

DBMaker

JServer Manager User's Guide

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

©Copyright 1995-2010 by CASEMaker Inc.
Document No.645049-234056/DBM52-M10302010-JSEM

Publication Date: 2010-10-30

All rights reserved. No part of this manual may be reproduced, stored in a retrieval system, or transmitted in any form, without the prior written permission of the manufacturer.

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 for 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	1-1
1.1	Additional Resources	1-2
1.2	Technical Support	1-3
1.3	Document Conventions	1-4
2	JServer Manager Basics	2-1
2.1	Summary of Features	2-2
2.2	The JServer Manager Workspace	2-4
	Main Console Structure	2-4
	Menu Bar	2-5
2.3	Select File / Select Path / Save Dialog Box.....	2-8
3	Creating a Database	3-1
3.1	Creating a Database: Basics	3-3
3.2	Creating a Database: Advanced Settings	3-7
	Setting Create Database Options	3-9
	Setting Storage Options	3-11
	Setting Backup Options	3-18
	Setting dmServer Options.....	3-24
3.3	Using the Create Database Wizard.....	3-27

4	Starting a Database	4-1
4.1	Starting a Database: Basics	4-2
4.2	Starting a Database: Advanced Settings.....	4-4
	Setting Cache and Control Options.....	4-6
	Setting Backup Options.....	4-10
	Setting Replication Options	4-16
	Setting Start Database Options.....	4-20
	Setting User File Options	4-24
	Setting DmServer Options	4-26
	Setting SQL Attributes	4-29
	Setting Distributed Database Environment Options	4-31
	Setting Group Commit Options	4-33
4.3	Using the Start Database Wizard	4-36
	Normal or Read Only Mode Startup	4-39
	Primary DB Mode Startup	4-50
	Slave DB Mode Startup	4-53
	Trouble Shooting Startup	4-55
5	Shutting Down a Database	5-1
6	Using an NT Service.....	6-1
6.1	Adding an NT Service.....	6-2
6.2	Viewing Installed NT Services	6-4
6.3	Removing NT Services	6-5
7	Altering Run Time Settings	7-1
7.1	Changing Backup Settings	7-4
	Start Backup Server.....	7-4
	Compress Backup files.....	7-5
	Backup Read only tablespace.....	7-5
	Directory of backup files.....	7-5
	Check Database	7-5
	Incremental Backup Mode	7-6
	File Object Backup Mode	7-7

	Settings for full backup process.....	7-7
7.2	Changing Differential and incremental Backup Settings	7-10
	Settings for differential backup process.....	7-10
	Settings for incremental backup process	7-11
7.3	Changing File Object Settings	7-14
7.4	Changing System Control Settings.....	7-16
	Journal File Synchronization	7-16
	Display Mode of SQL Command Monitor	7-17
	Enable Distributed Database Mode	7-17
	Pages to Extend when Extending a File.....	7-17
7.5	Changing Log System Settings	7-19
	Server Log Level	7-19
	Error Log Level	7-20
	Logging Slow Statements	7-21
	Log SYSINFO	7-21
	Maximum Log File Size	7-22
	Maximum Number of Log Files	7-22
	Log SQL Statements	7-23
	Log Execution Plan.....	7-23
	Log Input Parameter	7-23
	Log Lock Information.....	7-24
	Server Log Directory	7-24
	Log ZIP.....	7-25
	Log Keep Days	7-25
7.6	Changing Character Set Encoding	7-28
	Client LCode.....	7-28
	Error LCode	7-31

8	Renaming a Database	8-1
9	Diagnosing a Database	9-1
10	Deleting a Database	10-1
11	Backing up a Database	11-1
11.1	On-line Full Backup by Backup Server.....	11-3
11.2	On-line Differential Backup by Backup Server	11-6
11.3	On-line Incremental Backup by Backup Server	11-9
11.4	On-line Full Backup Interactively	11-12
11.5	On-line Full Backup Interactively to Tape.....	11-17
11.6	On-line Incremental Backup Interactively	11-21
11.7	On-line Incremental Backup Interactively to Current	11-24
11.8	Off-line Full Backup.....	11-27
11.9	Off-line Full Backup to Tape	11-30
12	Managing Log Files	12-1
13	Restoring a Database.....	13-1
13.1	Restoring a Database from Disk.....	13-2
13.2	Restoring a Database from Tape	13-9
	Glossary.....	Glossary- 1

1 Introduction

Welcome to JServer Manager User's guide. JServer manager is a cross-platform graphical user interface (GUI) for creating and managing databases using DBMaker—an SQL Database Management System.

