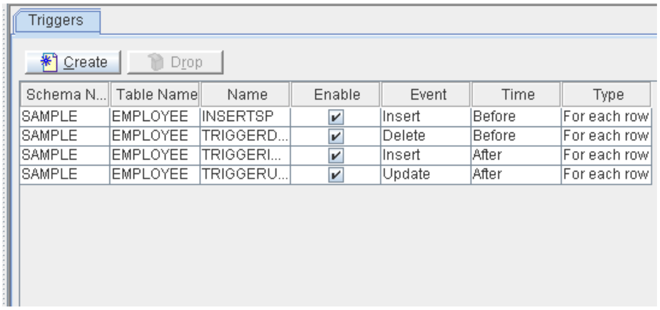
4. Click the **OK** button.

11.3 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, and trigger type.

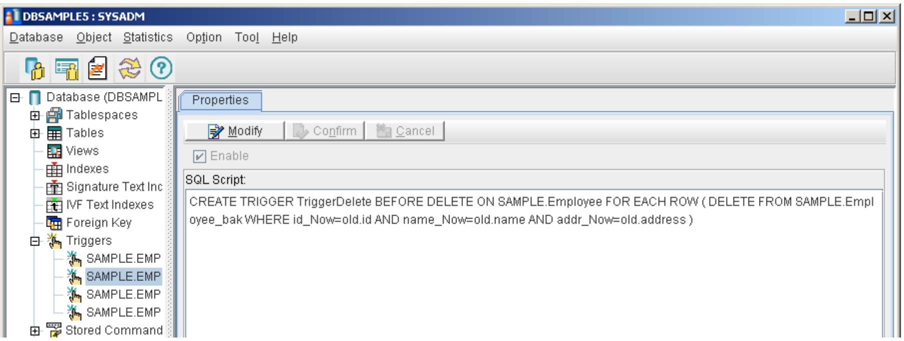
➲ Modifing a trigger:

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



Schema N...	Table Name	Name	Enable	Event	Time	Type
SAMPLE	EMPLOYEE	INSERTSP	☑	Insert	Before	For each row
SAMPLE	EMPLOYEE	TRIGGERD...	☑	Delete	Before	For each row
SAMPLE	EMPLOYEE	TRIGGERI...	☑	Insert	After	For each row
SAMPLE	EMPLOYEE	TRIGGERU...	☑	Update	After	For each row

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 - appears.

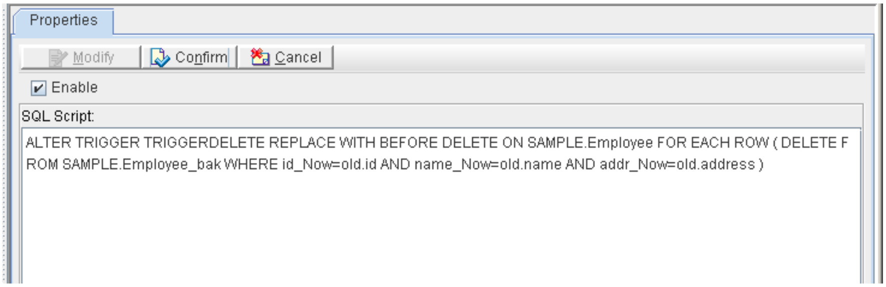


☑ Enable

SQL Script:

```
CREATE TRIGGER TriggerDelete BEFORE DELETE ON SAMPLE.Employee FOR EACH ROW ( DELETE FROM SAMPLE.Empl
oyee_bak WHERE id_Now=old.id AND name_Now=old.name AND addr_Now=old.address )
```

3. Click the **Modify** button.



☑ Enable

SQL Script:

```
ALTER TRIGGER TRIGGERDELETE REPLACE WITH BEFORE DELETE ON SAMPLE.Employee FOR EACH ROW ( DELETE F
ROM SAMPLE.Employee_bak WHERE id_Now=old.id AND name_Now=old.name AND addr_Now=old.address )
```

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 appears.

12 Working with Stored Commands

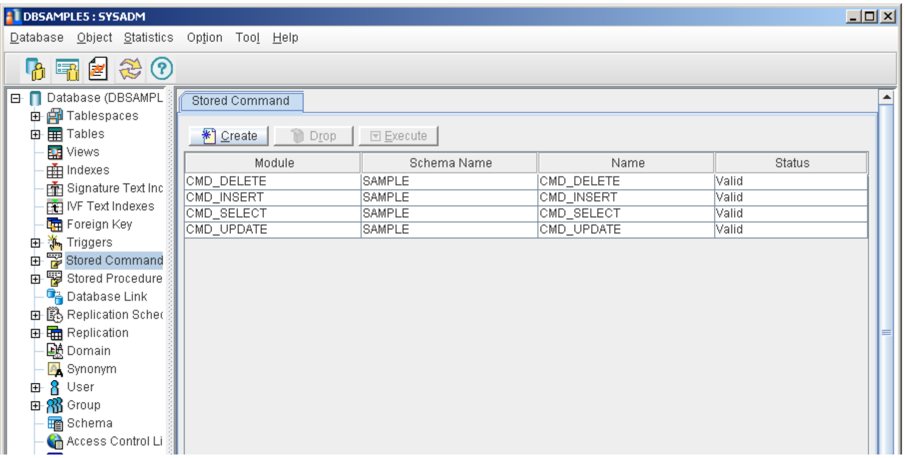
A stored command is a compiled SQL DML statement that is stored in the database. Stored commands are precompiled in an executable form. They can execute the same commands without repeated compiling and optimizing. It is possible to create a stored command for any frequently used SQL statement.

12.1 Creating a Stored Command

Create stored commands to achieve improved performance of frequently used statements.

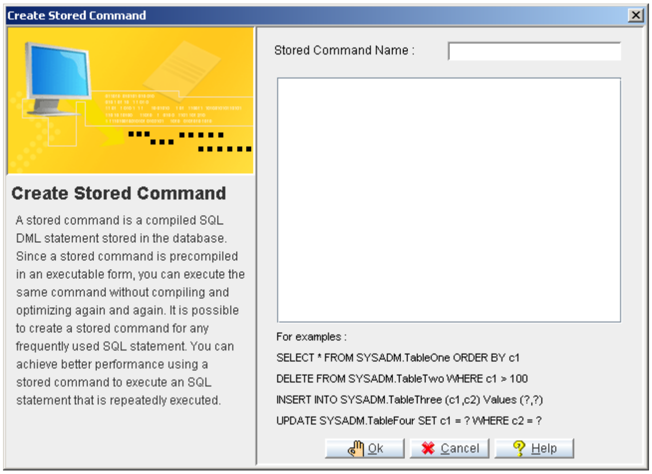
➲Creating a stored command:

1. Click the object **Stored Command** in the tree. The **`Stored Command`** window appears.



Module	Schema Name	Name	Status
CMD_DELETE	SAMPLE	CMD_DELETE	Valid
CMD_INSERT	SAMPLE	CMD_INSERT	Valid
CMD_SELECT	SAMPLE	CMD_SELECT	Valid
CMD_UPDATE	SAMPLE	CMD_UPDATE	Valid

2. Click the **`Create`** `button`. The **Create Stored Command** window appears.



Create Stored Command

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 compiling and optimizing again and again. It is possible to create a stored command for any frequently used SQL statement. You can achieve better performance using a stored command to execute an SQL statement that is repeatedly executed.

For examples :

```
SELECT * FROM SYSADM.TableOne ORDER BY c1
DELETE FROM SYSADM.TableTwo WHERE c1 > 100
INSERT INTO SYSADM.TableThree (c1,c2) Values (?,?)
UPDATE SYSADM.TableFour SET c1 = ? WHERE c2 = ?
```

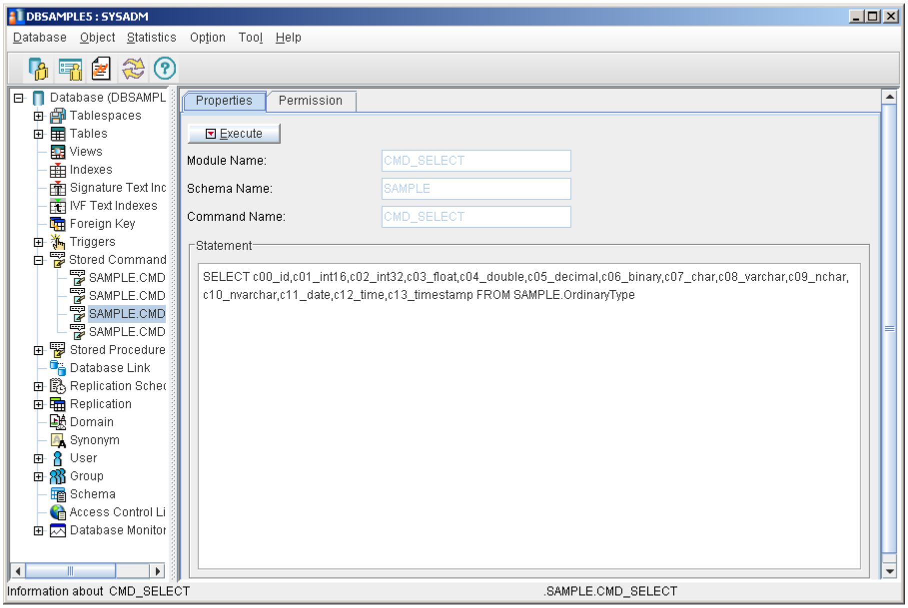
3. Enter a name for the stored command into the **Stored Command Name** field.
4. Enter the SQL command.
5. Click the **OK** button. The newly created stored command appears with the database’s existing stored commands.

12.2 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 prompts for its value during execution of the stored command. When you execute a stored command, the number of input parameters should be equal to number of input parameters in the stored command.

➲ Executing a stored command:

1. Open the object **Stored Command** in the tree and then select the stored command to execute. The stored command **Properties** pageappears.



Module Name: CMD_SELECT

Schema Name: SAMPLE

Command Name: CMD_SELECT

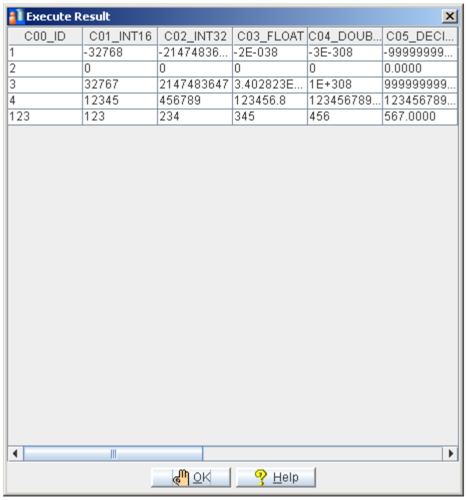
Statement

```
SELECT c00_id,c01_int16,c02_int32,c03_float,c04_double,c05_decimal,c06_binary,c07_char,c08_varchar,c09_nchar,
c10_nvarchar,c11_date,c12_time,c13_timestamp FROM SAMPLE.OrdinaryType
```

Information about CMD_SELECT .SAMPLE.CMD_SELECT

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

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



Execute Result

C00_ID	C01_INT16	C02_INT32	C03_FLOAT	C04_DOUB...	C05_DECI...
1	-32768	-21474836...	-2E-038	-3E-308	-99999999...
2	0	0	0	0	0.0000
3	32767	2147483647	3.402823E...	1E+308	999999999...
4	12345	456789	123456.8	123456789...	123456789...
123	123	234	345	456	567.0000

3. Click **OK** to dismiss the result window.

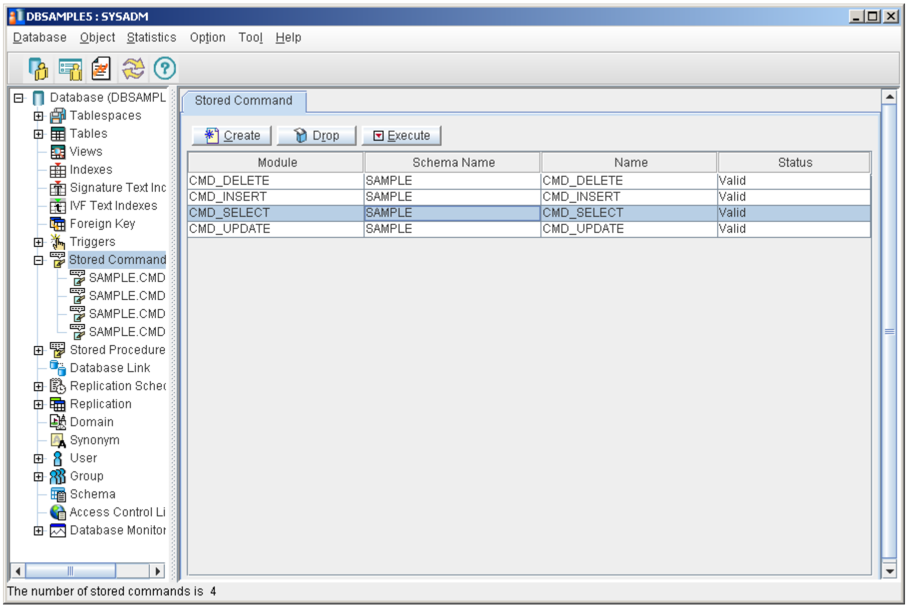
12.3 Dropping a Stored Command

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

➲ Dropping a stored command:

1. Click the object **Stored Command** in the tree. The **Stored Command** page appears.

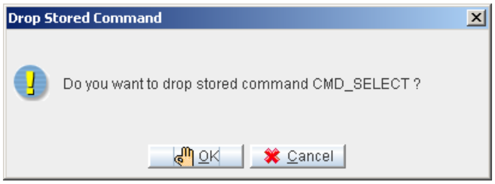
2. Click the stored command to drop. It will highlight blue.



Module	Schema Name	Name	Status
CMD_DELETE	SAMPLE	CMD_DELETE	Valid
CMD_INSERT	SAMPLE	CMD_INSERT	Valid
CMD_SELECT	SAMPLE	CMD_SELECT	Valid
CMD_UPDATE	SAMPLE	CMD_UPDATE	Valid

The number of stored commands is 4

3. Click the **Drop** button. The **Drop Stored Command** dialog box appears.



Do you want to drop stored command CMD_SELECT ?

4. Click the **OK** button. The remaining stored commands in the database are displayed.

12.4 Granting Stored Command Privileges

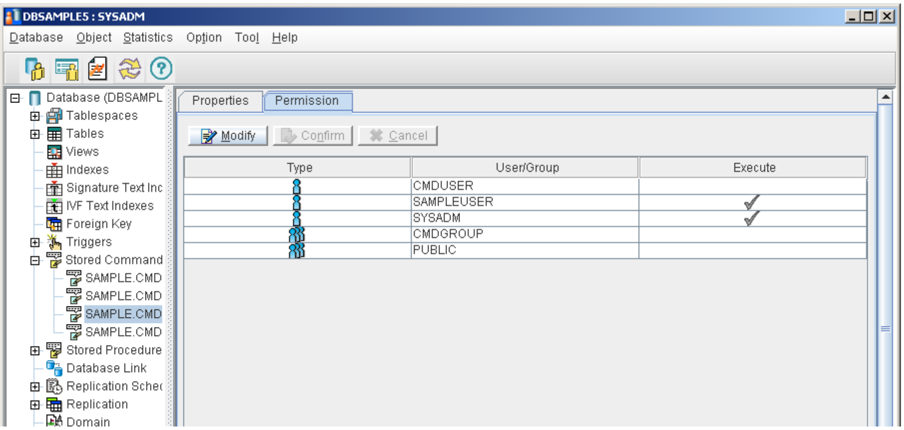
By default, owners of a stored command and users with DBA authority or higher, have execute privilege for the stored command. The owner or a user with DBA authority or higher may grant execute privileges to other users. The execute privilege can also be revoked from other users with resource authority 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.

➲ Granting and revoking privileges on a stored command:

1. Open the **Stored Command** node and then select a stored command. The stored command **Properties** page appears.

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

2. Click the **Permission** tab. The stored command **Permission** page appears.



Type	User/Group	Execute
👤	CMDUSER	
👤	SAMPLEUSER	✓
👤	SYSADM	✓
👥	CMDGROUP	
👥	PUBLIC	

3. Select a User/Group for granting or revoking privileges.
4. Click the **Modify** button.

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

13 Working with Stored Procedures

A stored procedure is a special kind of user-defined function. This increases the performance of frequently repeated tasks by bypassing repeated SQL compilation and optimization. Stored procedures are executed as commands 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 database access.

Stored procedures are stored as executable objects in the database and are available to every application running on the database. Application development time decreases because stored procedures are available to any application.

13.1 Creating a Stored Procedure

DBMaker provides three languages for creating stored procedures: ESQL/C, Java and SQL.

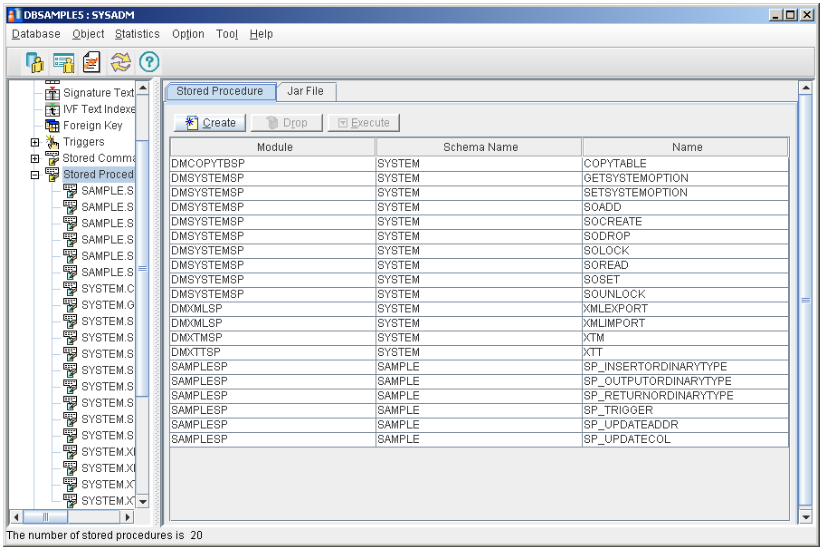
An ESQL/C stored procedure can perform any function a C application can perform, including calling other C functions and system calls. Writing ESQL/C stored procedures requires a C compiler. C compilers are usually already available in UNIX environments. Simply include the compiler in your path. Install Visual C++ version 4.2 or greater when using Windows.

Given Java's popularity today, it is certainly possible that members of a development team are more proficient in Java than ESQL. To create Java stored procedures, the value of the DB_SPDIR in the dmconfig.ini must define and next copy the physical jar file to the DB_SPDIR/jar/username/directory. Finally, execute the add jar file SQL statement to register the jar file with the database. For details about adding jar file SQL statements, please refer to chapter 13.

Creating stored procedures with ESQL and Java is inefficient. Stored procedures made directly from SQL statements make very efficient SQL stored procedures. An SQL stored procedure is a set of SQL statements that can be stored in a server. Once created, the user can refer to the SQL stored procedure rather than reissuing many individual statements. For more information on SQL stored procedures, please see the *DBMaker SQL Stored Procedure User's Guide*.

➲ Creating a stored procedure:

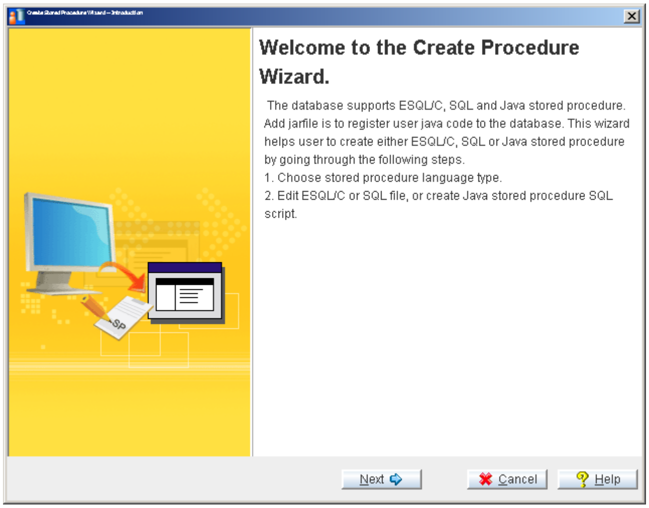
1. Click the object **Stored Procedure** in the tree. The **Stored Procedures** page appears.



Module	Schema Name	Name
DMCOPYTBSP	SYSTEM	COPYTABLE
DMSYSTEMSP	SYSTEM	GETSYSTEMOPTION
DMSYSTEMSP	SYSTEM	SETSYSTEMOPTION
DMSYSTEMSP	SYSTEM	SOADD
DMSYSTEMSP	SYSTEM	SOCREATE
DMSYSTEMSP	SYSTEM	SODROP
DMSYSTEMSP	SYSTEM	SOLOCK
DMSYSTEMSP	SYSTEM	SOREAD
DMSYSTEMSP	SYSTEM	SOSET
DMSYSTEMSP	SYSTEM	SOUNLOCK
DMXMLSP	SYSTEM	XMLEXPORT
DMXMLSP	SYSTEM	XMLIMPORT
DMXTMSP	SYSTEM	XTM
DMXTTSP	SYSTEM	XTT
SAMPLESP	SAMPLE	SP_INSERTORDINARYTYPE
SAMPLESP	SAMPLE	SP_OUTPUTORDINARYTYPE
SAMPLESP	SAMPLE	SP_RETURNORDINARYTYPE
SAMPLESP	SAMPLE	SP_TRIGGER
SAMPLESP	SAMPLE	SP_UPDATEADDR
SAMPLESP	SAMPLE	SP_UPDATECOL

The number of stored procedures is 20

2. Click the **Create** button. The **Introduction** window of the **Create Stored Procedure** wizard appears.

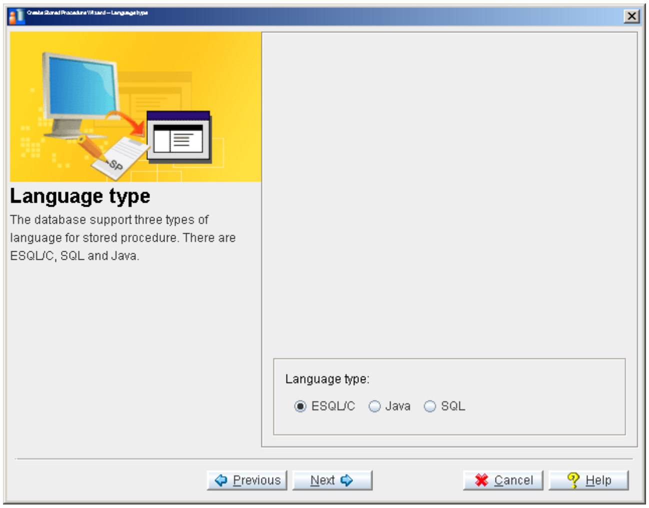


Welcome to the Create Procedure Wizard.

The database supports ESQL/C, SQL and Java stored procedure. Add jarfile is to register user java code to the database. This wizard helps user to create either ESQL/C, SQL or Java stored procedure by going through the following steps.

1. Choose stored procedure language type.
2. Edit ESQL/C or SQL file, or create Java stored procedure SQL script.

3. Click the **Next** button. The **Language Type** wizard appears.



Language type

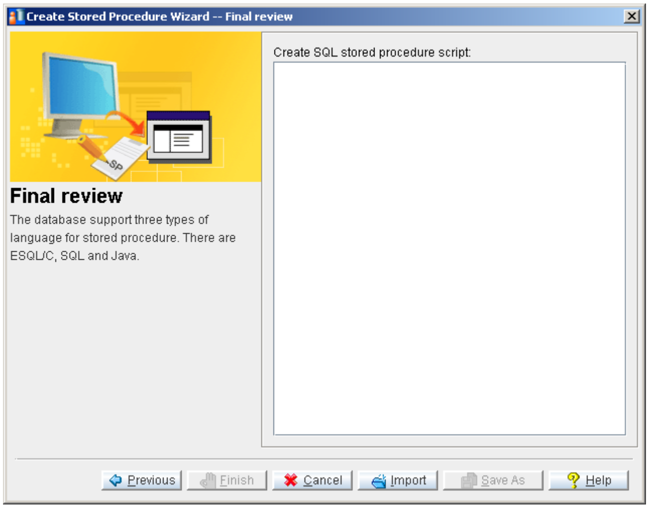
The database support three types of language for stored procedure. There are ESQL/C, SQL and Java.

Language type:

● ESQL/C ○ Java ○ SQL

4. Select a language to use for writing the stored procedure by clicking either the **ESQL/C** radio button, **Java** radio button or the **SQL** radio button. Skip to step five when **ESQL/C** is selected, skip to step nine when **Java** is selected and skip to step 10 when **SQL** is selected.

5. Click the **Next** button. The **Final Review** window appears. Input the ESQL/C statement or click the **Import button** to import an ESQL/C statement from a file.



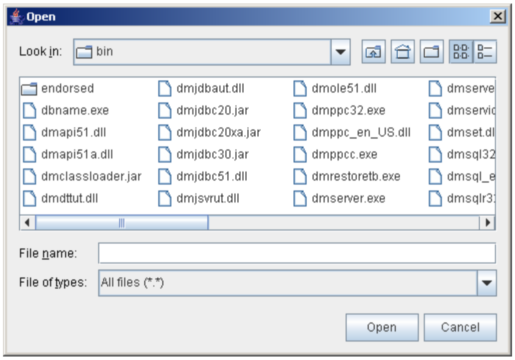
Create Stored Procedure Wizard -- Final review

Final review

The database support three types of language for stored procedure. There are ESQL/C, SQL and Java.

Create SQL stored procedure script:

6. Click the **Import** button. The **Open** window appears. Import files from any source, including the SPDIR directory of other databases on the server or network drives. Enter the path in the **File name** field, or browse the directory tree until the desired path is found.



Open

Look in: bin

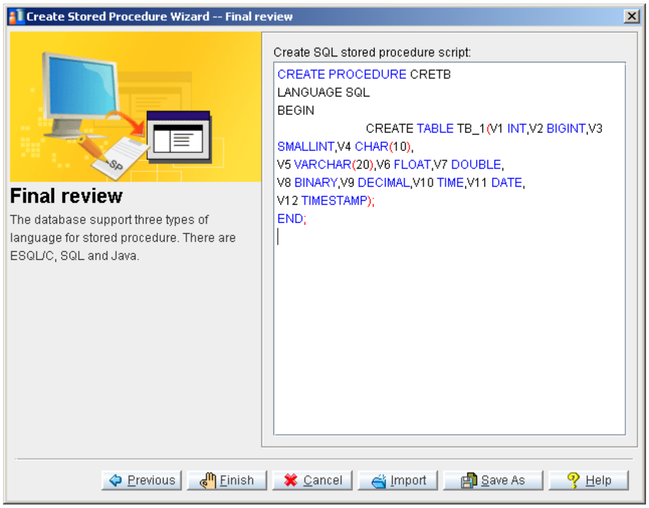
endorsed
dbname.exe
dmapi51.dll
dmapi51a.dll
dmclassloader.jar
dmdttut.dll
dmjdbaut.dll
dmjdbc20.jar
dmjdbc20xa.jar
dmjdbc30.jar
dmjdbc51.dll
dmjsvrut.dll
dmole51.dll
dmppc32.exe
dmppc_en_US.dll
dmppcc.exe
dmrestoretb.exe
dmserver.exe
dmserve
dmservic
dmset.dl
dmsql32
dmsql_e
dmsqlr3

File name:

File of types: All files (*.*)

7. Click the **Open** button.

8. The **Final review** window reappears as in the example below if the imported file contains properly formatted (ASCII) text, or if you choose to manually enter the code. Click the **Save As** button to save the stored procedure to another location, or click **Finish** to compile and save the stored procedure in the database.



Create Stored Procedure Wizard -- Final review

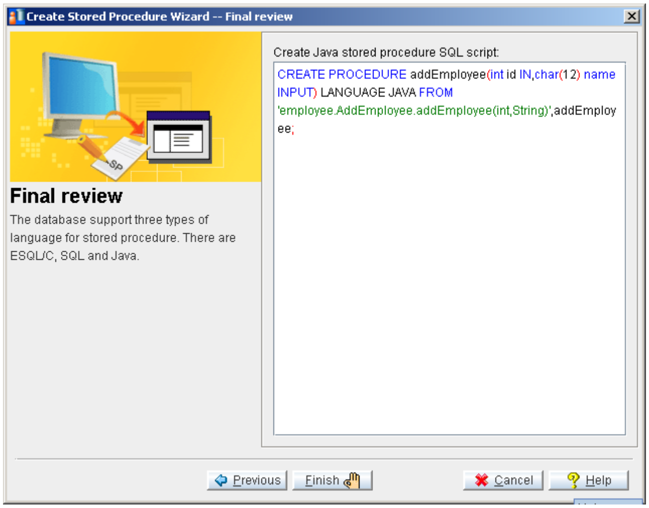
Final review

The database support three types of language for stored procedure. There are ESQL/C, SQL and Java.

Create SQL stored procedure script:

```
CREATE PROCEDURE CRETB
LANGUAGE SQL
BEGIN
                        CREATE TABLE TB_1(V1 INT,V2 BIGINT,V3
SMALLINT,V4 CHAR(10),
V5 VARCHAR(20),V6 FLOAT,V7 DOUBLE,
V8 BINARY,V9 DECIMAL,V10 TIME,V11 DATE,
V12 TIMESTAMP);
END;
```

9. Click the **Next** button. The **Final Review window appears.** Input the Java stored procedure SQL script. Click the **Finish** button.



Create Stored Procedure Wizard -- Final review

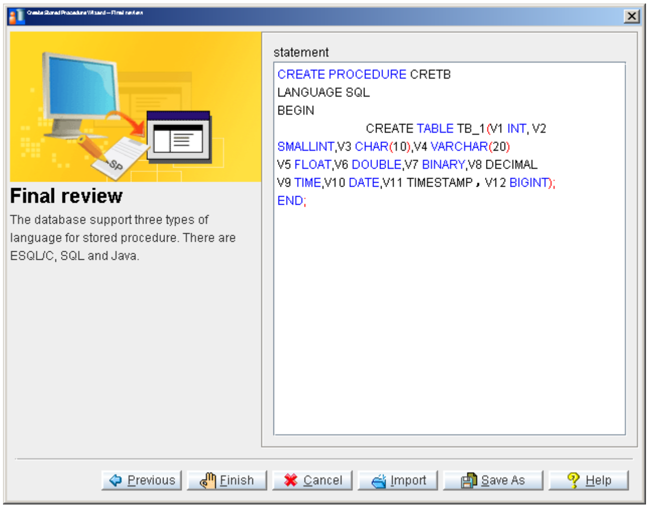
Final review

The database support three types of language for stored procedure. There are ESQL/C, SQL and Java.

Create Java stored procedure SQL script:

```
CREATE PROCEDURE addEmployee(int id IN,char(12) name
INPUT) LANGUAGE JAVA FROM
'employee.AddEmployee.addEmployee(int,String)',addEmploy
ee;
```

10. Click the **Next** button. The **Final Review** window appears. Input the SQL statement or click the **Import button** to import an SQL statement from a file.



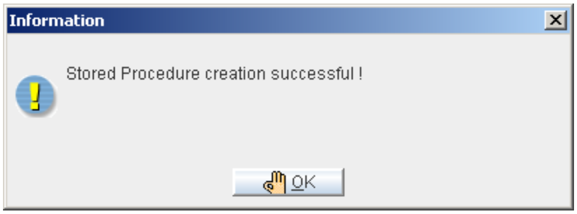
Final review

The database support three types of language for stored procedure. There are ESQL/C, SQL and Java.

statement

```
CREATE PROCEDURE CRETB
LANGUAGE SQL
BEGIN
                CREATE TABLE TB_1(V1 INT, V2
SMALLINT,V3 CHAR(10),V4 VARCHAR(20)
V5 FLOAT,V6 DOUBLE,V7 BINARY,V8 DECIMAL
V9 TIME,V10 DATE,V11 TIMESTAMP , V12 BIGINT);
END;
```

11. If the Stored Procedure compiles correctly, the following message appears.



Stored Procedure creation successful !

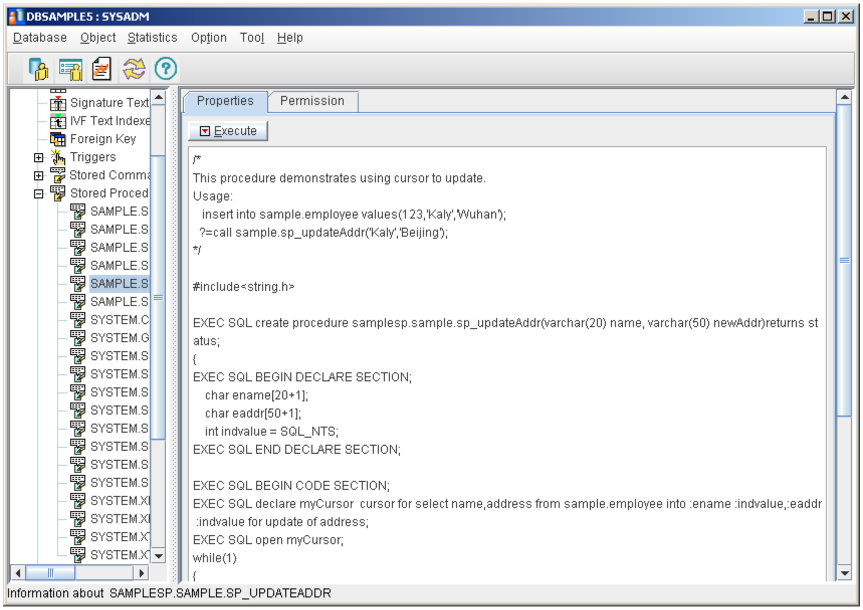
12. Click the **OK** button.

13.2 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 prompts for the parameter’s value during execution of the stored procedure. The number of input parameters should be equal to number of input parameters in the stored procedure.

➲ Executing a stored procedure:

1. Open the **Stored Procedure** node and select a stored procedure. The stored procedure **Properties** page appears.



```
/*
This procedure demonstrates using cursor to update.
Usage:
  insert into sample.employee values(123,'Kaly','Wuhan');
  ?=call sample.sp_updateAddr('Kaly','Beijing');
*/

#include<string.h>

EXEC SQL create procedure samplesp.sample.sp_updateAddr(varchar(20) name, varchar(50) newAddr)returns st
atus;
{
EXEC SQL BEGIN DECLARE SECTION;
   char ename[20+1];
   char eaddr[50+1];
   int indvalue = SQL_NTS;
EXEC SQL END DECLARE SECTION;

EXEC SQL BEGIN CODE SECTION;
EXEC SQL declare myCursor  cursor for select name,address from sample.employee into :ename :indvalue,:eaddr
 :indvalue for update of address;
EXEC SQL open myCursor;
while(1)
{
```

Information about SAMPLESP.SAMPLE.SP_UPDATEADDR

NOTE *Double-clicking a stored procedure in the right panel displays the same window.*

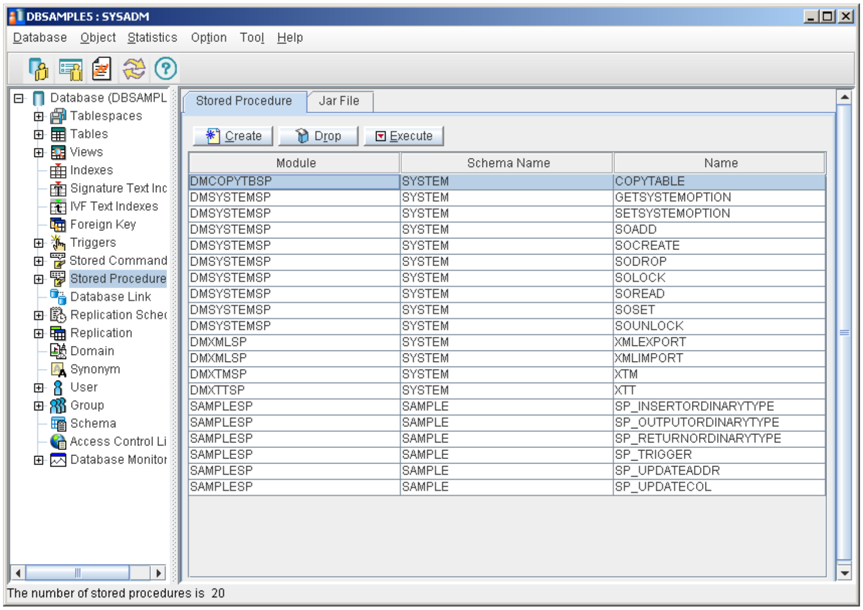
2. Click the **Execute** button. The result of the executed stored procedure appears.
3. Click the **OK** button.

13.3 Dropping a Stored Procedure

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

➲ Dropping a stored procedure:

1. Click the **Stored Procedure** object in the tree. All the stored procedures in the database appear.

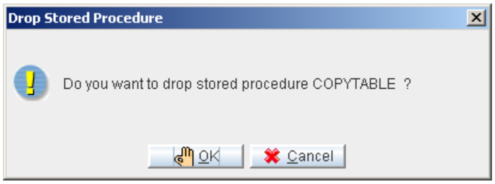


Module	Schema Name	Name
DMCOPYTBSP	SYSTEM	COPYTABLE
DMSYSTEMSP	SYSTEM	GETSYSTEMOPTION
DMSYSTEMSP	SYSTEM	SETSYSTEMOPTION
DMSYSTEMSP	SYSTEM	SOADD
DMSYSTEMSP	SYSTEM	SOCREATE
DMSYSTEMSP	SYSTEM	SODROP
DMSYSTEMSP	SYSTEM	SOLOCK
DMSYSTEMSP	SYSTEM	SOREAD
DMSYSTEMSP	SYSTEM	SOSET
DMSYSTEMSP	SYSTEM	SOUNLOCK
DMXMLSP	SYSTEM	XMLEXPORT
DMXMLSP	SYSTEM	XMLIMPORT
DMXTMSP	SYSTEM	XTM
DMXTTSP	SYSTEM	XTT
SAMPLESP	SAMPLE	SP_INSERTORDINARYTYPE
SAMPLESP	SAMPLE	SP_OUTPUTORDINARYTYPE
SAMPLESP	SAMPLE	SP_RETURNORDINARYTYPE
SAMPLESP	SAMPLE	SP_TRIGGER
SAMPLESP	SAMPLE	SP_UPDATEADDR
SAMPLESP	SAMPLE	SP_UPDATECOL

The number of stored procedures is 20

2. Select a Stored Procedure.

3. Click the **Drop** button. A confirmation dialog box appears.



Drop Stored Procedure

Do you want to drop stored procedure COPYTABLE ?

4. Click the **OK** button to drop the stored procedure or click the **Cancel** button to terminate the dropping process.

13.4 Granting Privileges on Stored Procedures

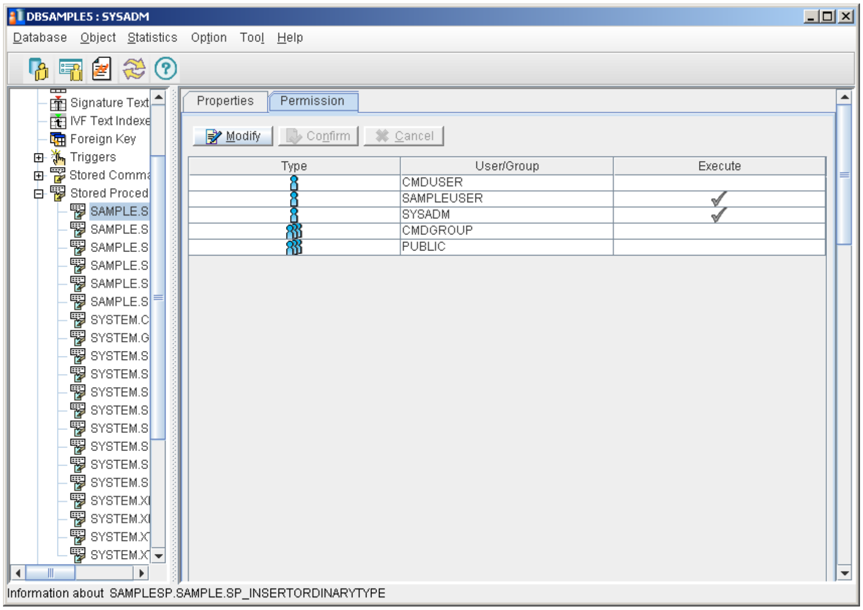
You must have execute privilege for a stored procedure to execute it. In order to allow a stored procedure to be used by others, you can grant execute privilege on a stored procedure to other database users. If you do not wish others to execute a stored procedure, you can revoke execute privilege of 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.

➲ Granting and revoking privileges on a stored procedure:

1. Open the **Stored Procedure** node and then select a stored procedure. The stored procedure **Properties** page appears.

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

2. Click the **Permission** tab. The **Permission** page appears.



Type	User/Group	Execute
👤	CMDUSER	
👤	SAMPLEUSER	✓
👤	SYSADM	✓
👥	CMDGROUP	
👥	PUBLIC	

Information about SAMPLESP.SAMPLE.SP_INSERTORDINARYTYPE

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**.

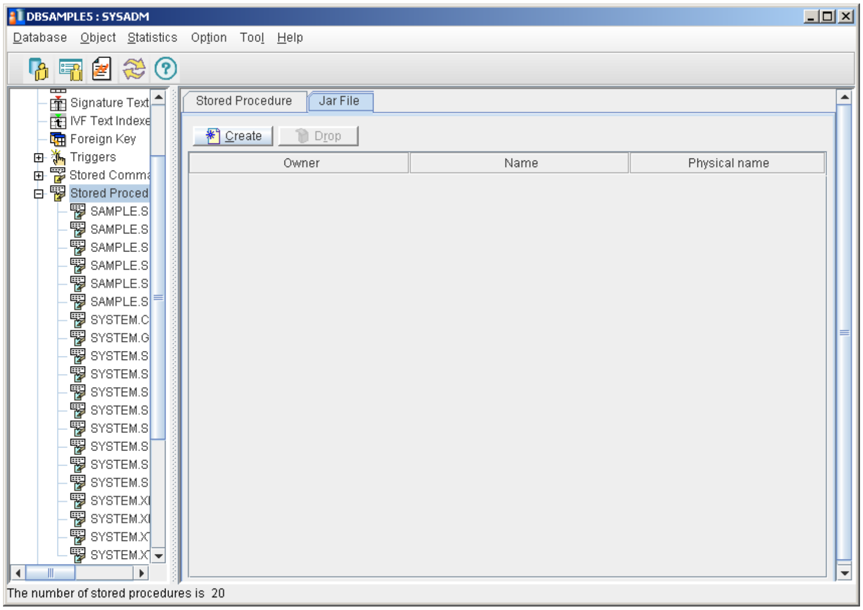
13.5 Add a Jar File for Java Stored Procedure

Before creating Java stored procedure, you should register you jar file into the database.

➲ Creating a Jarfile:

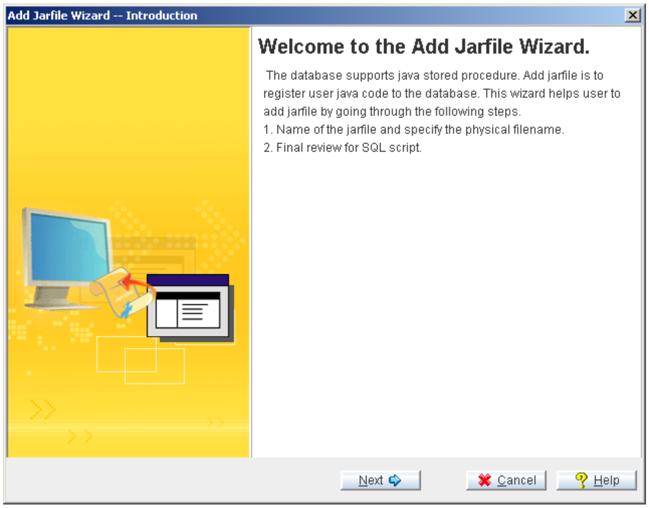
1. Click the object **Stored Procedure** in the tree. The Stored Procedures page will appear.

2. Click the **Jar File**. The Jar File page appears.



The number of stored procedures is 20

3. Click **Create**. The **Add Jarfile** wizard appears.

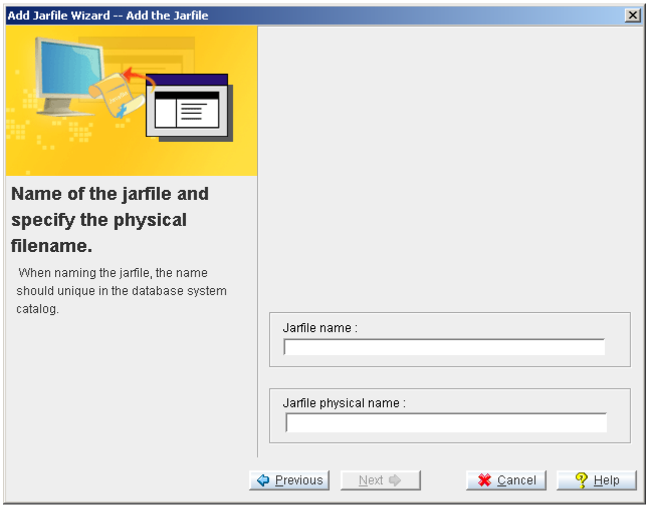


Add Jarfile Wizard -- Introduction

Welcome to the Add Jarfile Wizard.

The database supports java stored procedure. Add jarfile is to register user java code to the database. This wizard helps user to add jarfile by going through the following steps.
1. Name of the jarfile and specify the physical filename.
2. Final review for SQL script.

4. Click **Next**. The **Name of the jarfile and specify the physical filename** wizard appears.



Add Jarfile Wizard -- Add the Jarfile

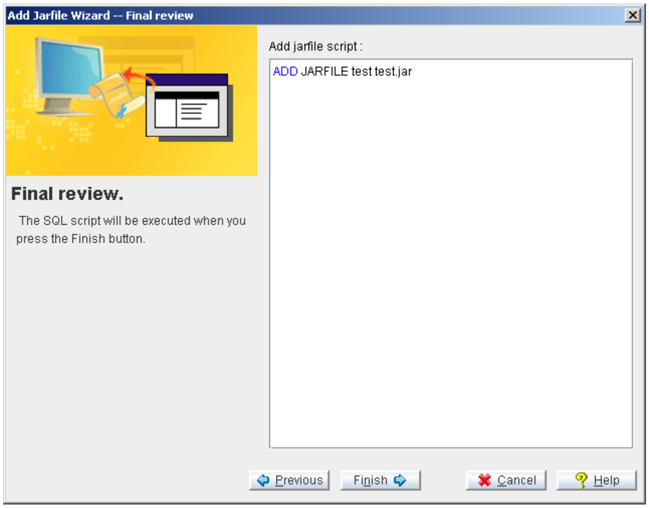
Name of the jarfile and specify the physical filename.

When naming the jarfile, the name should unique in the database system catalog.

Jarfile name :

Jarfile physical name :

5. Input the Jarfile name and the Jarfile physical path separately in the proper filed.
6. Click **Next**. The Final review wizard appears.



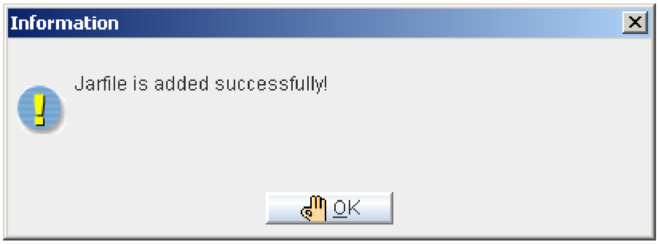
Final review.

The SQL script will be executed when you press the Finish button.

Add jarfile script :

```
ADD JARFILE test test.jar
```

7. Review the script in the right console and if you are sure about it, click **Finish.** The added Jarfile window will appear.



Jarfile is added successfully!