This manual gives systematic instructions on how to use JServer Manager for the creation and management of multiple databases. Program features are explained using procedures supported by screen images containing data from the sample databases provided with the software.

This book is intended for DBMaker administrators who will access and manage their databases from a variety of platforms and hardware configurations. Administrators using this manual may or may not be familiar with all of the features of DBMaker. It is assumed that the user does have some knowledge of working with the platform on which DBMaker is installed.

The JServer Manager User's Guide is organized according to the procedures used to create, start, stop, delete, and restore databases using the JServer Manager applet. The manual follows the structure of the interface in easy to understand steps.

1.1 Additional Resources

DBMaker provides many other user's guides and reference manuals in addition to this guide. You should consult one of the books shown below for more information on a particular subject.

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

For more information on the tools and utilities provided with DBMaker, refer to the *dmSQL User's Guide*, the *JDBA Tool User's Guide*, or the *Server Manager User's Guide*.

For more information on other Java tools provided with DBMaker, refer to the *JDBA Tool User's Guide* or the *JConfiguration Tool Reference*.

For more information on the SQL language implemented by DBMaker, refer to the *SQL Command and Function Reference*.

For more information on the ESQL/C language implemented by DBMaker, refer to the *ESQL/C Programmer's Reference*. For more information on error and warning messages, refer to the *Error and Message Reference*.

1.2 Technical Support

CASEMaker provides thirty days of complimentary email and phone support during the evaluation period. When software is registered, an additional thirty days of support will be included, thus extending the total support period for software to sixty days. However, CASEMaker will continue to provide email support for any bugs reported after the complimentary support or registered support has expired (free of 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 Command Line 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 be replaced by the actual name. Italics also are used to introduce new terms, and are occasionally used for emphasis.
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.
Command Line	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

Figure 1-1 Document Conventions Table

2 JServer Manager Basics

The following sections briefly describe some of the features of JServer Manager that are unique to the Java applications. Please take a moment to review this chapter before reading the following chapters of this manual. Section 2.2 describes the JServer Manager workspace and the different elements of the user interface: the menu bar and the main console. Section 2.3 explains how to use the Save / select file / select path dialog boxes. These three dialog boxes operate in a similar manner and appear any time a browse button is selected or a directory path needs to be specified.

2.1 Summary of Features

JServer Manager has a wide variety of features that help you effectively manage your database.

Creating Databases

- Create single-user or multi-user databases
- Link to the JConfiguration Tool for managing the **dmconfig.ini** file

Starting and Shutting Down a Database

- Start and shut down databases on local computers
- Link to JConfiguration Tool for managing database settings after startup
- Link to Server Manager for database connection management

Deleting Databases

- Delete a database
- Delete file objects within the database

Backing up Databases

- Perform full backups while the database is on-line or off-line
- Perform full backups to disk or to tape
- Perform differential backups
- Perform incremental backups

Restoring Databases

- Restore crashed databases from disk or from tape

Integrated User Interface

- Linked to JDBC Tool and JConfiguration Tool

2.2 The JServer Manager Workspace

The menu bar and the main console comprise JServer Manager's two primary workspace areas. Most common database server operations are on the main console. The following diagram illustrates the JServer Manager Workspace.



Figure 2-1 JServer Manager Workspace

Main Console Structure

The main console contains seven icons representing the main functions of JServer Manager. Clicking one of the icons opens a window that lets the user execute the desired function. The following chapter headings correspond to each of these functions. The chapters use procedures and examples to illustrate using each of these functions, and give troubleshooting tips and helpful information.

Menu Bar

The menu bar in JServer Manager houses all the functions that are contained in the main console. The Menu bar contains pull down menus: Database, Option, Wizards, Tool, and Help.



Figure 2-2 JServer Manager Tool Bar

The Database menu gives access to the same functions available from the JServer Manager main console. Please see Main Console Structure for additional information.

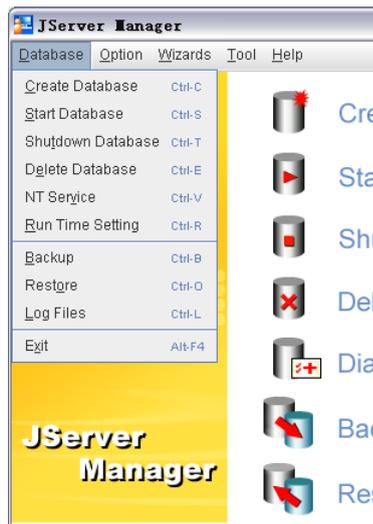


Figure 2-3 Database Menu

The Option menu lets users change languages. JServer Manager supports English, Traditional Chinese, and Japanese.



Figure 2-4 Option Menu

The Wizards menu provides access to wizards that help guide the user step-by-step through database creation, deletion, as well as renaming and database diagnostics.

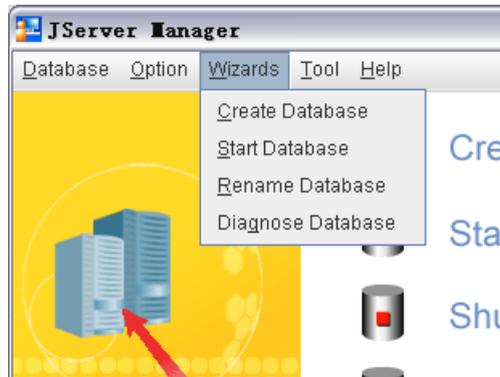


Figure 2-5 Wizards Menu

The Tool menu provides quick access to the JDBC Tool and JConfiguration Tool. Please see the *JDBC Tool User's Guide* and the *JConfiguration Tool Reference* for information regarding the use of these programs.

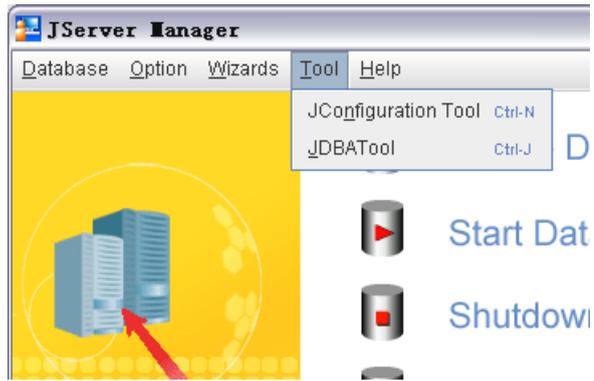
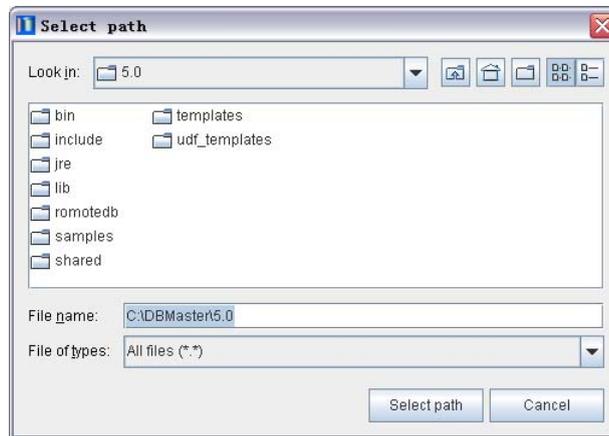


Figure 2-6 Tool Menu

The Help menu allows access to the help files.

2.3 Select File / Select Path / Save Dialog Box

Many settings require the entry of a file name or a directory path. These may be entered manually or selected by using the browse button . Selecting the browse button opens the following dialog box.



The figure above shows the **Select Path** dialog box. The **Select File** and **Save** dialog boxes differ in function, but are structurally the same. Directly underneath the title bar are the **Look in** menu and five buttons:



Up One Level button: makes the current directory one level closer to the root directory.



Home button: makes the active directory the user's home directory. In Windows 98, this is the My Documents folder. In Windows NT or Windows 2000, this is the user directory.



Create New Directory button: Creates a new folder, named **New Folder**, and located in the active directory.



The *List* and *Details* buttons are inactive in JServer Manager.

The main window is located directly below the menu and buttons. The main window displays the files and subdirectories of the current directory. A user can browse up and down the directory tree by double-clicking on a directory name (down one level), or clicking the *Up One Level* button. The user can select an individual file or directory from the main window box by left-clicking it. A user can also edit any directory or file name from the main window. To edit a file name or directory name, right-click the file or directory and make changes to the name. Pressing enter saves the new directory or file name. Be sure that the name of the selected file or directory appears in the *file name* text box below.

The **File name** and **Files of type** text boxes are displayed below the main window. The file name text box should display the name of the file the user wants to select. The user can then select or cancel the action by clicking the appropriate button to the right

3 Creating a Database

Databases can be run in a single-user or client-server mode in Windows. It is necessary to provide a port number and server address for databases that are to be started in client-server mode. A multi-user database may be started as a single-user database and vice versa. This option is provided during database creation to ensure that the port number and server address are specified if it is created as a multi-user database. Only one user may connect to a single user database.