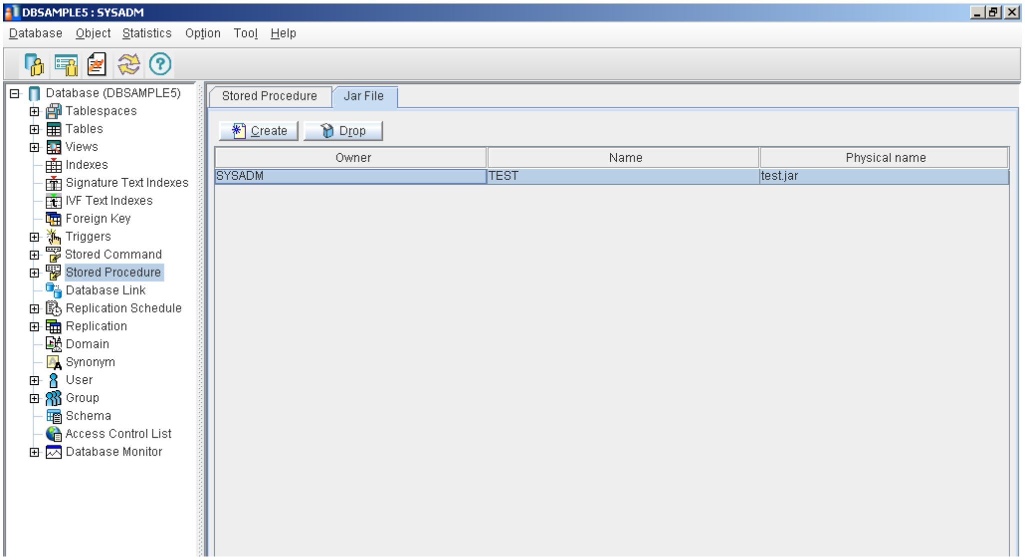
8. Click the **OK** button.

13.6 Dropping Jar Files

When a jarfile is no longer required, you can drop it.

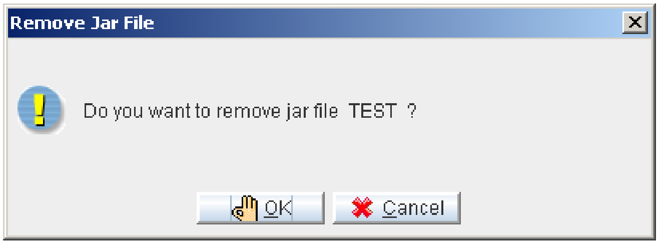
➲ Dropping a jar file:

1. Click the **Stored Procedure** icon from the tree, the Stored Procedure page will appear in the right panel, and then click the Jar File page. A list of the all the Jarfile you have added is displayed.



Owner	Name	Physical name
SYSADM	TEST	test.jar

2. Choose the Jarfile you want to drop, Click the **Drop** in the top of the page. The **Remove Jar File** confirmation dialog box appears.



Do you want to remove jar file TEST ?

3. Click the **OK** button. The remaining Jarfiles in the database are displayed.

14 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 that client's Coordinator Database. Through the Coordinator Database, the client can connect to other remote databases, which are called 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.

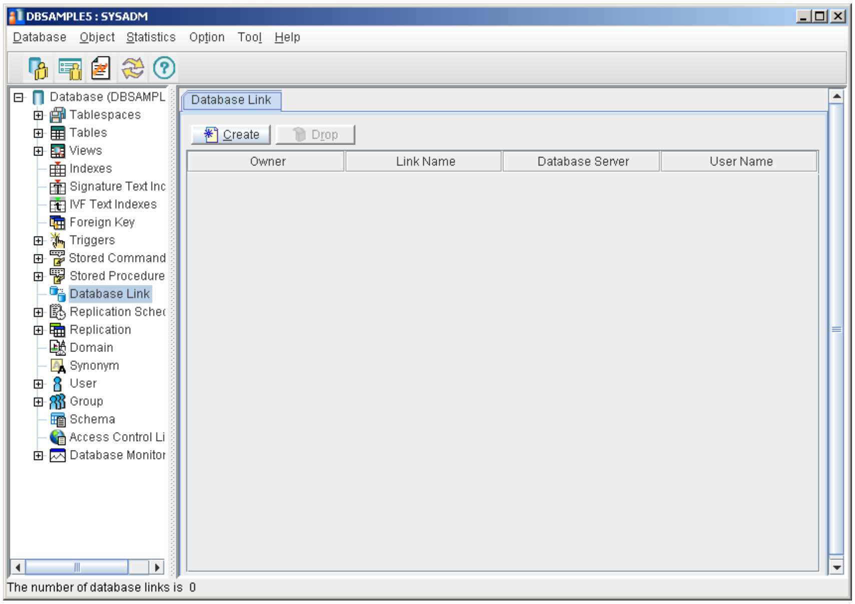
14.1 Creating a Database Link

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 you create a private database link 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 are used by default.

➲ Creating a database Link:

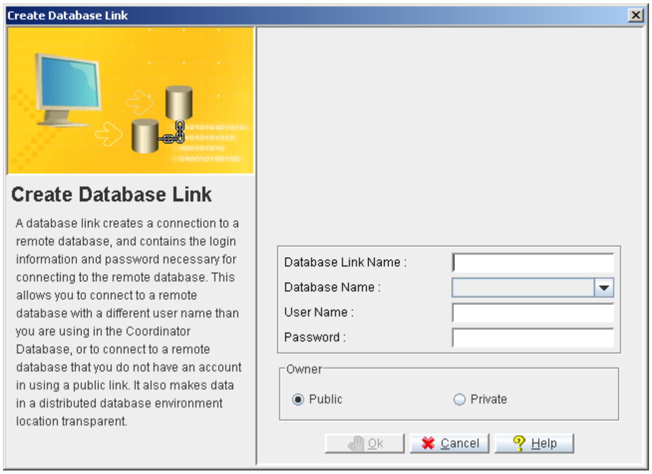
1. Select the object **Database Link** in the tree. A list of all database links in the database appears.



Owner	Link Name	Database Server	User Name

The number of database links is 0

2. Click **Create**. The **Create Database Link** window appears.



Create Database Link

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 that you do not have an account in using a public link. It also makes data in a distributed database environment location transparent.

Database Link Name :
Database Name :
User Name :
Password :

Owner
● Public ○ Private

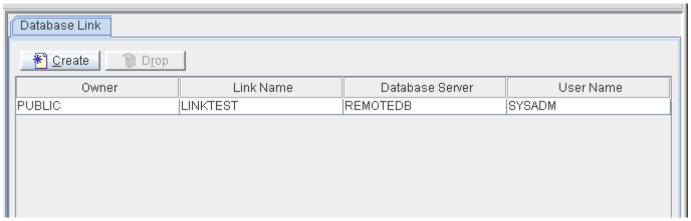
3. Enter the database link name.
4. Select the database to be the coordinator database from the **Database Name** menu.
5. The default user name for the selected coordinator database is displayed in the **User Name** field. A new name can be entered.
6. Enter the password.
7. If the owner of the database 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.

14.2 Dropping a Database Link

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

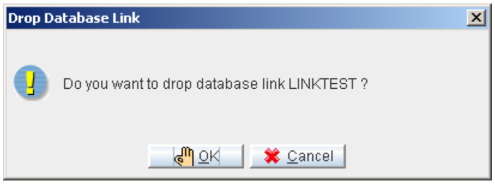
➲ Dropping a database link:

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



Owner	Link Name	Database Server	User Name
PUBLIC	LINKTEST	REMOTEDB	SYSADM

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



Do you want to drop database link LINKTEST ?

4. Click **OK**.

©Copyright 1995-2010 CASEMaker Inc.

15 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. DBMaker keeps trying to replicate until any failures are corrected or until the user-specified number of attempts to reconnect is reached. This allows asynchronous table replications to tolerate network and remote database failures.

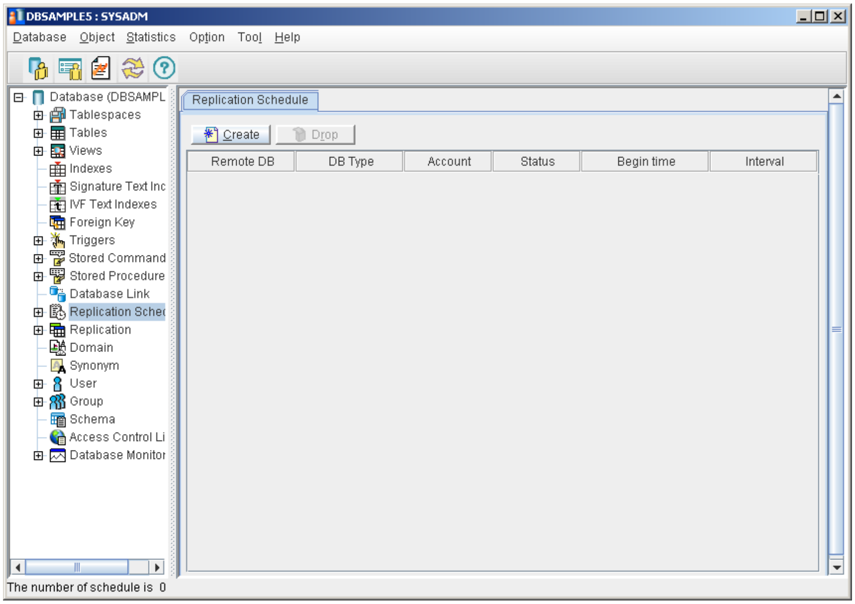
Asynchronous table replication depends on the establishment of a replication schedule. Likewise, developing a replication schedule without an asynchronous table replication that uses it is meaningless. A replication schedule should first be developed. Afterward, the asynchronous table replication can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

15.1 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.

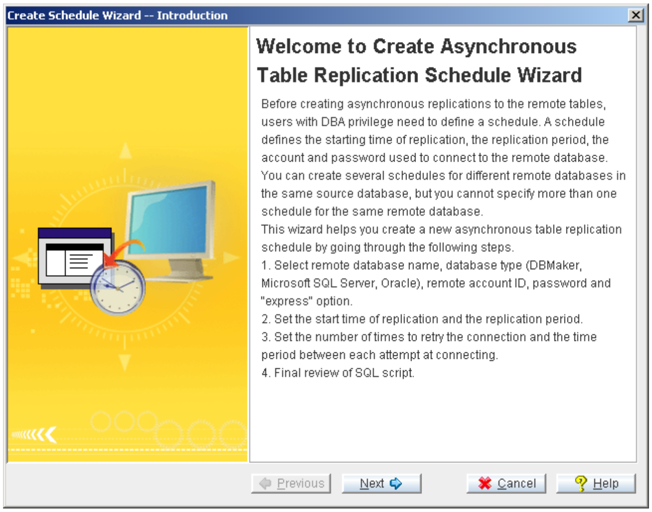
➲ Creating a Replication Schedule:

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



The number of schedule is 0

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



Create Schedule Wizard -- Introduction

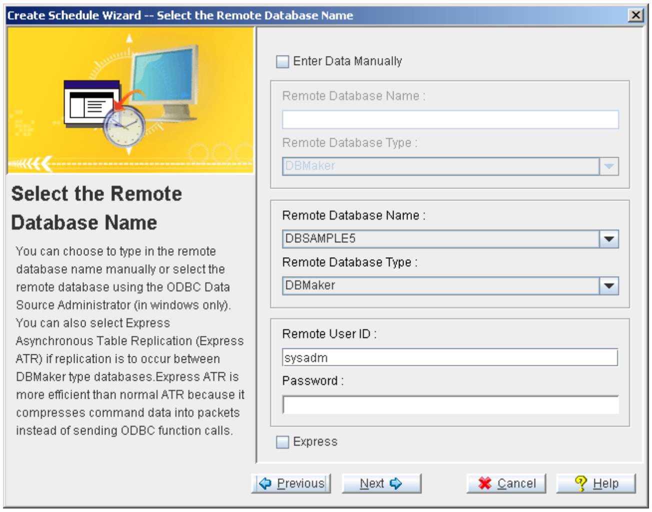
Welcome to Create Asynchronous Table Replication Schedule Wizard

Before creating asynchronous replications to the remote tables, users with DBA privilege need to define a schedule. A schedule defines the starting time of replication, the replication period, the account and password used to connect to the remote database. You can create several schedules for different remote databases in the same source database, but you cannot specify more than one schedule for the same remote database.

This wizard helps you create a new asynchronous table replication schedule by going through the following steps.

1. Select remote database name, database type (DBMaker, Microsoft SQL Server, Oracle), remote account ID, password and "express" option.
2. Set the start time of replication and the replication period.
3. Set the number of times to retry the connection and the time period between each attempt at connecting.
4. Final review of SQL script.

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



Create Schedule Wizard -- Select the Remote Database Name

Select the Remote Database Name

You can choose to type in the remote database name manually or select the remote database using the ODBC Data Source Administrator (in windows only). You can also select Express Asynchronous Table Replication (Express ATR) if replication is to occur between DBMaker type databases.Express ATR is more efficient than normal ATR because it compresses command data into packets instead of sending ODBC function calls.

□ Enter Data Manually

Remote Database Name :

Remote Database Type : DBMaker

Remote Database Name : DBSAMPLE5

Remote Database Type : DBMaker

Remote User ID : sysadm

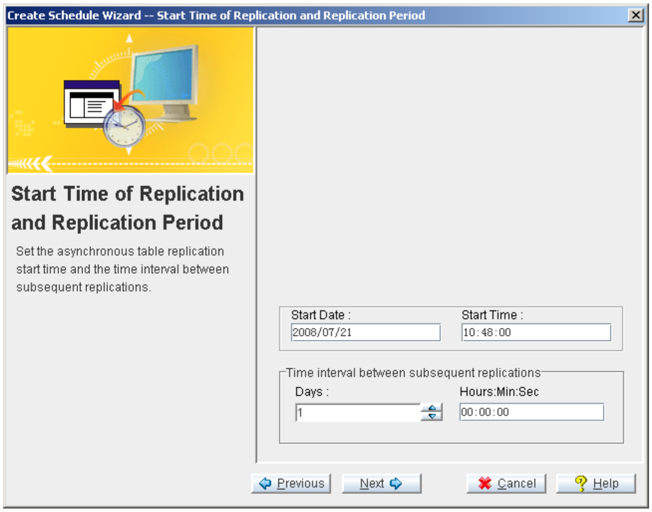
Password :

□ Express

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 appears.



Create Schedule Wizard -- Start Time of Replication and Replication Period

Start Time of Replication and Replication Period

Set the asynchronous table replication start time and the time interval between subsequent replications.

Start Date : 2008/07/21

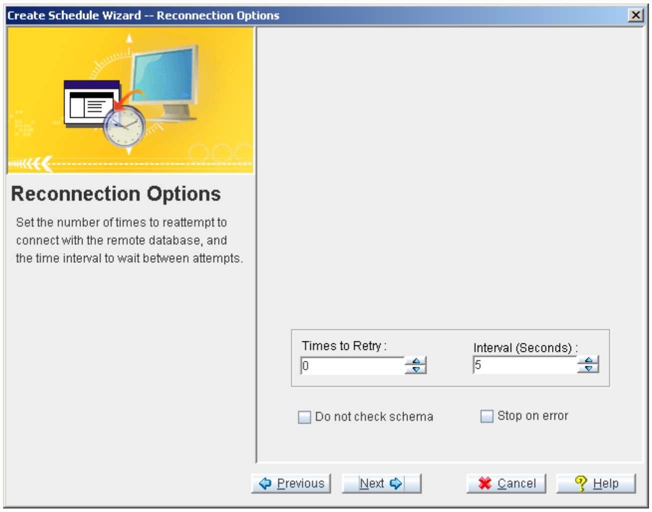
Start Time : 10:48:00

Time interval between subsequent replications

Days : 1

Hours:Min:Sec 00:00:00

11. Enter a date, in the **Start Date** box, when the Asynchronous Table Replication will begin.
12. Enter the starting time of Asynchronous Table Replication in the **Start Time** box.
13. In the **Time interval between subsequent replications** field: Enter the number of days between which the Asynchronous Table Replication will performed in the Days combo box.
14. 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.
15. Click **Next**. The **Reconnection Options** window appears.



Create Schedule Wizard -- Reconnection Options

Reconnection Options

Set the number of times to reattempt to connect with the remote database, and the time interval to wait between attempts.

Times to Retry : 0

Interval (Seconds) : 5

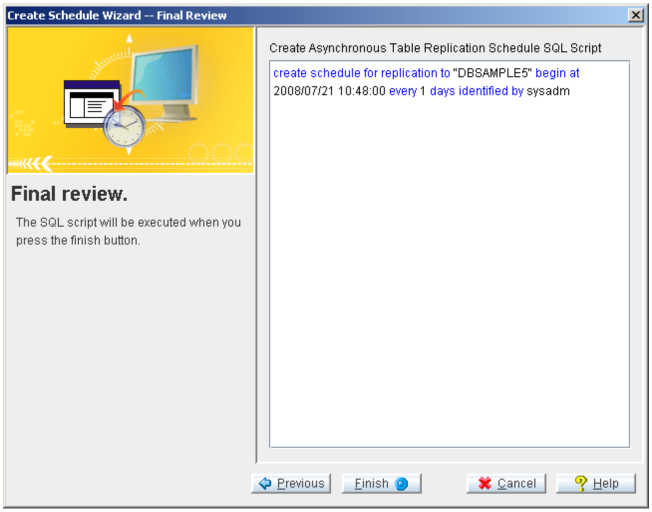
□ Do not check schema

□ Stop on error

16. 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.
17. 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.
18. Click the **Do not check schema** check box if you do not want the Replication process to check the table schema before replicating.
19. Click the **Stop on error** check box if you want the Replication process to stop if any error occures.

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

20. Click **Next**. The following window appears.

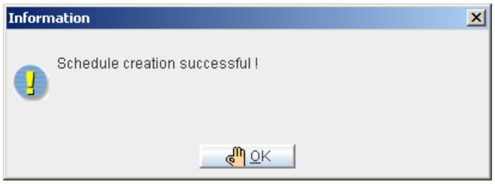


Create Schedule Wizard -- Final Review
Final review.
The SQL script will be executed when you press the finish button.
Create Asynchronous Table Replication Schedule SQL Script
Previous
Finish
Cancel
Help

```
create schedule for replication to "DBSAMPLE5" begin at
2008/07/21 10:48:00 every 1 days identified by sysadm
```

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

22. Click **Finish**. The **Information** dialog box appears.



Schedule creation successful !

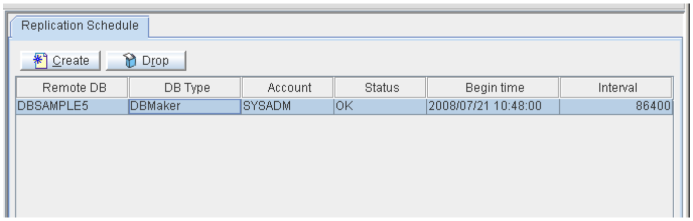
23. Click **OK** and the message disappears.

15.2 Dropping a Replication Schedule

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

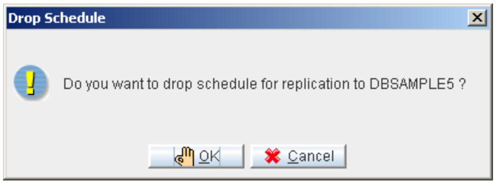
➲ Dropping a replication schedule:

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



Remote DB	DB Type	Account	Status	Begin time	Interval
DBSAMPLE5	DBMaker	SYSADM	OK	2008/07/21 10:48:00	86400

3. Click **Drop**. A confirmation window appears.



Do you want to drop schedule for replication to DBSAMPLE5 ?

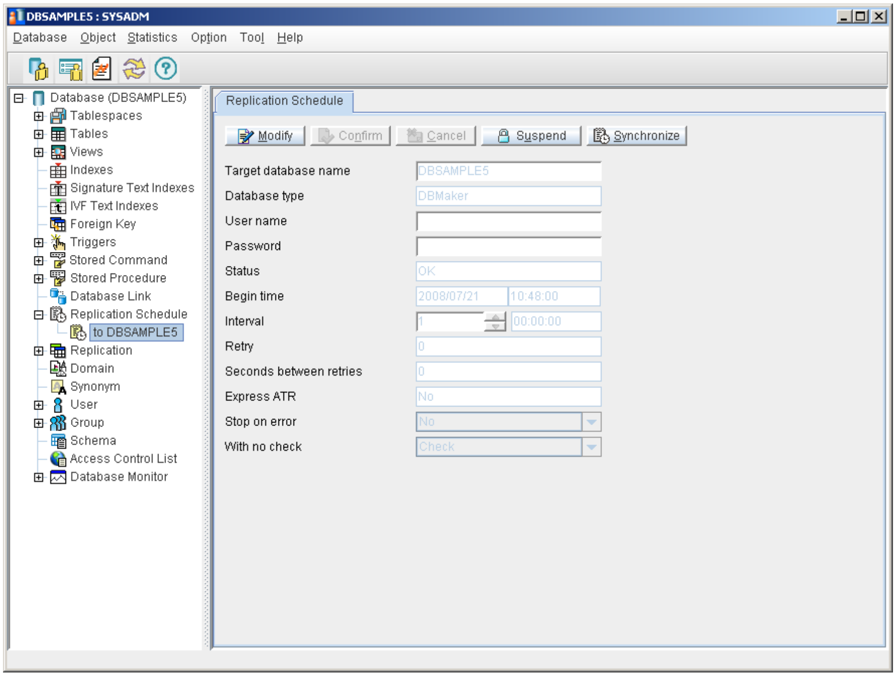
4. Click the **OK** button.

15.3 Modifying a Replication Schedule

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

➲ Modifying a replication schedule:

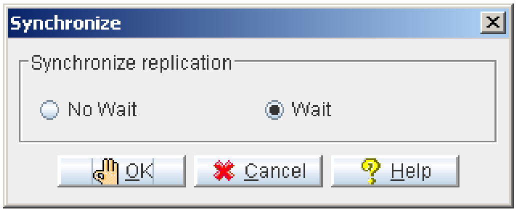
1. Open the node **Replication Schedule** in the tree and select a replication schedule. The following screen appears.



Target database name	DBSAMPLE5	
Database type	DBMaker	
User name		
Password		
Status	OK	
Begin time	2008/07/21	10:48:00
Interval	1	00:00:00
Retry	0	
Seconds between retries	0	
Express ATR	No	
Stop on error	No	
With no check	Check	

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 two 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.



Synchronize

Synchronize replication

○ No Wait ● Wait

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

- **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.

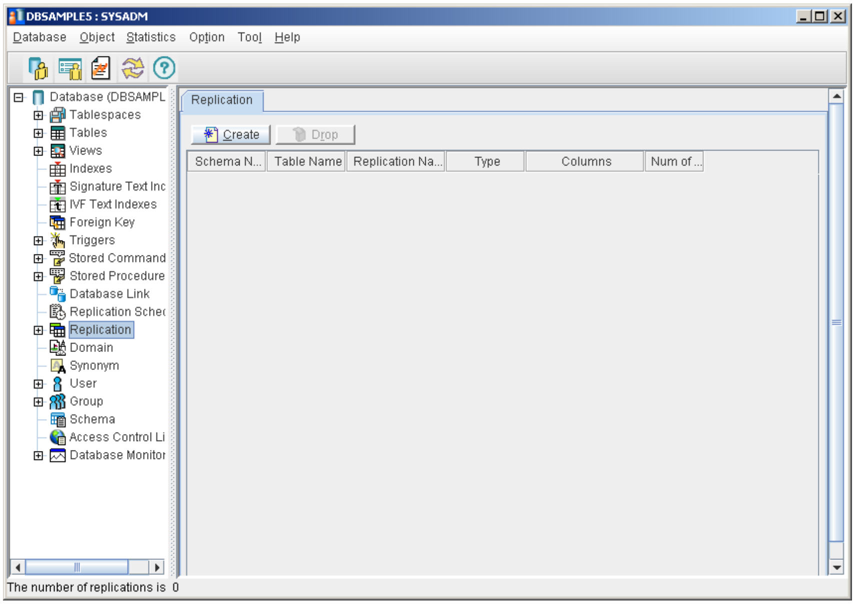
13. Click **OK**.

16 Managing Table Replication

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

➲ Viewing Table Replications:

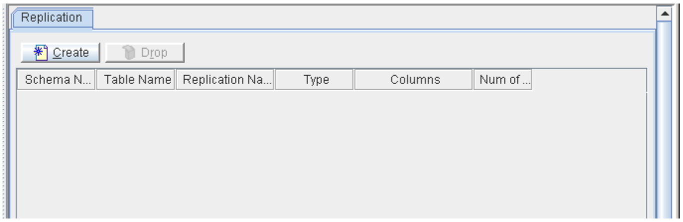
1. Select **Replication** from the tree, the **Replication** page will display. You can view a list of replications by owner, source table name, replication name, replication type (synchronous or asynchronous), source columns, and the number of subscribers to the replication (the number of destination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



Schema N...	Table Name	Replication Na...	Type	Columns	Num of ...

The number of replications is 0

2. Selecting a replication from the tree or the list on the **Replication** page opens 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.



Schema N...	Table Name	Replication Na...	Type	Columns	Num of ...

16.1 Synchronous Table Replication

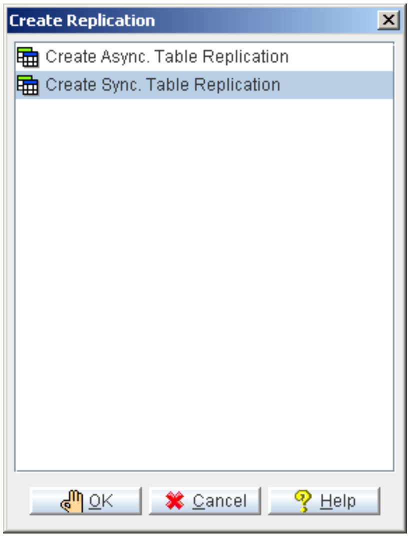
Table replication ensures the consistency of data across multiple databases. Synchronous means the modification to the remote site is immediate. The replication modifies the remote table at the same time it modifies the local table. Thus, after establishing a replication, any update on the source table 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 the local database will fail.

Adding Synchronous Table Replication

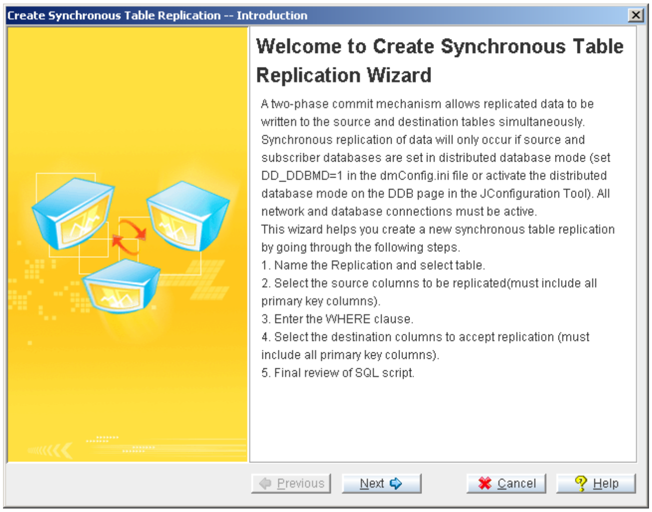
Synchronous table replication must be created for it to occur. Furthermore, distributed database mode must be enabled in all databases engaged in synchronous table replication. The schema of the destination columns must match the schema of the source columns. Primary key columns in the source table must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

➲ Creating a Synchronous Table Replication:

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


Create Replication
Create Async. Table Replication
Create Sync. Table Replication
OK
Cancel
Help

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



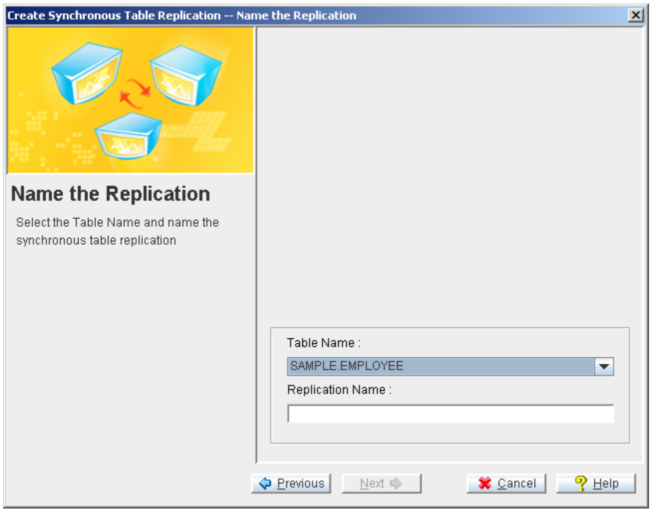
Welcome to Create Synchronous Table Replication Wizard

A two-phase commit mechanism allows replicated data to be written to the source and destination tables simultaneously. Synchronous replication of data will only occur if source and subscriber databases are set in distributed database mode (set DD_DDBMD=1 in the dmConfig.ini file or activate the distributed database mode on the DDB page in the JConfiguration Tool). All network and database connections must be active.

This wizard helps you create a new synchronous table replication by going through the following steps.

1. Name the Replication and select table.
2. Select the source columns to be replicated(must include all primary key columns).
3. Enter the WHERE clause.
4. Select the destination columns to accept replication (must include all primary key columns).
5. Final review of SQL script.

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



Create Synchronous Table Replication -- Name the Replication

Name the Replication

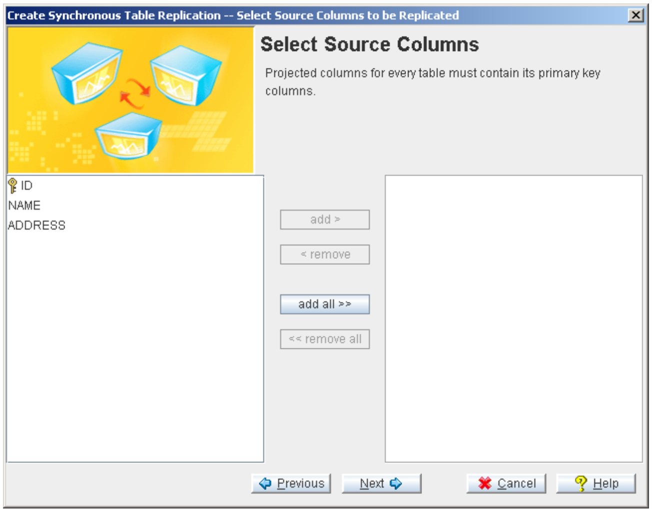
Select the Table Name and name the synchronous table replication

Table Name : SAMPLE.EMPLOYEE

Replication Name :

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.



Create Synchronous Table Replication -- Select Source Columns to be Replicated

Select Source Columns

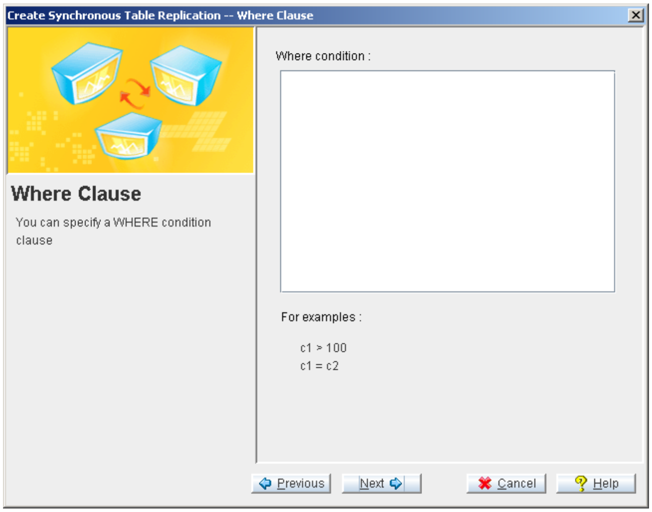
Projected columns for every table must contain its primary key columns.

ID
NAME
ADDRESS

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 adds all columns to the replication. The list box to the right displays all the selected source columns. Remove columns from the replication by double-clicking 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 appears.



Create Synchronous Table Replication -- Where Clause

Where Clause

You can specify a WHERE condition clause

Where condition :

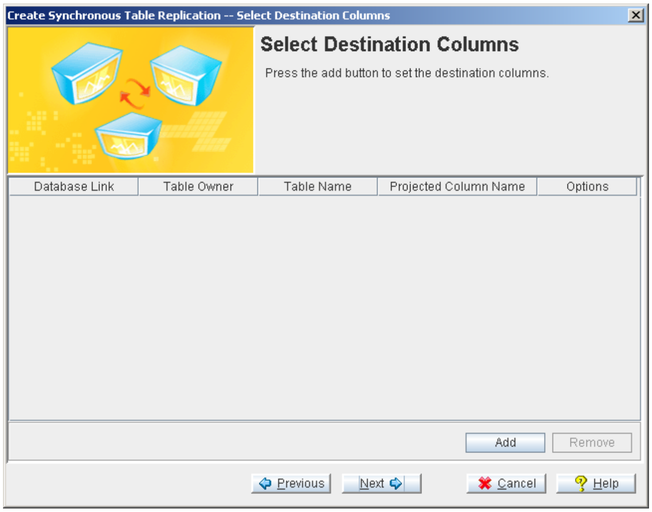
For examples :

c1 > 100
c1 = c2

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.*

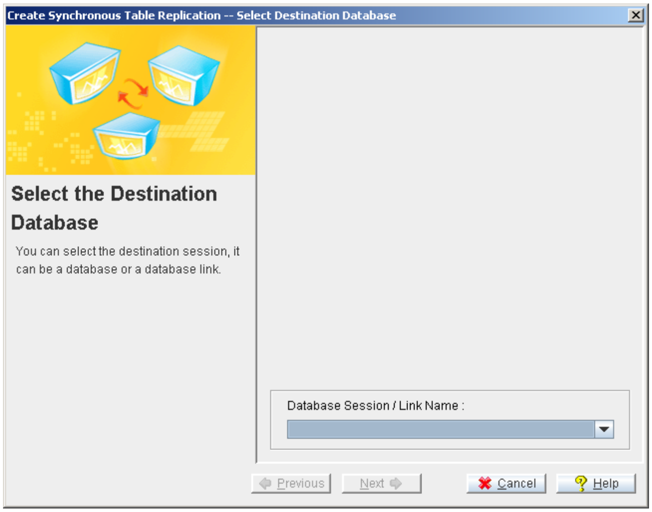
9. Click **Next**. The **Select Destination Columns** window will appear.



Select Destination Columns

Press the add button to set the destination columns.

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



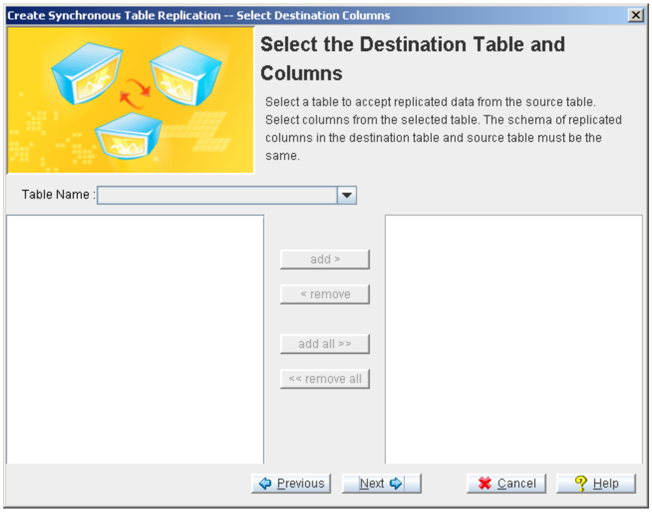
Create Synchronous Table Replication -- Select Destination Database

Select the Destination Database

You can select the destination session, it can be a database or a database link.

Database Session / Link Name :

11. Enter the destination database information from the **Database Session / Link Name** menu at the bottom of the page.
12. Click the **Next** button. The **Select the Destination Table and Columns** window appears.



Create Synchronous Table Replication -- Select Destination Columns

Select the Destination Table and Columns

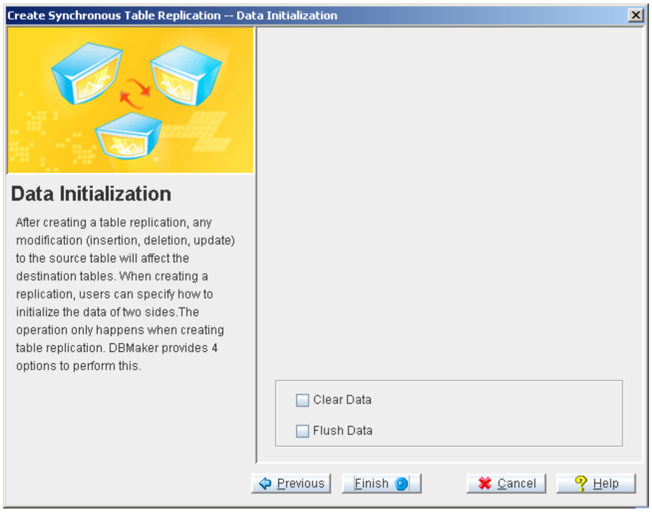
Select a table to accept replicated data from the source table. Select columns from the selected table. The schema of replicated columns in the destination table and source table must be the same.

Table Name :

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 adds all columns. All the selected destination columns are displayed in the list box to the right. Columns can be 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 appears.



Create Synchronous Table Replication -- Data Initialization

Data Initialization

After creating a table replication, any modification (insertion, deletion, update) to the source table will affect the destination tables. When creating a replication, users can specify how to initialize the data of two sides.The operation only happens when creating table replication. DBMaker provides 4 options to perform this.

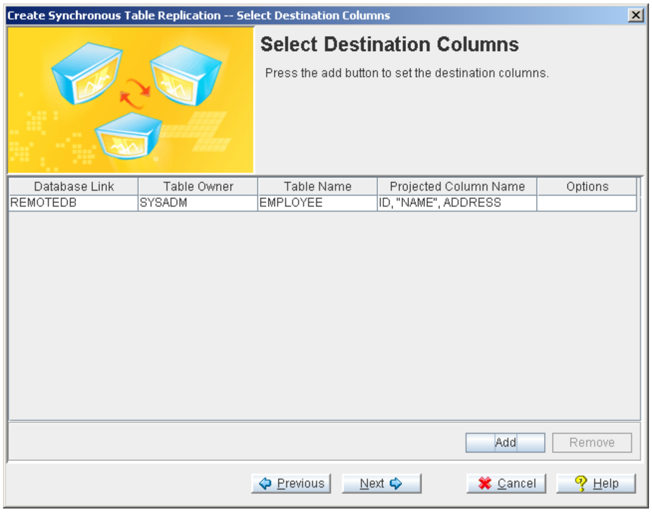
□ Clear Data

□ Flush Data

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.



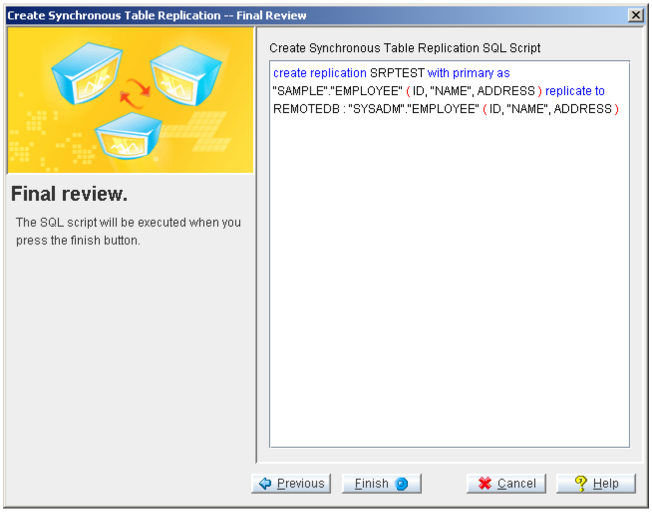
Create Synchronous Table Replication -- Select Destination Columns

Select Destination Columns

Press the add button to set the destination columns.

Database Link	Table Owner	Table Name	Projected Column Name	Options
REMOTEDB	SYSADM	EMPLOYEE	ID, "NAME", ADDRESS	

18. Click **Add** and repeat steps 7 through 13 to add more columns, otherwise click **Next**.



Create Synchronous Table Replication -- Final Review

Final review.

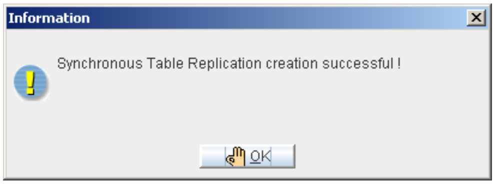
The SQL script will be executed when you press the finish button.

Create Synchronous Table Replication SQL Script

```
create replication SRPTEST with primary as
"SAMPLE"."EMPLOYEE" ( ID, "NAME", ADDRESS ) replicate to
REMOTEDB : "SYSADM"."EMPLOYEE" ( ID, "NAME", ADDRESS )
```

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



Synchronous Table Replication creation successful !

20. Click **OK**

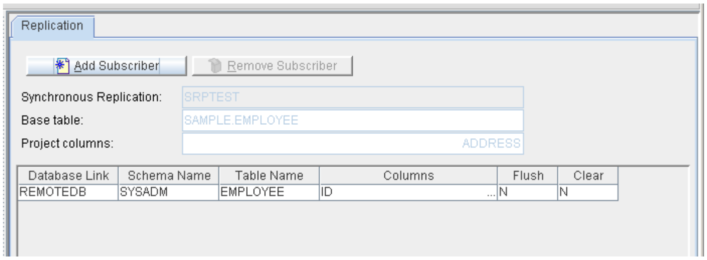
Adding Subscribers to a Synchronous Replication

After a replication is 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

➲ Adding 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.



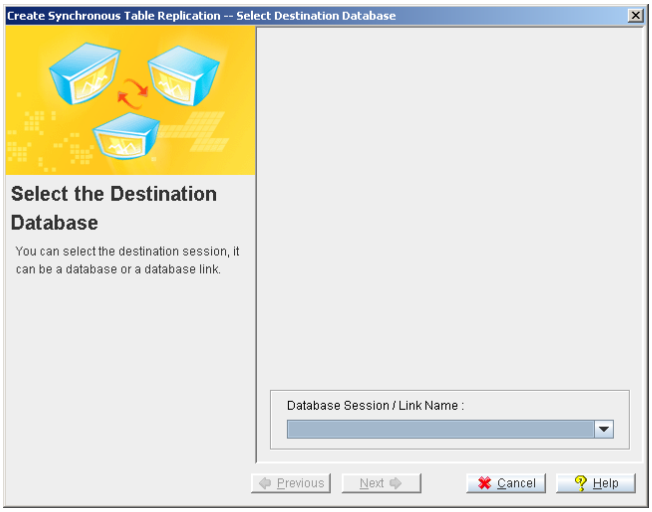
Synchronous Replication: SRPTEST

Base table: SAMPLE.EMPLOYEE

Project columns: ADDRESS

Database Link	Schema Name	Table Name	Columns	Flush	Clear
REMOTEDB	SYSADM	EMPLOYEE	ID ...	N	N

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

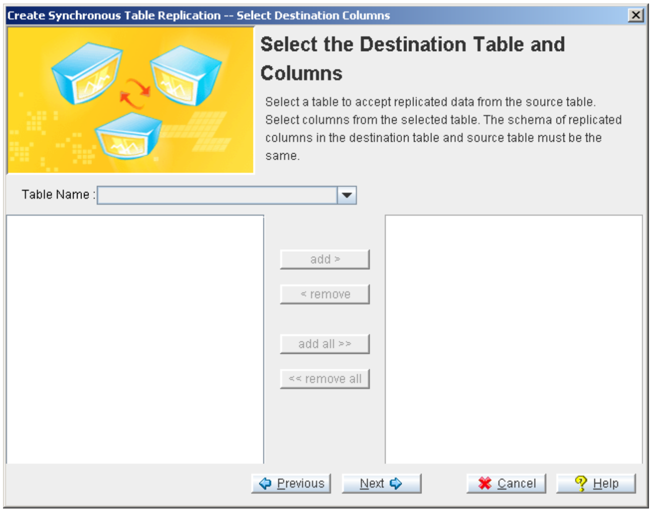

Create Synchronous Table Replication -- Select Destination Database
Previous
Next
Cancel
Help

Select the Destination Database

You can select the destination session, it can be a database or a database link.

Database Session / Link Name :

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



Create Synchronous Table Replication -- Select Destination Columns

Select the Destination Table and Columns

Select a table to accept replicated data from the source table. Select columns from the selected table. The schema of replicated columns in the destination table and source table must be the same.

Table Name :

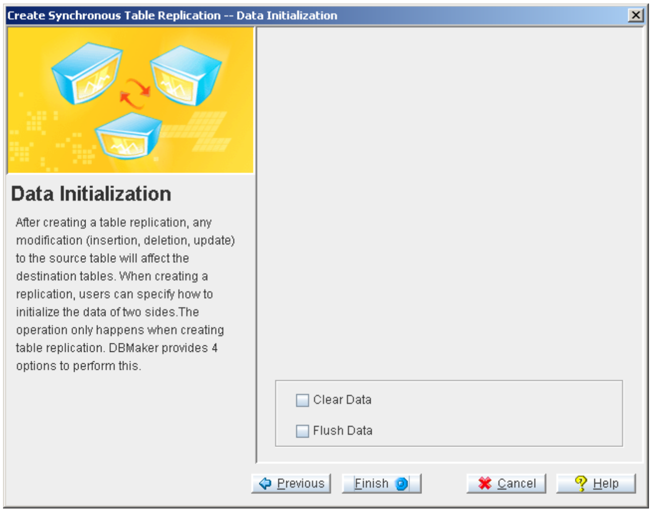
5. The **Select Destination Columns** window will prompt you to select destination columns and a destination table.

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 adds 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 are selected.



Create Synchronous Table Replication -- Data Initialization

Data Initialization

After creating a table replication, any modification (insertion, deletion, update) to the source table will affect the destination tables. When creating a replication, users can specify how to initialize the data of two sides.The operation only happens when creating table replication. DBMaker provides 4 options to perform this.

□ Clear Data

□ Flush Data

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 matching the search criteria into the destination table.

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

16.2 Asynchronous Table Replication

Asynchronous means that the modification to the remote site is 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 retry replication until no failures occur or until the user-specified number of attempts to reconnect is satisfied.

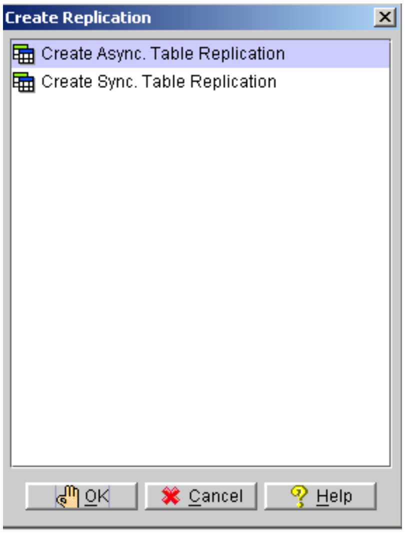
Asynchronous table replication uses ODBC function calls to communicate with remote databases. This might give poor performance in a Wide Area Network (WAN) environment. To achieve better performance in a WAN, DBMaker provides another mechanism named Express Asynchronous table replication whereby SQL 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

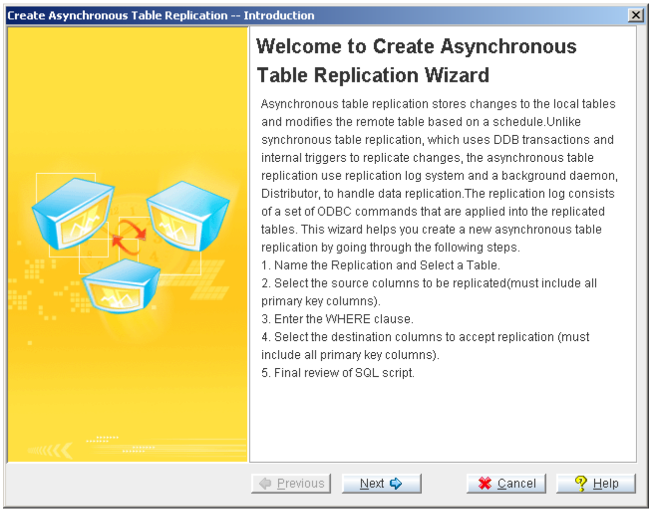
A replication schedule must exist for asynchronous table replication to occur, and all associated settings for the replication schedule set. Furthermore, the schema of the destination columns must match the schema of the source columns and primary key columns in the source table must be replicated.