For users that are not familiar with the settings involved in creating a database, the **Create Database Wizard** is provided. Please refer to Section 3.4 for instructions on using the wizard.

Before creating a database, you should consider the following guidelines for creating database names:

- Database names can be up to 32 characters long
- Database names can contain any alphanumeric characters, Chinese double-byte characters, and the underscore character. These characters can appear in any position of the filename.
- Database names are not case-sensitive
- Database names must be unique among all computers that will connect to the database. Since DBMaker stores these names and other configuration details for local and remote databases in the **dmconfig.ini** configuration file, using the same name for two databases will cause a conflict. If the same database

section appears twice in **dmconfig.ini**, DBMaker will use settings from the database section that appears first.

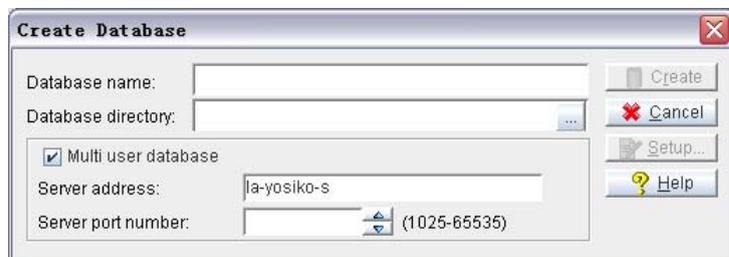
3.1 Creating a Database: Basics

The following procedure outlines how to create a database without using the **Create Database Wizard**. If you are new to DBMaker, the **Create Database Wizard** can help guide you through and explain the advanced database settings that may be necessary for your database. Please refer to section 3.3 for more information about the **Create Database Wizard**.

- ➔ Creating a database without using the **Create Database Wizard**:
 1. Select **Create Database** from the main console or the **Database** menu. The following dialog box appears:

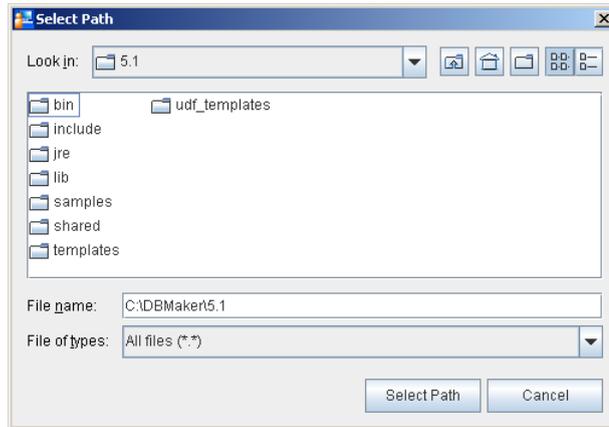


2. To use the wizard to create the database, select **Create database by wizard** and then refer to Section 3.3. Select **Create database** to avoid using the wizard. The following dialog box appears.

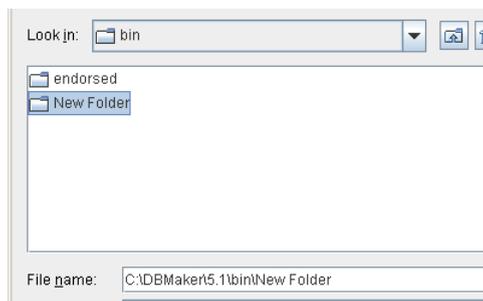


3. Enter a name for the new database in the **Database Name** field.

4. Enter a database directory into the **Database Directory** field, or select the browse button . If the browse button is selected, the following dialog box appears:



5. The default database directory path is *DBMaker\5.1\bin*. A new directory can be created for the database. To create a new database directory, first use the *Up One Level* button  and/or the **Look in** menu to select the root for the database directory. Then click on the **Create New Directory**  button to make a directory called **New Folder**. The new folder will appear as follows and appears in the **File Name** text field.