➲ Creating an Asynchronous Table Replication:

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


Create Replication
Create Async. Table Replication
Create Sync. Table Replication
OK
Cancel
Help

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

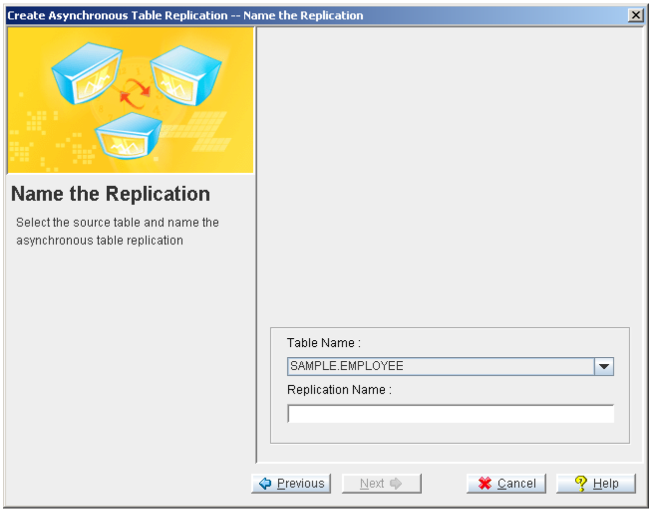


Welcome to Create Asynchronous Table Replication Wizard

Asynchronous table replication stores changes to the local tables and modifies the remote table based on a schedule.Unlike synchronous table replication, which uses DDB transactions and internal triggers to replicate changes, the asynchronous table replication use replication log system and a background daemon, Distributor, to handle data replication.The replication log consists of a set of ODBC commands that are applied into the replicated tables. This wizard helps you create a new asynchronous table replication by going through the following steps.

1. Name the Replication and Select a Table.
2. Select the source columns to be replicated(must include all primary key columns).
3. Enter the WHERE clause.
4. Select the destination columns to accept replication (must include all primary key columns).
5. Final review of SQL script.

3. Select **Next**. The **Name the Replication** window appears.



Create Asynchronous Table Replication -- Name the Replication

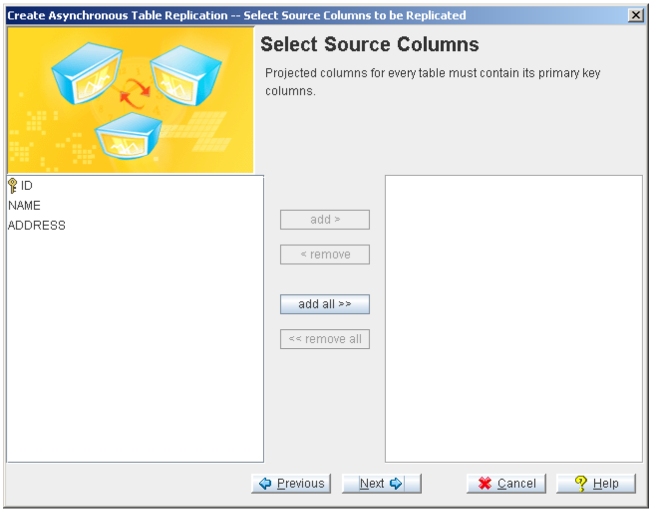
Name the Replication

Select the source table and name the asynchronous table replication

Table Name : SAMPLE.EMPLOYEE

Replication Name :

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**.



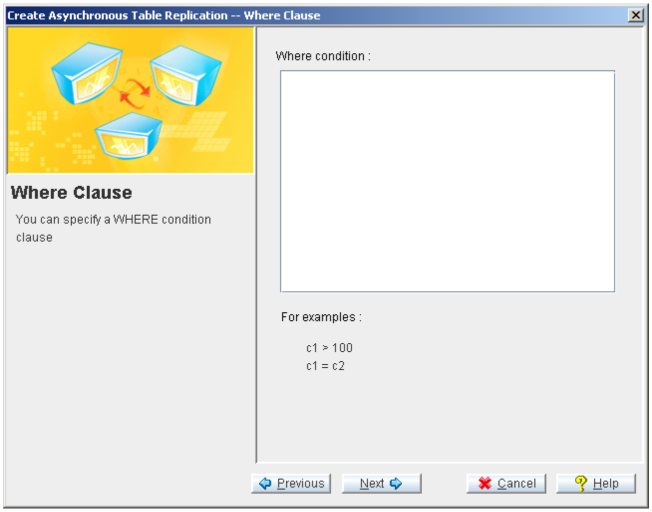
Create Asynchronous Table Replication -- Select Source Columns to be Replicated

Select Source Columns

Projected columns for every table must contain its primary key columns.

ID
NAME
ADDRESS

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.

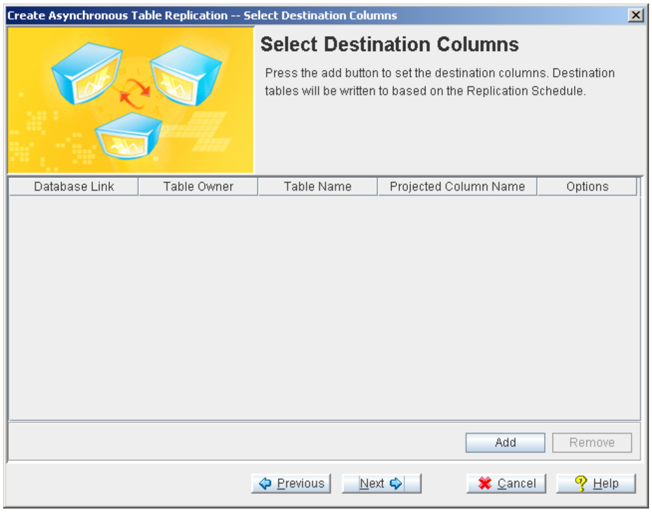

Create Asynchronous Table Replication -- Where Clause
Where Clause
You can specify a WHERE condition clause
Where condition :
For examples :
c1 > 100
c1 = c2
Previous
Next
Cancel
Help

NOTE *Selection of all primary key columns, from the source table is required 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** prompts to add destination columns to the replication.



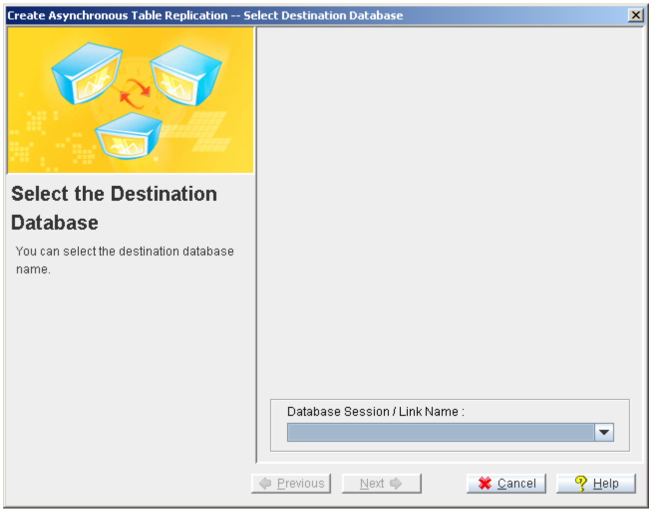
Create Asynchronous Table Replication -- Select Destination Columns

Select Destination Columns

Press the add button to set the destination columns. Destination tables will be written to based on the Replication Schedule.

Database Link	Table Owner	Table Name	Projected Column Name	Options

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



Create Asynchronous Table Replication -- Select Destination Database

Select the Destination Database

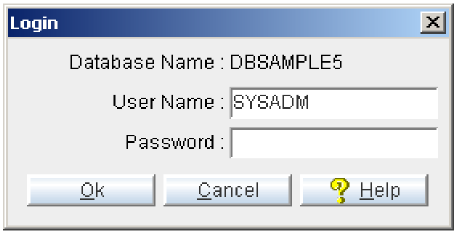
You can select the destination database name.

Database Session / Link Name :

10. 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 the **Next** button.



Login

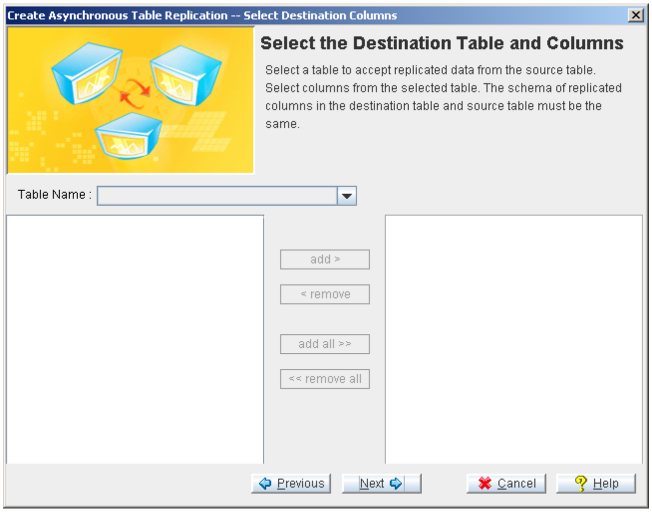
Database Name : DBSAMPLE5

User Name : SYSADM

Password :

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

13. Click **OK**, the **Select Destination Columns** window appears.



Create Asynchronous Table Replication -- Select Destination Columns

Select the Destination Table and Columns

Select a table to accept replicated data from the source table. Select columns from the selected table. The schema of replicated columns in the destination table and source table must be the same.

Table Name :

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

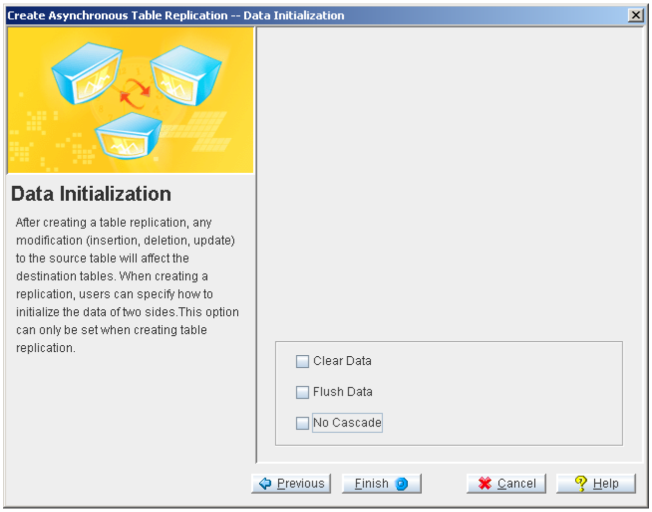
15. Select a destination table from the **Table Name** menu. A list of available columns on the selected table will appear on the left hand list box.

16. Select destination columns by double clicking on the column name or selecting the column and clicking the **add** button. The **add all** button adds all columns. All the selected destination columns are 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 are selected.


Create Asynchronous Table Replication -- Data Initialization
□ Clear Data
□ Flush Data
□ No Cascade
Previous
Finish
Cancel
Help

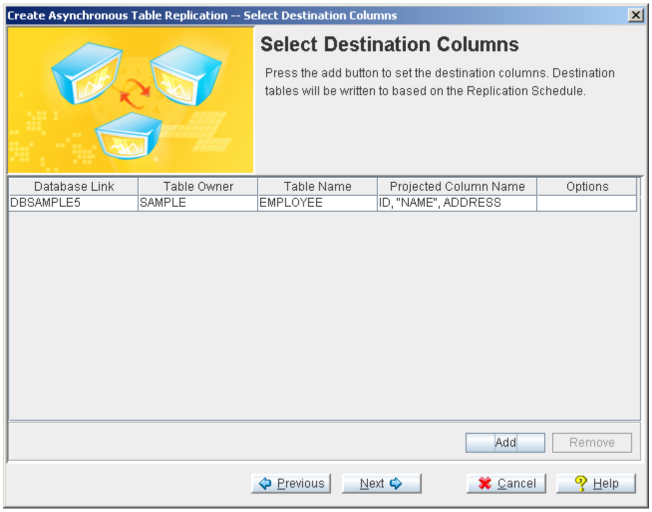
Data Initialization

After creating a table replication, any modification (insertion, deletion, update) to the source table will affect the destination tables. When creating a replication, users can specify how to initialize the data of two sides.This option can only be set when creating table replication.

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 reappears.



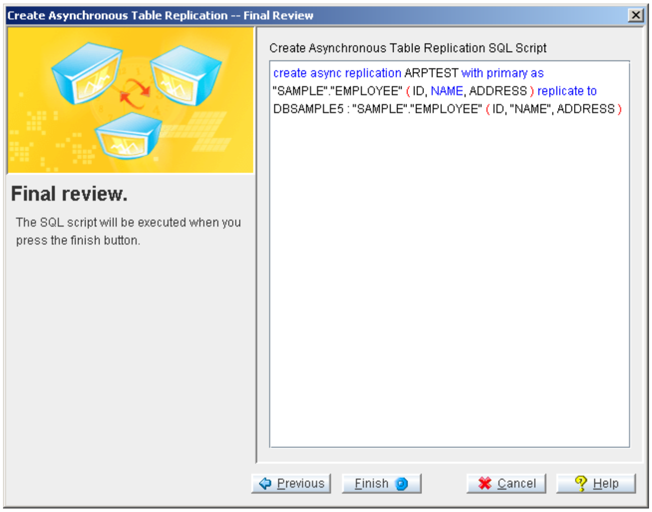
Create Asynchronous Table Replication -- Select Destination Columns

Select Destination Columns

Press the add button to set the destination columns. Destination tables will be written to based on the Replication Schedule.

Database Link	Table Owner	Table Name	Projected Column Name	Options
DBSAMPLE5	SAMPLE	EMPLOYEE	ID, "NAME", ADDRESS	

20. Click **Add** and repeat steps 7 through 13 to add more columns, otherwise click **Next**.



Create Asynchronous Table Replication -- Final Review

Final review.

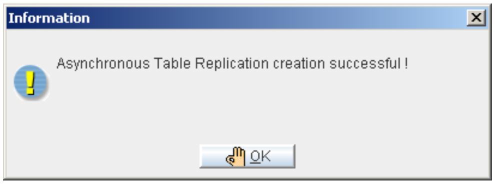
The SQL script will be executed when you press the finish button.

Create Asynchronous Table Replication SQL Script

```
create async replication ARPTEST with primary as
"SAMPLE"."EMPLOYEE" ( ID, NAME, ADDRESS ) replicate to
DBSAMPLE5 : "SAMPLE"."EMPLOYEE" ( ID, "NAME", ADDRESS )
```

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

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



Information

Asynchronous Table Replication creation successful !

22. Click **OK**.

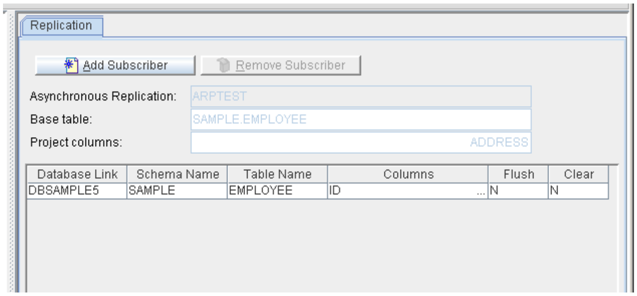
Adding Subscribers to an Asynchronous Replication

After creating a replication, users can be subscribed. 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.

➲ Adding 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** appears at the top of the page.



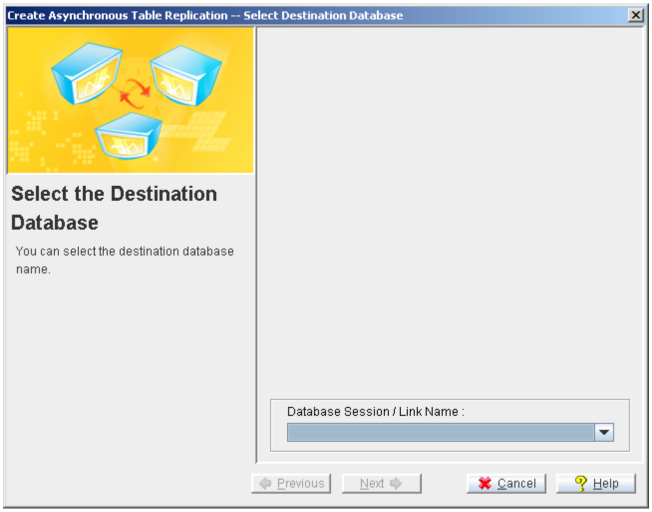
Asynchronous Replication: ARPTEST

Base table: SAMPLE.EMPLOYEE

Project columns: ADDRESS

Database Link	Schema Name	Table Name	Columns	Flush	Clear
DBSAMPLE5	SAMPLE	EMPLOYEE	ID ...	N	N

3. Click **Add Subscriber**. The **Select Destination Database** wizard appears.

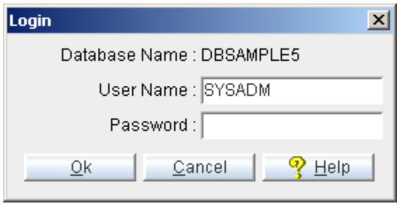

Create Asynchronous Table Replication -- Select Destination Database
Previous
Next
Cancel
Help

Select the Destination Database

You can select the destination database name.

Database Session / Link Name :

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

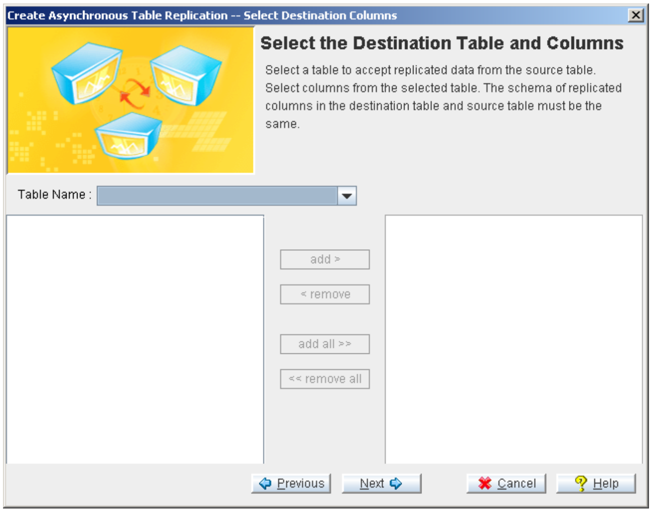


Database Name : DBSAMPLE5

User Name : SYSADM

Password :

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 appears.



Create Asynchronous Table Replication -- Select Destination Columns

Select the Destination Table and Columns

Select a table to accept replicated data from the source table. Select columns from the selected table. The schema of replicated columns in the destination table and source table must be the same.

Table Name :

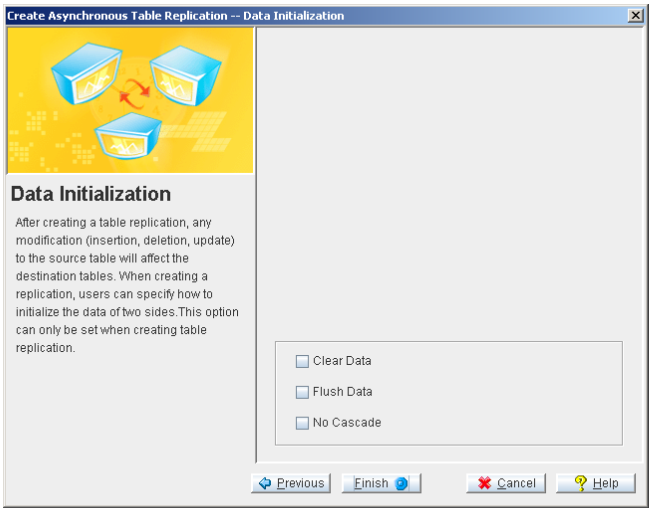
6. The **Select Destination Columns** window prompts for destination columns and a destination table selection.

7. Select a destination table from the **Table Name** menu. A list of available columns on the selected table appears 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 adds all columns. All the selected destination columns are 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. Click the **Next** button after selecting the desired columns.


Create Asynchronous Table Replication -- Data Initialization

Data Initialization

After creating a table replication, any modification (insertion, deletion, update) to the source table will affect the destination tables. When creating a replication, users can specify how to initialize the data of two sides.This option can only be set when creating table replication.

□ Clear Data

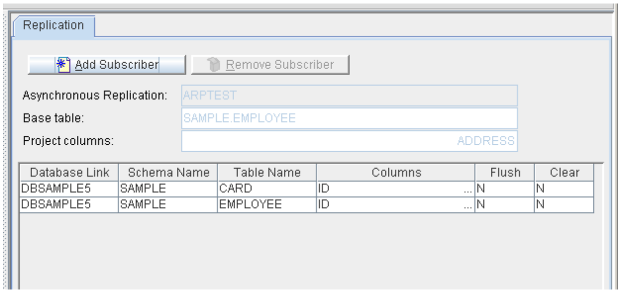
□ Flush Data

□ No Cascade

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 is deleted from the destination table 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.



Asynchronous Replication: ARPTEST

Base table: SAMPLE.EMPLOYEE

Project columns: ADDRESS

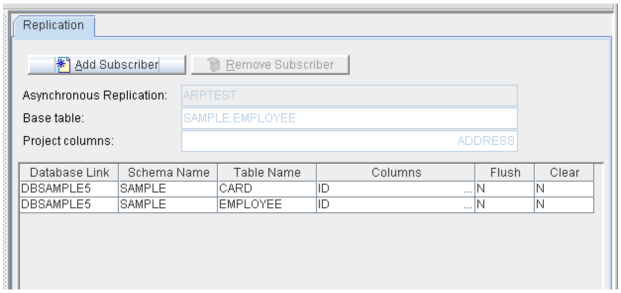
Database Link	Schema Name	Table Name	Columns	Flush	Clear
DBSAMPLE5	SAMPLE	CARD	ID ...	N	N
DBSAMPLE5	SAMPLE	EMPLOYEE	ID ...	N	N

16.3 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.

➲ Dropping a subscriber:

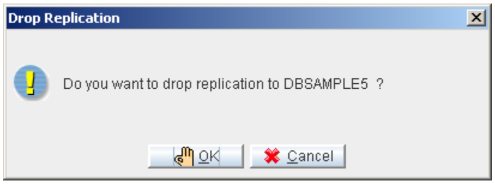
1. Select the **Replication** node from the tree. The tree will expand, showing a list of replications below the **Replication** node on the tree.
2. Select a replication from the tree. The replication page will show subscriber information about the selected database in a table. Furthermore, the buttons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



Asynchronous Replication: ARPTEST
Base table: SAMPLE.EMPLOYEE
Project columns: ADDRESS

Database Link	Schema Name	Table Name	Columns	Flush	Clear
DBSAMPLE5	SAMPLE	CARD	ID ...	N	N
DBSAMPLE5	SAMPLE	EMPLOYEE	ID ...	N	N

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.



Do you want to drop replication to DBSAMPLE5 ?

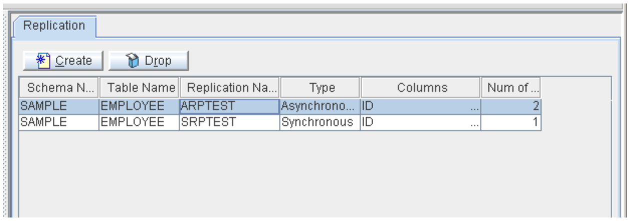
5. Click the **OK** button**.**

16.4 Dropping a Replication

Consider dropping a replication if there are no subscribers and no further need to replicate that data.

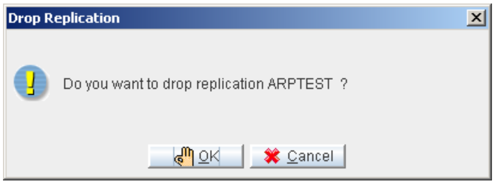
➲ Dropping a replication:

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.



Schema N...	Table Name	Replication Na...	Type	Columns	Num of ...
SAMPLE	EMPLOYEE	ARPTEST	Asynchrono...	ID ...	2
SAMPLE	EMPLOYEE	SRPTEST	Synchronous	ID ...	1

3. Click the **Drop** button. A conformation dialog box appears.



Do you want to drop replication ARPTEST ?

4. Click the **OK** button.

17 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. Domains may also specify a default value and a value constraint. When defining a column using a domain, the column inherits the properties of the domain (e.g., data type, default value, and value constraint) without requiring explicit specification.

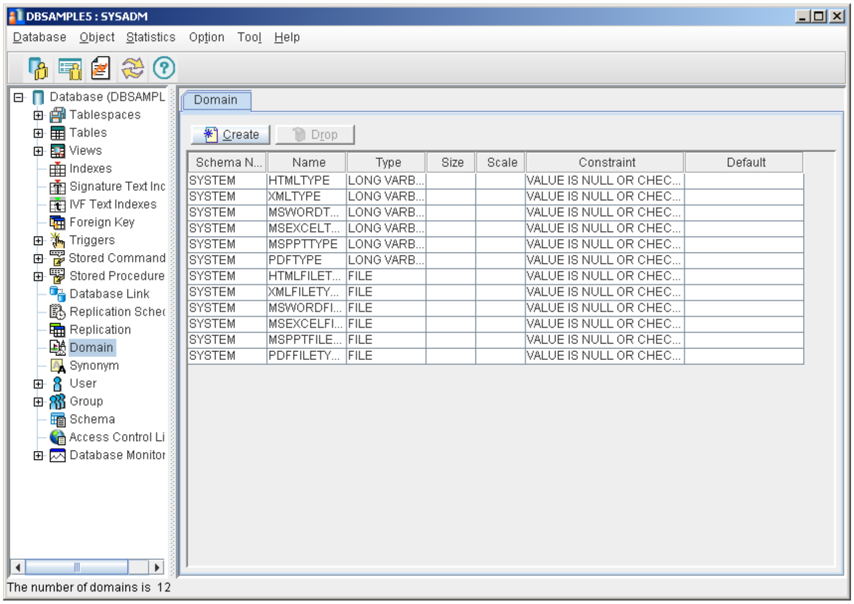
Specifying the default value and value constraint using domains achieves the same results as specifying them in a standard column definition. If you specify a default value for a column, it will override the default value specified in a domain. Value constraints specified in the column definition, are 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.

17.1 Creatng 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.

➲ Creating a domain:

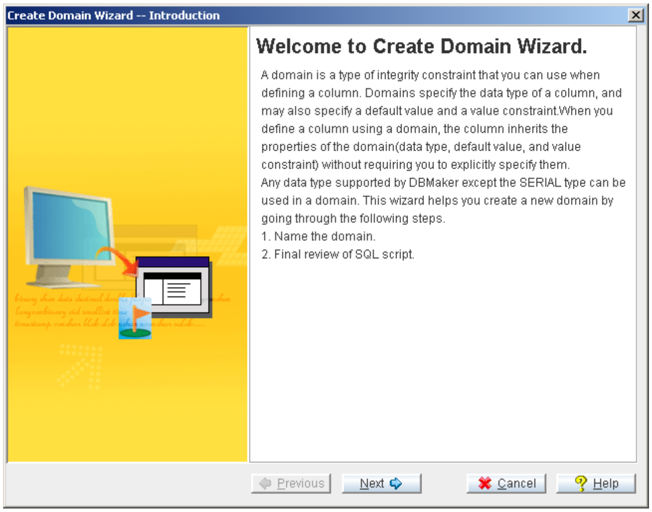
1. Select the object **Domain** from the tree. The following window appears.



Schema N...	Name	Type	Size	Scale	Constraint	Default
SYSTEM	HTMLTYPE	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	XMLTYPE	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	MSWORDT...	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	MSEXCELT...	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	MSPPTTYPE	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	PDFTYPE	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	HTMLFILET...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	XMLFILETY...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	MSWORDFI...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	MSEXCELFI...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	MSPPTFILE...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	PDFFILETY...	FILE			VALUE IS NULL OR CHEC...	

The number of domains is 12

2. Click the **Create** button. The **Create Domain Wizard** appears.

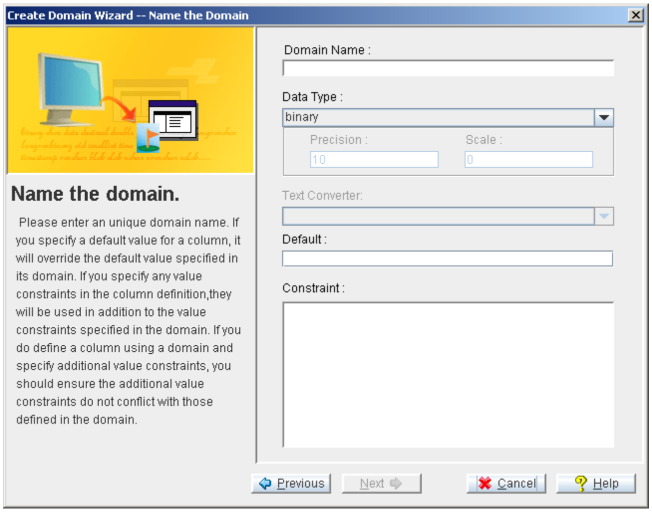


Create Domain Wizard -- Introduction

Welcome to Create Domain Wizard.

A domain is a type of integrity constraint that you can use when defining a column. Domains specify the data type of a 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.
Any data type supported by DBMaker except the SERIAL type can be used in a domain. This wizard helps you create a new domain by going through the following steps.
1. Name the domain.
2. Final review of SQL script.

3. Click the **Next** button.



Create Domain Wizard -- Name the Domain

Name the domain.

Please enter an unique domain name. If you specify a default value for a column, it will override the default value specified in its 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.

Domain Name :

Data Type : binary

Precision : 10

Scale : 0

Text Converter:

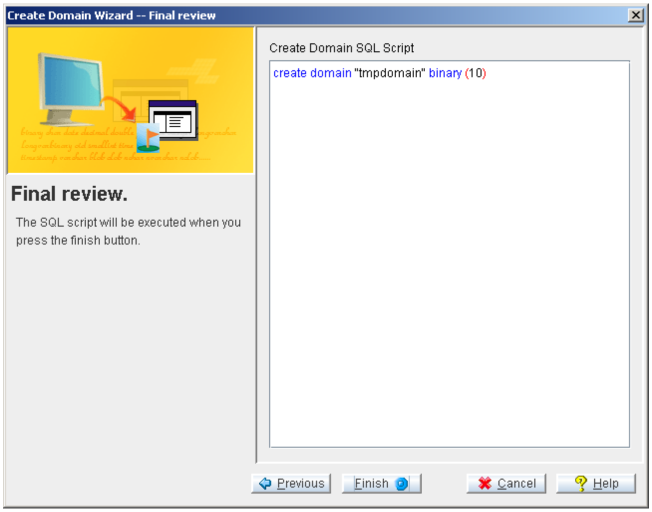
Default :

Constraint :

4. Enter the **Domain Name**.
5. Select a data type from the **Data Type** menu.
6. Enter **Precision** and **Scale** information corresponding to the selected data type.
7. Enter a **Default** value.

8. Enter the **Constraint** of the column. Constraints must comply with SQL syntax rules. Please see *Constraint Syntax* at the end of Chapter 4 for additional information.

9. Click the **Next** button.



Create Domain Wizard -- Final review

Final review.

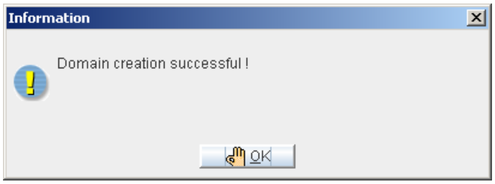
The SQL script will be executed when you press the finish button.

Create Domain SQL Script

```
create domain "tmpdomain" binary (10)
```

10. Check the SQL script and make changes if required.

11. Click the **Finish** button. The **Information** dialog box appears.



Information

Domain creation successful !

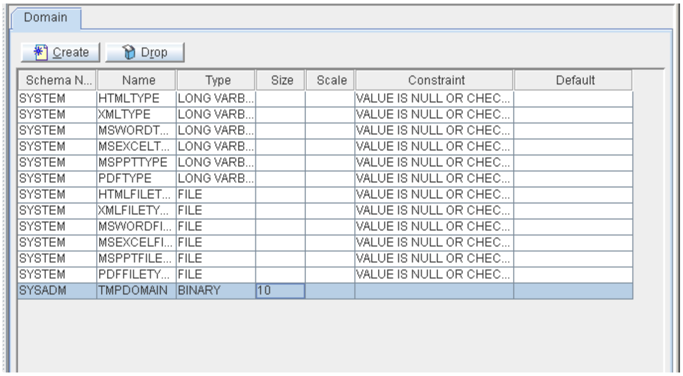
12. Click the **OK** button.

17.2 Dropping a Domain

Consider dropping domains that are no longer required.

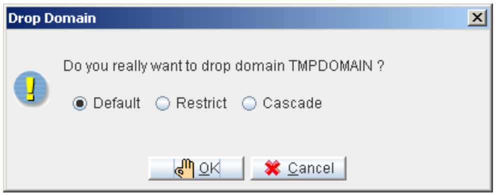
➲ Dropping a Domain:

1. Select the object **Domain** from the tree. The list of all the Domains in the database appears.
2. Select a domain.



Schema N...	Name	Type	Size	Scale	Constraint	Default
SYSTEM	HTMLTYPE	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	XMLTYPE	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	MSWORDT...	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	MSEXCELT...	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	MSPPTTYPE	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	PDFTYPE	LONG VARB...			VALUE IS NULL OR CHEC...	
SYSTEM	HTMLFILET...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	XMLFILETY...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	MSWORDFI...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	MSEXCELFI...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	MSPPTFILE...	FILE			VALUE IS NULL OR CHEC...	
SYSTEM	PDFFILETY...	FILE			VALUE IS NULL OR CHEC...	
SYSADM	TMPDOMAIN	BINARY	10			

3. Click the **Drop** button. A confirmation window appears.



Do you really want to drop domain TMPDOMAIN ?

● Default ○ Restrict ○ Cascade

4. There are three options for dropping domains. The **Default** option deletes the domain without considering dependent object within the domain. Dropping a domain renders dependent objects become invalid. The **Restrict** option considers table, domain, and procedure dependencies. This option does not permit a domain drop when one or more objects depend on the domain. The **Cascade** option drops the domain and all of its dependent objects. Click a radio button to select a domain dropping option.
5. Click the **OK** button.

18 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.

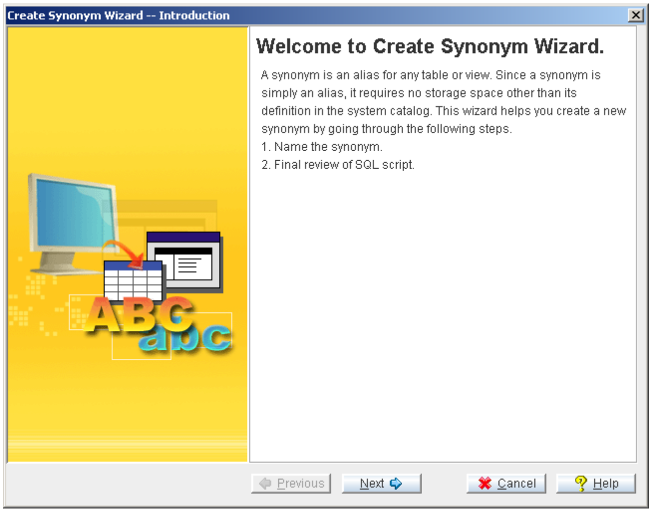
Synonyms are useful for simplifying a fully qualified table or view name. DBMaker normally identifies tables and views with fully qualified names that are composites of the owner and object names. By using a synonym, anyone can access a table or view through the corresponding synonym without having to 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 them.

18.1 Creating a Synonym

You can create a synonym for a table or view.

➲ Creating 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 appears.
3. Click **Next**. The **Name the Synonym** window appears.



Create Synonym Wizard -- Introduction

Welcome to Create Synonym Wizard.

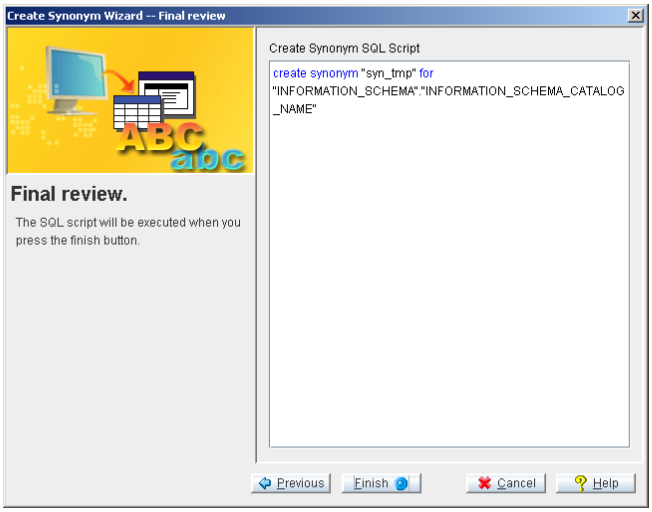
A synonym is an alias for any table or view. Since a synonym is simply an alias, it requires no storage space other than its definition in the system catalog. This wizard helps you create a new synonym by going through the following steps.

1. Name the synonym.

2. Final review of SQL script.

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

6. Click **Next**. The **Final Review** window appears.



Create Synonym Wizard -- Final review

Final review.

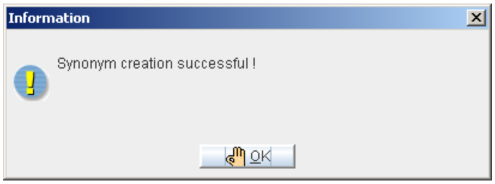
The SQL script will be executed when you press the finish button.

Create Synonym SQL Script

```
create synonym "syn_tmp" for
"INFORMATION_SCHEMA"."INFORMATION_SCHEMA_CATALOG
_NAME"
```

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

8. The **Information** dialog box appears.



Information

Synonym creation successful !

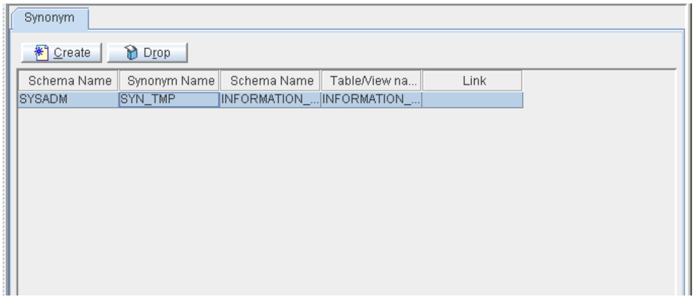
9. Click the **OK** button.

18.2 Dropping a Synonym

Consider dropping synonyms that are no longer required.

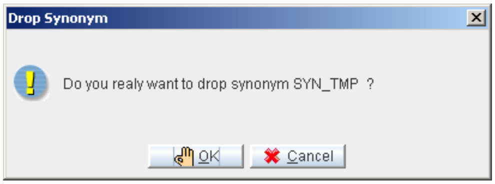
➲ Dropping a synonym:

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



Schema Name	Synonym Name	Schema Name	Table/View na...	Link
SYSADM	SYN_TMP	INFORMATION_...	INFORMATION_...	

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



Drop Synonym

Do you realy want to drop synonym SYN_TMP ?

3. Click **OK**.

19 Query by Example (QBE)

QBE is a query language included with DBMaker. The query by example (QBE) interface provides a streamline, user-friendly, and efficient interface for creating SQL query strings. The QBE interface guides you through the creation of SQL query strings. SQL programming knowledge is not required. Both advanced users and beginners will find the interface makes the creation of SQL query strings a quick and easy process by offering these functions:

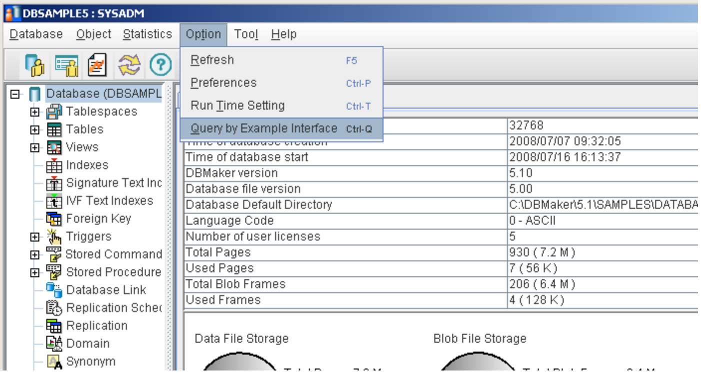
- Single and multi table queries
- Table/Column alias names
- WHERE CLAUSE improvement: LIKE, MATCH, CONTAIN, IN/NOT IN, IS NULL/IS NOT NULL, BETWEEN
- Wildcards
- DISTINCT function for removing duplicate records
- Projection column operations: +, -, ×, /, ||
- Projection column aggregate/UDF functions: AVG, COUNT, MAX, MIN, SUM
- GROUP BY/HAVING clause
- ORDER BY CLAUSE
- Manual SQL query editing

The QBE interface guides you through each step while creating an SQL string. Base the SQL string on a single table or multiple tables. The interface provides a complete list of all tables and views in the database.

In subsequent steps, columns are selected in tables that are affected by the query string. You can set an alias name for the tables or the columns. The GROUP BY clause will produce summary data within a group. The table and column order can also be set.

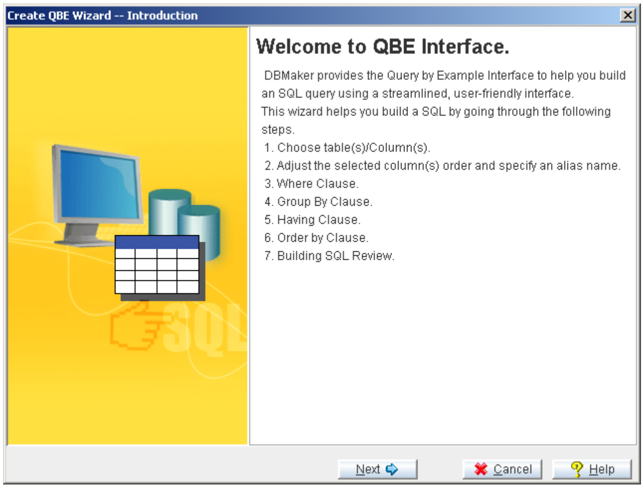
➲ Creating an SQL string using QBE:

1. Select **Query by Example Interface** from the **Option** menu.



	32768
Time of database creation	2008/07/07 09:32:05
Time of database start	2008/07/16 16:13:37
DBMaker version	5.10
Database file version	5.00
Database Default Directory	C:\DBMaker\5.1\SAMPLES\DATABA
Language Code	0 - ASCII
Number of user licenses	5
Total Pages	930 (7.2 M)
Used Pages	7 (56 K)
Total Blob Frames	206 (6.4 M)
Used Frames	4 (128 K)

2. The **Create QBE Wizard Introduction** dialog box appears.

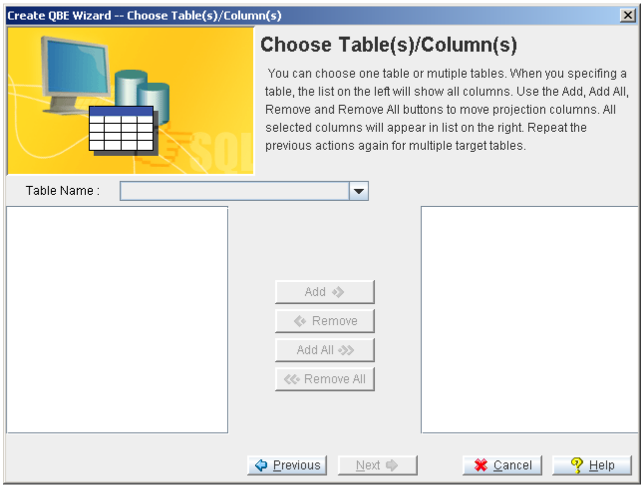


Create QBE Wizard -- Introduction

Welcome to QBE Interface.

DBMaker provides the Query by Example Interface to help you build an SQL query using a streamlined, user-friendly interface.
This wizard helps you build a SQL by going through the following steps.
1. Choose table(s)/Column(s).
2. Adjust the selected column(s) order and specify an alias name.
3. Where Clause.
4. Group By Clause.
5. Having Clause.
6. Order by Clause.
7. Building SQL Review.

3. Click the **Next** button. The **Choose Table(s)/Column(s)** window appears.



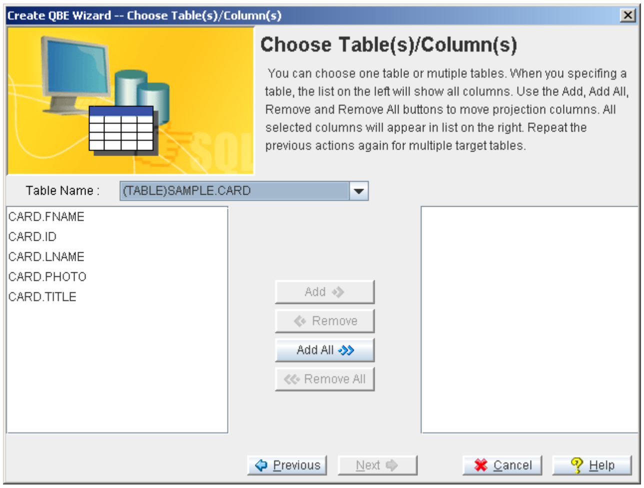
Create QBE Wizard -- Choose Table(s)/Column(s)

Choose Table(s)/Column(s)

You can choose one table or mutiple tables. When you specifing a table, the list on the left will show all columns. Use the Add, Add All, Remove and Remove All buttons to move projection columns. All selected columns will appear in list on the right. Repeat the previous actions again for multiple target tables.

Table Name :

4. Choose one table/view or multiple tables/views from the **Table Name** drop down list. When specifying a table or view, the columns contained in the selected table(s)/view(s) appear on the left side of the window.



Create QBE Wizard -- Choose Table(s)/Column(s)

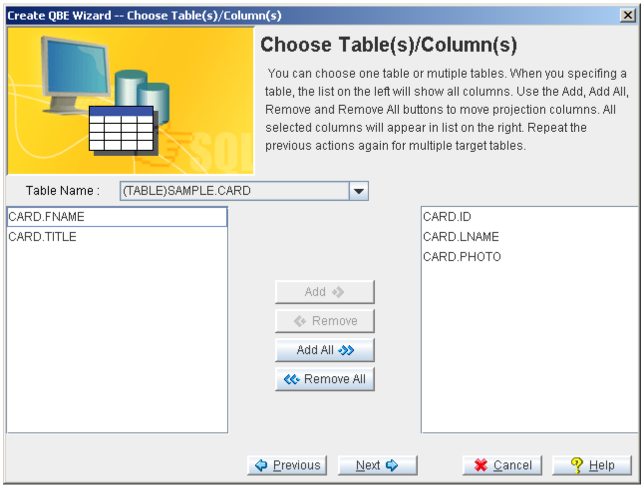
Choose Table(s)/Column(s)

You can choose one table or mutiple tables. When you specifing a table, the list on the left will show all columns. Use the Add, Add All, Remove and Remove All buttons to move projection columns. All selected columns will appear in list on the right. Repeat the previous actions again for multiple target tables.

Table Name : (TABLE)SAMPLE.CARD

CARD.FNAME
CARD.ID
CARD.LNAME
CARD.PHOTO
CARD.TITLE

5. Double-clicking a column name selects the column. Alternatively, click once on a column then click the **Add** button. The **Add all** button adds all columns to the replication. All the selected columns appear in the list box to the right. Likewise, columns are removed from the replication by double-clicking the column name or selecting the column and clicking the **Remove** button.



Create QBE Wizard -- Choose Table(s)/Column(s)

Choose Table(s)/Column(s)

You can choose one table or mutiple tables. When you specifing a table, the list on the left will show all columns. Use the Add, Add All, Remove and Remove All buttons to move projection columns. All selected columns will appear in list on the right. Repeat the previous actions again for multiple target tables.

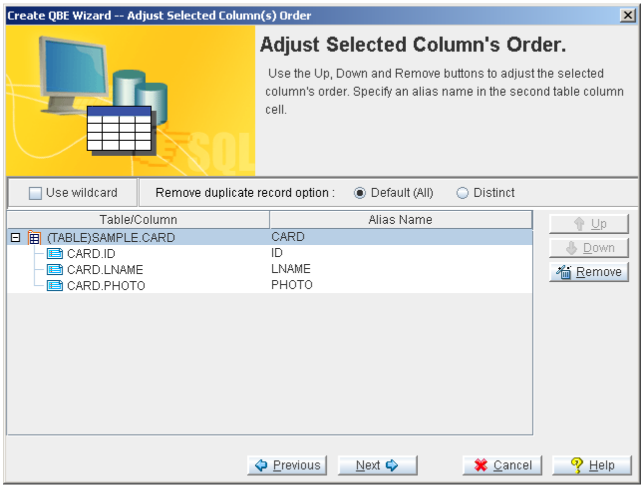
Table Name : (TABLE)SAMPLE.CARD

CARD.FNAME
CARD.TITLE

CARD.ID
CARD.LNAME
CARD.PHOTO

NOTE *To select multiple columns press and hold the **Ctrl** key and select the column with the mouse.*

6. Click the **Next** button. The Adjust **Selected Column's Order** dialog box appears.



Create QBE Wizard -- Adjust Selected Column(s) Order

Adjust Selected Column's Order.

Use the Up, Down and Remove buttons to adjust the selected column's order. Specify an alias name in the second table column cell.

□ Use wildcard

Remove duplicate record option : ● Default (All) ○ Distinct

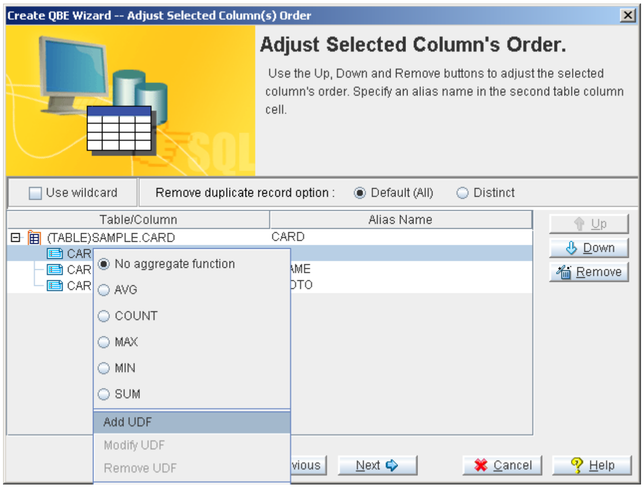
Table/Column	Alias Name
⊟ (TABLE)SAMPLE.CARD	CARD
CARD.ID	ID
CARD.LNAME	LNAME
CARD.PHOTO	PHOTO

7. Enabling the **Use wildcard** setting is helpful for fetching all the table's records using wildcards. Optionally click the **Use wildcard** checkbox to enable this feature.

8. Select a **Remove duplicate record** option. The **Default (ALL)** radio button fetches all records. In other words, duplicate records are not exclused from the results. The **Distinct** radio button option removes duplicate data records from the results of a query (i.e., only unique data record are included in the results).

9. To expand the table to view to include all of the available columns double-click the table or click the expanding icon (⊞) at the side of table name.

10. Select a column from the column list. The selected column highlights. Right-click the column to set an aggregate function or remove it from the projected column using a pop-up menu. QBE supports five aggregate functions: AVG, COUNT, MAX, MIN and SUM.



Create QBE Wizard -- Adjust Selected Column(s) Order

Adjust Selected Column's Order.

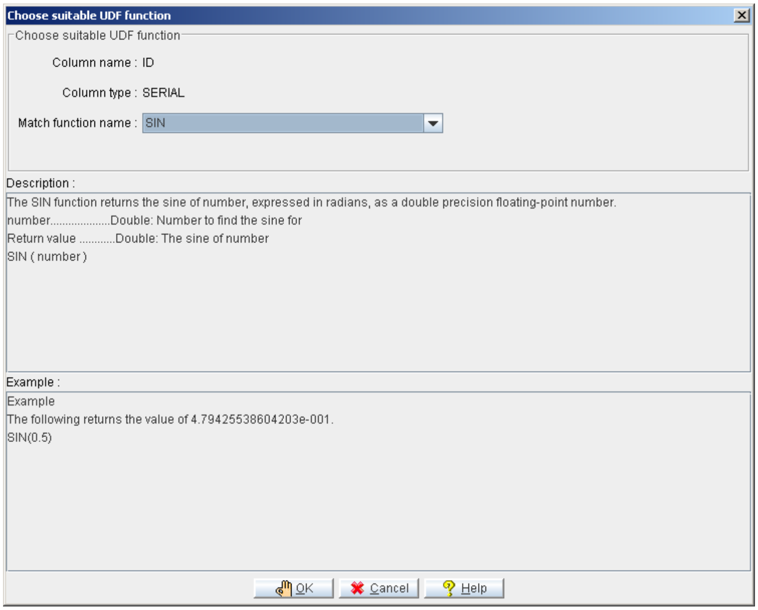
Use the Up, Down and Remove buttons to adjust the selected column's order. Specify an alias name in the second table column cell.

□ Use wildcard

Remove duplicate record option : ● Default (All) ○ Distinct

Table/Column	Alias Name
(TABLE)SAMPLE.CARD	CARD
CAR	
CAR	ME
CAR	OTO

11. QBE can apply UDF functions into projected columns. Click the **Add UDF** button and the following window appears. Column Name and Column Type fields show the selected column’s name and type. Select a UDF from the **Match Function Name** drop down menu, the selected UDF appears in the **Description** area. To help illustrate the UDF, examples appear in the **Example** area. Click the **OK** button.



Column name : ID

Column type : SERIAL

Match function name : SIN

Description :

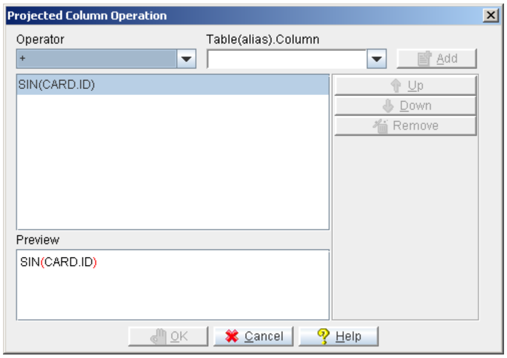
The SIN function returns the sine of number, expressed in radians, as a double precision floating-point number.
number...................Double: Number to find the sine for
Return value ....Double: The sine of number
SIN (number)

Example :

Example
The following returns the value of 4.79425538604203e-001.
SIN(0.5)

12. Click the **Modify UDF/Remove UDF** buttons to modify or remove the selected UDF function.

13. Click the **Setting** button. The **Projected Column Operation** window appears as shown below. You can choose one operator from the **Operator** drop down list. Choose one column from the **Table(alias).Column** drop down list. Click the **Add** button. The operation appears in the Preview area below the window. Use the **Up**, **Down** and **Remove** buttons to adjust the selected column's order. Click the **OK** button.



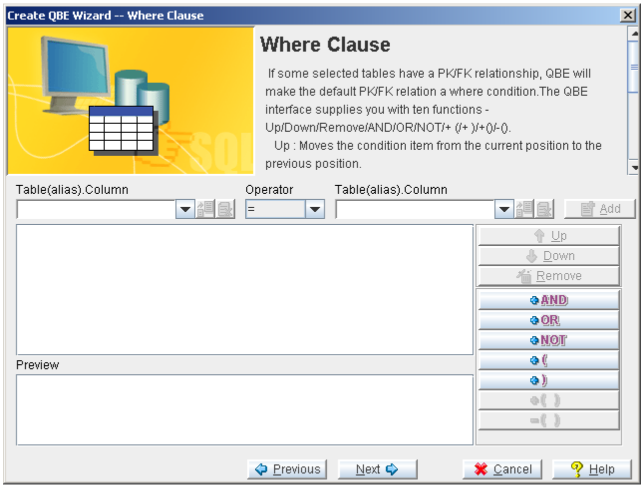
SIN(CARD.ID)

Preview: SIN(CARD.ID)

14. Double-click the **Alias Name** field for a column to specify an alias name.

15. Select a column from the **Table/Column** field using the mouse. The selected columns highlight. Click the **Up**, **Down,** or **Remove** buttons to adjust or drop the selected column.

16. Click the **Next** button. The **Where Clause** window appears.



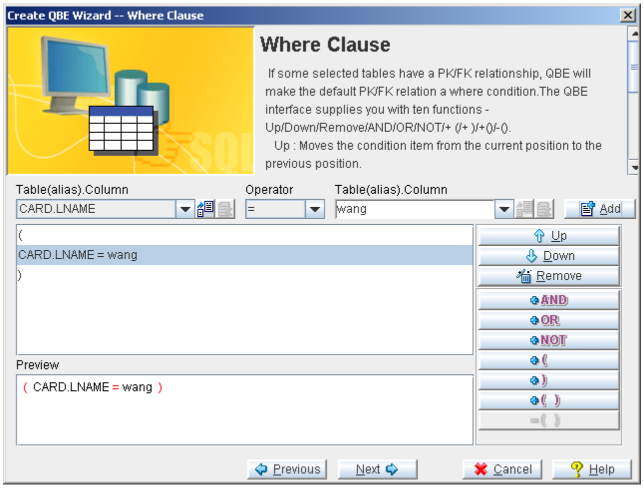
If some selected tables have a PK/FK relationship, QBE will make the default PK/FK relation a where condition.The QBE interface supplies you with ten functions - Up/Down/Remove/AND/OR/NOT/+ (/+)/+()/-().
Up : Moves the condition item from the current position to the previous position.

17. Select a column from the **Table(alias).Column(alias)** drop down list on the left.

18. Select an operator from the Operator drop down list. QBE supports following operators: =, >, <, >=, <=, <>, +, -, *, /, ||, between, is null/is not null, in/not in, like/not like, match/not match, and contain/not contain.

19. Enter a value in the **Table(alias).Column(alias)** field on the right or select a column from the drop down list.

20. Click the **Add** button. The where condition appears in the panel below the window.



Where Clause

If some selected tables have a PK/FK relationship, QBE will make the default PK/FK relation a where condition.The QBE interface supplies you with ten functions - Up/Down/Remove/AND/OR/NOT/+ (/+)/+()/-().
Up : Moves the condition item from the current position to the previous position.

Table(alias).Column: CARD.LNAME
Operator: =
Table(alias).Column: wang

(
CARD.LNAME = wang
)

Preview
(CARD.LNAME = wang)

NOTE *If some selected tables have a PK/FK relationship, QBE makes the default PK/FK relation a where condition.*

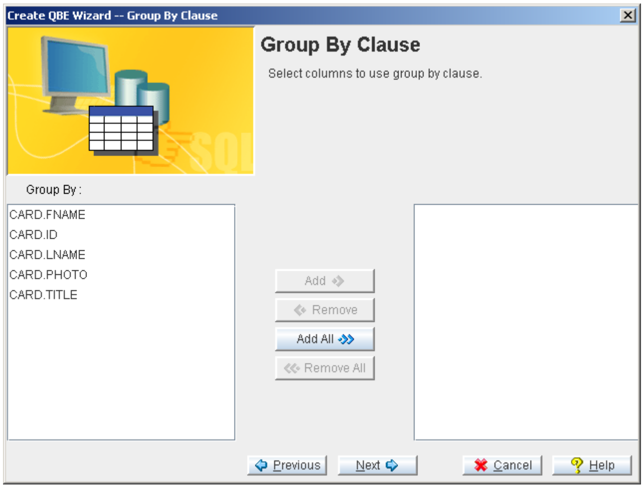
21. Optionally, add a UDF function by clicking the **Import** button () on the right of the **Table(alias).Column(alias)** field to add a UDF function. Click the **Export** button () to cancel the selected UDF function.

22. The QBE interface supports ten functions:

- Up: Moves the condition item from the current position to the previous position
- Down: Moves the condition item from current position to the next position
- Remove: Removes the selected condition item
- AND : Adds an AND operator into the next position
- OR: Adds an OR operator into the next position
- NOT: Adds a NOT operator into the previous position

- + (: Adds a left bracket into the previous position
- +): Adds a right bracket into the next position
- + (): Adds brackets into the selected condition item
- - (): Removes brackets from the selected condition item

23. Click the **Next** button. The **Group by Clause** window appears.



Create QBE Wizard -- Group By Clause

Group By Clause

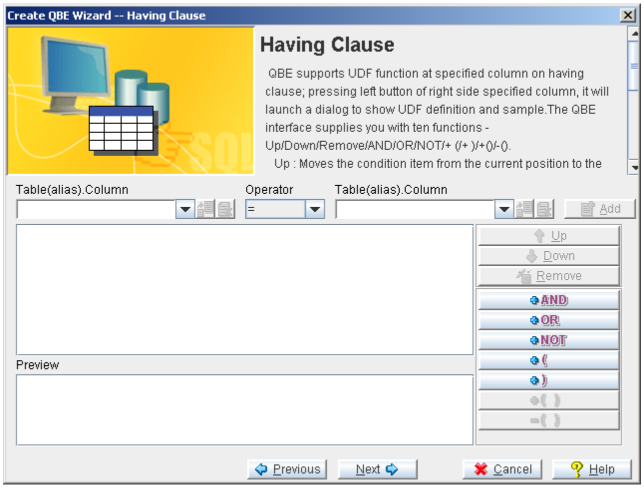
Select columns to use group by clause.

Group By :

CARD.FNAME
CARD.ID
CARD.LNAME
CARD.PHOTO
CARD.TITLE

24. Select columns to use **Group by Clause**. A group is a set of rows having the same values of group by columns.A single row of aggregate results is generated for each group.

25. Click the **Next** button. The **having clause** window appears. Please refer to steps 12 through 16 for details.



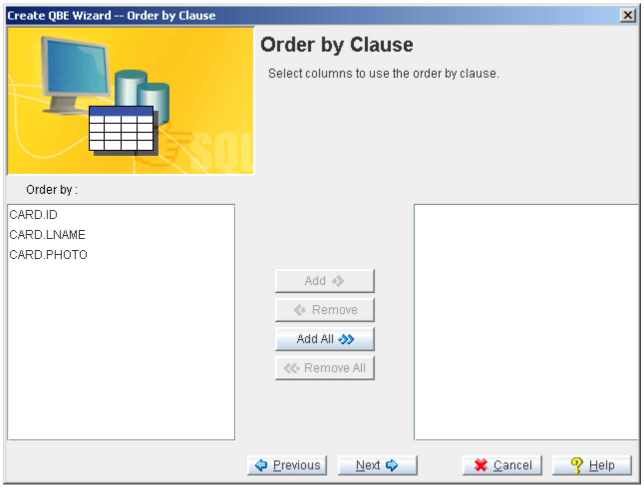
Create QBE Wizard -- Having Clause

Having Clause

QBE supports UDF function at specified column on having clause; pressing left button of right side specified column, it will launch a dialog to show UDF definition and sample.The QBE interface supplies you with ten functions - Up/Down/Remove/AND/OR/NOT/+ (/+)/+()/-().

Up : Moves the condition item from the current position to the

26. Click the **Next** button when you are satisfied with the clauses. The **Order by Clause** window appears.



Create QBE Wizard -- Order by Clause

Order by Clause

Select columns to use the order by clause.

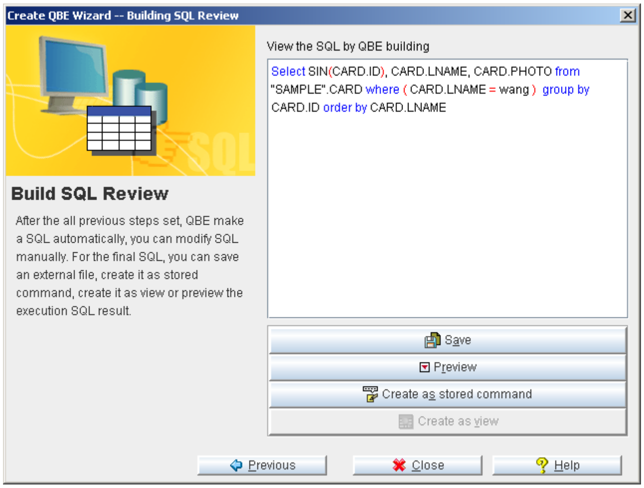
Order by :

CARD.ID
CARD.LNAME
CARD.PHOTO

27. Select the sort order for the columns using the **Order by Clause** dialog box. The order by clause can sort query results by the values contained in selected columns.

28. Click the **Next** button. The **Building SQL Review** dialog box appears. QBE make a SQL script automatically, you can modify it manually. For the final

SQL, you can save an external file, create it as stored command, create it as view, or preview the execution SQL result.



Create QBE Wizard -- Building SQL Review

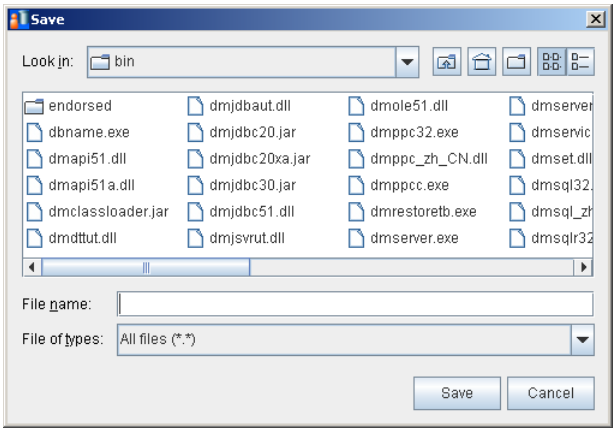
View the SQL by QBE building

```
Select SIN(CARD.ID), CARD.LNAME, CARD.PHOTO from
"SAMPLE".CARD where ( CARD.LNAME = wang )  group by
CARD.ID order by CARD.LNAME
```

Build SQL Review

After the all previous steps set, QBE make a SQL automatically, you can modify SQL manually. For the final SQL, you can save an external file, create it as stored command, create it as view or preview the execution SQL result.

29. Click the **Save** button to save the query. The **Save** dialog box appears.



Save

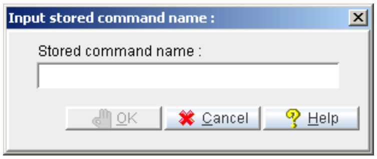
Look in: bin

File name:

File of types: All files (*.*)

- Enter a name and location for the query in the dialog box and click the **Save** button.

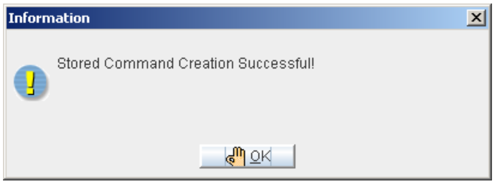
30. To save the query as a stored command, click the **Save as stored command** button. The **Input stored command** name dialog box appears.



Input stored command name :

Stored command name :

- Enter a name in the **Stored command name** field.
- Click the **OK** button. A message saying the stored command is created successfully is displayed.

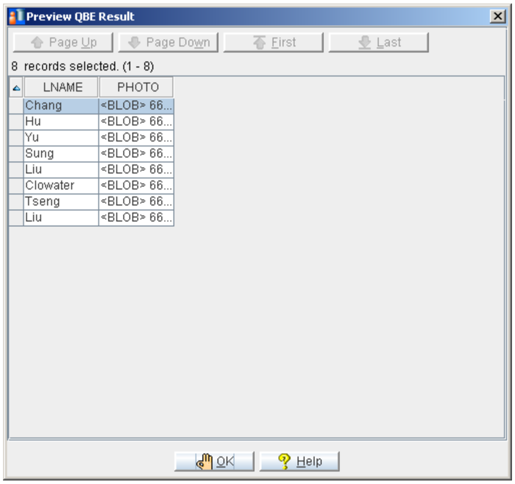


Information

Stored Command Creation Successful!

- Click **OK**. The created SQL command saves as stored command appears in the stored command list.

31. To preview the query, click the **Preview** button. The Preview **QBE Result** dialog box appears.

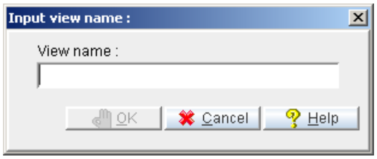


8 records selected. (1 - 8)

LNAME	PHOTO
Chang	<BLOB> 66...
Hu	<BLOB> 66...
Yu	<BLOB> 66...
Sung	<BLOB> 66...
Liu	<BLOB> 66...
Clowater	<BLOB> 66...
Tseng	<BLOB> 66...
Liu	<BLOB> 66...

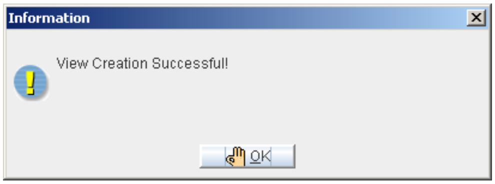
- Browse the results using the scroll bars. For large results use the **Page Up**, **Page Down**, **First** and **Last** buttons.

32. To create a view on the query, click the **Create as view** button. The **Input View Name** dialog box appears.



View name :

- Enter a name for the view in the View name field.
- Click **OK**. A message confirming successful creation appears.



View Creation Successful!

- Click the **OK** button.

33. Click **Previous** to return to prior windows or click the **Close** button when you are satisfied with the QBE.

20 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.

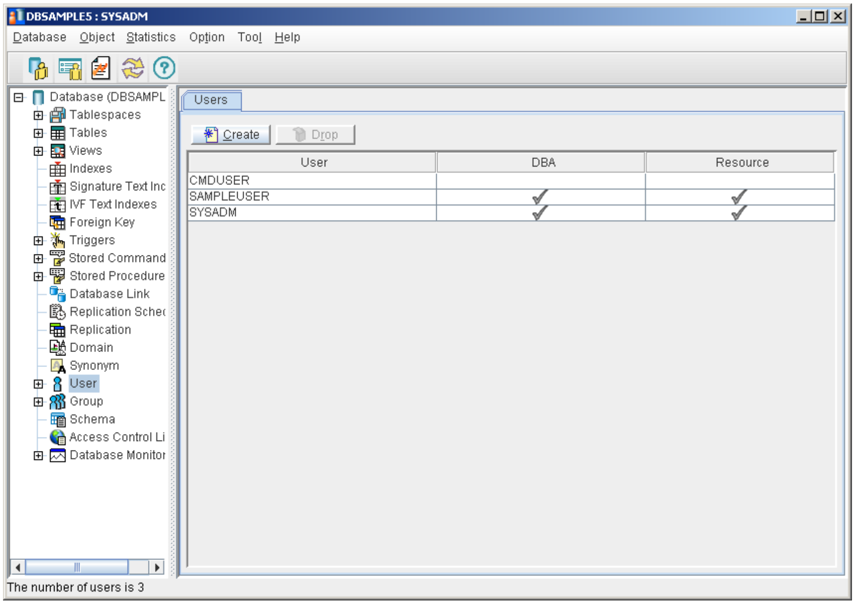
Object privileges include the SELECT, INSERT, DELETE, UPDATE, INDEX, ALTER, and REFERENCE for tables and views, and the EXECUTE privilege for stored commands and stored procedures. Object privileges may be granted or revoked by the user that created the object, or by a user with DBA authority or higher.

20.1 Creating a New User Account

Create new accounts when specific access profiles are 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. Both user names and password may contain letters, numbers, the underscore character, and the symbols $ and #. The first character may not be a number.

➲ Creating a new user account:

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

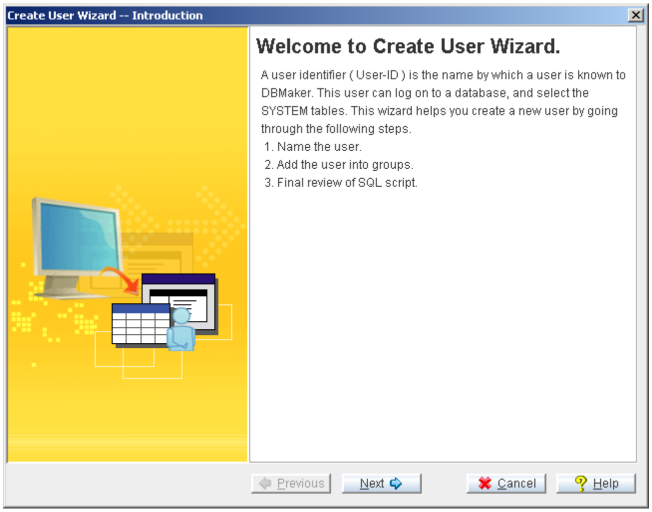


User	DBA	Resource
CMDUSER		
SAMPLEUSER	✓	✓
SYSADM	✓	✓

The number of users is 3

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

3. Click **Next**. The **Name the User** window appears.



Create User Wizard -- Introduction

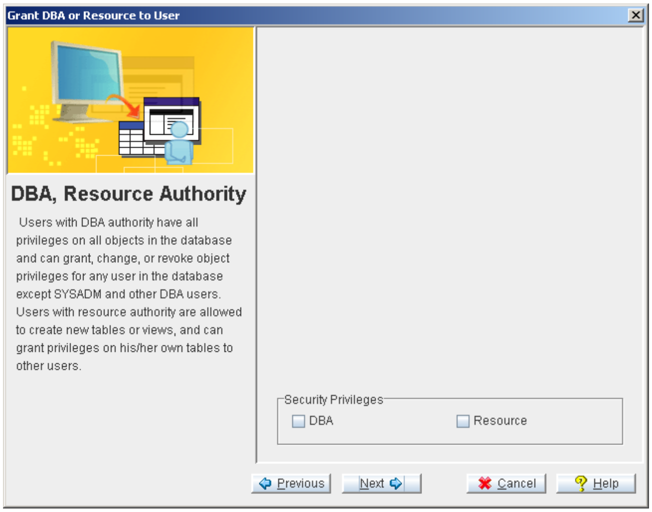
Welcome to Create User Wizard.

A user identifier (User-ID) is the name by which a user is known to DBMaker. This user can log on to a database, and select the SYSTEM tables. This wizard helps you create a new user by going through the following steps.

1. Name the user.
2. Add the user into groups.
3. Final review of SQL script.

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 appears.



Grant DBA or Resource to User

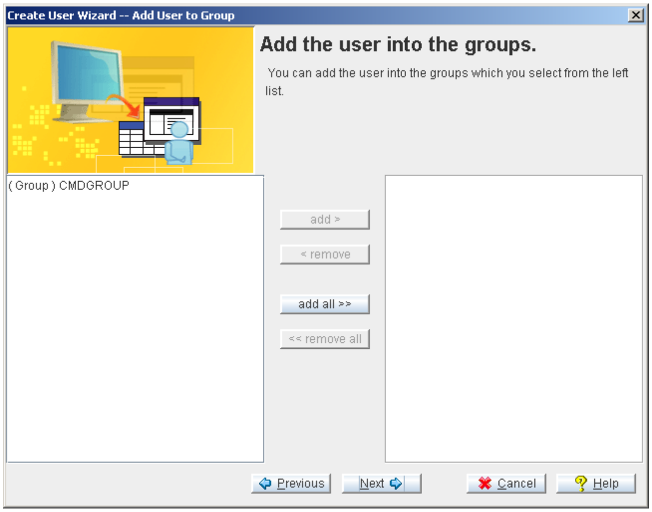
DBA, Resource Authority

Users with DBA authority have all privileges on all objects in the database and can grant, change, or revoke object privileges for any user in the database except SYSADM and other DBA users. Users with resource authority are allowed to create new tables or views, and can grant privileges on his/her own tables to other users.

Security Privileges

□ DBA □ Resource

6. Select the type of database authority the new user will have. Connect authority is granted to the new user if no authority level is selected. Click the **Next** button, the following window appears.



Create User Wizard -- Add User to Group

Add the user into the groups.

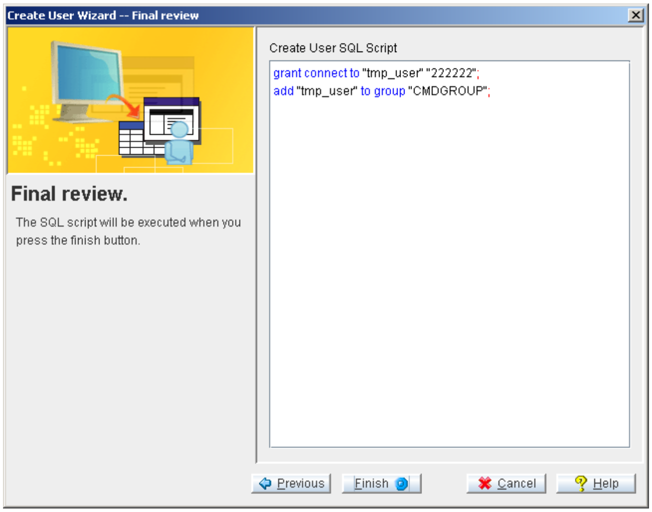
You can add the user into the groups which you select from the left list.

(Group) CMDGROUP

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

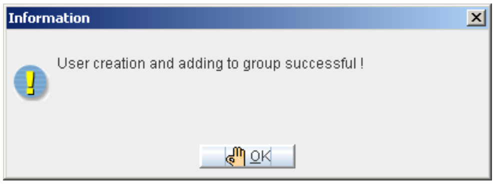
add button. The **add all** button can be used to select all groups to add the user into. All the selected groups are displayed in the list box to the right. Groups can be, likewise removed by double clicking on the group name, or by selecting the name and clicking the **remove** button.

8. Click the **Next** button, the **Final review** page appears.


Create User Wizard -- Final review
Final review.
The SQL script will be executed when you press the finish button.
Create User SQL Script
Previous
Finish
Cancel
Help

```
grant connect to "tmp_user" "222222";
add "tmp_user" to group "CMDGROUP";
```

9. Review the new SQL script. 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.



User creation and adding to group successful !

10. Click **OK**.

20.2 Changing Security Level

There are two security mechanisms in DBMaker. The first is user authority level and the second is 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.

Users who exist but are not 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.

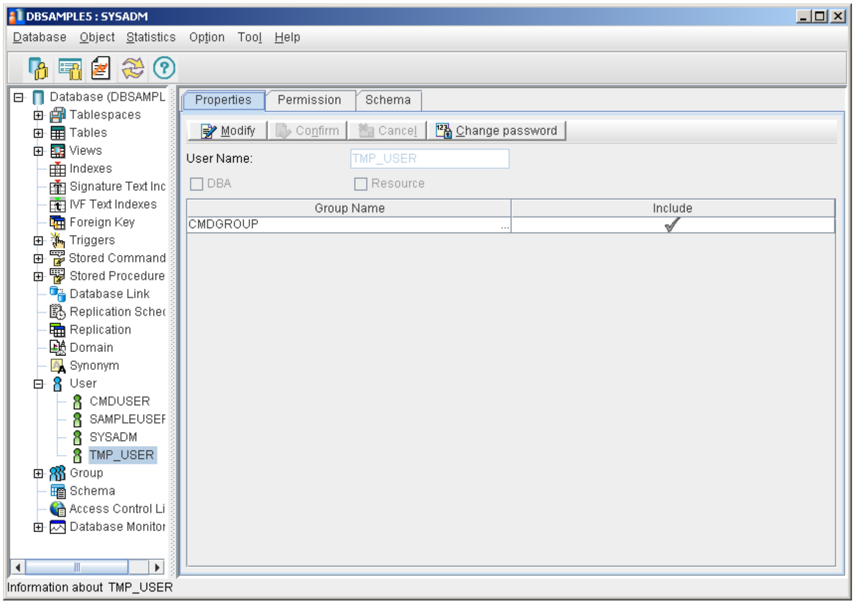
A user that is created and granted RESOURCE authority can create, alter, and drop tables, domains, and indexes. As the owner of any objects they create, users with RESOURCE authority may grant and revoke object privileges to other users and create synonyms and views for any objects they own.

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).

➲ Setting a user's authority level:

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

2. The users table displays the current authority level of the user. Double-click the user name in the tree or on the users table. The **Properties** page appears.



User Name: TMP_USER

□ DBA □ Resource

Group Name	Include
CMDGROUP	✓

Information about TMP_USER

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. This 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. It 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.

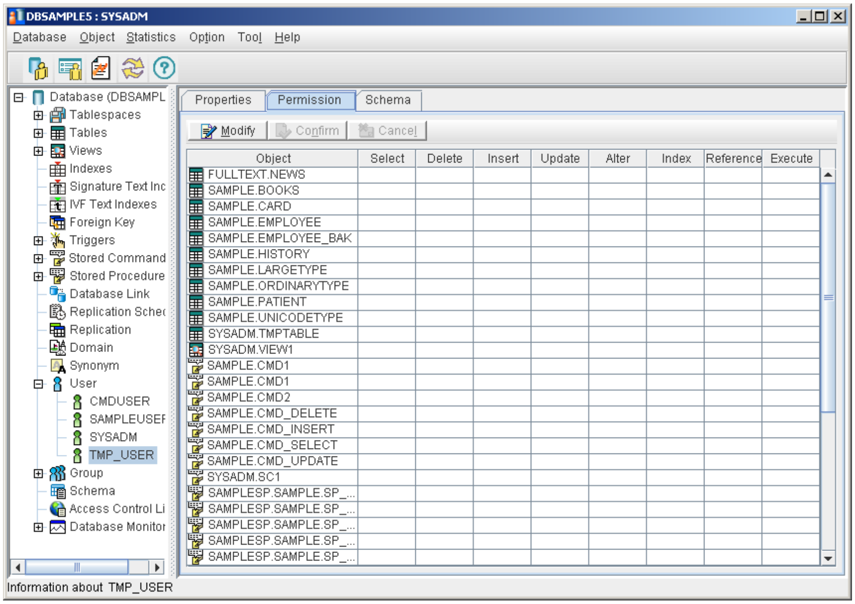
You cannot grant privileges on specific columns and on the entire database object at the same time. If you wish to do so you must use the command twice, once to grant privileges on specific columns, and once to grant privileges on the entire table. It is possible to grant object privileges to all users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

➲ Granting and removing object privileges:

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

2. Double-click the user name in the tree or on the users table. The **Properties** page appears.

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

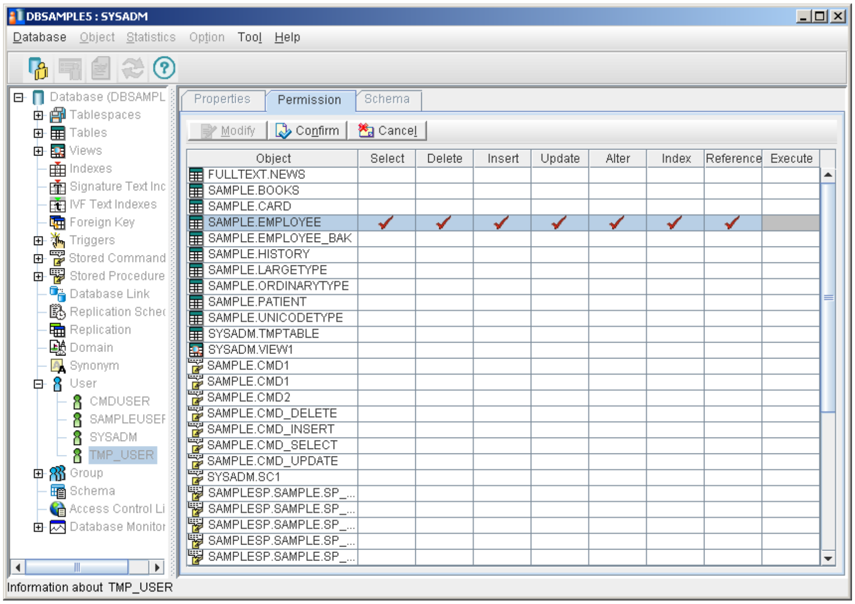


Object	Select	Delete	Insert	Update	Alter	Index	Reference	Execute
FULLTEXT.NEWS								
SAMPLE.BOOKS								
SAMPLE.CARD								
SAMPLE.EMPLOYEE								
SAMPLE.EMPLOYEE_BAK								
SAMPLE.HISTORY								
SAMPLE.LARGETYPE								
SAMPLE.ORDINARYTYPE								
SAMPLE.PATIENT								
SAMPLE.UNICODETYPE								
SYSADM.TMPTABLE								
SYSADM.VIEW1								
SAMPLE.CMD1								
SAMPLE.CMD1								
SAMPLE.CMD2								
SAMPLE.CMD_DELETE								
SAMPLE.CMD_INSERT								
SAMPLE.CMD_SELECT								
SAMPLE.CMD_UPDATE								
SYSADM.SC1								
SAMPLESP.SAMPLE.SP_...								
SAMPLESP.SAMPLE.SP_...								
SAMPLESP.SAMPLE.SP_...								
SAMPLESP.SAMPLE.SP_...								
SAMPLESP.SAMPLE.SP_...								

Information about TMP_USER

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.



Object	Select	Delete	Insert	Update	Alter	Index	Reference	Execute
FULLTEXT.NEWS								
SAMPLE.BOOKS								
SAMPLE.CARD								
SAMPLE.EMPLOYEE	✓	✓	✓	✓	✓	✓	✓	
SAMPLE.EMPLOYEE_BAK								
SAMPLE.HISTORY								
SAMPLE.LARGETYPE								
SAMPLE.ORDINARYTYPE								
SAMPLE.PATIENT								
SAMPLE.UNICODETYPE								
SYSADM.TMPTABLE								
SYSADM.VIEW1								
SAMPLE.CMD1								
SAMPLE.CMD1								
SAMPLE.CMD2								
SAMPLE.CMD_DELETE								
SAMPLE.CMD_INSERT								
SAMPLE.CMD_SELECT								
SAMPLE.CMD_UPDATE								
SYSADM.SC1								
SAMPLESP.SAMPLE.SP_...								
SAMPLESP.SAMPLE.SP_...								
SAMPLESP.SAMPLE.SP_...								
SAMPLESP.SAMPLE.SP_...								
SAMPLESP.SAMPLE.SP_...								

Information about TMP_USER

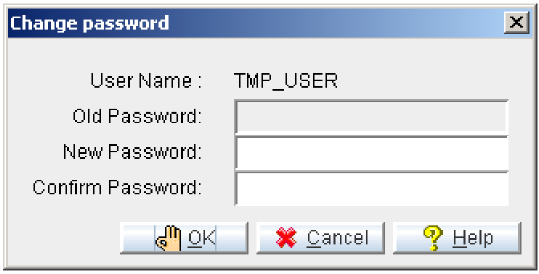
5. Click **Confirm** to save the changes, or **Cancel** to discard them.

20.3 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. Passwords have a maximum length of sixteen characters and may contain letters, numbers, the underscore character, and the symbols $ and #. The first character may not be a number.

➲ Changing a user’s password:

1. Open the **User** node on the object tree. The **Users** page appears.
2. Double click the user name in the tree or on the users table. The **Propertie**s page appears.
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.



User Name : TMP_USER

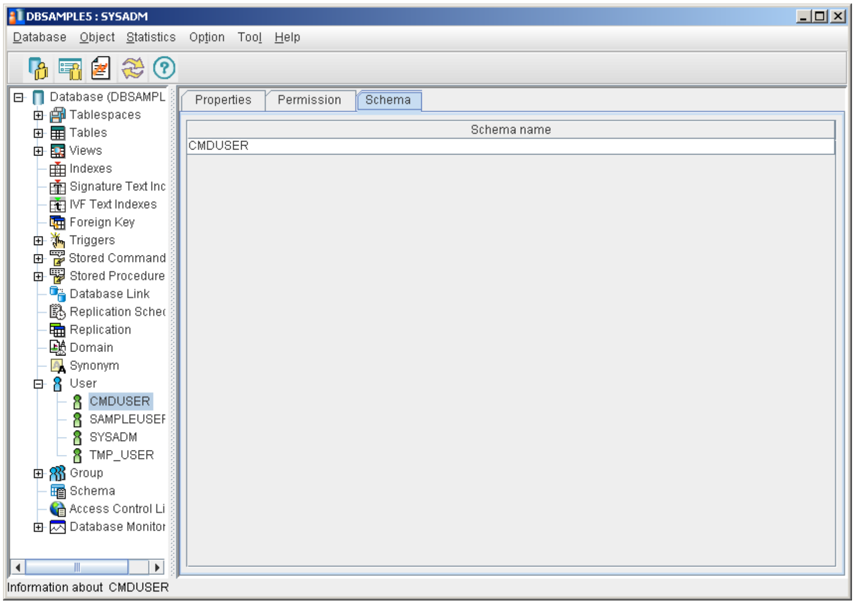
4. Click the **OK** button to change the password.

20.4 Browsing Assigned Schema

You can quickly browse the schemas that are assigned to each user in the Schema page.

➲ Browsing schema assigned to users:

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



Schema name
CMDUSER

Information about CMDUSER

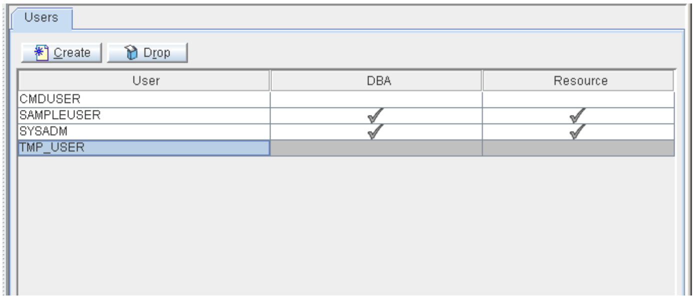
4. All of the schemas in the database that are assigned to the user are displayed on this page.

20.5 Deleting a User Account

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

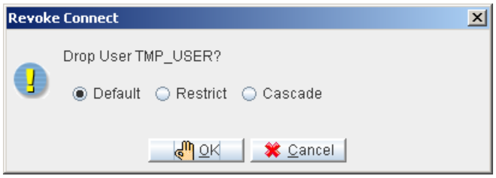
➲ Removing a user account from the database:

1. Open the **User** node on the object tree. The Users page appears.
2. Select the user account to be deleted from the table by clicking once (the user account will highlight blue).



User	DBA	Resource
CMDUSER		
SAMPLEUSER	✓	✓
SYSADM	✓	✓
TMP_USER		

3. Click **Drop**. A window will appear asking confirmation.



Drop User TMP_USER?

● Default ○ Restrict ○ Cascade

4. DBMaker provides three options for dropping user. The **Default** option drops the user without considering dependent objects. The **Restrict** option checks the user's schema and schema's table for dependencies. The **Cascad** option drops the user and the user’s shema and the schema’s table together.
5. Click the **OK** button to remove the user from the database.

21 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 are granted gain those object privileges in addition to object privileges are granted to them directly. Members removed from a group lose object privileges that are granted to the group, but retain any privileges that are to them directly.

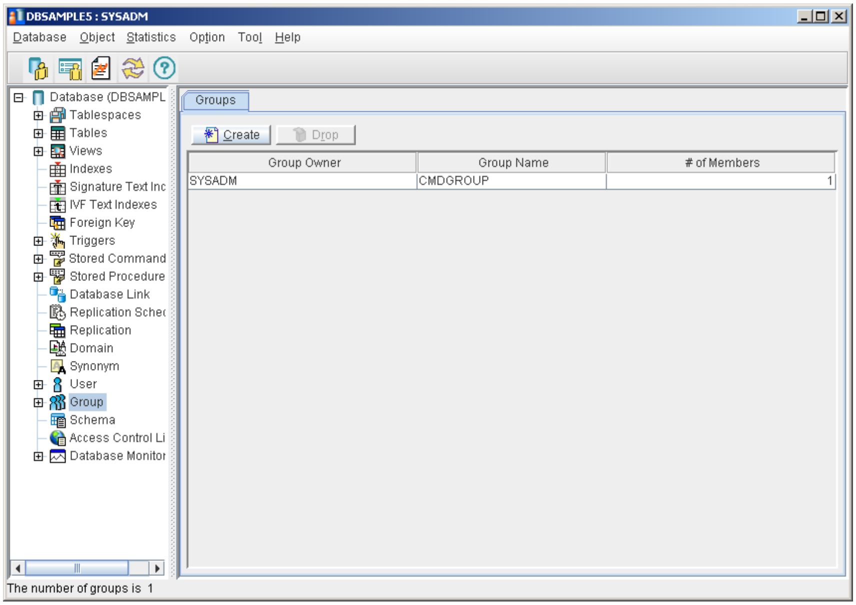
You can also specify a group name in place of a user name, as long as the group you are trying to add does not contain the group you are adding it to. User and group names have a maximum length of eight characters, and may contain letters, numbers, the underscore character, and the symbols $ and #. The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

21.1 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 or business.

➲ Creating a new group:

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

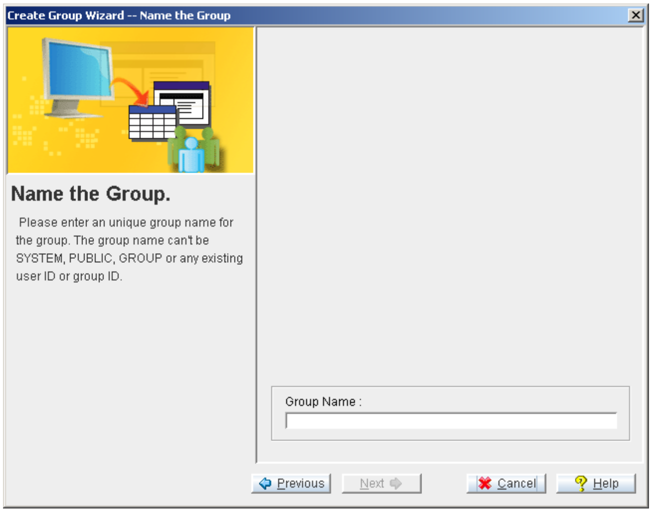


Group Owner	Group Name	# of Members
SYSADM	CMDGROUP	1

The number of groups is 1

2. Click **Create**. The **Create New Group** wizard starts.

3. Click **Next**. The **Name the Group** window appears.



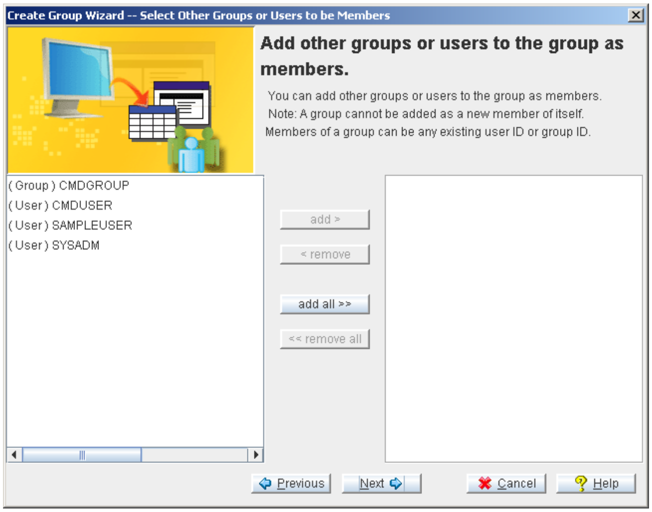
Create Group Wizard -- Name the Group

Name the Group.

Please enter an unique group name for the group. The group name can't be SYSTEM, PUBLIC, GROUP or any existing user ID or group ID.

Group Name :

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



Create Group Wizard -- Select Other Groups or Users to be Members

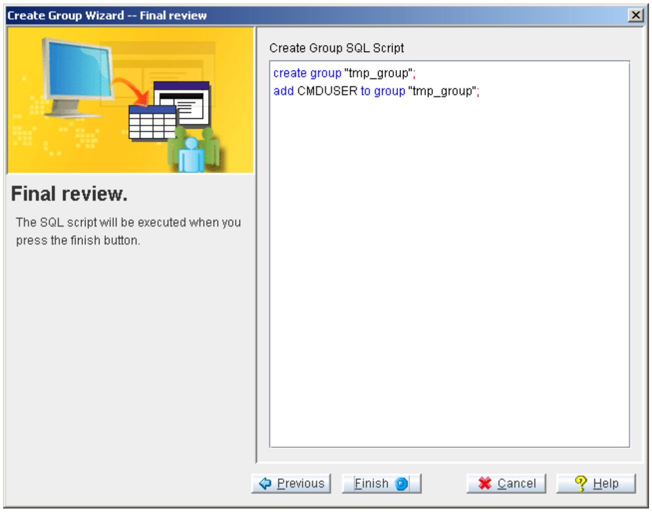
Add other groups or users to the group as members.

You can add other groups or users to the group as members.
Note: A group cannot be added as a new member of itself.
Members of a group can be any existing user ID or group ID.

(Group) CMDGROUP
(User) CMDUSER
(User) SAMPLEUSER
(User) SYSADM

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 are 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 appears.



Create Group Wizard -- Final review

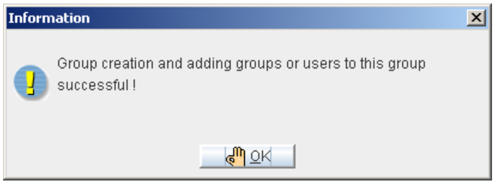
Final review.

The SQL script will be executed when you press the finish button.

Create Group SQL Script

```
create group "tmp_group";
add CMDUSER to group "tmp_group";
```

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 appears.



Information

Group creation and adding groups or users to this group successful !

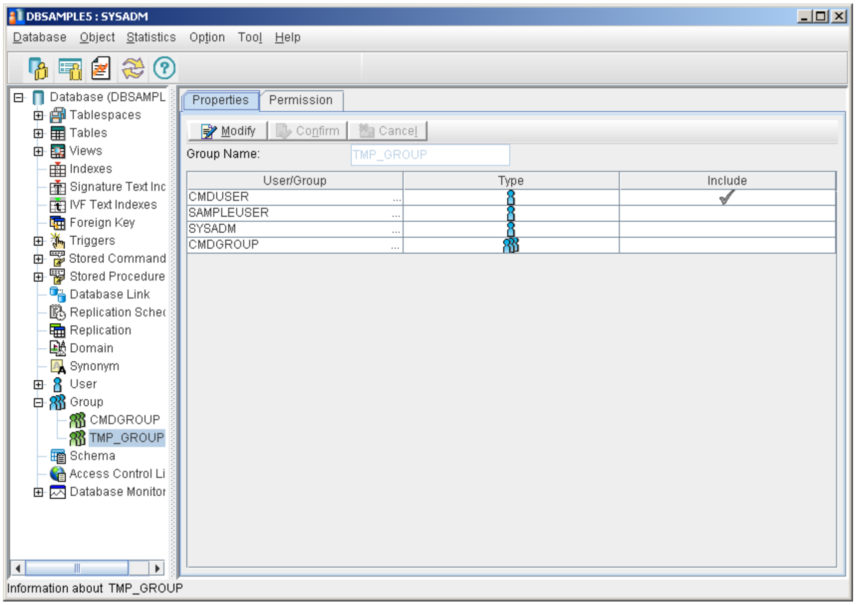
7. Click **OK**.

21.2 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.

➲ Adding and removing 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.



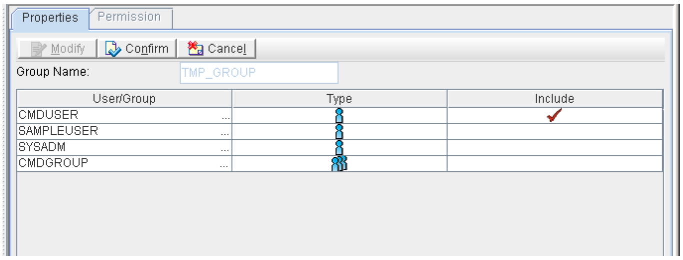
Group Name: TMP_GROUP

User/Group	Type	Include
CMDUSER		✓
SAMPLEUSER		
SYSADM		
CMDGROUP		

Information about TMP_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.



Group Name: TMP_GROUP

User/Group	Type	Include
CMDUSER ...	👤	✓
SAMPLEUSER ...	👤	
SYSADM ...	👤	
CMDGROUP ...	👥	

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.

21.3 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.

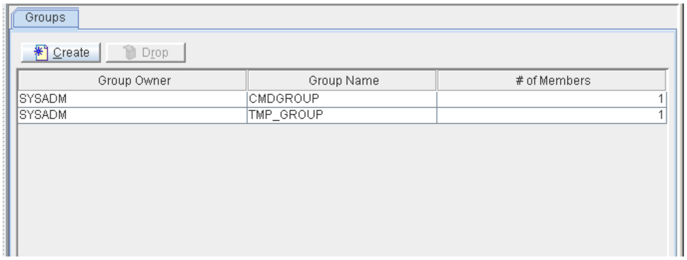
REFERENCE privilege allows you to create referential constraints (such as foreign keys) on a database object. This privilege can also be restricted to specific columns.

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.

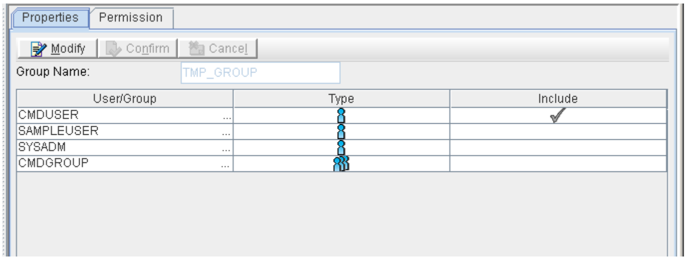
➲ Granting and removing group object privileges:

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.



Group Owner	Group Name	# of Members
SYSADM	CMDGROUP	1
SYSADM	TMP_GROUP	1

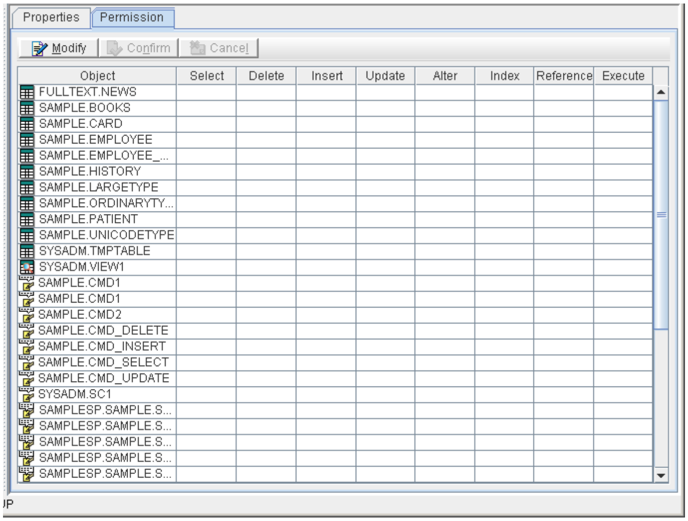
2. Double-click the group name in the tree or in the group table. The **Properties** page appears.



Group Name: TMP_GROUP

User/Group	Type	Include
CMDUSER ...	👤	✓
SAMPLEUSER ...	👤	
SYSADM ...	👤	
CMDGROUP ...	👥	

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



Object	Select	Delete	Insert	Update	Alter	Index	Reference	Execute
FULLTEXT.NEWS								
SAMPLE.BOOKS								
SAMPLE.CARD								
SAMPLE.EMPLOYEE								
SAMPLE.EMPLOYEE_...								
SAMPLE.HISTORY								
SAMPLE.LARGETYPE								
SAMPLE.ORDINARYTY...								
SAMPLE.PATIENT								
SAMPLE.UNICODETYPE								
SYSADM.TMPTABLE								
SYSADM.VIEW1								
SAMPLE.CMD1								
SAMPLE.CMD1								
SAMPLE.CMD2								
SAMPLE.CMD_DELETE								
SAMPLE.CMD_INSERT								
SAMPLE.CMD_SELECT								
SAMPLE.CMD_UPDATE								
SYSADM.SC1								
SAMPLESP.SAMPLE.S...								
SAMPLESP.SAMPLE.S...								
SAMPLESP.SAMPLE.S...								
SAMPLESP.SAMPLE.S...								
SAMPLESP.SAMPLE.S...								

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

22 Controling Access

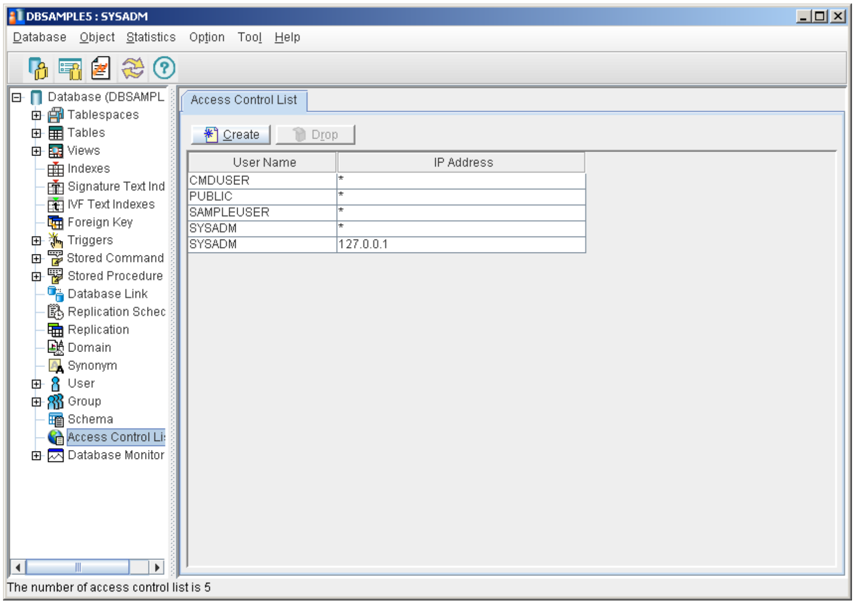
Access Control List provides an ability to determine whether allow some users from some IPs to connect to the database. DBA can use it to allow or block some users come from any network address. You can easily create or drop the Access Control List with the JTOOL interface. After creating a new database, there are three defaulted tuples in the list. They are (PUBLIC,*, allow) (SYSADM, 127.0.0.1, allow) and (SYSADM,*, allow), respectively. Please notice that the record (SYSADM,127.0.0.1,allow) cannot be removed because the SYSADM is always allowed to connect to database from the local machine.

22.1 Broswing Access Control List

You can quickly browse the Access Control List (ACL) in a database by selecting the Access Control List icon from the data structure tree. The schema's name and owner appearin the panel to the right of the data structure tree.

➲ Browing ACLs:

1. Click the Access Control List icon from the tree. A list of all users' name and the IP address in the database appears.



User Name	IP Address
CMDUSER	*
PUBLIC	*
SAMPLEUSER	*
SYSADM	*
SYSADM	127.0.0.1

The number of access control list is 5

2. Browse the listed on the right panel by using the upward and downward arrows.

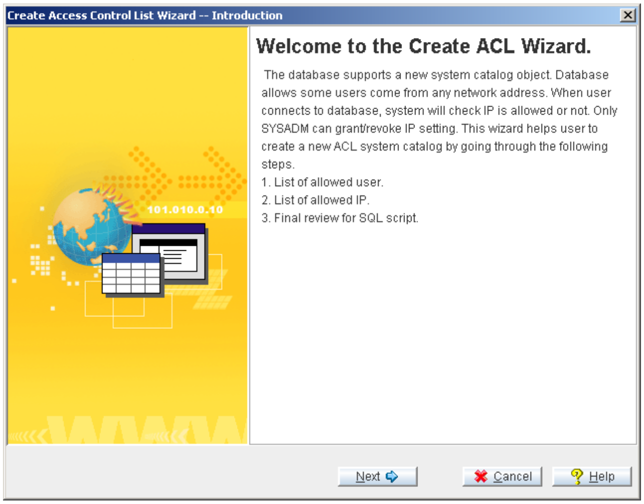
22.2 Creating Access Controling List

Creating a schema using JDBATool is a very intuitive process. You are guided through the process by a user-friendly user interface.

DBMaker uses the new keyword “PUBLIC” to mean that all users that want to connect to database. DBA cannot create a user name with “PUBLIC.”

➲ Creating an ACL:

1. Click the Access Control List icon from the tree.The Access Control List page appears.
2. Click the Create button in the Access Control List window. The Create Access Control List Wizard Introduction window appears.



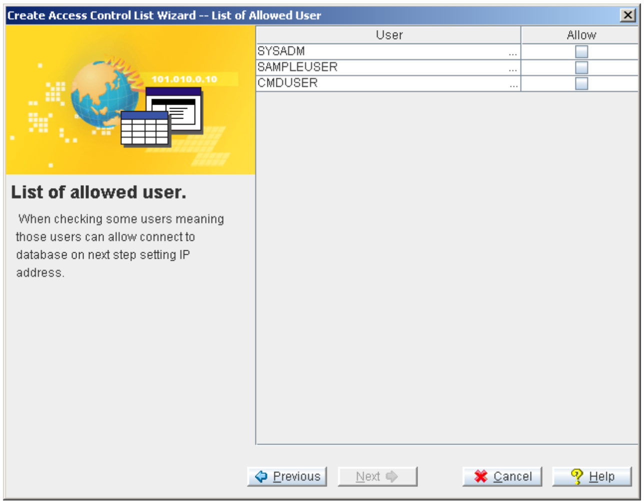
Create Access Control List Wizard -- Introduction

Welcome to the Create ACL Wizard.

The database supports a new system catalog object. Database allows some users come from any network address. When user connects to database, system will check IP is allowed or not. Only SYSADM can grant/revoke IP setting. This wizard helps user to create a new ACL system catalog by going through the following steps.

1. List of allowed user.
2. List of allowed IP.
3. Final review for SQL script.

3. Click the **Next** button in the **Create Access Control List** Wizard Introduction window. The List of Allowed User window appears.

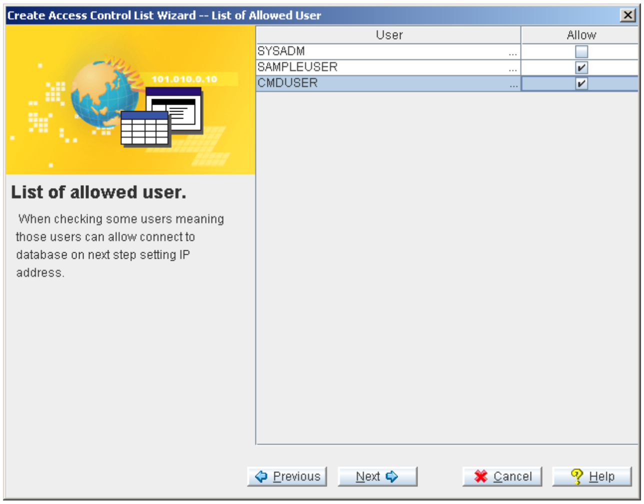

Create Access Control List Wizard -- List of Allowed User

User	Allow
SYSADM ...	□
SAMPLEUSER ...	□
CMDUSER ...	□

List of allowed user.

When checking some users meaning those users can allow connect to database on next step setting IP address.

4. Choose the user you want to allow Allow field.

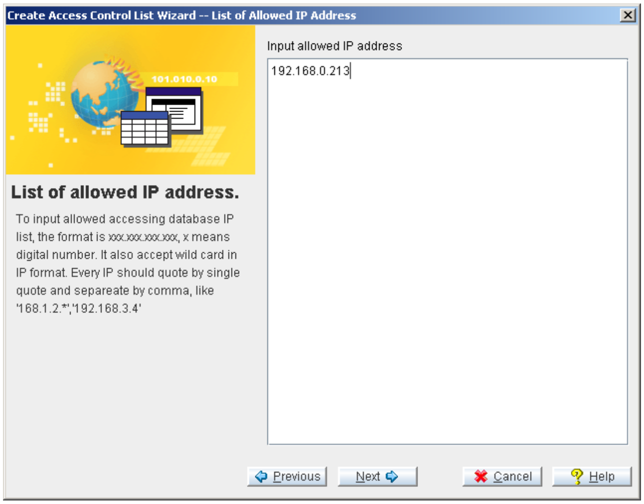

Create Access Control List Wizard -- List of Allowed User

User	Allow
SYSADM ...	□
SAMPLEUSER ...	☑
CMDUSER ...	☑

List of allowed user.

When checking some users meaning those users can allow connect to database on next step setting IP address.

5. Click the **Next** button in the List of Allowed User window. The List of Allowed IP Address window appears.



Create Access Control List Wizard -- List of Allowed IP Address

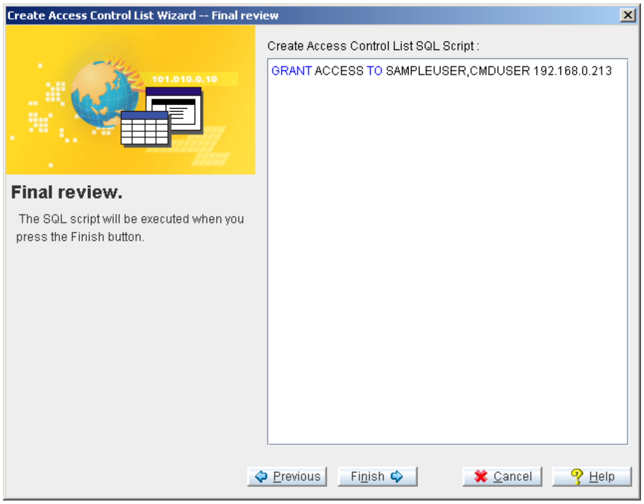
Input allowed IP address

192.168.0.213

List of allowed IP address.

To input allowed accessing database IP list, the format is xxx.xxx.xxx.xxx, x means digital number. It also accept wild card in IP format. Every IP should quote by single quote and separeate by comma, like '168.1.2.*','192.168.3.4'

6. Input the IP address you want to allow in the Input allowed IP address field.
7. Click the **Next** button the Final review window appears. From this window the user can edit the SQL code in the Create Schema SQL Script field



Create Access Control List Wizard -- Final review

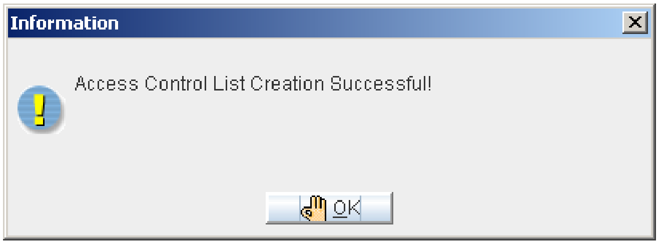
Create Access Control List SQL Script :

```
GRANT ACCESS TO SAMPLEUSER,CMDUSER 192.168.0.213
```

Final review.

The SQL script will be executed when you press the Finish button.

8. Click the **Finish** button when you are satisfied with the result SQL Script. A conformation message appears stating that the creation of the ACL was successful. The created Access Control List appears in the Access Control List window to the right.



Information

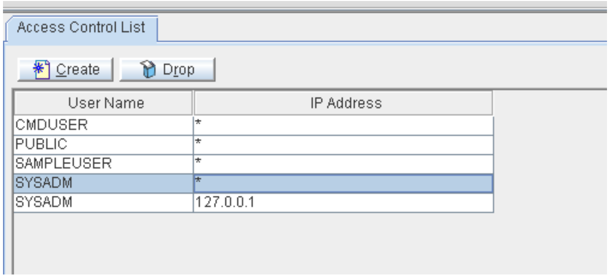
Access Control List Creation Successful!

22.3 Dropping Access Controling List

Consider dropping an ACL if it is no longer required.

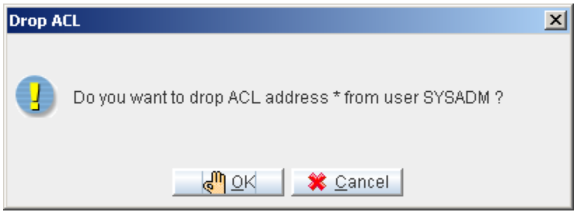
➲ Dropping an ACL:

1. Select the object Access Controling List from the tree and select the ACL from the list.



User Name	IP Address
CMDUSER	*
PUBLIC	*
SAMPLEUSER	*
SYSADM	*
SYSADM	127.0.0.1

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



Do you want to drop ACL address * from user SYSADM ?

3. Click the **OK** button.

23 Data Transfer

The Data Transfer Tool provides a user-friendly interface for transferring data in and out of the database. The tool performs the following functions:

- Import from text
- Import from XML file
- Import from ODBC
- Export to text
- Export to XML
- Batch transfer

A wizard performs each type of data transformation. 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 5.1>DataTransfer, or opened from within JDBA Tool. It consists of a main console and a menu bar, as illustrated in *Figure 23-1*. 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 changes the language that the UI is displayed in. Currently English, Japanese, and Chinese (traditional) are supported. The help menu provides access to the help system for JDBA Tool.

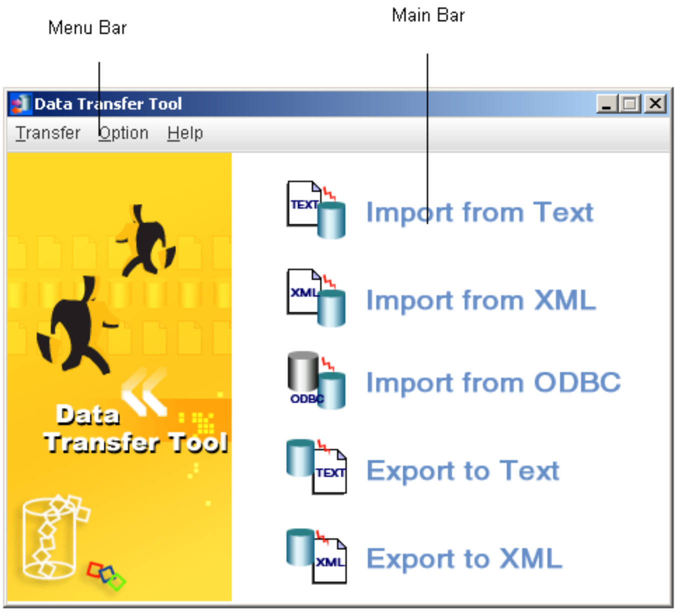


Figure 23-1 The Data Transfer Tool main window

➲ Opening 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 appears.

23.1 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 column delimiter characters include: 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.

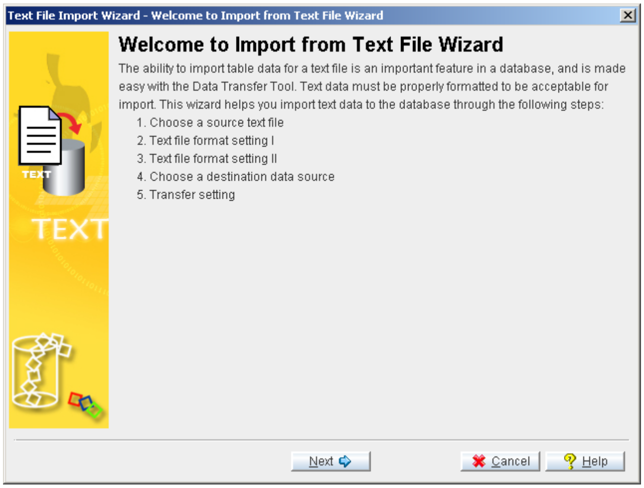
➲ Example

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
12:47:05.100';'C:\\DBMAKER\\5.1\\BIN\\WEBDB\\FO\\ZZ00000B.GIF';"10000000000000000
000"9;10.250
'A_HOWARD            ';'blobtmpdir2\\blbtmpf1.txt';'2001-09-22
10:14:21.000';'C:\\DBMAKER\\5.1\\BIN\\WEBDB\\FO\\ZZ00000C.GIF';"20000000000000000
000"9;13.550
'A_HOWARD            ';'blobtmpdir2\\blbtmpf2.txt';'2001-10-04
16:22:06.000';'C:\\DBMAKER\\5.1\\BIN\\WEBDB\\FO\\ZZ00000D.GIF';"30000000000000000
000"9;27.333
```

➲ Importing 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



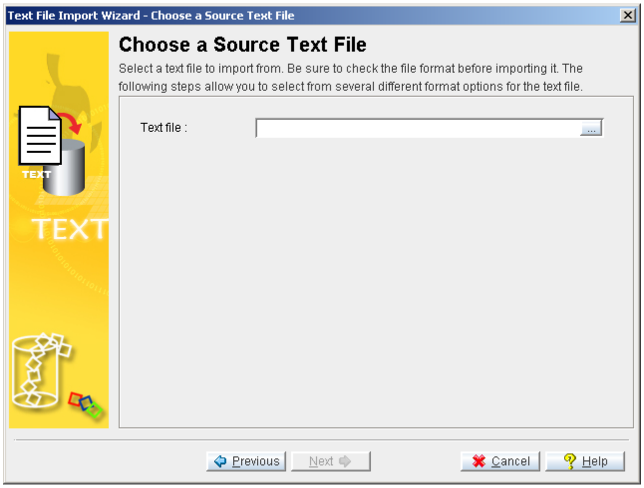
Text File Import Wizard - Welcome to Import from Text File Wizard

Welcome to Import from Text File Wizard

The ability to import table data for 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. This wizard helps you import text data to the database through the following steps:

1. Choose a source text file
2. Text file format setting I
3. Text file format setting II
4. Choose a destination data source
5. Transfer setting

3. Click **Next**. The **Choose a Source Text File** window appears.



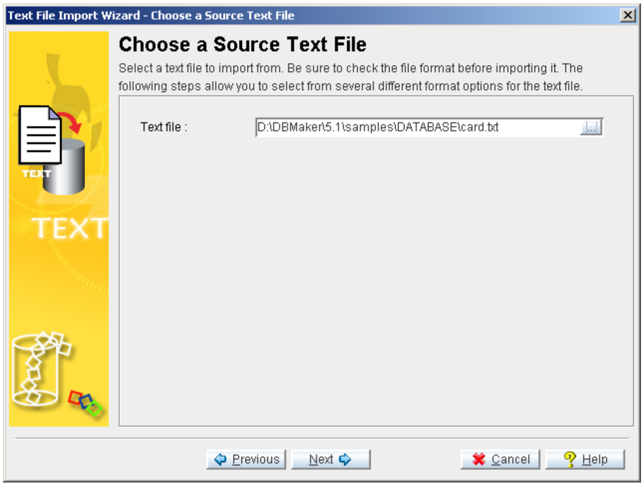
Text File Import Wizard - Choose a Source Text File

Choose a Source Text File

Select a text file to import from. Be sure to check the file format before importing it. The following steps allow you to select from several different format options for the text file.

Text file :

4. Enter the full path of a text file to import or click the browse button to search for a text file.



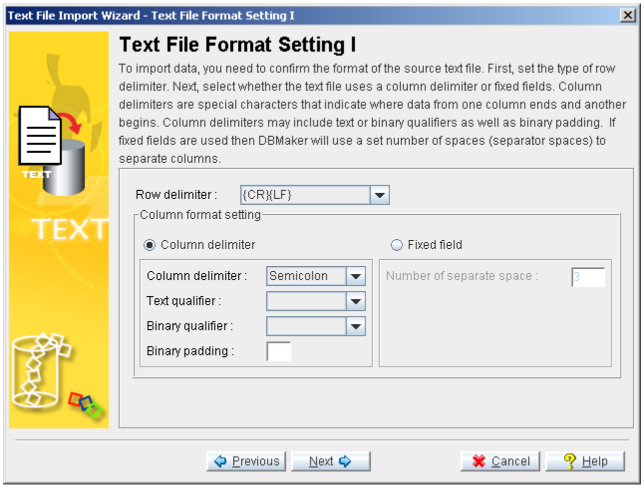
Text File Import Wizard - Choose a Source Text File

Choose a Source Text File

Select a text file to import from. Be sure to check the file format before importing it. The following steps allow you to select from several different format options for the text file.

Text file : D:\DBMaker\5.1\samples\DATABASE\card.txt

5. After you have selected a text file, click **Next**, the **Text File Format Setting 1** window appears.



Text File Import Wizard - Text File Format Setting I

Text File Format Setting I

To import data, you need to confirm the format of the source text file. First, set the type of row delimiter. Next, select whether the text file uses a column delimiter or fixed fields. Column delimiters are special characters that indicate where data from one column ends and another begins. Column delimiters may include text or binary qualifiers as well as binary padding. If fixed fields are used then DBMaker will use a set number of spaces (separator spaces) to separate columns.

Row delimiter : {CR}{LF}

Column format setting

● Column delimiter ○ Fixed field

Column delimiter : Semicolon

Text qualifier :

Binary qualifier :

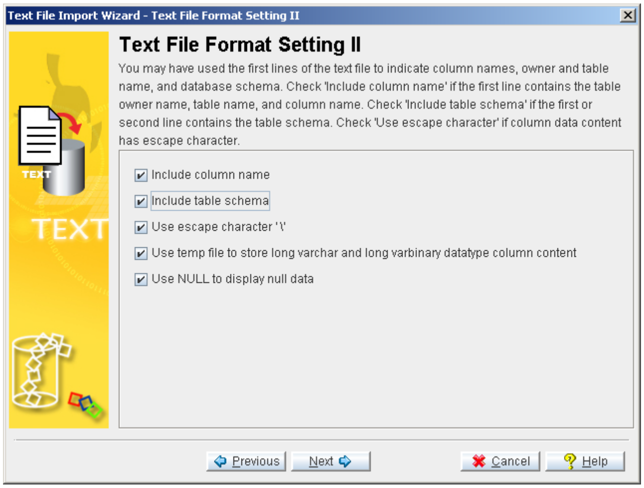
Binary padding :

Number of separate space : 3

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 appears.



Text File Import Wizard - Text File Format Setting II

Text File Format Setting II

You may have used the first lines of the text file to indicate column names, owner and table name, and database schema. Check 'Include column name' if the first line contains the table owner name, table name, and column name. Check 'Include table schema' if the first or second line contains the table schema. Check 'Use escape character' if column data content has escape character.

☑ Include column name

☑ Include table schema

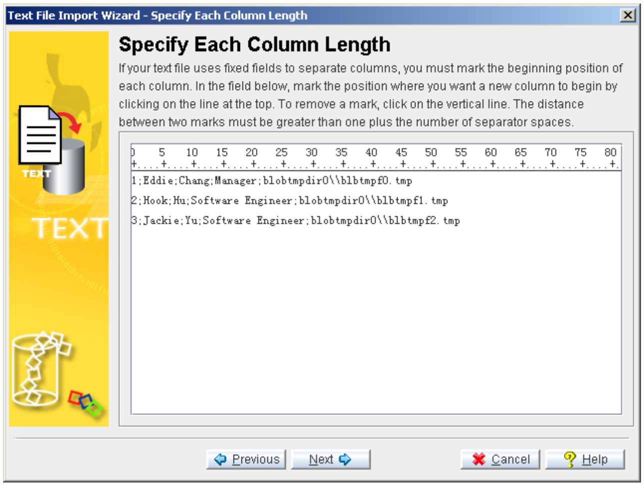
☑ Use escape character '\'

☑ Use temp file to store long varchar and long varbinary datatype column content

☑ Use NULL to display null data

9. Finish selecting the appropriate settings for the format of the text file you are importing. Click the **Next** button.

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.



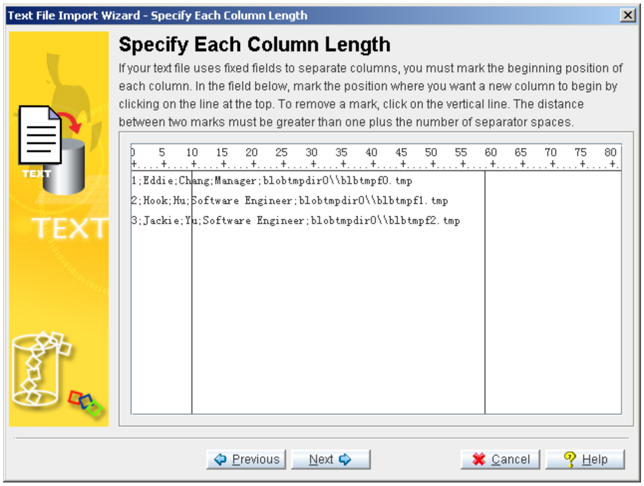
Text File Import Wizard - Specify Each Column Length

Specify Each Column Length

If your text file uses fixed fields to separate columns, you must mark the beginning position of each column. In the field below, mark the position where you want a new column to begin by clicking on the line at the top. To remove a mark, click on the vertical line. The distance between two marks must be greater than one plus the number of separator spaces.

```
0    5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+.
1;Eddie;Chang;Manager;blobtmpdir0\\blbtmpf0.tmp
2;Hook;Hu;Software Engineer;blobtmpdir0\\blbtmpf1.tmp
3;Jackie;Yu;Software Engineer;blobtmpdir0\\blbtmpf2.tmp
```

11. Click the horizontal line where you want to indicate the beginning of a column. A vertical line appears to indicate the column break. Use the scroll bar at the bottom to advance to other columns. Click the **Next** button.



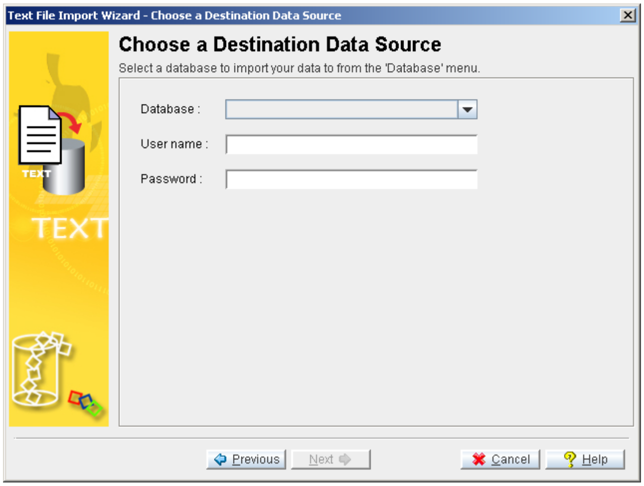
Text File Import Wizard - Specify Each Column Length

Specify Each Column Length

If your text file uses fixed fields to separate columns, you must mark the beginning position of each column. In the field below, mark the position where you want a new column to begin by clicking on the line at the top. To remove a mark, click on the vertical line. The distance between two marks must be greater than one plus the number of separator spaces.

```
0    5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+.
1;Eddie;Chang;Manager;blobtmpdir0\\blbtmpf0.tmp
2;Hook;Hu;Software Engineer;blobtmpdir0\\blbtmpf1.tmp
3;Jackie;Yu;Software Engineer;blobtmpdir0\\blbtmpf2.tmp
```

12. The **Choose Destination Data Source** window appears.



Text File Import Wizard - Choose a Destination Data Source

Choose a Destination Data Source

Select a database to import your data to from the 'Database' menu.

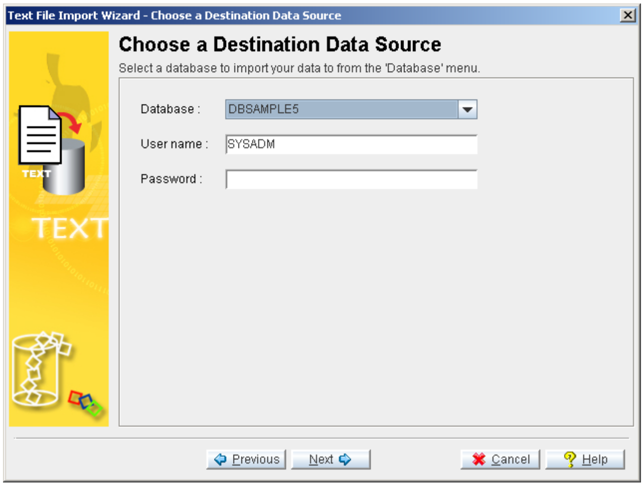
Database :

User name :

Password :

13. Select the database to import data to from the **Database** menu.

14. Enter a user name and password into the appropriate fields.



Text File Import Wizard - Choose a Destination Data Source

Choose a Destination Data Source

Select a database to import your data to from the 'Database' menu.

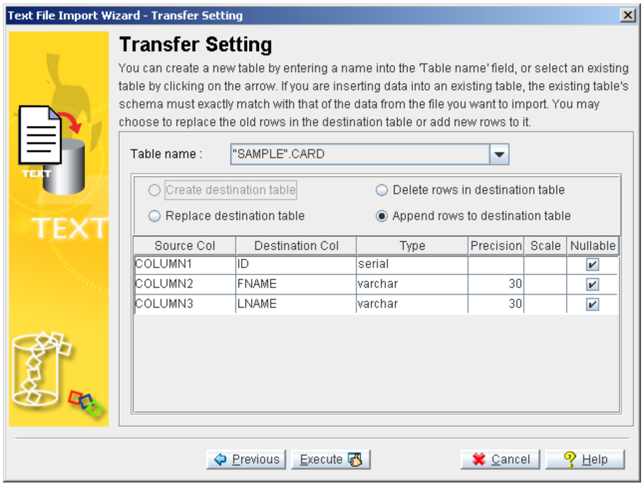
Database : DBSAMPLE5

User name : SYSADM

Password :

NOTE *A user must have INSERT privilege to import a text file.*

15. Click **Next**. The **Transfer Setting** window appears.



Text File Import Wizard - Transfer Setting

Transfer Setting

You can create a new table by entering a name into the 'Table name' field, or select an existing table by clicking on the arrow. If you are inserting data into an existing table, the existing table's schema must exactly match with that of the data from the file you want to import. You may choose to replace the old rows in the destination table or add new rows to it.

Table name : "SAMPLE".CARD

○ Create destination table ○ Delete rows in destination table

○ Replace destination table ● Append rows to destination table

Source Col	Destination Col	Type	Precision	Scale	Nullable
COLUMN1	ID	serial			☑
COLUMN2	FNAME	varchar	30		☑
COLUMN3	LNAME	varchar	30		☑

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 appears.

17. Click the **OK** button.

23.2 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.

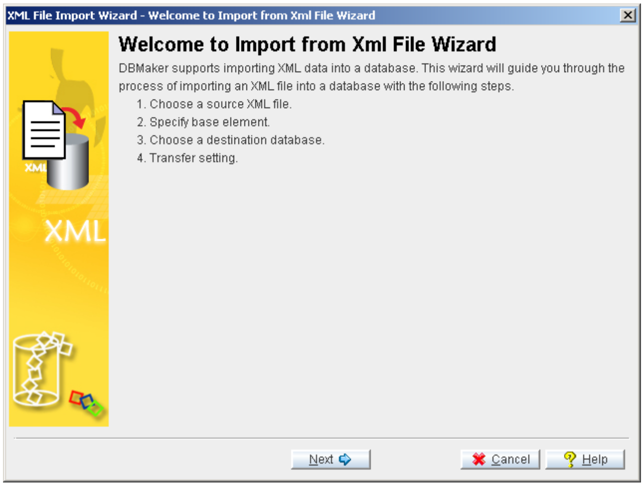
It is important to consider the structure of the XML file you wish to import. To ensure that the structure of the XML file and associated DTD have compatible structure, examine the structure of XML files produced by the Data Transfer Tool: Export to XML File wizard. Examples may be found in section 23.5, *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.

➲ Importing 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 starts.



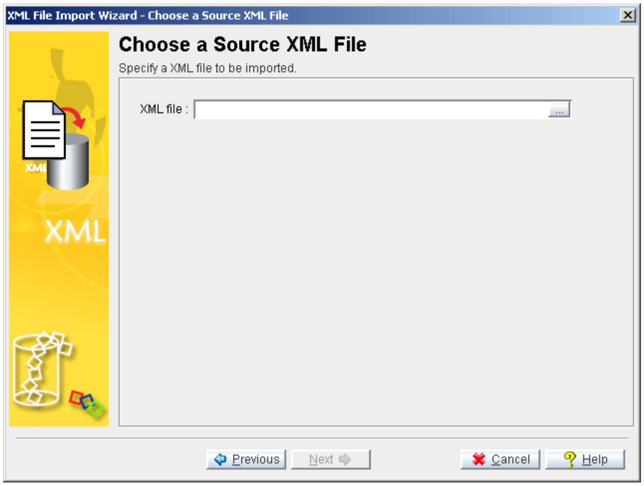
XML File Import Wizard - Welcome to Import from Xml File Wizard

Welcome to Import from Xml File Wizard

DBMaker supports importing XML data into a database. This wizard will guide you through the process of importing an XML file into a database with the following steps.

1. Choose a source XML file.
2. Specify base element.
3. Choose a destination database.
4. Transfer setting.

3. Click **Next**. The **Choose a Source XML File** window appears.



XML File Import Wizard - Choose a Source XML File

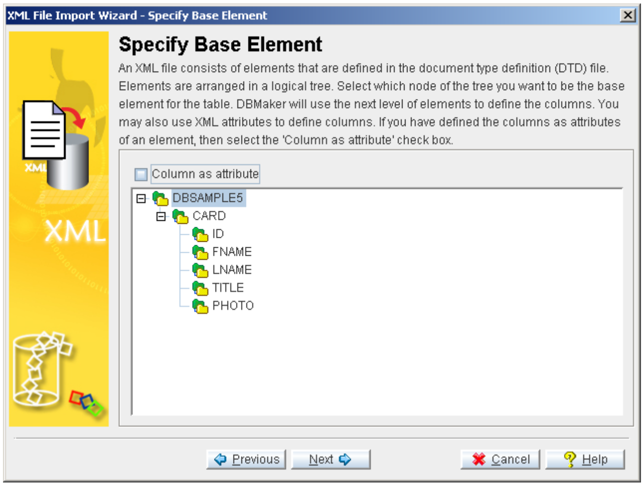
Choose a Source XML File

Specify a XML file to be imported.

XML file :

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 appears.



XML File Import Wizard - Specify Base Element

Specify Base Element

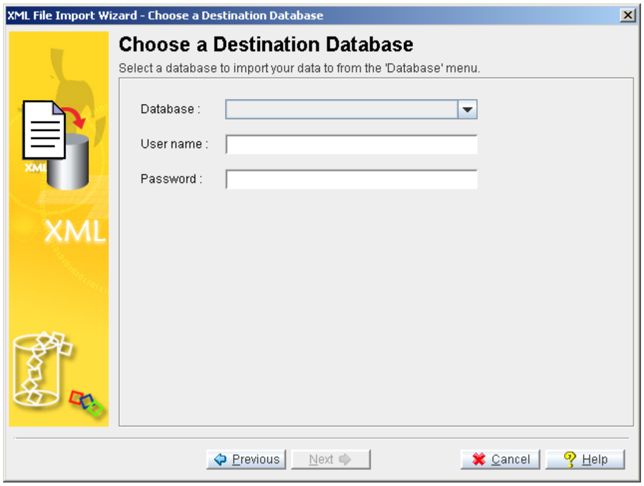
An XML file consists of elements that are defined in the document type definition (DTD) file. Elements are arranged in a logical tree. Select which node of the tree you want to be the base element for the table. DBMaker will use the next level of elements to define the columns. You may also use XML attributes to define columns. If you have defined the columns as attributes of an element, then select the 'Column as attribute' check box.

□ Column as attribute

- DBSAMPLE5
  - CARD
    - ID
    - FNAME
    - LNAME
    - TITLE
    - PHOTO

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 appears.



XML File Import Wizard - Choose a Destination Database

Choose a Destination Database

Select a database to import your data to from the 'Database' menu.

Database :

User name :

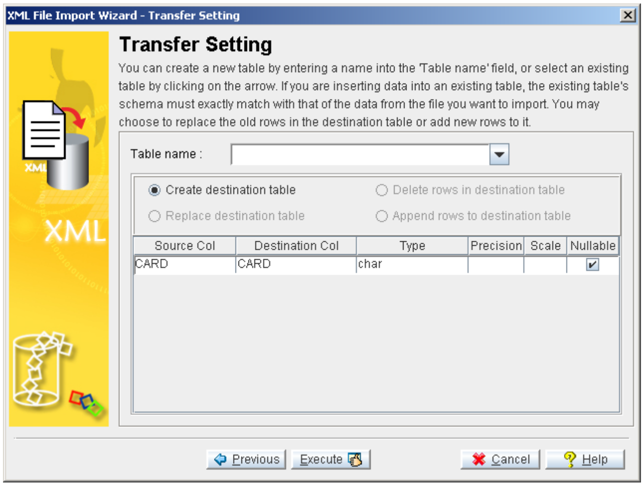
Password :

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 appears.



XML File Import Wizard - Transfer Setting

Transfer Setting

You can create a new table by entering a name into the 'Table name' field, or select an existing table by clicking on the arrow. If you are inserting data into an existing table, the existing table's schema must exactly match with that of the data from the file you want to import. You may choose to replace the old rows in the destination table or add new rows to it.

Table name :

● Create destination table
○ Delete rows in destination table
○ Replace destination table
○ Append rows to destination table

Source Col	Destination Col	Type	Precision	Scale	Nullable
CARD	CARD	char			☑

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 appears.

12. Click **OK**.

23.3 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 using three methods:

- directly from tables
- writing one or more SQL SELECT statements
- through an XML batch file

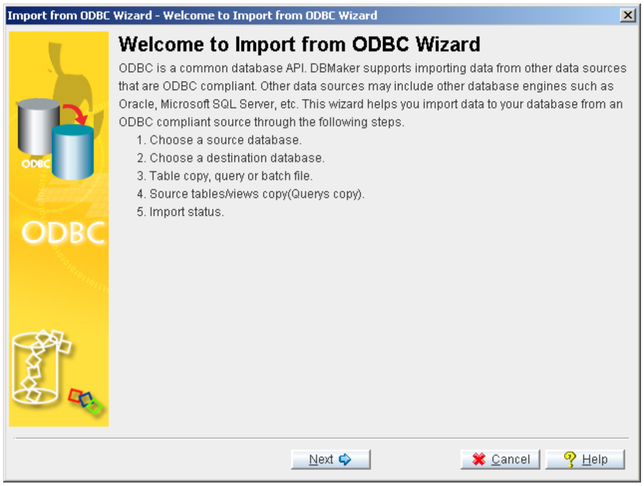
Furthermore, you may specify the mapping of column data through the transformation function. The transformation function supports direct column-to-column mapping or mapping through SQL SELECT and SQL INSERT statements.

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.

➲ Importing 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 appears. This window displays a summary of the wizard’s steps.



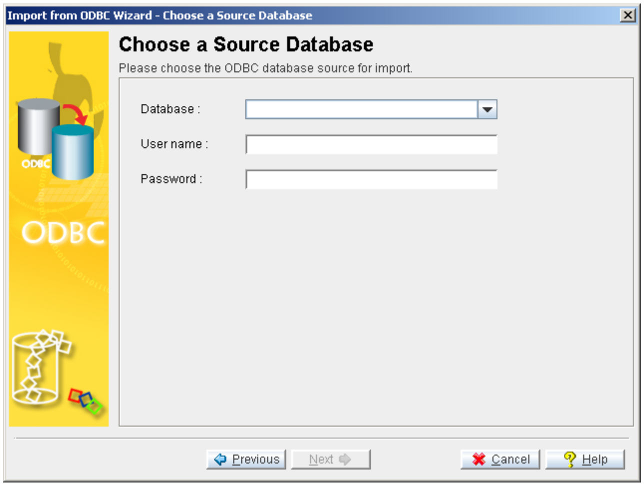
Import from ODBC Wizard - Welcome to Import from ODBC Wizard

Welcome to Import from ODBC Wizard

ODBC is a common database API. DBMaker supports importing data from other data sources that are ODBC compliant. Other data sources may include other database engines such as Oracle, Microsoft SQL Server, etc. This wizard helps you import data to your database from an ODBC compliant source through the following steps.

1. Choose a source database.
2. Choose a destination database.
3. Table copy, query or batch file.
4. Source tables/views copy(Querys copy).
5. Import status.

3. Click **Next**. The **Choose a Source Database** window appears.



Import from ODBC Wizard - Choose a Source Database

Choose a Source Database

Please choose the ODBC database source for import.

Database :

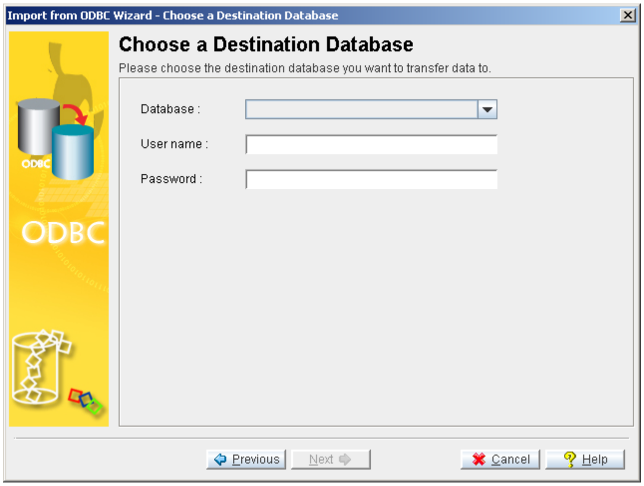
User name :

Password :

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 appears.



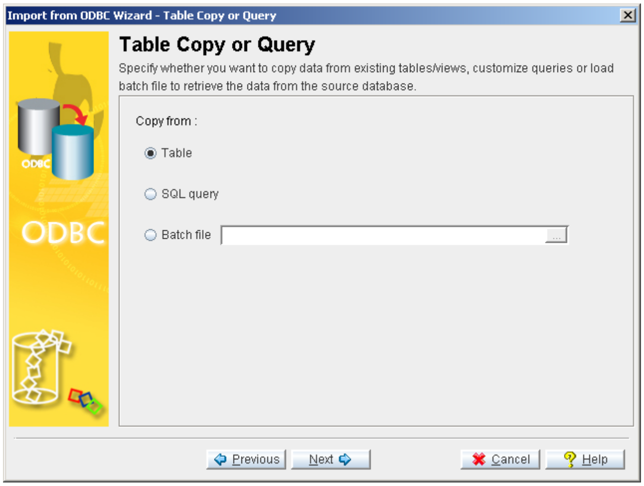
Import from ODBC Wizard - Choose a Destination Database
Choose a Destination Database
Please choose the destination database you want to transfer data to.
Database :
User name :
Password :
ODBC
Previous
Next
Cancel
Help

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 the **Next** button. The **Table Copy or Query** window appears.



Import from ODBC Wizard - Table Copy or Query

Table Copy or Query

Specify whether you want to copy data from existing tables/views, customize queries or load batch file to retrieve the data from the source database.

Copy from :

● Table

○ SQL query

○ Batch file

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**

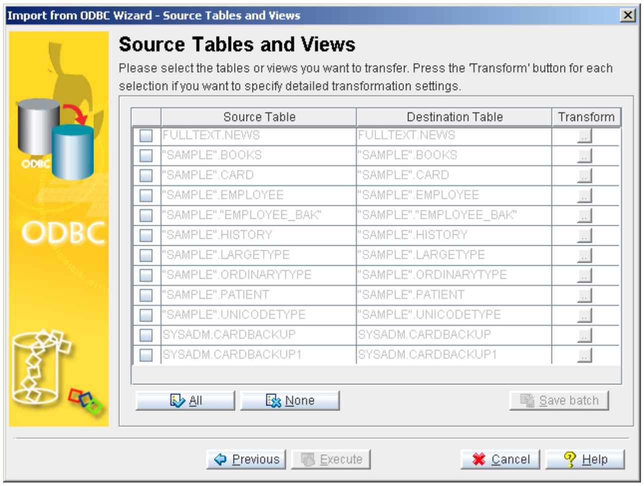
Importing ODBC Data from Tables

Select tables in the source database to import ODBC data. The **Import from ODBC Wizard** provides an intuitive graphical interface for selecting tables and setting how data is transformed during the import process.

➲ Importing 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 appears.



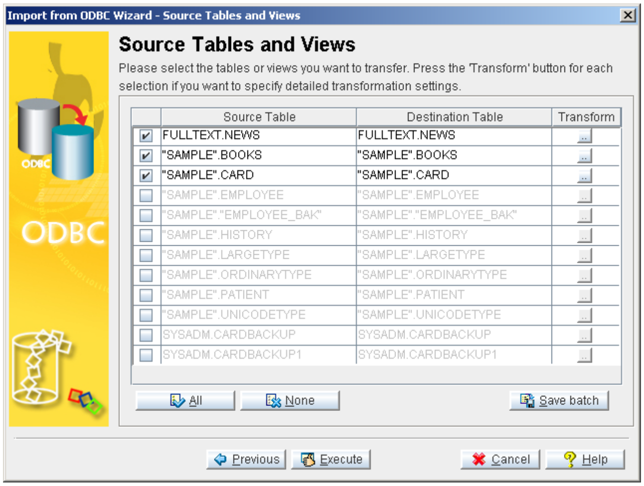
Import from ODBC Wizard - Source Tables and Views

Source Tables and Views

Please select the tables or views you want to transfer. Press the 'Transform' button for each selection if you want to specify detailed transformation settings.

	Source Table	Destination Table	Transform
□	FULLTEXT.NEWS	FULLTEXT.NEWS	...
□	"SAMPLE".BOOKS	"SAMPLE".BOOKS	...
□	"SAMPLE".CARD	"SAMPLE".CARD	...
□	"SAMPLE".EMPLOYEE	"SAMPLE".EMPLOYEE	...
□	"SAMPLE"."EMPLOYEE_BAK"	"SAMPLE"."EMPLOYEE_BAK"	...
□	"SAMPLE".HISTORY	"SAMPLE".HISTORY	...
□	"SAMPLE".LARGETYPE	"SAMPLE".LARGETYPE	...
□	"SAMPLE".ORDINARYTYPE	"SAMPLE".ORDINARYTYPE	...
□	"SAMPLE".PATIENT	"SAMPLE".PATIENT	...
□	"SAMPLE".UNICODETYPE	"SAMPLE".UNICODETYPE	...
□	SYSADM.CARDBACKUP	SYSADM.CARDBACKUP	...
□	SYSADM.CARDBACKUP1	SYSADM.CARDBACKUP1	...

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.



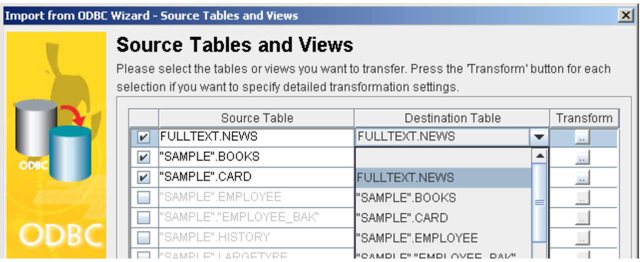
Import from ODBC Wizard - Source Tables and Views

Source Tables and Views

Please select the tables or views you want to transfer. Press the 'Transform' button for each selection if you want to specify detailed transformation settings.

	Source Table	Destination Table	Transform
☑	FULLTEXT.NEWS	FULLTEXT.NEWS	...
☑	"SAMPLE".BOOKS	"SAMPLE".BOOKS	...
☑	"SAMPLE".CARD	"SAMPLE".CARD	...
□	"SAMPLE".EMPLOYEE	"SAMPLE".EMPLOYEE	...
□	"SAMPLE"."EMPLOYEE_BAK"	"SAMPLE"."EMPLOYEE_BAK"	...
□	"SAMPLE".HISTORY	"SAMPLE".HISTORY	...
□	"SAMPLE".LARGETYPE	"SAMPLE".LARGETYPE	...
□	"SAMPLE".ORDINARYTYPE	"SAMPLE".ORDINARYTYPE	...
□	"SAMPLE".PATIENT	"SAMPLE".PATIENT	...
□	"SAMPLE".UNICODETYPE	"SAMPLE".UNICODETYPE	...
□	SYSADM.CARDBACKUP	SYSADM.CARDBACKUP	...
□	SYSADM.CARDBACKUP1	SYSADM.CARDBACKUP1	...

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.



Import from ODBC Wizard - Source Tables and Views

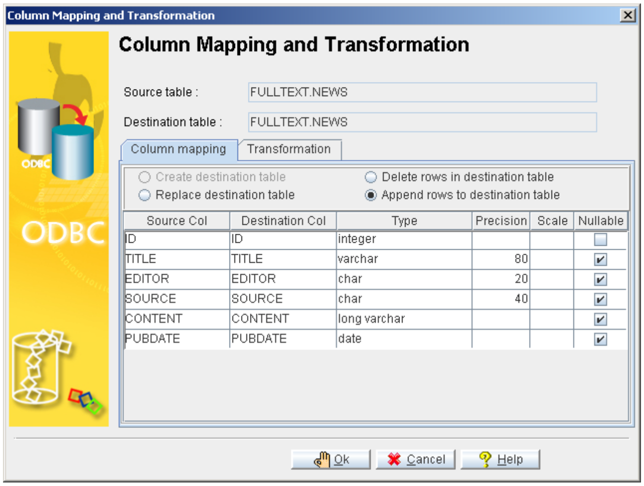
Source Tables and Views

Please select the tables or views you want to transfer. Press the 'Transform' button for each selection if you want to specify detailed transformation settings.

	Source Table	Destination Table	Transform
☑	FULLTEXT.NEWS	FULLTEXT.NEWS	...
☑	"SAMPLE".BOOKS		...
☑	"SAMPLE".CARD		...
□	"SAMPLE".EMPLOYEE		...
□	"SAMPLE"."EMPLOYEE_BAK"		...
□	"SAMPLE".HISTORY		...
□	"SAMPLE".LARGETYPE		...

FULLTEXT.NEWS
"SAMPLE".BOOKS
"SAMPLE".CARD
"SAMPLE".EMPLOYEE
"SAMPLE"."EMPLOYEE_BAK"

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.



Column Mapping and Transformation

Column Mapping and Transformation

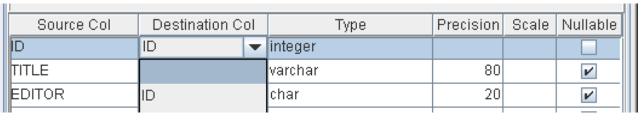
Source table : FULLTEXT.NEWS

Destination table : FULLTEXT.NEWS

○ Create destination table ○ Delete rows in destination table
○ Replace destination table ● Append rows to destination table

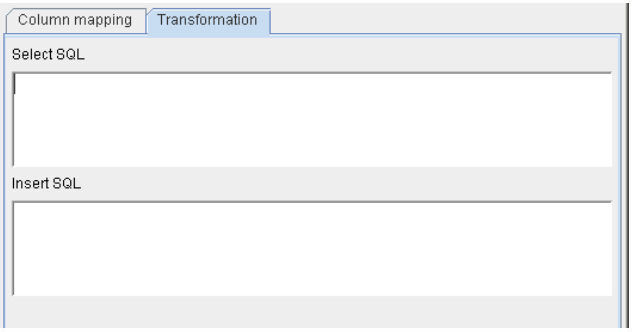
Source Col	Destination Col	Type	Precision	Scale	Nullable
ID	ID	integer			□
TITLE	TITLE	varchar	80		☑
EDITOR	EDITOR	char	20		☑
SOURCE	SOURCE	char	40		☑
CONTENT	CONTENT	long varchar			☑
PUBDATE	PUBDATE	date			☑

6. Change the name of the destination column by selecting a new column from the menu or entering a new name.



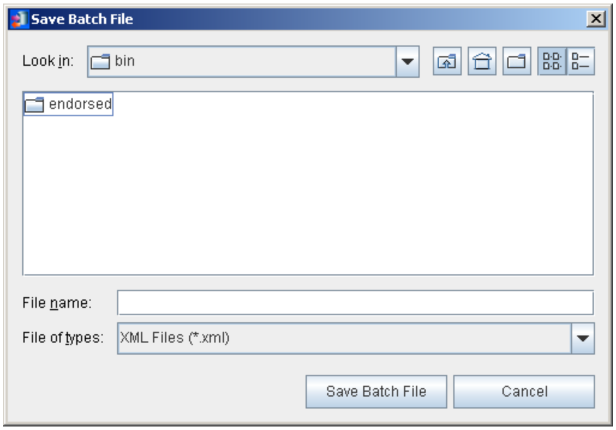
Source Col	Destination Col	Type	Precision	Scale	Nullable
ID	ID	integer			□
TITLE		varchar	80		☑
EDITOR	ID	char	20		☑

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.



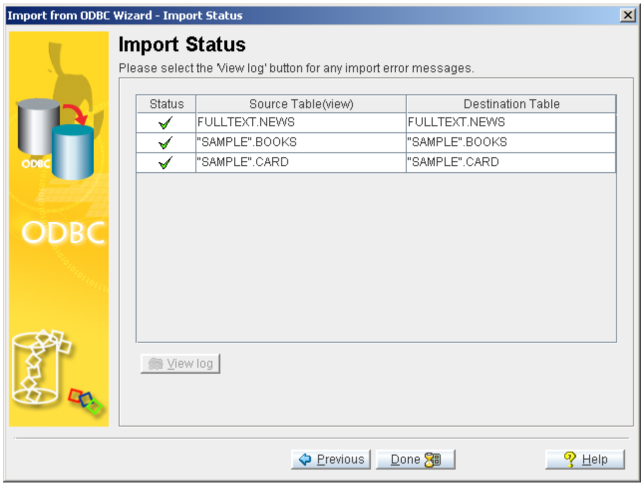
8. Click **OK** to return to the **Source Tables and Views** window.

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 appears.



Import from ODBC Wizard - Import Status

Import Status

Please select the 'View log' button for any import error messages.

Status	Source Table(view)	Destination Table
✓	FULLTEXT.NEWS	FULLTEXT.NEWS
✓	"SAMPLE".BOOKS	"SAMPLE".BOOKS
✓	"SAMPLE".CARD	"SAMPLE".CARD

12. If errors appear, click **View log** and scroll to the bottom to see the error message. If no errors occurred, click **Done**.

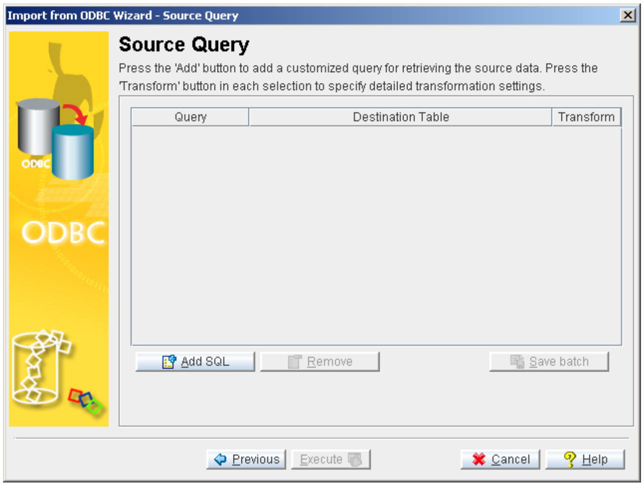
Importing ODBC Data Using SQL SELECT Statements

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.

➲ Importing 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 appears.



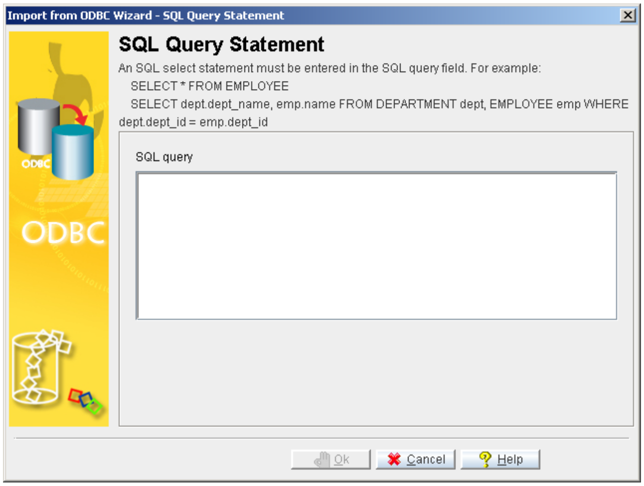
Import from ODBC Wizard - Source Query

Source Query

Press the 'Add' button to add a customized query for retrieving the source data. Press the 'Transform' button in each selection to specify detailed transformation settings.

Query	Destination Table	Transform

3. Click **Add SQL**. The **SQL Query Statement** window appears.



Import from ODBC Wizard - SQL Query Statement

SQL Query Statement

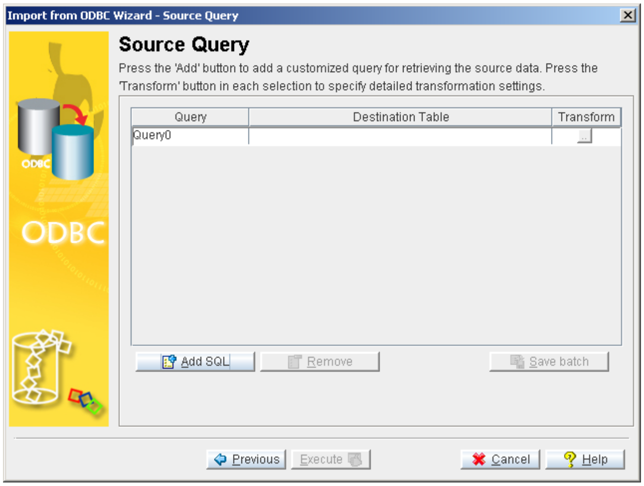
An SQL select statement must be entered in the SQL query field. For example:

```
   SELECT * FROM EMPLOYEE
   SELECT dept.dept_name, emp.name FROM DEPARTMENT dept, EMPLOYEE emp WHERE
dept.dept_id = emp.dept_id
```

SQL query

4. Enter a valid SQL SELECT statement into the **SQL Query** field.

5. Click **OK**. The **Source Query** window reappears.



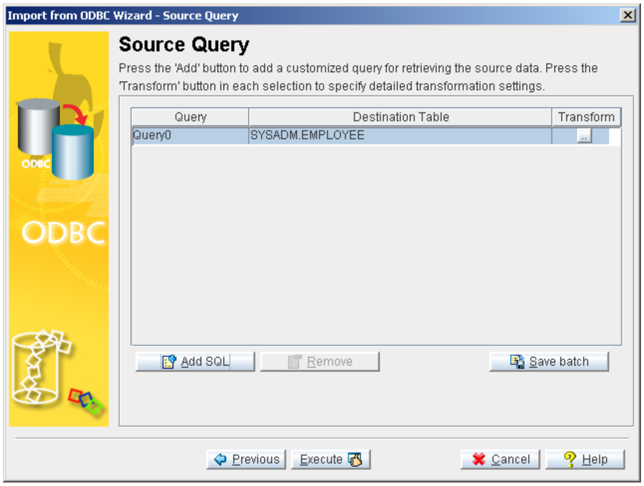
Import from ODBC Wizard - Source Query

Source Query

Press the 'Add' button to add a customized query for retrieving the source data. Press the 'Transform' button in each selection to specify detailed transformation settings.

Query	Destination Table	Transform
Query0		...

6. Select or create a destination table from the **Destination Table** column.



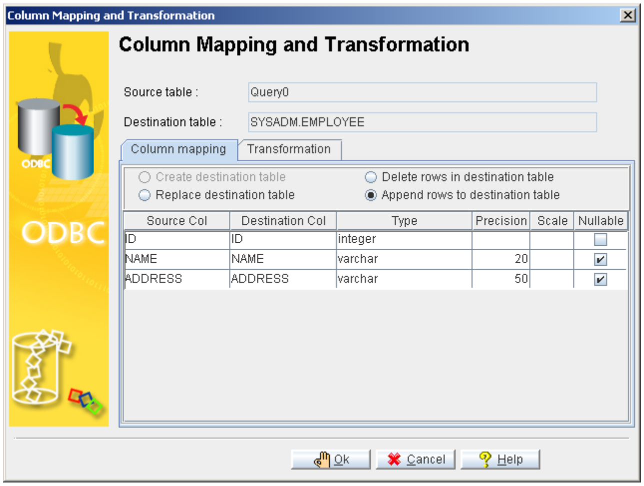
Import from ODBC Wizard - Source Query

Source Query

Press the 'Add' button to add a customized query for retrieving the source data. Press the 'Transform' button in each selection to specify detailed transformation settings.

Query	Destination Table	Transform
Query0	SYSADM.EMPLOYEE	...

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.



Column Mapping and Transformation

Source table : Query0

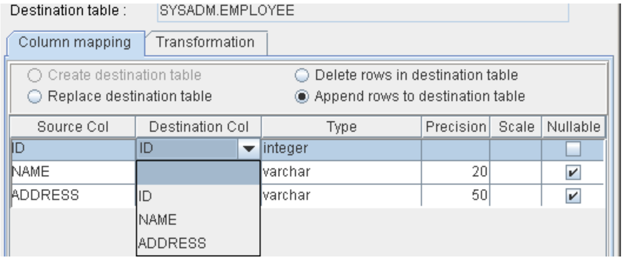
Destination table : SYSADM.EMPLOYEE

Column mapping | Transformation

○ Create destination table
○ Replace destination table
○ Delete rows in destination table
● Append rows to destination table

Source Col	Destination Col	Type	Precision	Scale	Nullable
ID	ID	integer			□
NAME	NAME	varchar	20		☑
ADDRESS	ADDRESS	varchar	50		☑

8. Change the name of the destination column by selecting a new column from the menu or entering a new name.



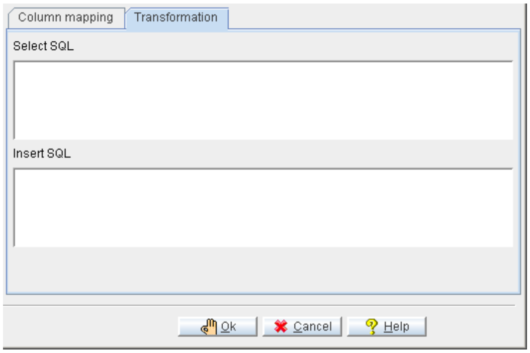
Destination table : SYSADM.EMPLOYEE

Column mapping | Transformation

○ Create destination table
○ Replace destination table
○ Delete rows in destination table
● Append rows to destination table

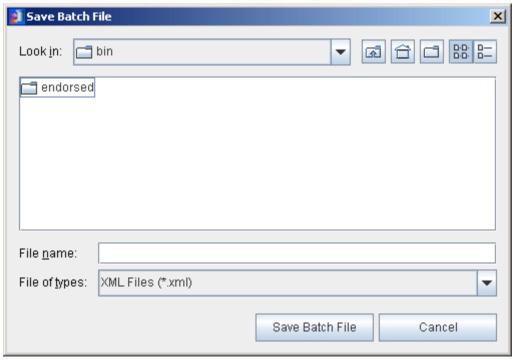
Source Col	Destination Col	Type	Precision	Scale	Nullable
ID	ID	integer			□
NAME		varchar	20		☑
ADDRESS	ID	varchar	50		☑

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.



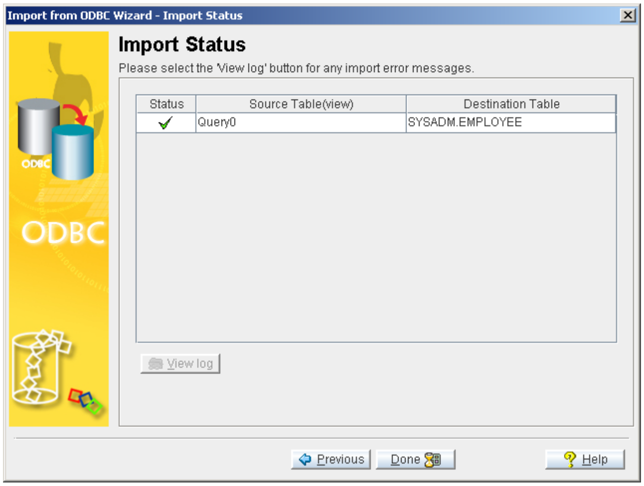
10. Click **OK** to return to the **Source Tables and Views** window.

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 reappears.

13. Click **Execute** to import the source data. The Import Status window appears.



Import from ODBC Wizard - Import Status
Import Status
Please select the 'View log' button for any import error messages.
ODBC
View log
Previous
Done
Help

Status	Source Table(view)	Destination Table
✓	Query0	SYSADM.EMPLOYEE

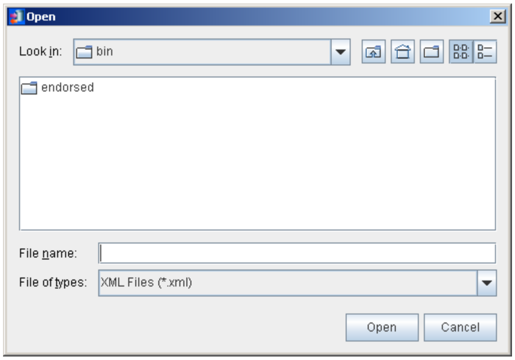
14. If errors appear, click **View log** and scroll to the bottom to see the error message. If no errors occurred, click **Done**.

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

➲ Importing 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 appears.



Look in: bin

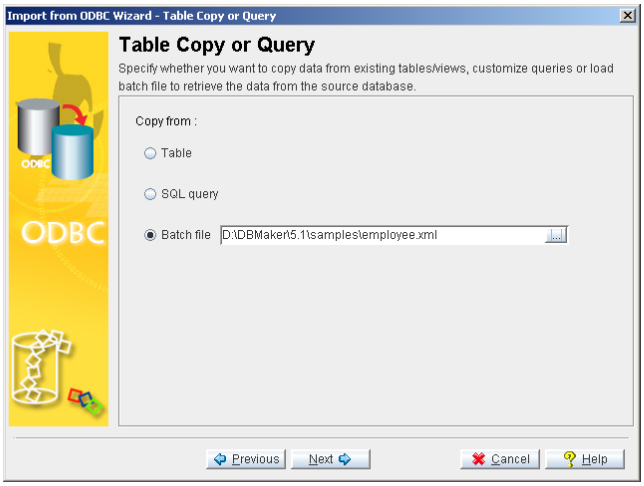
endorsed

File name:

File of types: XML Files (*.xml)

2. Select an XML file from which to import the ODBC map schema.

3. Click **Open**. The **Table Copy or Query** window reappears.



Import from ODBC Wizard - Table Copy or Query

Table Copy or Query

Specify whether you want to copy data from existing tables/views, customize queries or load batch file to retrieve the data from the source database.

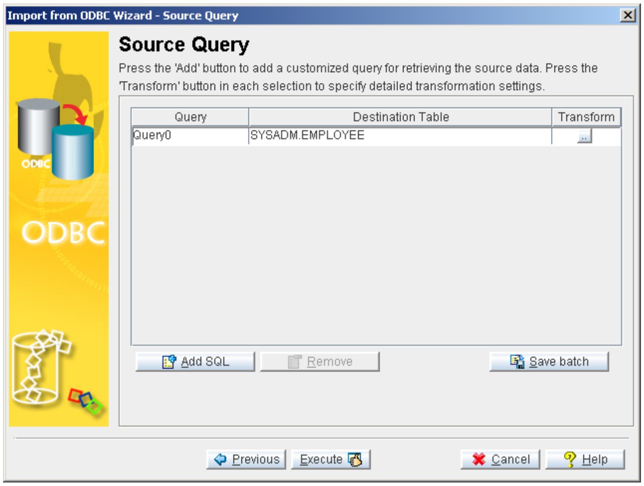
Copy from :

○ Table

○ SQL query

● Batch file D:\DBMaker\5.1\samples\employee.xml

4. Click **Next**. The **Source Query** window will open, displaying a mapping schema according to the XML file.



Import from ODBC Wizard - Source Query

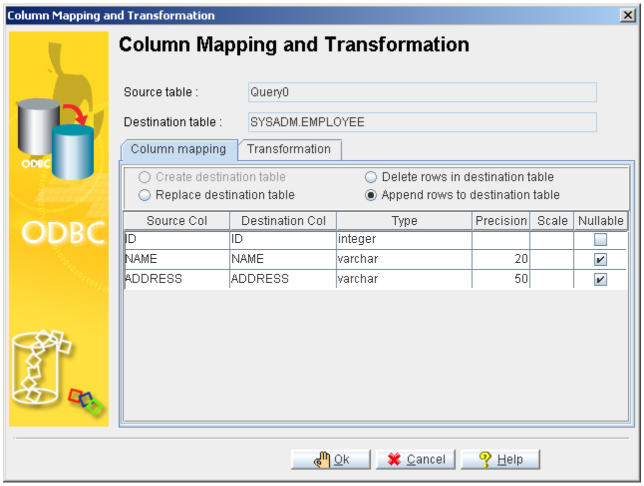
Source Query

Press the 'Add' button to add a customized query for retrieving the source data. Press the 'Transform' button in each selection to specify detailed transformation settings.

Query	Destination Table	Transform
Query0	SYSADM.EMPLOYEE	...

5. 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.

6. You may modify column mapping or the result set to import by clicking on the **Transform** button of the corresponding source and destination table.



Column Mapping and Transformation

Column Mapping and Transformation

Source table : Query0

Destination table : SYSADM.EMPLOYEE

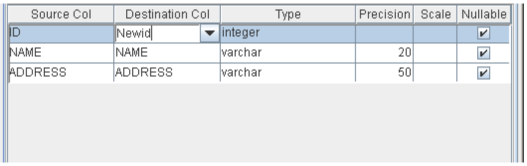
Column mapping | Transformation

○ Create destination table ○ Delete rows in destination table

○ Replace destination table ● Append rows to destination table

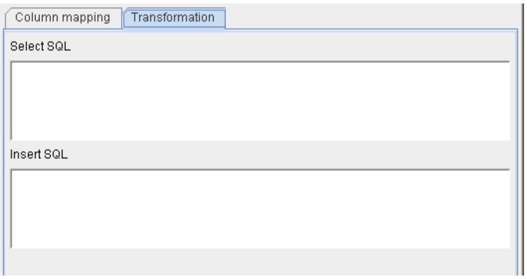
Source Col	Destination Col	Type	Precision	Scale	Nullable
ID	ID	integer			□
NAME	NAME	varchar	20		☑
ADDRESS	ADDRESS	varchar	50		☑

7. Change the name of the destination column by selecting a new column from the menu or entering a new name.



Source Col	Destination Col	Type	Precision	Scale	Nullable
ID	Newid	integer			☑
NAME	NAME	varchar	20		☑
ADDRESS	ADDRESS	varchar	50		☑

8. 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.



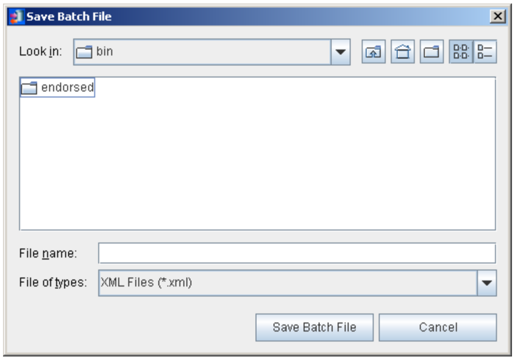
Column mapping | Transformation

Select SQL

Insert SQL

9. Click **OK** to return to the **Source Tables and Views** window.

10. 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.



Save Batch File

Look in: bin

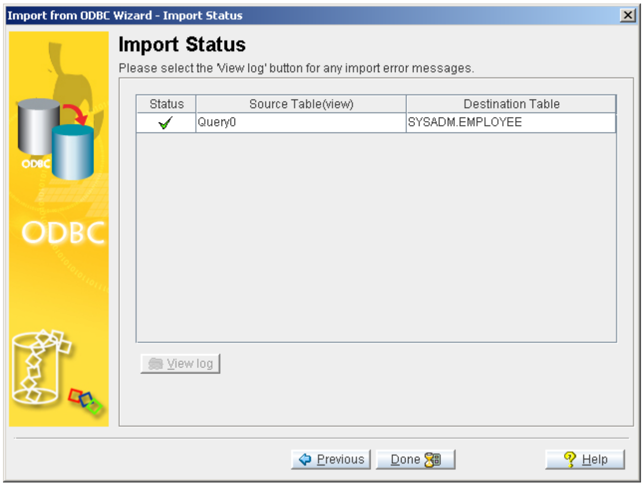
endorsed

File name:

File of types: XML Files (*.xml)

11. 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.

12. Click **Execute** to import the source data. The Import Status window will appear.



Import from ODBC Wizard - Import Status

Import Status

Please select the 'View log' button for any import error messages.

Status	Source Table(view)	Destination Table
✓	Query0	SYSADM.EMPLOYEE

13. If errors appear, click **View log** and scroll to the bottom to see the error message. If no errors occurred, click **Done**.

23.4 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*.

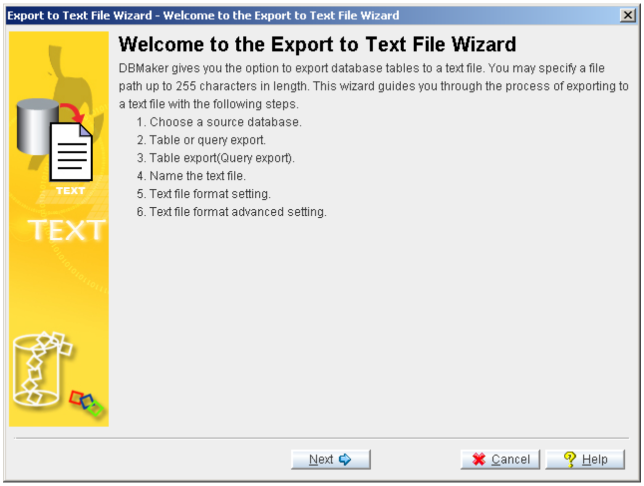
➲ Example

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
```

➲ Exporting 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.



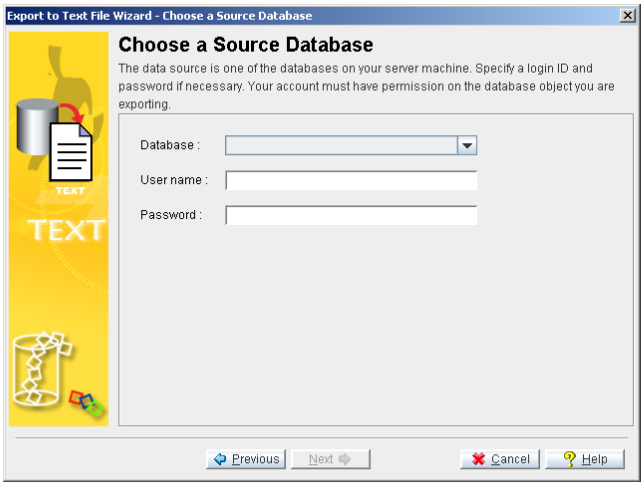
Export to Text File Wizard - Welcome to the Export to Text File Wizard

Welcome to the Export to Text File Wizard

DBMaker gives you the option to export database tables to a text file. You may specify a file path up to 255 characters in length. This wizard guides you through the process of exporting to a text file with the following steps.

1. Choose a source database.
2. Table or query export.
3. Table export(Query export).
4. Name the text file.
5. Text file format setting.
6. Text file format advanced setting.

3. Click **Next**. The **Choose a Source Database** window will open.



Export to Text File Wizard - Choose a Source Database

Choose a Source Database

The data source is one of the databases on your server machine. Specify a login ID and password if necessary. Your account must have permission on the database object you are exporting.

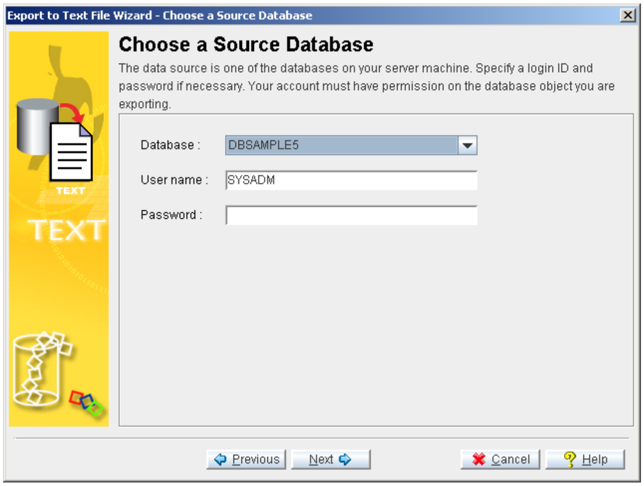
Database :

User name :

Password :

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.*



Export to Text File Wizard - Choose a Source Database

Choose a Source Database

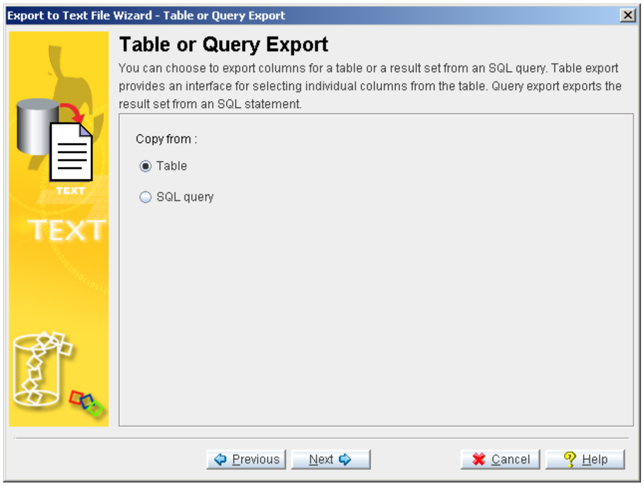
The data source is one of the databases on your server machine. Specify a login ID and password if necessary. Your account must have permission on the database object you are exporting.

Database : DBSAMPLE5

User name : SYSADM

Password :

6. After you have selected a database, click **Next**, the Table or Query Export window will open.



Export to Text File Wizard - Table or Query Export

Table or Query Export

You can choose to export columns for a table or a result set from an SQL query. Table export provides an interface for selecting individual columns from the table. Query export exports the result set from an SQL statement.

Copy from :

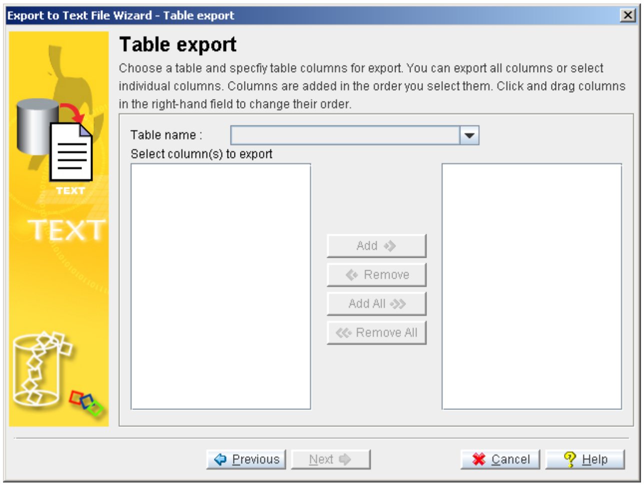
● Table

○ SQL query

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.



Export to Text File Wizard - Table export

Table export

Choose a table and specfiy table columns for export. You can export all columns or select individual columns. Columns are added in the order you select them. Click and drag columns in the right-hand field to change their order.

Table name :

Select column(s) to export

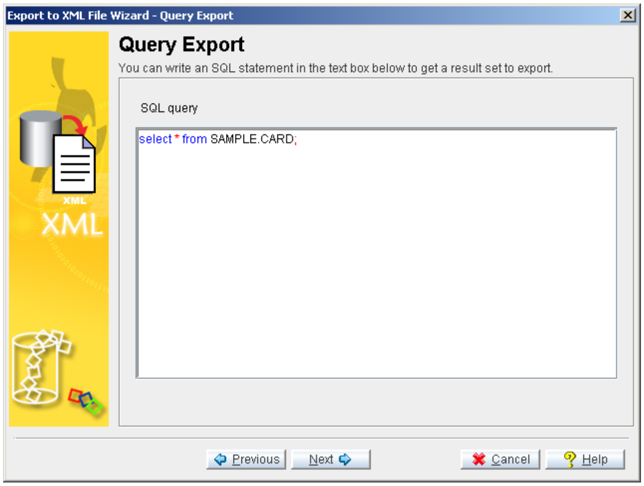
10. Select a table to export from the **Table name** menu. A list of columns in the table will appear in the Select columns to export field.

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.



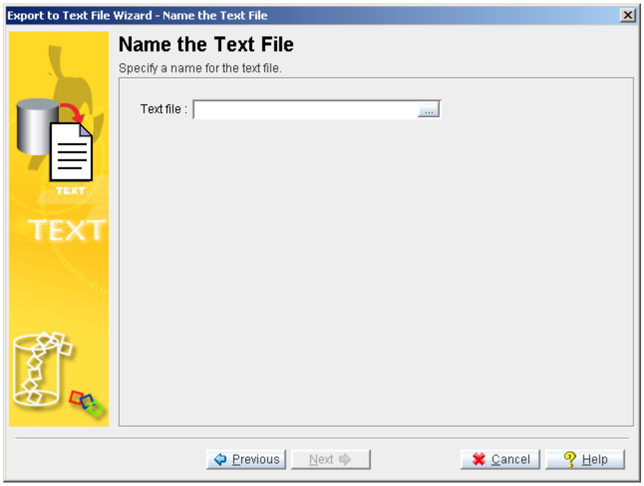
Query Export

You can write an SQL statement in the text box below to get a result set to export.

SQL query

```
select * from SAMPLE.CARD;
```

15. Click **Next**. The Name the Text File window appears.



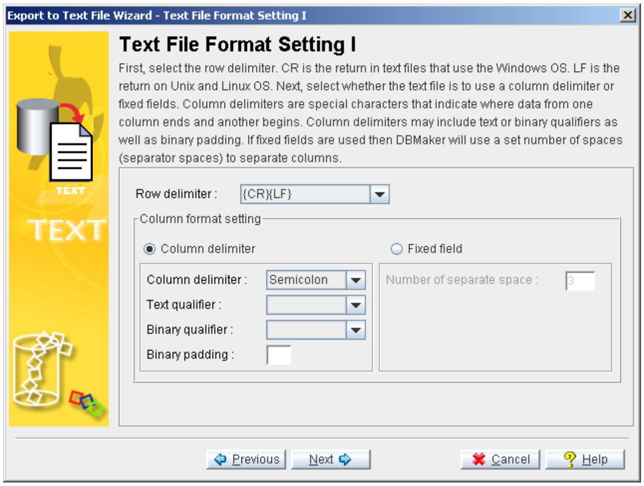
Name the Text File

Specify a name for the text file.

Text file :

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 appears.


Export to Text File Wizard - Text File Format Setting I

Text File Format Setting I

First, select the row delimiter. CR is the return in text files that use the Windows OS. LF is the return on Unix and Linux OS. Next, select whether the text file is to use a column delimiter or fixed fields. Column delimiters are special characters that indicate where data from one column ends and another begins. Column delimiters may include text or binary qualifiers as well as binary padding. If fixed fields are used then DBMaker will use a set number of spaces (separator spaces) to separate columns.

Row delimiter : {CR}{LF}

Column format setting

● Column delimiter ○ Fixed field

Column delimiter : Semicolon

Text qualifier :

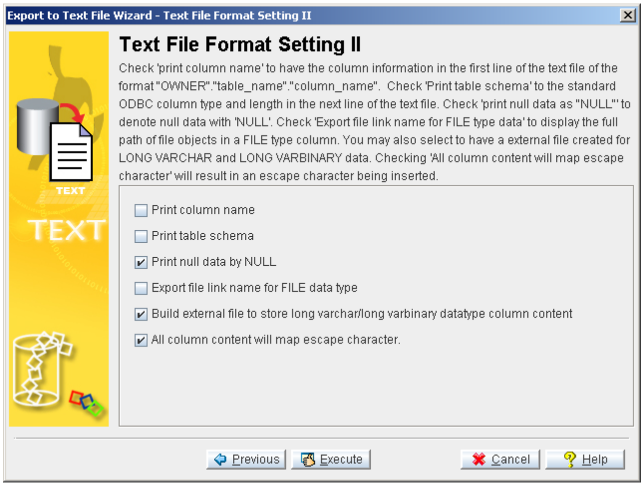
Binary qualifier :

Binary padding :

Number of separate space : 3

18. Select the appropriate settings for the format of the new text file.

19. Click **Next**. The **Text File Format Setting 2** window appears.


Export to Text File Wizard - Text File Format Setting II

Text File Format Setting II

Check 'print column name' to have the column information in the first line of the text file of the format "OWNER"."table_name"."column_name". Check 'Print table schema' to the standard ODBC column type and length in the next line of the text file. Check 'print null data as "NULL"' to denote null data with 'NULL'. Check 'Export file link name for FILE type data' to display the full path of file objects in a FILE type column. You may also select to have a external file created for LONG VARCHAR and LONG VARBINARY data. Checking 'All column content will map escape character' will result in an escape character being inserted.

☐ Print column name

☐ Print table schema

☑ Print null data by NULL

☐ Export file link name for FILE data type

☑ Build external file to store long varchar/long varbinary datatype column content

☑ All column content will map escape character.

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 appears.

21. Click the **OK** button**.**

23.5 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: Select this original for the full path to reference system and user file objects, otherwise, when unselected file type data is 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 is:

```
<?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 is:

```
<!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\5.1\BIN\WEBDB\FO\ZZ000000.GIF">
<!ENTITY BLBTMP_TXT1 SYSTEM "blobtmpdir0\blbtmpf1.txt">
<!ENTITY DBMAKER_FO_1 SYSTEM "C:\DBMAKER\5.1\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 is:

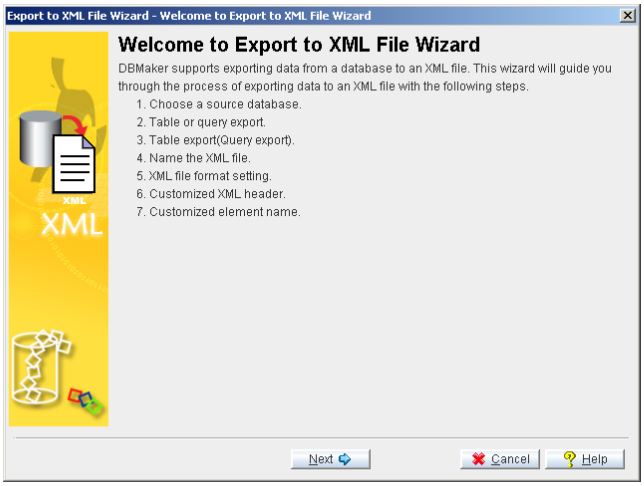
```
<?xml version="1.0" encoding="BIG5"?>
<!DOCTYPE WEBDB SYSTEM "Support.dtd">
<WEBDB>
  <SUPPORTQUERIES
LOGINID="A_HOWARD&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x2
0;&#x20;" REQUESTTIME="2001-09-09 12:47:05.000"
ATTACHMENT="C:\DBMAKER\5.1\BIN\WEBDB\FO\ZZ000000.GIF"
BINARY_C="10000000000000000000" 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\5.1\BIN\WEBDB\FO\ZZ000001.GIF"
BINARY_C="20000000000000000000" DECIMAL_C="13.550"/>
```

The associated DTD is:

```
<!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*)>
```

➲ Exporting 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 appears.



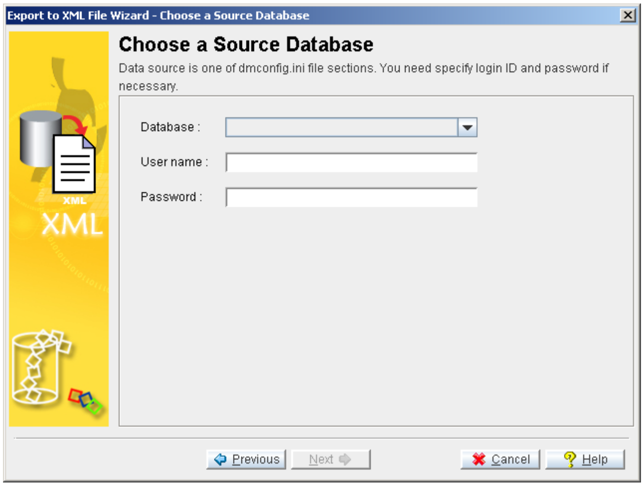
Export to XML File Wizard - Welcome to Export to XML File Wizard

Welcome to Export to XML File Wizard

DBMaker supports exporting data from a database to an XML file. This wizard will guide you through the process of exporting data to an XML file with the following steps.

1. Choose a source database.
2. Table or query export.
3. Table export(Query export).
4. Name the XML file.
5. XML file format setting.
6. Customized XML header.
7. Customized element name.

3. Click **Next**. The **Choose a Source Database** window appears.



Export to XML File Wizard - Choose a Source Database

Choose a Source Database

Data source is one of dmconfig.ini file sections. You need specify login ID and password if necessary.

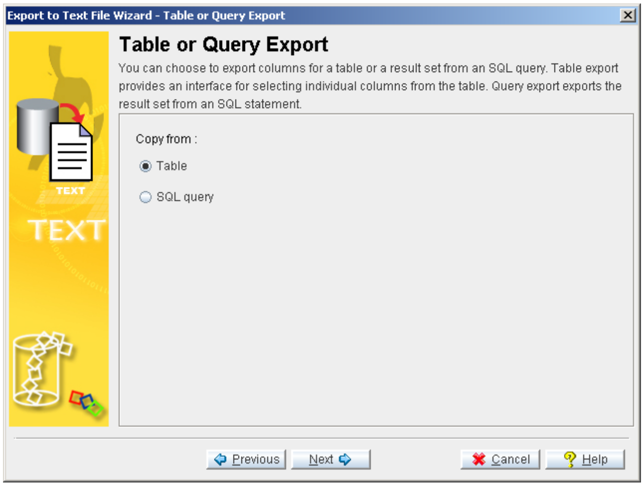
Database :

User name :

Password :

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 appears.



Export to Text File Wizard - Table or Query Export

Table or Query Export

You can choose to export columns for a table or a result set from an SQL query. Table export provides an interface for selecting individual columns from the table. Query export exports the result set from an SQL statement.

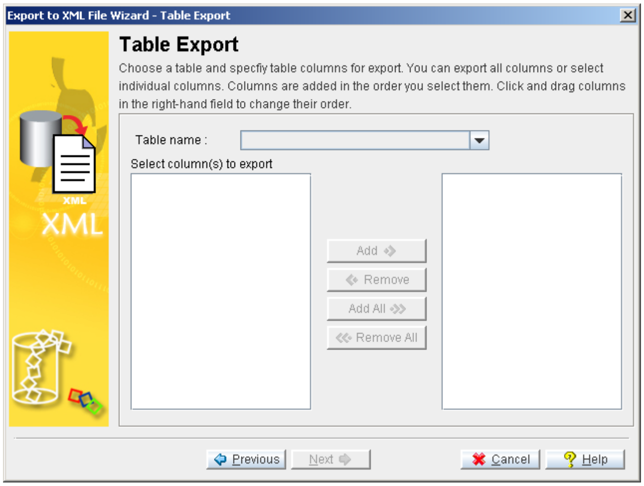
Copy from :

● Table

○ SQL query

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 appears.



Export to XML File Wizard - Table Export

Table Export

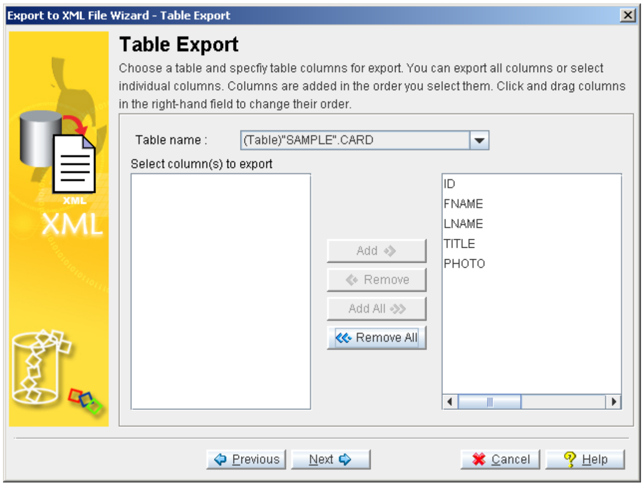
Choose a table and specfiy table columns for export. You can export all columns or select individual columns. Columns are added in the order you select them. Click and drag columns in the right-hand field to change their order.

Table name :

Select column(s) to export

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.



Export to XML File Wizard - Table Export

Table Export

Choose a table and specfiy table columns for export. You can export all columns or select individual columns. Columns are added in the order you select them. Click and drag columns in the right-hand field to change their order.

Table name : (Table)"SAMPLE".CARD

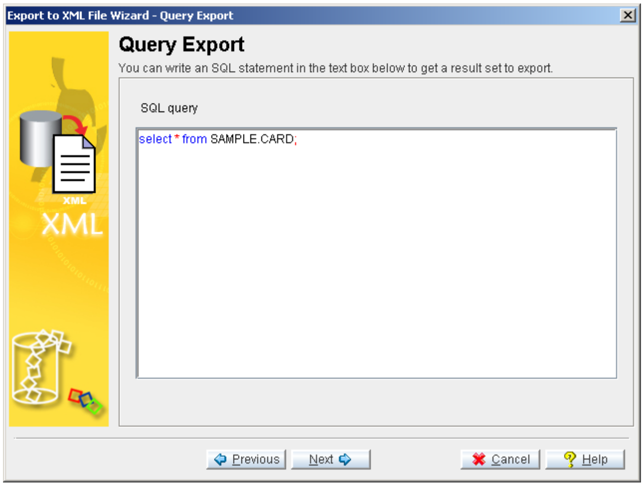
Select column(s) to export

ID
FNAME
LNAME
TITLE
PHOTO

10. Click **Next**. The **Name the XML File** window appears. Please 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.



Export to XML File Wizard - Query Export

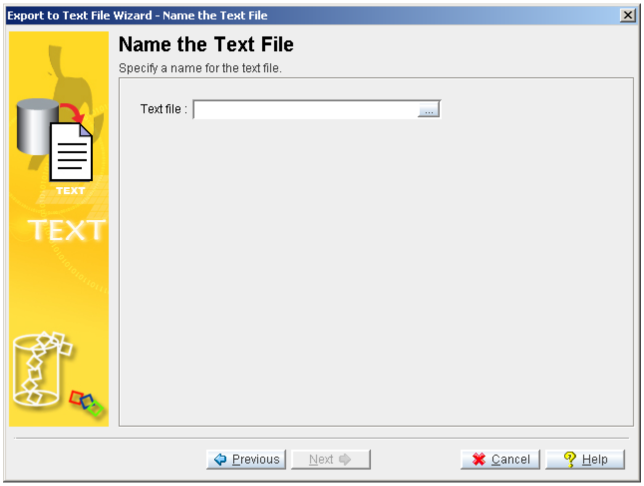
Query Export

You can write an SQL statement in the text box below to get a result set to export.

SQL query

```
select * from SAMPLE.CARD;
```

13. Click **Next**. The **Name the XML File** window appears.



Export to Text File Wizard - Name the Text File

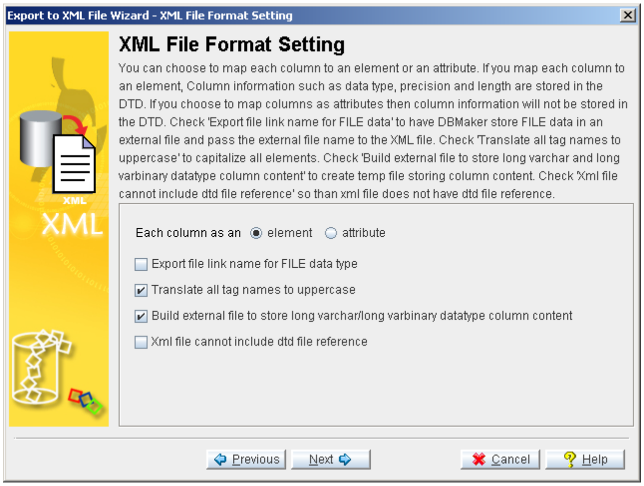
Name the Text File

Specify a name for the text file.

Text file :

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 appears.



Export to XML File Wizard - XML File Format Setting

XML File Format Setting

You can choose to map each column to an element or an attribute. If you map each column to an element, Column information such as data type, precision and length are stored in the DTD. If you choose to map columns as attributes then column information will not be stored in the DTD. Check 'Export file link name for FILE data' to have DBMaker store FILE data in an external file and pass the external file name to the XML file. Check 'Translate all tag names to uppercase' to capitalize all elements. Check 'Build external file to store long varchar and long varbinary datatype column content' to create temp file storing column content. Check 'Xml file cannot include dtd file reference' so than xml file does not have dtd file reference.

Each column as an ● element ○ attribute

☐ Export file link name for FILE data type

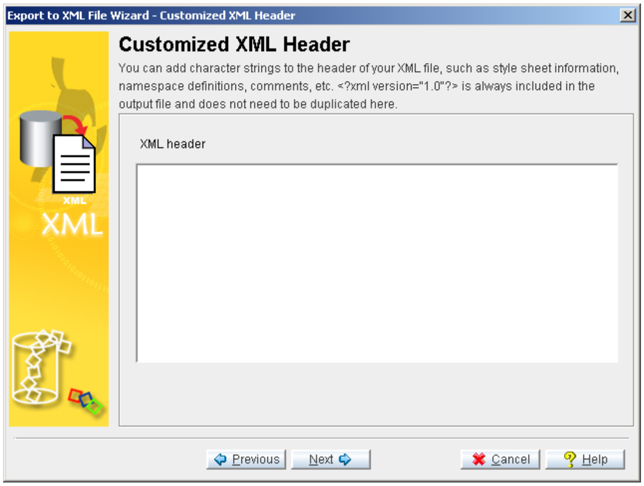
☑ Translate all tag names to uppercase

☑ Build external file to store long varchar/long varbinary datatype column content

☐ Xml file cannot include dtd file reference

16. Select the appropriate settings for the format of the new XML file.

17. Click **Next**. The **Customized XML Header** window appears.



Export to XML File Wizard - Customized XML Header

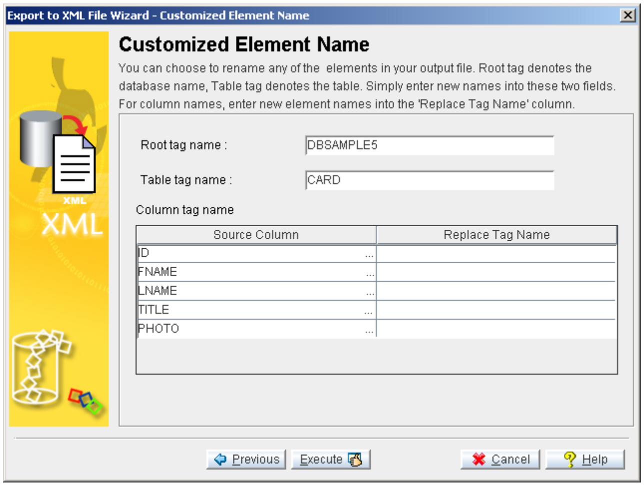
Customized XML Header

You can add character strings to the header of your XML file, such as style sheet information, namespace definitions, comments, etc. <?xml version="1.0"?> is always included in the output file and does not need to be duplicated here.

XML header

18. Enter appropriate information, such as namespace and style sheet definitions, if relevant.

19. Click **Next**. The **Customized Element Name** window appears.



Export to XML File Wizard - Customized Element Name

Customized Element Name

You can choose to rename any of the elements in your output file. Root tag denotes the database name, Table tag denotes the table. Simply enter new names into these two fields. For column names, enter new element names into the 'Replace Tag Name' column.

Root tag name : DBSAMPLE5

Table tag name : CARD

Column tag name

Source Column	Replace Tag Name
ID ...	
FNAME ...	
LNAME ...	
TITLE ...	
PHOTO ...	

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**.

24 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.

24.1 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 performance-related statistical parameters. Table 20-1 summarizes the statistical parameters.

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.

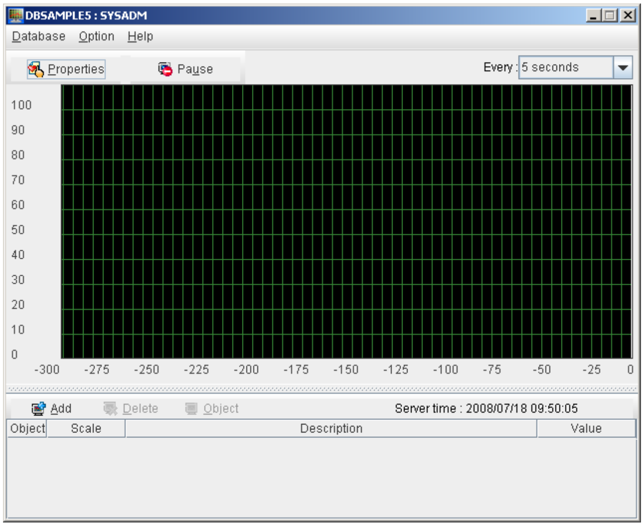
Number of SELECT Operations	The number of SELECT statements within the sample period.
Number of INSERT Operations	The number of INSERT statements within the sample period.
Number of UPDATE Operations	The number of UPDATE statements within the sample period.
Number of DELETE Operations	The number of DELETE 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 24-1 Database Monitor Tool Statistical Parameters

*Windows 2000 will only display the percentage of the first CPU in multi-processor systems.

➲ Viewing 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.

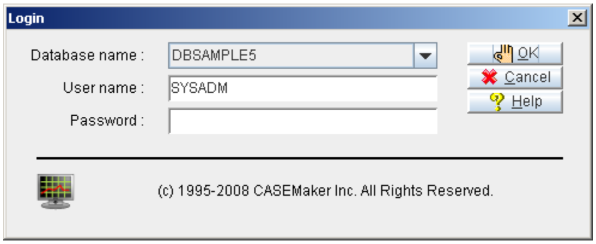


DBSAMPLE5 : SYSADM

Every : 5 seconds

Server time : 2008/07/18 09:50:05

2. If the **Database Monitor Tool** is started from the Tool menu in JDBA Tool it is automatically connected to the database. If the Database Monitor Tool is started from the Windows Start menu, connecting to the database must be performed manually. Connect to a database by selecting **Connect Database** from the **Database** menu. The **Login** window appears.



Login

Database name : DBSAMPLE5

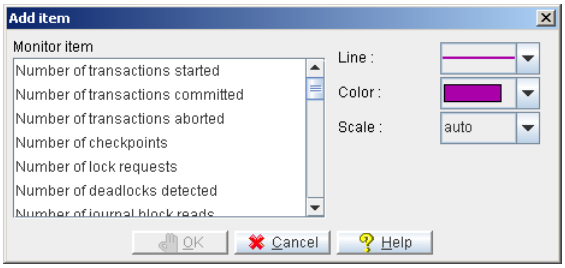
User name : SYSADM

Password :

(c) 1995-2008 CASEMaker Inc. All Rights Reserved.

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.

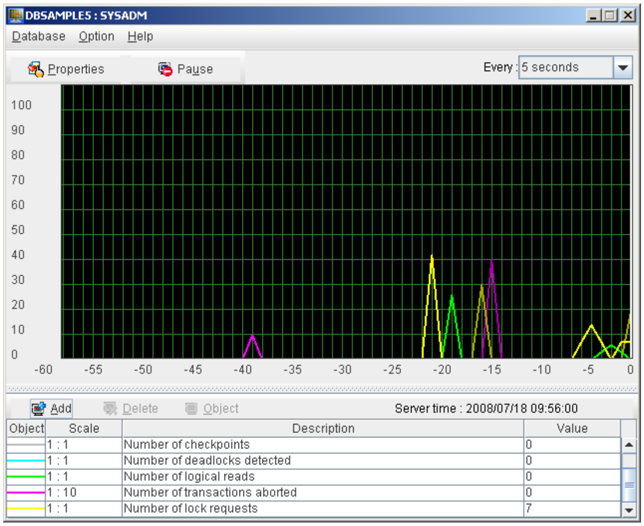

Add item
Monitor item
Number of transactions started
Number of transactions committed
Number of transactions aborted
Number of checkpoints
Number of lock requests
Number of deadlocks detected
Number of journal block reads
Line :
Color :
OK
Cancel
Help

Scale : auto

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.

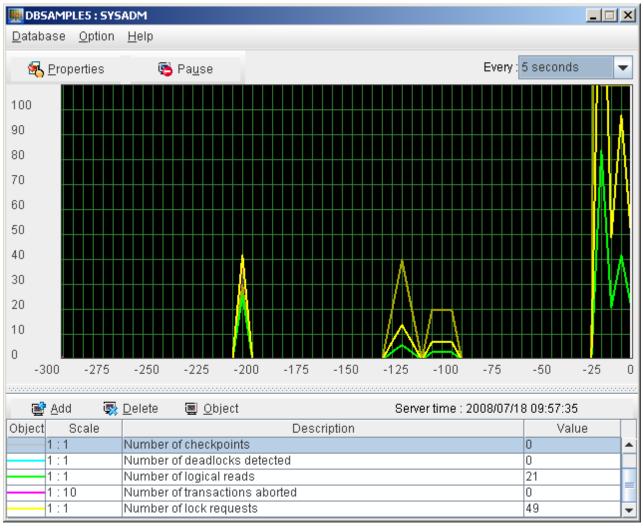


Every : 5 seconds

Server time : 2008/07/18 09:56:00

Object	Scale	Description	Value
⬜	1 : 1	Number of checkpoints	0
🟦	1 : 1	Number of deadlocks detected	0
🟩	1 : 1	Number of logical reads	0
🟪	1 : 10	Number of transactions aborted	0
🟨	1 : 1	Number of lock requests	7

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.



Every : 5 seconds

Server time : 2008/07/18 09:57:35

Object	Scale	Description	Value
⬜	1 : 1	Number of checkpoints	0
🟦	1 : 1	Number of deadlocks detected	0
🟩	1 : 1	Number of logical reads	21
🟪	1 : 10	Number of transactions aborted	0
🟨	1 : 1	Number of lock requests	49

9. To change the properties of the chart:

a) Click **Properties**. The **Properties** dialog appears.

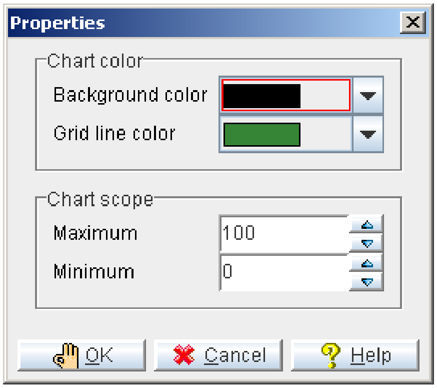


Chart color

Background color ■

Grid line color 🟩

Chart scope

Maximum 100

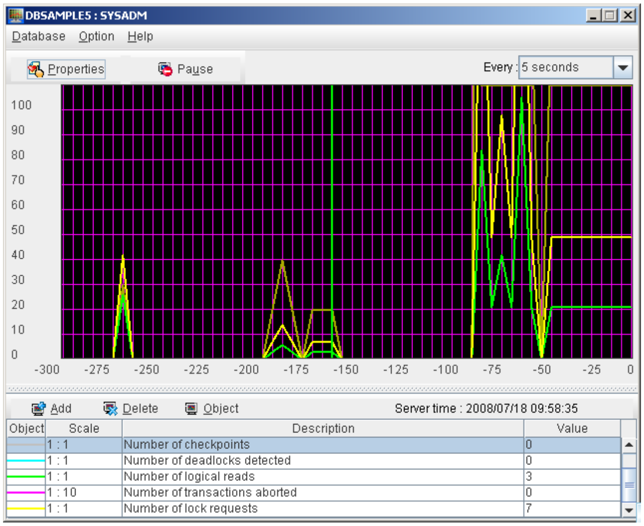
Minimum 0

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 chart changes to reflect in the changes.



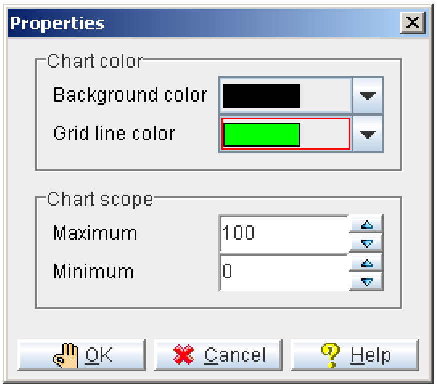
Server time : 2008/07/18 09:58:35

Object	Scale	Description	Value
⬜	1 : 1	Number of checkpoints	0
🟦	1 : 1	Number of deadlocks detected	0
🟩	1 : 1	Number of logical reads	3
🟪	1 : 10	Number of transactions aborted	0
🟨	1 : 1	Number of lock requests	7

10. To change the line type of an item:

a) Select one of the items.

b) Click **Object**. The **Line Style** window will open.



Background color ⬛

Grid line color 🟩

Maximum 100

Minimum 0

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.

24.2 Export to File Options

The export to file feature allows DBA’s to look at log files created by Jmonitor over an extended period of time. The DBA can then analyise the data and then adjust his database for optimal performance.

You have several options when setting the view of the created log files.

- Display the log data in column or row format
- Set the caption break for headings of monitored items by row or day
- Set the number of log files that will be created
- Set the size of the log files that be created

When setting the output of the log file to display in a column or row format. In a column format, the log file will list the monitored items in columns. For example:

```
                            [Free swap space (KB)]    [Total swap space (KB)]
===============================================================================
2004/06/21 19:45:20               459216                      633044
2004/06/21 19:45:25               459408                      633044
```

In a row format, the monitored items in the log file will be listed in a row. For example:

```
 Item Name                             Data                 Time
=================================================================================
[Free swap space (KB)]                457176         2004/06/21 19:59:25
[Total swap space (KB)]               633044         2004/06/21 19:59:25
```

You can set the caption break for headings of monitored items, like Date, Data and Item Names, to appear in the log file after a specified number of rows or days. When specifying the caption break for a number of rows the valid range is between 10 and 200. The default value is 25. When specifiying caption break value in days the valid range is from 1 to 30, with the default value set at 2.

The JMonitor export monitor item to external file function supports multi file exporting. You can set the number of log files for exporting. The exporting file function will create a new file and rename the previous exporting file with another file name. The current monitor item information will write into the new log file. The file count follows an ascending value. The valid range is between 10 and 20, with the default value set at 10.

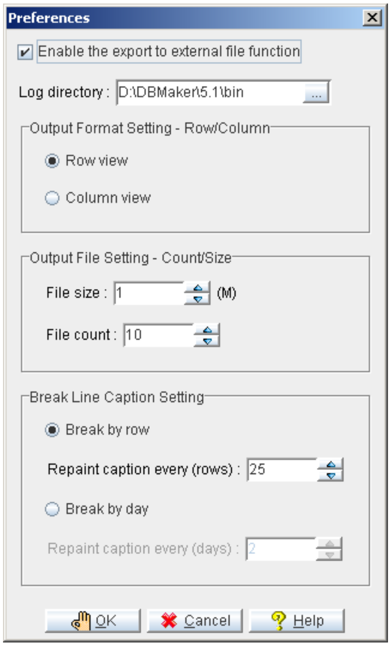
➲ Example

If we set the file count as 10, the first log created will be **Message.log**. After **Message.log** reaches its maximum size (which is set in the File Size field), the system will create a new log **Message0.log**. The contents of **Message.log** are transferred to **Message0.log**. The most current data is stored in **Message.log**.Once the log **Message.log** reaches its specified maximum size **Message1.log** is created. The contents of Message.log are transferred to **Message0.log**, and the contents of **Message0.log** are transferred to **Message1.log**. This process continues.

The file size option allows you to limit the size of the logs to a manageable size. The valid range is between 1MB to 10MB, with the default value set at 1 MB.

➲ Accessing the export to file options:

1. Select **Option** from the menu bar. A menu list appears.
2. Select **Preferences** from the menu list. The **Preferences** dialog box appears.



Preferences

☑ Enable the export to external file function

Log directory : D:\DBMaker\5.1\bin

Output Format Setting - Row/Column

● Row view

○ Column view

Output File Setting - Count/Size

File size : 1 (M)

File count : 10

Break Line Caption Setting

● Break by row

Repaint caption every (rows) : 25

○ Break by day

Repaint caption every (days) : 2

3. Set the **Enable the export to external file function** check box. The fields in the **Preferences** dialog box become active. This check box is checked by default. To disable the feature clear the check box.

4. Enter a file path for the log files in the **Log directory** field. Or select a path by clicking the **Browse** button. The default path is *…\DBMaker\5.1\bin.*

5. Select an output format from the **Output Format Setting – Row\Column** portion of the dialog box.

6. Set the out put file size number of files to be created in the **Output File Settings – Count\Size** portion of the dialog box.

7. Select the caption break for the log files in the **Break Line Caption Setting** portion of the dialog box.

8. Click **OK** after setting all options. You are returned to the **JMonitor** screen.

24.3 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 five database-monitoring functions provided by JDBA Tool: Session information, Lock status, Database status, Pending Transaction status and System Configuration.

- 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.
- System Configuration information allows the DBA to view all server configuration setting.

Monitoring Database Sessions

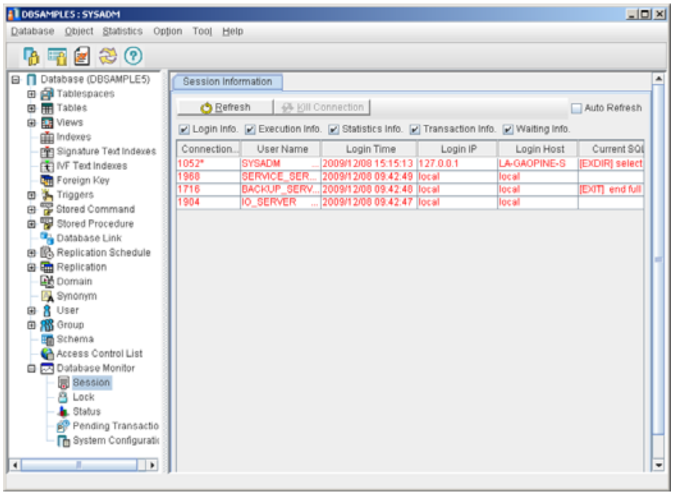
Session information allows you to view information that relates to users connected to the database and resources used by each user. Session Information is organized by five categories:

- Login Information
- Execution Information
- Statistical Information

- Transaction Information
- Waiting Information

➲ Viewing session information:

1. Select the **Database Monitor** node from the tree. Four nodes will appear, select the **Session** node. The following window appears.



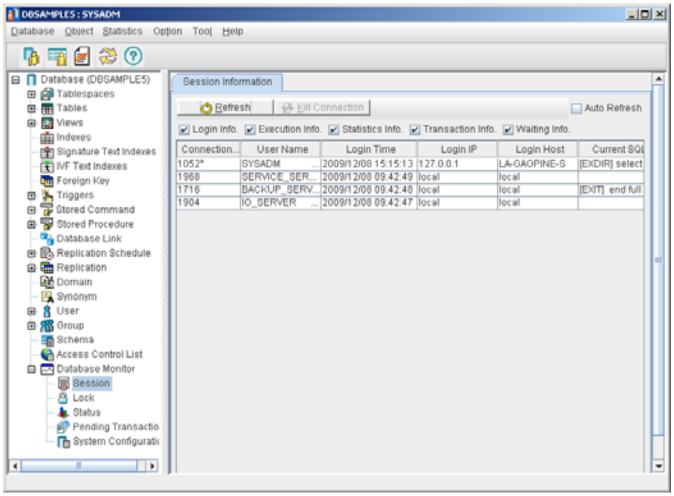
□ Auto Refresh

☑ Login Info. ☑ Execution Info. ☑ Statistics Info. ☑ Transaction Info. ☑ Waiting Info.

Connection...	User Name	Login Time	Login IP	Login Host	Current SQL
1052*	SYSADM ...	2009/12/08 15:15:13	127.0.0.1	LA-GAOPINE-S	[EXDIR] select
1960	SERVICE_SER...	2009/12/08 09:42:49	local	local	
1716	BACKUP_SERV...	2009/12/08 09:42:48	local	local	[EXIT] end full
1904	IO_SERVER ...	2009/12/08 09:42:47	local	local	

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 appearsin red. Information that remains the same appearsin black.



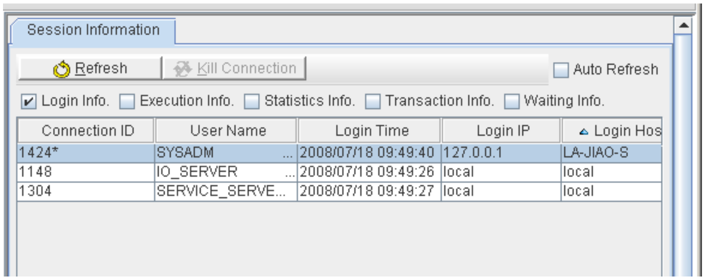
□ Auto Refresh

☑ Login Info. ☑ Execution Info. ☑ Statistics Info. ☑ Transaction Info. ☑ Waiting Info.

Connection...	User Name	Login Time	Login IP	Login Host	Current SQL
1052*	SYSADM ...	2009/12/08 15:15:13	127.0.0.1	LA-GAOPINE-S	[EXDIR] select
1968	SERVICE_SER...	2009/12/08 09:42:49	local	local	
1716	BACKUP_SERV...	2009/12/08 09:42:48	local	local	[EXIT] end full
1904	IO_SERVER ...	2009/12/08 09:42:47	local	local	

Login Information

This category contains the user’s name, information relevant to the connection and information about the computer their using.



□ Auto Refresh

☑ Login Info. □ Execution Info. □ Statistics Info. □ Transaction Info. □ Waiting Info.

Connection ID	User Name	Login Time	Login IP	Login Hos
1424*	SYSADM ...	2008/07/18 09:49:40	127.0.0.1	LA-JIAO-S
1148	IO_SERVER ...	2008/07/18 09:49:26	local	local
1304	SERVICE_SERVE...	2008/07/18 09:49:27	local	local

Figure 24-1 Session Information: Login Information

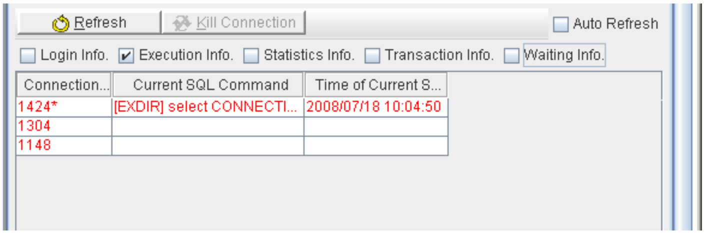
Column Name	Description
Connection ID:	The ID of the session.
User_Name:	The name of the user using the session.
Login_Time:	The time that the user connected to the database.
Login_IP_Addr:	The IP address of the user using the session.

COLUMN NAME	DESCRIPTION
Login_Host:	The host computer using the session.

Table 24-2 Login Information

EXECUTION INFORMATION

This category describes the type and status of SQL commands being executed by each user.



□ Auto Refresh

□ Login Info. ☑ Execution Info. □ Statistics Info. □ Transaction Info. □ Waiting Info.

Connection...	Current SQL Command	Time of Current S...
1424*	[EXDIR] select CONNECTI...	2008/07/18 10:04:50
1304		
1148		

Figure 24-2 Session Information: Execution Information

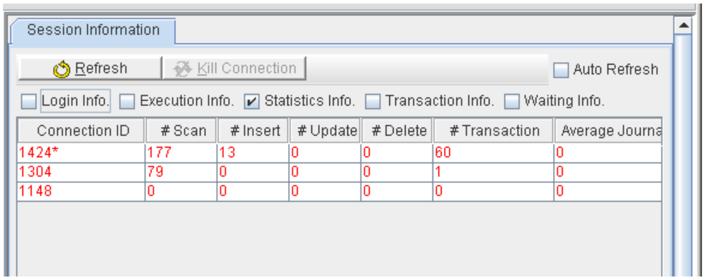
COLUMN NAME	DESCRIPTION
Connection ID:	The ID of the session.
Current SQL Command	The most recently executed SQL command and the command status. The command status can be the following: [PRE] - The SQL command is being prepared. [EXEC] - The SQL command is being executed from SQLExecute call. [EXDIR] - 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.

COLUMN NAME	DESCRIPTION
Time of Current SQL Command	The time when the most recently used SQL command was executed.

Table 24-3 Execution Information

STATISTICAL INFORMATION

This category displays statistics relevant to the data.



□ Auto Refresh

□ Login Info. □ Execution Info. ☑ Statistics Info. □ Transaction Info. □ Waiting Info.

Connection ID	# Scan	# Insert	# Update	# Delete	# Transaction	Average Journa
1424*	177	13	0	0	60	0
1304	79	0	0	0	1	0
1148	0	0	0	0	0	0

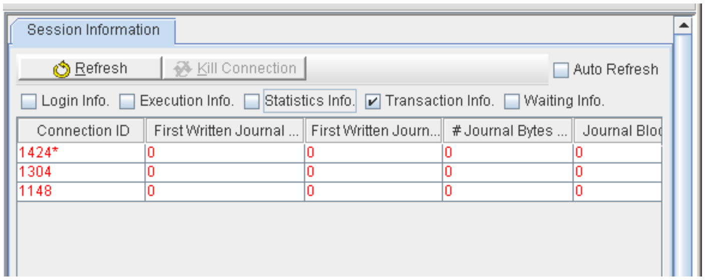
Figure 24-3 Session Information: Statistical Information

COLUMN NAME	DESCRIPTION
Connection ID:	The ID of the session.
#Scan:	The number of SELECT operations in the session.
#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 24-4 Statistical Information

TRANSACTION INFORMATION

Displays journal file activity, specifically the start and end points of transactions within the journal.



□ Auto Refresh

□ Login Info. □ Execution Info. □ Statistics Info. ☑ Transaction Info. □ Waiting Info.

Connection ID	First Written Journal ...	First Written Journ...	# Journal Bytes ...	Journal Bloc
1424*	0	0	0	0
1304	0	0	0	0
1148	0	0	0	0

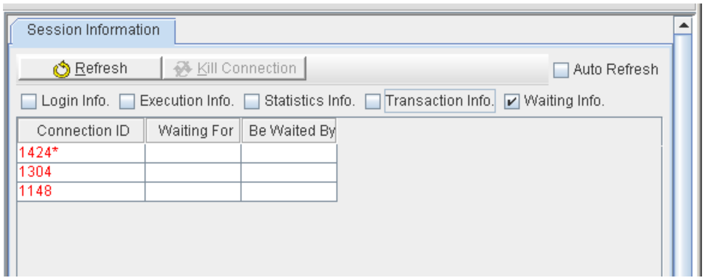
Figure 24-4 Session Information: Transaction Information

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.
# 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 24-5 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.



□ Auto Refresh

□ Login Info. □ Execution Info. □ Statistics Info. □ Transaction Info. ☑ Waiting Info.

Connection ID	Waiting For	Be Waited By
1424*		
1304		
1148		

Figure 24-5 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 24-6 Waiting Information

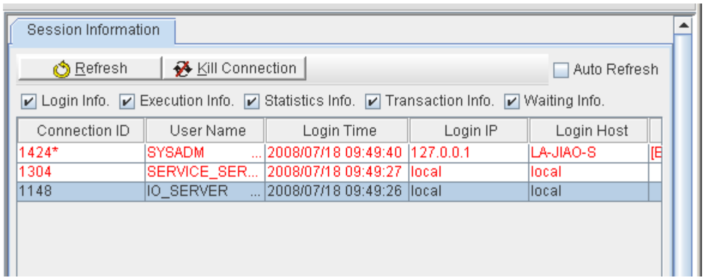
Killing Connections

You can choose to terminate an active connection.

➲ Killing 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.



Connection ID	User Name	Login Time	Login IP	Login Host	
1424*	SYSADM ...	2008/07/18 09:49:40	127.0.0.1	LA-JIAO-S	[E
1304	SERVICE_SER...	2008/07/18 09:49:27	local	local	
1148	IO_SERVER ...	2008/07/18 09:49:26	local	local	

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.

The following table summarizes the items on the lock page:

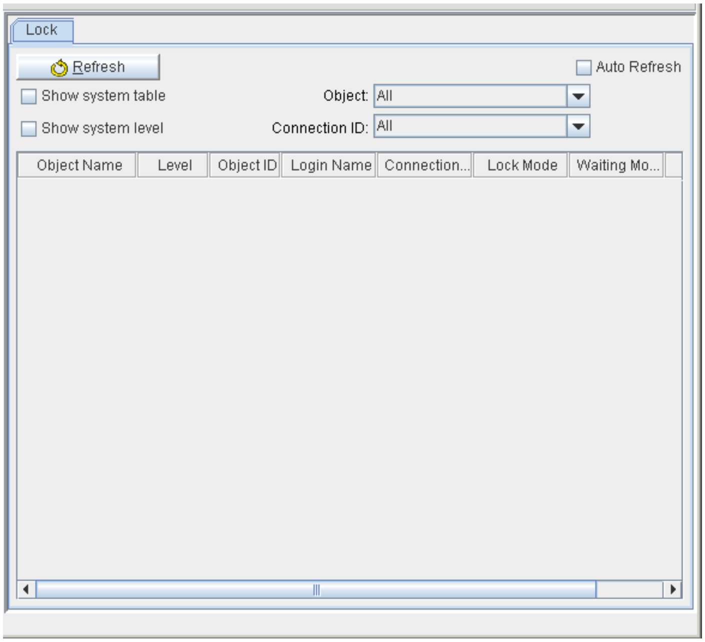
ITEM	DESCRIPTION
Object Name:	The object name for a locked object, or an object that a lock is requested for.
Level:	The current lock level. Four lock levels exist for an object: System, Table, Page, and Row.
Object ID	An identifying number unique to each table, page, or row in the database.
Login Name	Name of the user that has the lock on the object.
Connection ID:	The ID of the session

ITEM	DESCRIPTION
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 24-7 Database Lock items

➲ Viewing database lock status:

1. Select the **Lock** node from the **Database Monitor** node in the tree. The following window appears.



Object: All

Connection ID: All

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:

- 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. Table 24-11 summarizes the lock status items and their corresponding SYSINFO.INFO values.

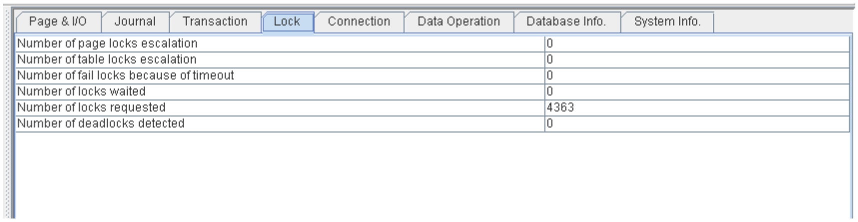
Lock Status Item	Description
Number of Page Locks Escalation	The accumulated number of row locks that have been escalated to page locks. (NUM_ROW_LOCK_UPG)
Number of Table Locks Escalation	The accumulated number of page locks that have been escalated to table locks. (NUM_PAGE_LOCK_UPG)
Number of fail 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)

LOCK STATUS ITEM	DESCRIPTION
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 24-8 Lock Status Items

➲ Viewing lock status:

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Lock** tab. The following page appears.



Number of page locks escalation	0
Number of table locks escalation	0
Number of fail locks because of timeout	0
Number of locks waited	0
Number of locks requested	4363
Number of deadlocks detected	0

VIEWING CONNECTION STATUS

The connection status page displays connection information statistics from the SYSINFO table. Table 24-11 summarizes the connection status items and their corresponding SYSINFO.INFO values.

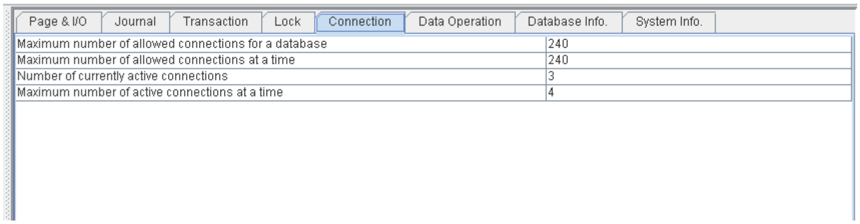
CONNECTION STATUS ITEM	DESCRIPTION
Maximum number of allowed connections for a database	The hard limitation of connections. This value is determined by the DB_MaxCo keyword when the database is created or started with a new journal. (NUM_MAX_HARD_CONNECT)
Maximum number of allowed connections at a time	The soft limitation of connections, which is less than or equal to the number of hard connections (NUM_MAX_SOFT_CONNECT)

CONNECTION STATUS ITEM	DESCRIPTION
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 24-9 Connection Status items

➲ Viewing connection status:

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Connection** tab. The following page appears.



Maximum number of allowed connections for a database	240
Maximum number of allowed connections at a time	240
Number of currently active connections	3
Maximum number of active connections at a time	4

VIEWING DATA OPERATION STATUS

The data operation status page displays data operation statistics from the SYSINFO table. Table 24-11 summarizes the data operation status items and their corresponding SYSINFO.INFO values.

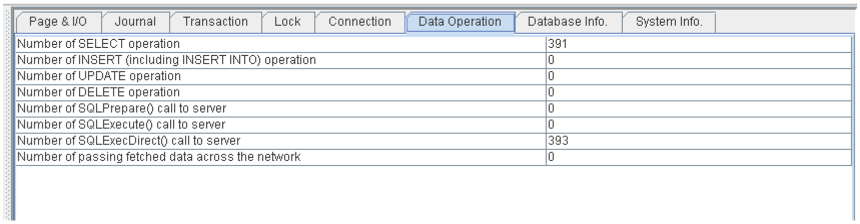
DATA OPERATION STATUS ITEM	DESCRIPTION
Number of SELECT Operations	The total number of SELECT operations performed since the database was started. (NUM_SQL_SELECT)
Number of INSERT (including INSERT INTO) Operations	The total number of SELECT operations performed since the database was started. (NUM_SQL_ INSERT)

DATA OPERATION STATUS ITEM	DESCRIPTION
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 24-10 Data Operations Status items

➲ Viewing 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 appears.



Number of SELECT operation	391
Number of INSERT (including INSERT INTO) operation	0
Number of UPDATE operation	0
Number of DELETE operation	0
Number of SQLPrepare() call to server	0
Number of SQLExecute() call to server	0
Number of SQLExecDirect() call to server	393
Number of passing fetched data across the network	0

VIEWING DATABASE INFORMATION

The database information page displays database information statistics from the SYSINFO table. Table 24-11 summarizes the database information items and their corresponding SYSINFO.INFO values.

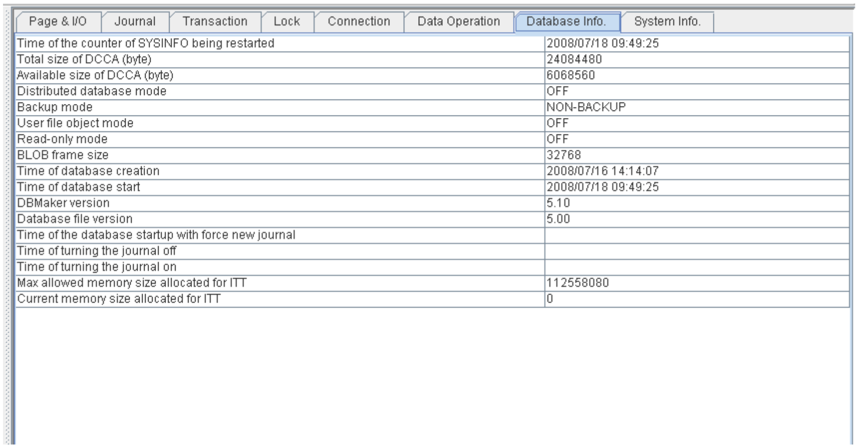
DATABASE INFORMATION ITEMS	DESCRIPTION
Time of the counter of SYSINFO being started	The last time and date that data in the SYSINFO table was reset. The SYSINFO table gets reset when one of the counters overflows, or the SET SYSINFO CLEAR command is used. (SYSINFO_RESET_TIME)
Total size of DCCA	Total 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 24-11 Database Information Items

➲ Viewing Database Information:

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Database Info** tab. The following page appears.



Time of the counter of SYSINFO being restarted	2008/07/18 09:49:25
Total size of DCCA (byte)	24084480
Available size of DCCA (byte)	6068560
Distributed database mode	OFF
Backup mode	NON-BACKUP
User file object mode	OFF
Read-only mode	OFF
BLOB frame size	32768
Time of database creation	2008/07/16 14:14:07
Time of database start	2008/07/18 09:49:25
DBMaker version	5.10
Database file version	5.00
Time of the database startup with force new journal	
Time of turning the journal off	
Time of turning the journal on	
Max allowed memory size allocated for ITT	112558080
Current memory size allocated for ITT	0

VIEWING SYSTEM INFORMATION

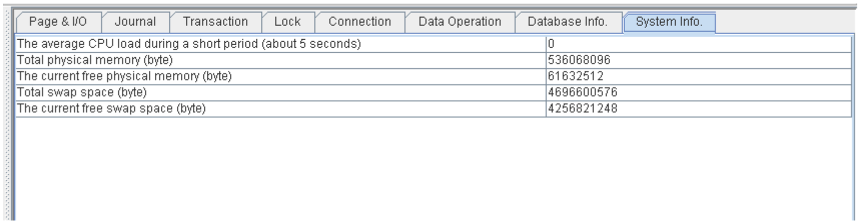
The system information page displays operating system statistics from the SYSINFO table. Table 24-11 summarizes the system information items and their corresponding SYSINFO.INFO values.

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 24-12 System Information items

➲ Viewing system information:

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **System Info** tab. The following page appears.



The average CPU load during a short period (about 5 seconds)	0
Total physical memory (byte)	536068096
The current free physical memory (byte)	61632512
Total swap space (byte)	4696600576
The current free swap space (byte)	4256821248

VIEWING PAGE AND I/O STATUS

The page and I/O status page displays operating system statistics from the SYSINFO table. Table 24-11 summarizes the page and I/O status items and their corresponding SYSINFO.INFO values.

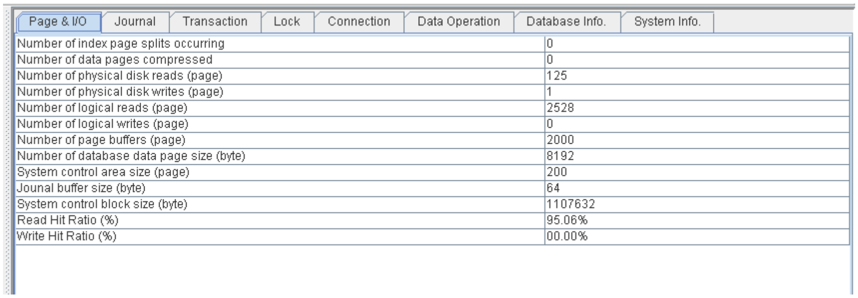
PAGE AND I/O STATUS ITEMS	DESCRIPTION
Number of Index Page Splits Occurring	The accumulated number of index pages split in the current run time. (NUM_IDX_SPLIT)
Number of Data Pages Compressed	The accumulated number of data pages reorganized in the current run time. (NUM_PAGE_COMPRESS)
Number of Physical 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 24-13 Page and I/O Status items

➲ Viewing 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 appears.



Number of index page splits occurring	0
Number of data pages compressed	0
Number of physical disk reads (page)	125
Number of physical disk writes (page)	1
Number of logical reads (page)	2528
Number of logical writes (page)	0
Number of page buffers (page)	2000
Number of database data page size (byte)	8192
System control area size (page)	200
Jounal buffer size (byte)	64
System control block size (byte)	1107632
Read Hit Ratio (%)	95.06%
Write Hit Ratio (%)	00.00%

VIEWING JOURNAL STATUS

The journal status page displays operating system statistics from the SYSINFO table. Table 24-11 summarizes the journal status items and their corresponding SYSINFO.INFO values.

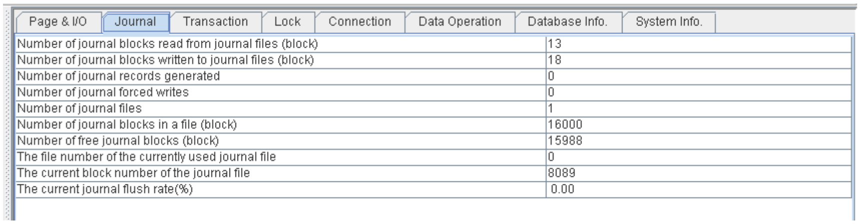
JOURNAL STATUS ITEMS	DESCRIPTION
Number of journal blocks read from journal files	The accumulated number of journal block reads in the current run time. (NUM_JNL_BLK_READ)
Number of journal blocks written to journal files	The accumulated number of journal block writes in the current run time. (NUM_JNL_BLK_WRITE)
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)

JOURNAL STATUS ITEMS	DESCRIPTION
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 24-14 Journal Status items

➲ Viewing journal status:

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Journal** tab. The following page appears.



Number of journal blocks read from journal files (block)	13
Number of journal blocks written to journal files (block)	18
Number of journal records generated	0
Number of journal forced writes	0
Number of journal files	1
Number of journal blocks in a file (block)	16000
Number of free journal blocks (block)	15988
The file number of the currently used journal file	0
The current block number of the journal file	8089
The current journal flush rate(%)	0.00

VIEWING TRANSACTION STATUS

The transaction status page displays operating system statistics from the SYSINFO table. Table 24-11 summarizes the transaction status items and their corresponding SYSINFO.INFO values.

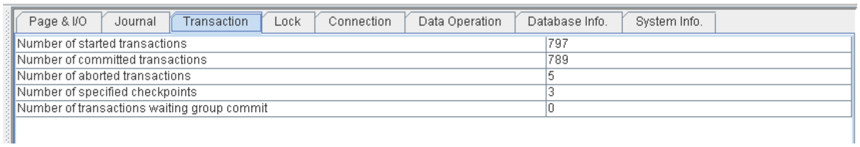
TRANSACTION STATUS ITEMS	DESCRIPTION
Number of started transactions	The cumulative number of transactions started during the current runtime. (NUM_STARTED_TRANX)
Number of committed transactions	The cumulative number of committed transactions during the current runtime. (NUM_COMMITED_TRANX)

TRANSACTION STATUS ITEMS	DESCRIPTION
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 24-15 Transaction Status items

➲ Viewing 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.



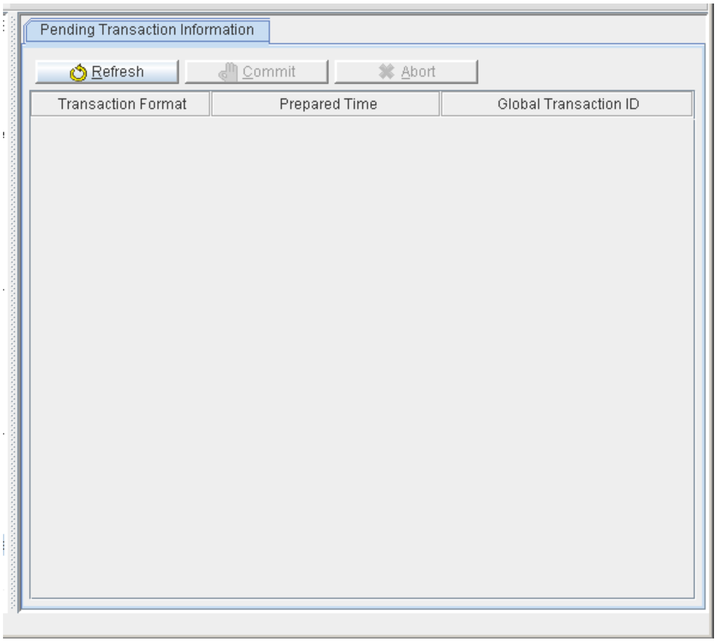
Number of started transactions	797
Number of committed transactions	789
Number of aborted transactions	5
Number of specified checkpoints	3
Number of transactions waiting group commit	0

Monitoring Pending Transactions

In a distributed database environment, transactions can be made across multiple databases. DBMaker effectively manage this sort of transaction using 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.

- Viewing pending transactions:

   1. Select Pending Transaction from the Database Monitor node of the tree
   2. A list of current pending transactions will appear.

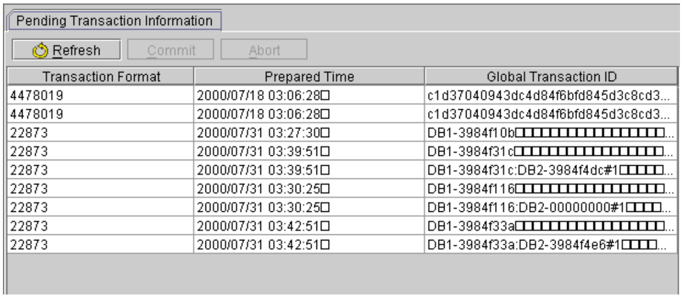


   3. Select the **Refresh** button to view the current pending transaction status.

- Commiting pending transactions:

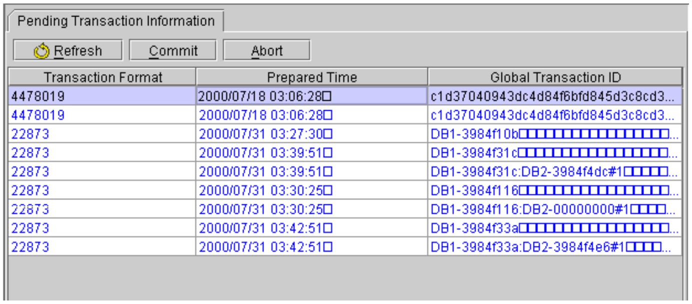
   1. Select **Pending Transaction** from the **Database Monitor** node of the tree

2. A list of current pending transactions will appear.



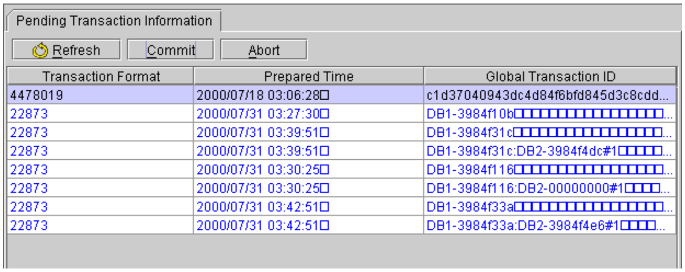
Transaction Format	Prepared Time	Global Transaction ID
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cd3...
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cd3...
22873	2000/07/31 03:27:30□	DB1-3984f10b□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c:DB2-3984f4dc#1□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116:DB2-00000000#1□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a:DB2-3984f4e6#1□□□□...

3. Select the transaction to commit from the list in the main console. It will highlight blue.



Transaction Format	Prepared Time	Global Transaction ID
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cd3...
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cd3...
22873	2000/07/31 03:27:30□	DB1-3984f10b□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c:DB2-3984f4dc#1□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116:DB2-00000000#1□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a:DB2-3984f4e6#1□□□□...

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.

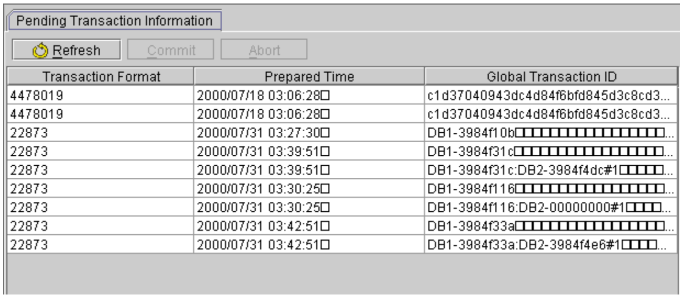


Transaction Format	Prepared Time	Global Transaction ID
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cdd...
22873	2000/07/31 03:27:30□	DB1-3984f10b□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c:DB2-3984f4dc#1□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116:DB2-00000000#1□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a□□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a:DB2-3984f4e6#1□□□□...

➲ Aborting pending transactions:

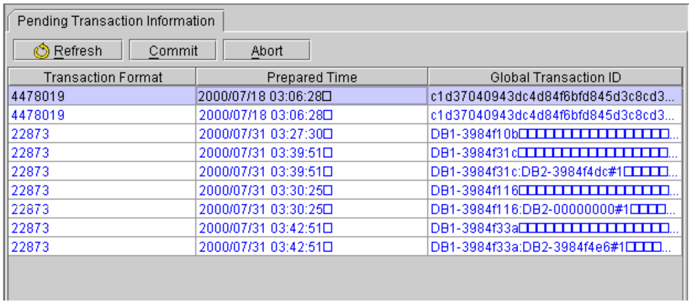
1. Select **Pending Transaction** from the **Database Monitor** node of the tree

2. A list of current pending transactions will appear.



Transaction Format	Prepared Time	Global Transaction ID
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cd3...
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cd3...
22873	2000/07/31 03:27:30□	DB1-3984f10b□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c:DB2-3984f4dc#1□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116:DB2-00000000#1□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a:DB2-3984f4e6#1□□□□...

3. Select the transaction to abort from the list in the main console. It will highlight blue.



Transaction Format	Prepared Time	Global Transaction ID
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cd3...
4478019	2000/07/18 03:06:28□	c1d37040943dc4d84f6bfd845d3c8cd3...
22873	2000/07/31 03:27:30□	DB1-3984f10b□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:39:51□	DB1-3984f31c:DB2-3984f4dc#1□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:30:25□	DB1-3984f116:DB2-00000000#1□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a□□□□□□□□□□□□□□□□...
22873	2000/07/31 03:42:51□	DB1-3984f33a:DB2-3984f4e6#1□□□□...

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.

Monitoring System Configurations

During runtime, database settings may be altered. This makes it even more important for users, on the client site, to know the current server settings. Users can do this with either the JDBA tool or the SYSCONFIG system table.

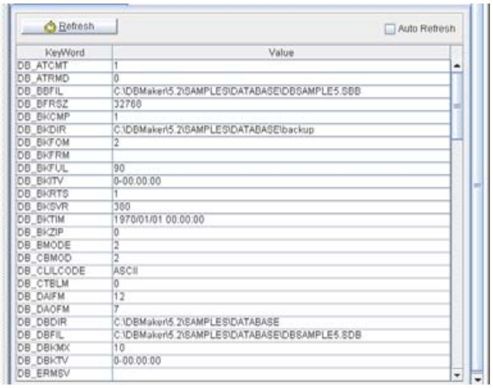
The following table summarizes items found on the system configuration page:

ITEM	DESCRIPTION
Keyword	The keyword used in dmconfig.ini
Value	Current value of the setting

Table 24-16 System Configuration Items

- Viewing system configuration information:

1. From the **Database Monitor** node select **System Configuration** .
2. A list of current server system settings appears.



KeyWord	Value
DB_ATCMT	1
DB_ATRMD	0
DB_BBFIL	C:\DBMaker\5.2\SAMPLES\DATABASE\DBSAMPLE5.SBB
DB_BFRSZ	32768
DB_BKCMP	1
DB_BKDIR	C:\DBMaker\5.2\SAMPLES\DATABASE\backup
DB_BKFOM	2
DB_BKFRM	
DB_BKFUL	90
DB_BKITV	0-00:00:00
DB_BKRTS	1
DB_BKSVR	380
DB_BKTIM	1970/01/01 00:00:00
DB_BKZIP	0
DB_BMODE	2
DB_CBMOD	2
DB_CLILCODE	ASCII
DB_CTBLM	0
DB_DAIFM	12
DB_DAOFM	7
DB_DBDIR	C:\DBMaker\5.2\SAMPLES\DATABASE
DB_DBFIL	C:\DBMaker\5.2\SAMPLES\DATABASE\DBSAMPLE5.SDB
DB_DBKMX	10
DB_DBKTV	0-00:00:00
DB_ERMSV	

3. Click the **Auto Refresh** button to refresh the system configuration information.

25 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.

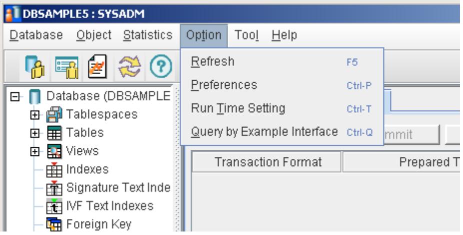
25.1 Altering Run Time Settings

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:

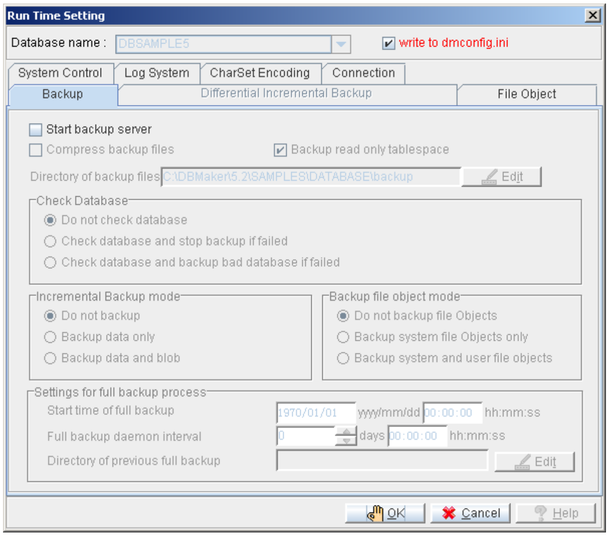
- Backup settings: Allows you to modify automatic full backup frequency and location, set file object backup mode, or disable the backup server daemon.
- Differential and Incremental settings: Allows you to modify automatic differential and incremental backup frequency.
- File object settings: Allows you to enable or disable user file objects and modify the location of system file objects.
- System control settings: Allows you to enable distributed database mode for synchronous table replication and set the number of pages or frames to extend autoextend tablespaces.
- Log system settings: Allows you to enable and set options for the Log System.
- Character Set settings: Allows client side users to select the language code for the database and error messages.
- Connection settings: Provides the option to block new connections.

➲ Accessing 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.



Database name : DBSAMPLE5 ☑ write to dmconfig.ini

□ Start backup server

□ Compress backup files ☑ Backup read only tablespace

Directory of backup files C:\DBMaker\5.2\SAMPLES\DATABASE\backup

Check Database
- ● Do not check database
- ○ Check database and stop backup if failed
- ○ Check database and backup bad database if failed

Incremental Backup mode
- ● Do not backup
- ○ Backup data only
- ○ Backup data and blob

Backup file object mode
- ● Do not backup file Objects
- ○ Backup system file Objects only
- ○ Backup system and user file objects

Settings for full backup process
- Start time of full backup 1970/01/01 yyyy/mm/dd 00:00:00 hh:mm:ss
- Full backup daemon interval 0 days 00:00:00 hh:mm:ss
- Directory of previous full backup

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.*

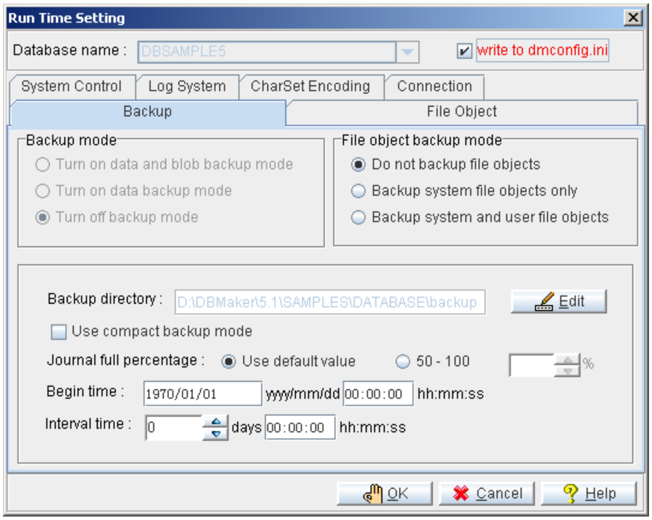
5. Set the following options:

- Backup Settings
- Differential and Incremental Settings
- File Object Settings

- System Control
- Log System
- Character Set
- Connection

Changing Backup Settings

The followingf backup settings may be altered during run-time: check database, incremental backup mode, file object backup mode, backup location, deactivation of backup BLOB, deactivation of backup data and full backup process setting. During run-time, backup modes can be changed but only to a state that backs up less data than does the current state. In the case of file objects, this exception does not apply. For example, this may be usefulto free up resources by changing the backup mode from "backup data and BLOB" to "backup data".



Run Time Setting

Database name : DBSAMPLE5 ☑ write to dmconfig.ini

Backup mode
○ Turn on data and blob backup mode
○ Turn on data backup mode
● Turn off backup mode

File object backup mode
● Do not backup file objects
○ Backup system file objects only
○ Backup system and user file objects

Backup directory : D:\DBMaker\5.1\SAMPLES\DATABASE\backup

□ Use compact backup mode

Journal full percentage : ● Use default value ○ 50 - 100 %

Begin time : 1970/01/01 yyyy/mm/dd 00:00:00 hh:mm:ss

Interval time : 0 days 00:00:00 hh:mm:ss

Figure 25-1 Backup Run-Time Settings Dialog

START BACKUP SERVER

Enabling the **Start Backup Server** check box activates the Backup Server demaon. Some settings can be changed during run-time. Do this from the **Backup** tab and the **Differential Incremental Backup** tab. Please note, settings for **Incremental Backup Mode** cannot be adjusted during run-time. . All other items found in the **Backup** and **Differential Incremental Backup** tabs are disabled.

COMPRESS BACKUP FILES

Selecting **Compress Backup files** activates backup file compression. When activated, the Backup Server compresses backup files during full backups.

BACKUP READ-ONLY TABLESPACE

Selecting this check box tells Backup Server to backup read-only tablespace files when performing a full backup. Deselecting this check box tells Backup Server to skip backup of read-only tablespace files. Please note, a full backup must be performed after setting a tablespace to read-only. If you don’t backup the latest files, serious errors may result when restoring, from a full backup, databases containing a read-only tablespace.

DIRECTORY OF BACKUP FILES

The Backup Server stores back up files such as journal files in the backup directory. The backup directory sand the database files should not be located on the same disk. The default path for backup files is the database directory */backup* and DBMaker automatically creates it. The total length of the backup directory path must not exceed 255 characters.

CHECK DATABASE

You can choose to check databases before full or differential backups. The following table describes how the check modes operate.

CHECK MODE	DESCRIPTION
Do not check database	Backup Server does not check databases

CHECK MODE	DESCRIPTION
	before full backup or differential backups.
Check database and stop backup if failed	Backup Server checks databases before full and differential backups. Error messages are logged and the backup is terminated when a database is damaged.
Check database and backup bad database if failed	Backup Server checks databases before full and differential backups. Error message are logged and when a damaged database is first detected, the backup continues with the damaged files copied to the BKDIR/BADDB directory. During subsequent back up operations, Backup Server logs error messages but does not copy damaged databases. Damaged databases that subsequently report no damage are backed up normally and Backup Server removes the damaged copy of the database from the BKDIR/BADDB directory.

Incremental backup files are always generated follow full and differential backups. Backup Server places incremental backup files associated with a damaged database into BKDIR/BADDB. The incremental backup files are moved back to BKDIR when the associated database is no longer damaged.

INCREMENTAL 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
Do not backup	Disables the incremental backup daemon. In this mode, journal files are not backed up.
Backup data only	Sets the incremental backup daemon to backup all data journal pages.
Backup data and blob	BLOB data is written to the journal and the incremental backup daemon backs up all journal files.

Table 25-1Incremental 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 only	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 25-2 File Object Backup Modes

START TIME OF FULL BACKUP

The date and time values that you set in the yyyy/mm/dd and hh:mm:ss fields specifie when the database’s first full backup will begin. Please careful input this two values.

FULL BACKUP DAEMON INTERVAL

This value specifies, in days, the full backup time interval. The adjacent field accepts a value, in hours, minutes, and seconds, and specifies this portion of the full backup time interval. The total time interval between full backups is determined by adding the two values. For example, inputting **1** for the **days** field and inputting **12:00:00** for the **hh:mm:ss** field specifies a full backup execution every day and a half.

DIRECTORY OF PREVIOUS FULL BACKUP

This Directory of previous full backup specifies the directorie where backup server saved previous versions of full backup files.

➲ Changing backup settings:

1. Select the **Start backup server** check box.
2. Select the **Compress backup files** check box to enable backup compression.
3. Select the **Backup read-only tablespace** check box to include read-only tablespace files when performing a full backup.
4. Enter a path into the **Directory of backup files** field or click the adjacent **Browse** button Edit to specify a destination for the backup files.
5. Select one of the check modes from the **Check Database** box.

    - Select **Do not check database** to skip checking databases before full and differential backups.

    - Select **Check database and stop backup if failed** to check database before full and differential backups. Additionally, with this option, Backup Server logs error messages, and upon discovering a damaged database, terminates the backup operation.

    - Select **Check database and backup bad database if failed** to check database before full and differential backups. Additionally, Backup Server logs error messages but continues to back up damaged databases, placing them in the BKDIR/BADDB directory.
6. Select a backup mode from the **incremental backup mode** box.

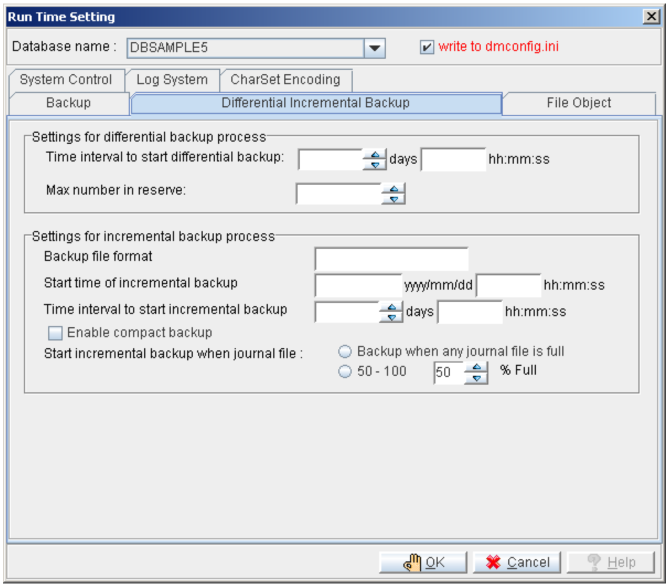
    - To change the backup mode to backup data only, select the **Backup data only**option button.

- To change the backup mode to backup data and blob, select the **Backup data and blob** option button.
- To turn off the backup mode, select the **Do not backup** option button.

7. Select a file object backup mode from the **Backup file object mode** box.
   - Select **Do not backup file objects** to exclude file objects from the backup.
   - Select **Backup system file objects only** to backup system file objects.
   - Select **Backup system and user file objects** to backup system and user file objects.
8. Establishing settings for a full backup process from the **Settings for full backup process** area.
   - Use the **Start time of full backup** fields to specify a date and a time when full backups will begin .
   - Enter the number of days, hours, minutes, and seconds between each successive full backup in the **Full backup daemon interval** fields.
   - Enter a path in the **Directory of previous full backup** field or select the adjacent browse button Edit to specify a destination for Backup Server to put previous version of full backup files.
9. Select one of the other tabs at the top (to change other runtime settings) or click the OK button at the bottom of the Run Time Settings window to save your changes.

Changing Differential and Incremental Backup Settings

Differential and incremental backup settings page allows users to specify details for their differential and incremental backup processes.



Run Time Setting

Database name : DBSAMPLE5 ☑ write to dmconfig.ini

Settings for differential backup process

Time interval to start differential backup: days hh:mm:ss

Max number in reserve:

Settings for incremental backup process

Backup file format

Start time of incremental backup yyyy/mm/dd hh:mm:ss

Time interval to start incremental backup days hh:mm:ss

□ Enable compact backup

Start incremental backup when journal file : ○ Backup when any journal file is full ○ 50 - 100 50 % Full

Figure 25-2 The Differential and Incremental Backup page of the Run Time Settings dialog

TIME INTERVAL TO START DIFFERENTIAL BACKUP

This number specifies, in days, the interval at which differential backups occur. Beside the combo box is a field to input an hours, minutes and seconds time value. The sum of these three values determines the total time interval between differential backups. For example, entering 1 in the **days** field and entering 12:00:00 in the **hh:mm:ss** field will cause Backup Server to execute a differential backup every day and a half.

MAX NUMBER IN RESERVE

The value specifies the maximum number of differential backups, after a full backup (i.e., the differential base), that Backup Server retains. When the number of saved differential backups exceeds this value, Backup Server removes the oldest differential backup. The default value is 10. Set this value to 0 to save an unlimited number of differential backups.

BACKUP FILE FORMAT

This value specifies the naming format Backup Server uses for incremental backup journal files. The backup filename format may include both text constants and format sequences (e.g., escape sequences), that represent special character strings.

START TIME OF INCREMENTAL BACKUP

This value specifies when the database' s first incremental backup begins. Enter your desired date in the yyyy/mm/dd field and time in the hh:mm:ss field.

TIME INTERVAL TO START INCREMENTAL BACKUP

This number specifies, in days, the interval at which incremental backups occur. Beside the combo box is a field to input an hours, minutes and seconds time value. The sum of these three values determines the total time interval between incremental backups. For example, entering 1 in the **days** field and entering 12:00:00 in the **hh:mm:ss** field will cause Backup Server to execute a incremental backup every day and a half.

ENABLE COMPACT BACKUP

Enable Compact Backup Mode ensures that Backup Server only copies needed journal files when performing incremental backups. As a result, less storage space is required at the backup location, however, restoring a database may take more time.

START INCREMENTAL BACKUP WHEN JOURNAL FILE

DBMaker can initiate an incremental backup before the journal file is full. Backup Server performs an online incremental backup when a journal file fills to the percent value specified by the journal trigger. Journal trigger values between 50 and 100 are valid. You can combine a journal trigger value with the backup schedule to ensure a regular schedule of database backups.

➲ Changing differential and incremental backup settings:

1. Setting up a differential backup process from the **Settings for differential backup process** area.

- Enter the number of days, hours, minutes and seconds between each successive full backup in the **Time interval to start differential backup** time fields.
- Enter a maximum value in the **Max number in reserve** fields.

2. Setting up an incremental backup process from the **Settings for incremental backup process** area.

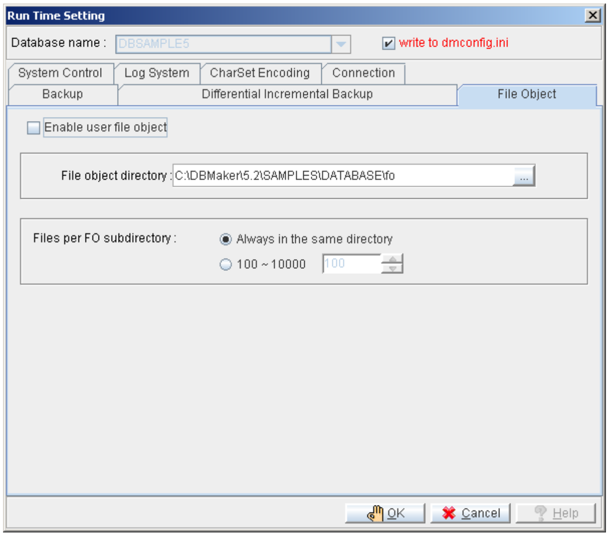
   - Enter a format the Backup Server will use to name incremental backup journal files.
   - Indicate a date and a time in the **Start time of incremental backup** fields to specify when incremental backups will begin.
   - Enter the number of days, hours, minutes and seconds between each successive full backup in the **Time interval to start incremental backup** time fields.

3. To enable compact backups, select the **Enable Compact Backup** check box.
4. Incremental backups can automatically execute when journal files have filled to a predetermined percent. Next to **Start incremental backup when journal file**:

   - Select the **Backup when any journal file is full** option button to set incremental backups to execute when any journal file is full.
   - Enter a value from 50 to 100 in the 50 – 100 % field to set incremental backups to execute when any journal file has filled to the value entered.

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.



Run Time Setting

Database name : DBSAMPLE5

☑ write to dmconfig.ini

□ Enable user file object

File object directory : C:\DBMaker\5.2\SAMPLES\DATABASE\fo

Files per FO subdirectory :

● Always in the same directory

○ 100 ~ 10000   100

Figure 25-3 The File Object page of the Run Time Settings dialog

➲ Changing file object settings:

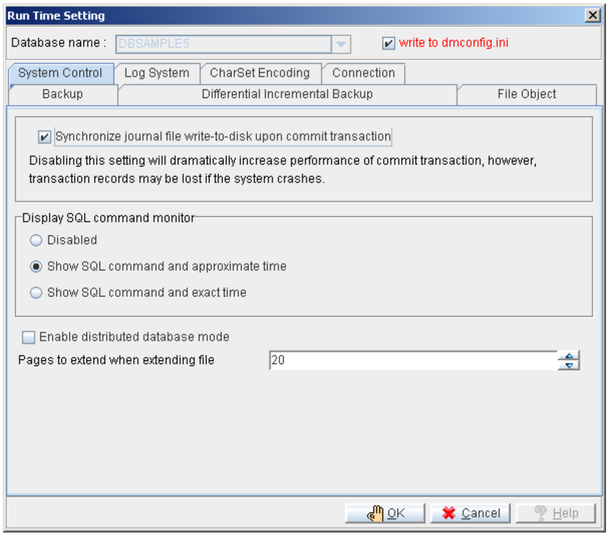
1. Select the **File Object** tab from the **Run Time Setting** window.
2. To enable the database to use external file objects, click on the **Enable User File Object** check box.
3. Enter a path into or select the browse button .. next to the **System File Object Directory** field to indicate the location 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.



Database name : DBSAMPLE5

☑ write to dmconfig.ini

☑ Synchronize journal file write-to-disk upon commit transaction

Disabling this setting will dramatically increase performance of commit transaction, however, transaction records may be lost if the system crashes.

Display SQL command monitor

○ Disabled

● Show SQL command and approximate time

○ Show SQL command and exact time

□ Enable distributed database mode

Pages to extend when extending file: 20

Figure 25-4The 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.

Omitting the second step may greatly increase the performance of the commit transaction process. 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 is 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 use of the database 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

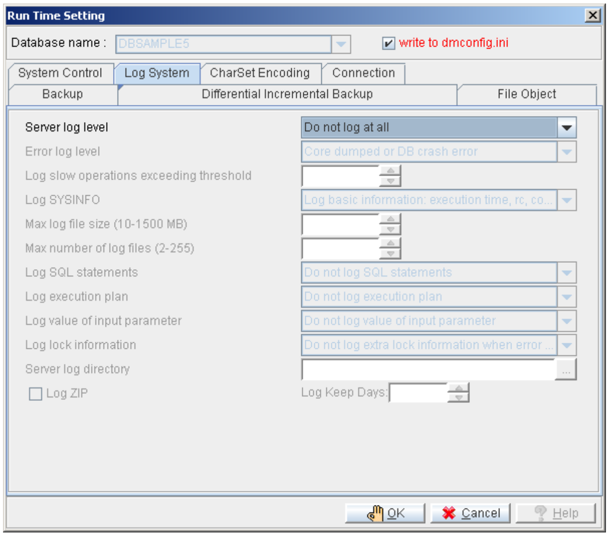
When all pages in a data file or BLOB file are full, DBMaker can automatically extend the number of pages or frames in the file to allow the database to grow. The **Number of Pages to Extend While Extending File** setting tells DBMaker how many pages or frames to add to the full file in the event that it is full. Select a greater number, to reduce the frequency of file appendages, if the database administrator expects that the database will grow very quickly. One page can be 4 KB, 8 KB, 16 KB, or 32 KB.

➲ Changing 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 Log System Settings

Selecting the **Log System** tab in the **Run Time Settings** window allows the user to set the log correlative setting. The options on this page apply only to the server.



Database name : DBSAMPLE5
☑ write to dmconfig.ini

Server log level: Do not log at all
Error log level: Core dumped or DB crash error
Log slow operations exceeding threshold:
Log SYSINFO: Log basic information: execution time, rc, co...
Max log file size (10-1500 MB):
Max number of log files (2-255):
Log SQL statements: Do not log SQL statements
Log execution plan: Do not log execution plan
Log value of input parameter: Do not log value of input parameter
Log lock information: Do not log extra lock information when error ...
Server log directory:
□ Log ZIP
Log Keep Days:

SERVER LOG LEVEL

Users specify when and what operations are logged according to set the Server log level. DBMaker provides seven log levels as follows:

- **Do not log at all**: disables the log system
- **Log error detail in DB_LGERR**: logs errors and the default errors according to the logging level specified in **Error log level**
- **Log slow operation, detail in DB_LGSTM**: logs operations that exceed the threshold specified by **Log slow stmt over the number of seconds**
- **Log error and slow operation**: logs both errors (see **Log error detail in DB_LGERR** above) and slow operations (see **Log slow operation, detail in DB_LGSTM** above)
- **Log SQL, detail in DB_LGSQL**: logs connects, disconnects, commits, rollbacks, SQL command executions, errors, and slow operations
- **Log all operation when exit**: logs all operations during the exit process

- **Log all operation when exit/enter**: logs all operations during the enter process and the exit process.

ERROR LOG LEVEL

Users can specify the level of detail that will be recorded when server logging is enabled as specified by the **Server log level** setting including log error. The **Error log level** indicates what kind of errors will be logged according to the five levels described here:

- **Core dumped or DB crash error:** logs all errors that occur having error codes > 30000
- **Disconnect error or DB crash error:** logs all errors that occur having error codes > 20000
- **Abort, disconnect or DB crash error:** logs all errors that occur having error codes > 10000
- **Normal, abort, disconnect or DB crash error:** logs all errors that occur having error codes > 100
- **Warning or any error:** logs all errors (i.e., all errors that occur having error codes > 0)

This setting corresponds to the **DB_LGERR** keyword in the **dmconfig.ini** file. The default level is **Normal, abort, disconnect or DB crash error**.

LOG SLOW STMT OVER THE NUMBER OF SECONDS

Users can specify the threshold value in seconds to record statements that execute too slowly (i.e., execution time exceeds the specified threshold) when server logging is enabled as specified by the **Server log level** setting including log slow statement. Users can enter a value into the field or adjust the field's value using the up and down arrows located to the right of the field.

LOG SYSINFO

Users can specify what system information will be logged using **Log SYSINFO** when server logging is enabled. There are three SYSINFO logging settings for recording different levels of information detail:

- **Log basic information such as execution time, rc and connection id.**
- **Log all the information as the above setting (DB_LGSYS = 0) and SYSUSER and SYSINFO information**
- **Log all the information as the above two settings (DB_LGSYS = 1) and SYSTEM memory information if detectable**

MAXIMUM LOG FILE SIZE

This setting specifies, in megabytes, the maximum file size for log and text files. DBMaker stores the log as DBNAME_1.LOG. When the log file size reaches the specified value, subsequent logs are generated as follows: DBSAMPLE5_2.LOG, DBSAMPLE5_3.LOG, …, DBSAMPLE5_n.LOG, where n is the value specified by DB_LGFNO or when DB_LGFNO is unspecified the default value of 20 is used. Log files are over written starting from DBNAME_1.LOG after the final log file reaches the maximum file size. Users can enter a value into the field or adjust the value using the up and down arrows located to the right of the field.

MAXIMUM NUMBER OF LOG FILES

This setting specifies the number of the log files that are generated before the log rolls-over (i.e., the first log file is overwritten with new log information). DBMaker stores the log as DBNAME_1.LOG. When the log file size reaches the default value of 100 MB or the size specified by DB_LGFSZ, the log information is logged to a subsequent log file as follows: DBSAMPLE4_2.LOG, DBSAMPLE4_3.LOG, …, DBSAMPLE4_n.LOG where n is the number of log files. The log information will be written to next log file until either the default of 20 log files have been generated or the number of log files generated is equal to the value specified by this setting. At this point, the log rolls-over. In other words, log files are erased and recreated with the new log information beginning with log file

DBNAME_1.LOG. Users can enter a value into the field or adjust the existing value using the up and down arrows located to the right of the field.

LOG SQL STATEMENT

Users can specify what system information will be logged using **Log SQL** when server logging is enabled for **Log SQL, detail in DB_LGSQL (DB_LGSVR = 4)**. There are three SQL logging settings for recording different levels of information detail:

- **Do not log SQL statement**
- **Log any non-select SQL statement**
- **Log all SQL statement**

LOG EXECUTION PLAN

This setting specifies whether to log the execution plan for select, update, and delete statements. There are two logging options:

- **Do not log execution plan**
- **Log execution plan**

LOG INPUT PARAMETER’S VALUE

User can specify whether to log the input parameter’s value. DBMaker provides five logging options. Each offers a different amount of information detail:

- **Do not log input parameter’s value**
- **Log input parameter’s value**
- **Log input parameter’s value and stored procedure’s executed SQL command**
- **Log trigger's SQL statement and parameter**
- **Log both stored procedure and trigger's SQL statement and parameter**

LOG LOCK INFORMATION

User can specify whether to log lock timeout and deadlock events when the log server is enabled for error logging. DBMaker provides two logging options each offering a different amount of information detail:

- **Do not log extra lock information when error code is lock time out or deadlock**
- **Log extra lock time out information**

SERVER LOG DIRECTORY

The directory path of the log files can be specified according to a user’s needs. It may be entered manually or selected by using the browse button (..). Selecting the browse button (..) opens a dialog box.

LOG ZIP

Consider that the number of log files might grow, so packing/zipping the earlier closed log files would be necessary in order to save some storage. DBMaker provides the log zip function to compress the earlier log files.

LOG KEEP DAYS

Server log can include current date in the log filename, so the log filename would be unique and will not be overwritten. User can specify the number of days that the log files will be kept available. The expired log file would be removed by the daemon service.

➲ Changing the Log System Settings:

1. Select a Level for Log Server:
    - Select **Do not log at all** to shut down the Log System function
    - Select **Log errors, see DB_LGERR for details** to log the error informations
    - Select **Log slow operations, see DB_LGSTM for details** to log the operations those runs too slowly

- Select **Log errors and slow operations** to log both the errors and slow operations
- Select **Log SQL, see DB_LGSQL for details** to log the SQL command, connection, disconnetions, commit,rollback, errors and slow operations
- Select **Log all operations on exit** to log all operations when exit from database
- Select **Log all operations on startup/exit** to log all operations when both connection and disconnect the server

2. Select a error log level for logging errors:

   - Select **Core dumped or DB crash error** to log the error which code > 30000
   - Select **Disconnect error or DB crash error** to log the error which code > 20000
   - Select **Abort, disconnect, or DB crash error** to log the error which code > 10000
   - Select **Normal, abort, disconnect, or DB crash error** to log the error which code > 100
   - Select **Warning or other error** to log all errors which code > 0

3. Specify the threshold for the slow operations. The unit is second.
4. Select Log SYSINFO options:

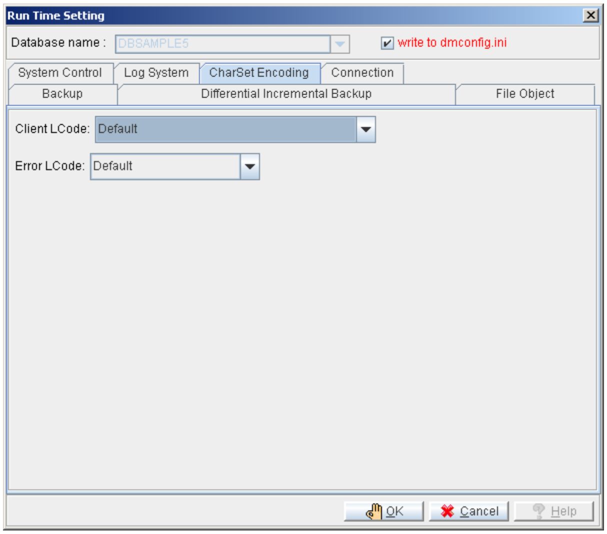
   - Select **Log basic information such as execution time, rc and connection id.** to log the basic system information.
   - select **Log all the information as the above setting (DB_LGSYS = 0) and SYSUSER and SYSINFO information** to log SYSUSER and SYSINFO as well as basic information

- Select **Log all the information as the above two settings (DB_LGSYS = 1) and SYSTEM memory information if detectable** to log both the above and the SYSTEM memory information.

5. Specify the maximum size of the Log file and the unit is MB.
6. Specify the maximum numbers of the Log file.
7. Select the Log SQL action:
    - Select **Do not log SQL statement** to do not log the SQL statement
    - Select **Log any non-select SQL statement** to log the SQL statement except the SELET statement
    - Select **Log all SQL statement** to log all kinds of SQL statement
8. Specify whether to log the execution plan.
9. Specify whether to log the input parameter’s value:
    - Do not want to log the input parameter’s value, select **Do not log input parameter’s value**
    - Select **Log input parameter’s value** to log the input parameter’s value
    - Select **Log input parameter’s value and stored procedure’s executed SQL command** to log both input parameter’s value and the stored procedure’s SQL command.
10. Specify whether to log the lock information when time out or deadlock.
11. Specify the log directory in the Server Log Directory filed or click the borwse button (..) to select a path.
12. Specify whether to zip the earlier log file to save the storage.select the check box to start the zip function.
13. Input how many days the log file can be kept.

Changing Character Set Encoding Settings

The **CharSet Encoding** tab found on the **Run Time Settings** window allows client-side users to set a database’s language code and error message code.



Database name : DBSAMPLE5

☑ write to dmconfig.ini

Client LCode: Default

Error LCode: Default

Figure 25-5 The CharSet Encoding page of the Run Time Setting dialog

CLIENT LCODE

Client-side users can choose a language code for connecting to UTF-8 Database Servers (i.e., the Database Server's *Lcode* is set to **10**) when using multilingual database. The client-side language codes are:

- DEFAULT
- European Language (ASCII)
- European Language (ISO-8859-1)
- European Language (ISO-8859-2)
- European Language (ISO-8859-3)
- European Language (ISO-8859-4)
- European Language (ISO-8859-5)
- European Language (ISO-8859-7)
- European Language (ISO-8859-9)

- European Language (ISO-8859-10)
- European Language (ISO-8859-13)
- European Language (ISO-8859-14)
- European Language (ISO-8859-15)
- European Language (ISO-8859-16)
- European Language (KO18-R)
- European Language (KO18-U)
- European Language (KO18-RU)
- European Language (CP1250)
- European Language (CP1251)
- European Language (CP1252)
- European Language (CP1253)
- European Language (CP1254)
- European Language (CP1257)
- European Language (CP850)
- European Language (CP866)
- European Language (MACROMAN)
- European Language (MACCENTRALEUROPE)
- European Language (MACICELAND)
- European Language (MACCROATIAN)
- European Language (MACROMANIA)
- European Language (MACCYRILLIC)
- European Language (MACUKRAINE)
- European Language (MACGREEK)

- European Language (MACTURKISH)
- European Language (MACINTOSH)
- Semitic Language (ISO-8859-6)
- Semitic Language (ISO-8859-8)
- Semitic Language (CP1255)
- Semitic Language (CP1256)
- Semitic Language (CP862)
- Semitic Language (MACHEBREW)
- Semitic Language (MACARABIC)
- Japanese (SHIFT-JIS)
- Japanese (SHIFT-JP)
- Japanese (CP932)
- Japanese (ISO-2022-JP)
- Japanese (ISO-2022-JP-1)
- Japanese (ISO-2022-JP-2)
- Chinese (GBK)
- Chinese (GB18030)
- Chinese (BIG5)
- Chinese (EUC-CN)
- Chinese (EUC-TW)
- Chinese (CP936)
- Chinese (CP950)
- Korean (EUC-KR)
- Korean (CP949)

- Korean (JOHAB)
- Georgian (GEORGIAN-ACADEMY)
- Georgian (GEORGIAN-PS)
- Tajik (KO18-T)
- Kazakh (PT154)
- Thai (TIS-620)
- Thai (CP874)
- Thai (MACTHAI)
- Laotian (MULELAO-1)
- Laotian (CP1133)
- Vietnamese (VISCII)
- Vietnamese (TCVN)
- Vietnamese (CP1258)
- Unicode (UTF-8)
-

ERROR LCODE

In multilingual database, client side can set their own output locale codes of error message. They can be:

- DEFAULT
- ASCII (English)
- ISO-8859-1 (English code)
- ISO-8859-2 (English code)
- ISO-8859-5 (English code)
- ISO-8859-7 (English code)

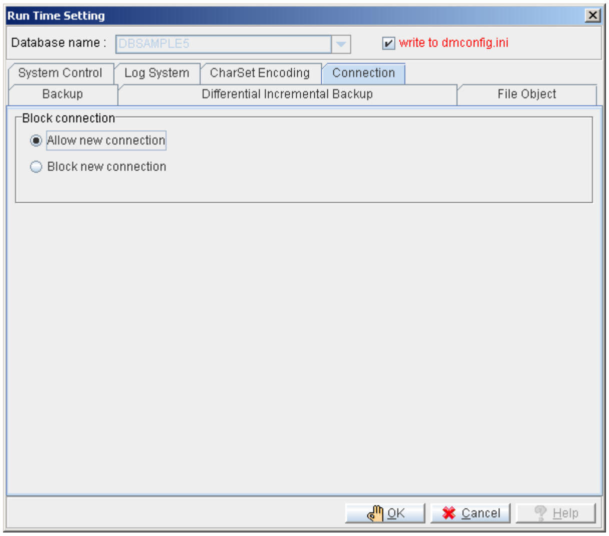
- UTF-8 (English code)
- Shift-JIS (Japanese code)
- EUC-JP (Japanese code)
- UTF-8 (Japanese code)
- GBK (Simplified Chinese)
- GB18030 (Simplified Chinese)
- UTF-8 (Simplified Chinese)
- BIG5 (Traditional Chinese)
- UTF-8 (Traditional Chinese)

➲ Changing the Character Set setting:

1. Select the **Client LCode**.
2. Select the **Error LCode** for the client side.

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.



Run Time Setting

Database name : DBSAMPLE5 ☑ write to dmconfig.ini

Block connection

● Allow new connection

○ Block new connection

Figure 25-6 The Connection page of the Run Time Setting dialog

➲ Changing 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**.

25.2 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.

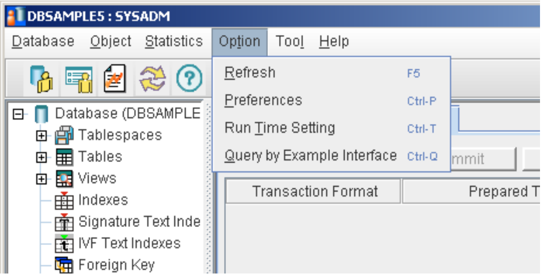
Refresh rate: Determines the rate at which database monitor information is updated to the screen.

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.

➲ Accessing the Preferences window:

1. Select **Preferences** from the **Option** menu on the menu bar.

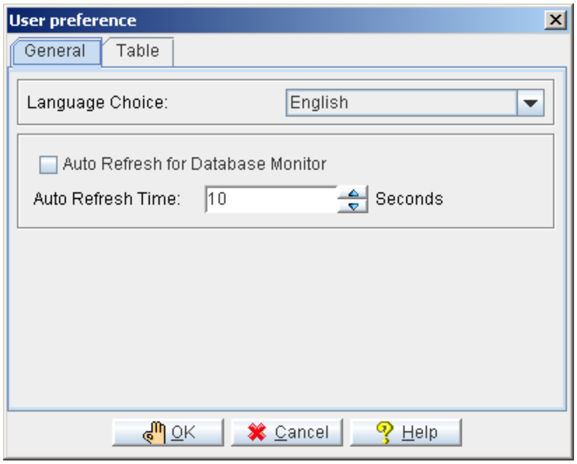


2. The **User Preference** window will appear with the **General** page displayed.

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.



User preference

General | Table

Language Choice: English

□ Auto Refresh for Database Monitor

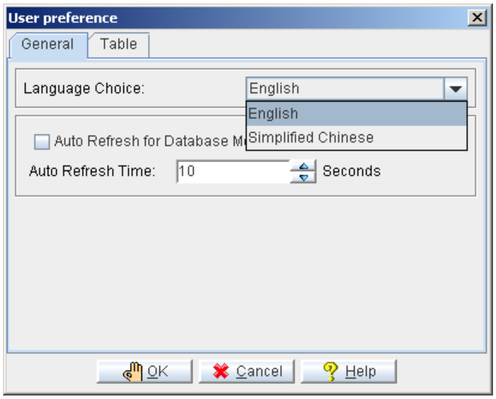
Auto Refresh Time: 10 Seconds

Figure 25-7 The General page of the User Preference Window

➲ Changing 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.*



Language Choice: English

English
Simplified Chinese

□ Auto Refresh for Database M

Auto Refresh Time: 10

➲ 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.

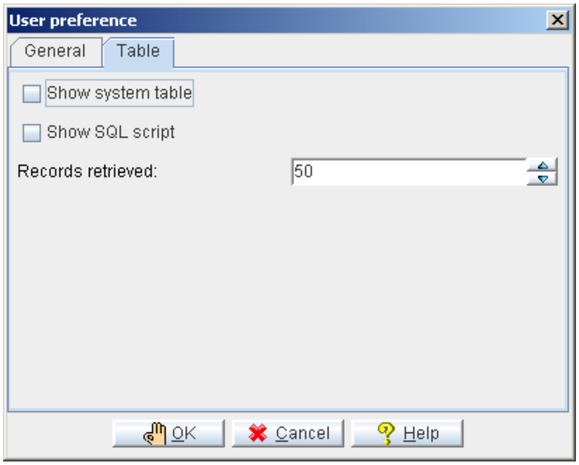
NOTE *Setting a low **Auto Refresh Time** will use more processor resources and can substantially slow down the database.*

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 commands** at the bottom of the **Alter SQL Command** window.



□ Show system table

□ Show SQL script

Records retrieved: 50

Figure 25-8 The Table page of the User Preference Window.

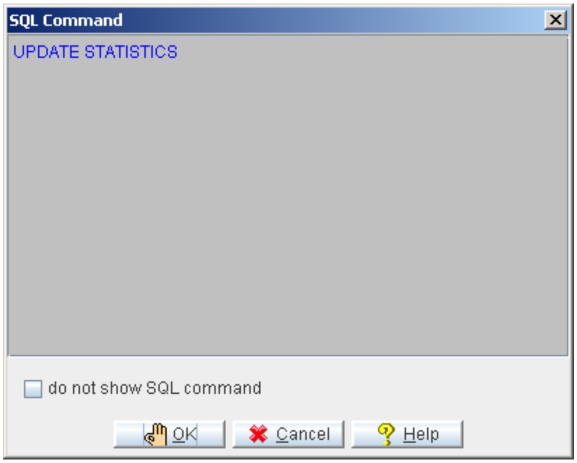
➲ Viewing system tables in the tree view:

1. Select the check box next to **Show system table**. A check mark in the box indicates that system tables are shown.

➲ Viewing 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 shown the following figure.



SQL Command

UPDATE STATISTICS

□ do not show SQL command

➲ Altering 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. Clicking the **OK** button at the end of the procedure displays the **Alter SQL Command** window. You can edit the SQL command in the text field as needed.
4. Clicking the **OK** button executes the command. Clicking the Cancel button aborts the 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).