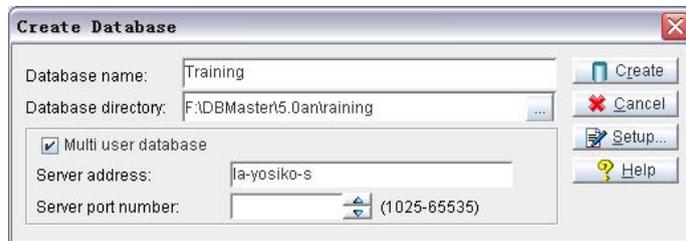
6. To change the directory name, type over the original name. Note that these changes are made directly to the operating system (i.e., use caution when changing the names of existing directories). After typing the new directory name it appears as follows:



7. Press the enter key to complete the creation of the new directory. The list will reappear in alphabetical order. Note that new name will appear in the **File name** field.



8. Click the **Select Path** button, the **Create Database** dialog box reappears.



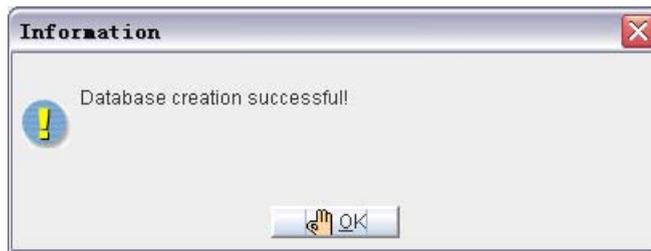
9. To start the database as a multi-user database, enable the **Multi User Database** check box. To start the database as a single user database, disable the **Multi User Database** check box.
10. Enter the IP address or host name of the server in the **Server Address** field. This is only required for multi-user databases.

NOTE *If a host name is specified, ensure that a Domain Name Server is set up on the server.*

11. Enter the server port number in the **Server Port Number** field. This must be an integer between 1025 and 65535. This is only required for multi-user databases.



12. Now you may choose to create the database or alter advanced database settings for the new database. To finish creating the new database, click the **Create** button. To change settings in the **Create Database Advanced Settings** window, select **Setup**. If no errors are encountered upon selecting **Create**, DBMaker displays the following dialog box.



NOTE Please refer to *Creating a Database: Advanced Settings* for information on settings that apply to database creation.

13. Click the **OK** button.

3.2 Creating a Database: Advanced Settings

You can alter the following advanced settings before creating a database:

- Create Database – settings that may only be applied before database creation
- Storage – location and initial size of data, journals, and temporary files
- Backup – frequency and destination of backup files
- dmServer – network address, port number, idle time-out value, and encryption
- Cache and Control – Upper memory caching of data pages and system control, SQL command upper memory caching, CHAR data output padding, system file object mapping, cursor behavior, and browse mode
- DDB Environment – distributed database behavior
- SQL Attributes – date and time formatting for SQL statements and stored procedure settings

The tabbed pages available in this window correspond exactly with the settings in **JConfiguration Tool**. To learn more about how these settings affect the creation and subsequent management of databases, please refer to the *JConfiguration Tool Reference*. Database administrators should be aware of database requirements before creating the database. The language code, file name of code order, case sensitivity, and BLOB frame size are important functions of a database that can only be set before the database is created. Cache and control, DDB environment, and SQL attributes are discussed in *Starting a Database: Advanced Settings*.

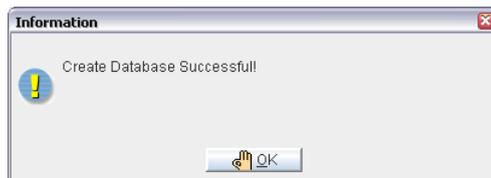
- ➔ Creating a database with advanced settings:
 1. Enter the database name, directory, IP address and Port Number, if applicable, as outlined in Section 3.1



2. Click the **Setup** button. The **Create Database Advanced Settings** dialog box appears.
3. Enter advanced setting options. See the following procedures for details.

NOTE *The following list should be linked to the appropriate headers of the underlying topics.*

- Setting Create Database Options
 - Setting Storage Options
 - Setting Backup Options
 - Setting dmServer Options
 - Setting Cache and Control Options (Please refer to Chapter 4, *Setting Cache and Control Options*)
 - Setting DDB Options (Please refer to Chapter 4, *Setting Distributed Database Environment Options*)
 - Setting SQL Attributes (Please refer to Chapter 4, *Setting SQL Attributes*)
4. Now the user may choose to create the database or alter advanced database settings for the new database. To finish creating the new database, click the **Create** button.



5. Click the **OK** button.

Setting Create Database Options

You can set the following Create Database settings. The Create Database page corresponds exactly to the Create Database page in the **JConfiguration Tool**. Please refer to Chapter 14 of the *JConfiguration Tool Reference* for more information.

NOTE *The settings on the Create Database page can only be set before creation of the database.*

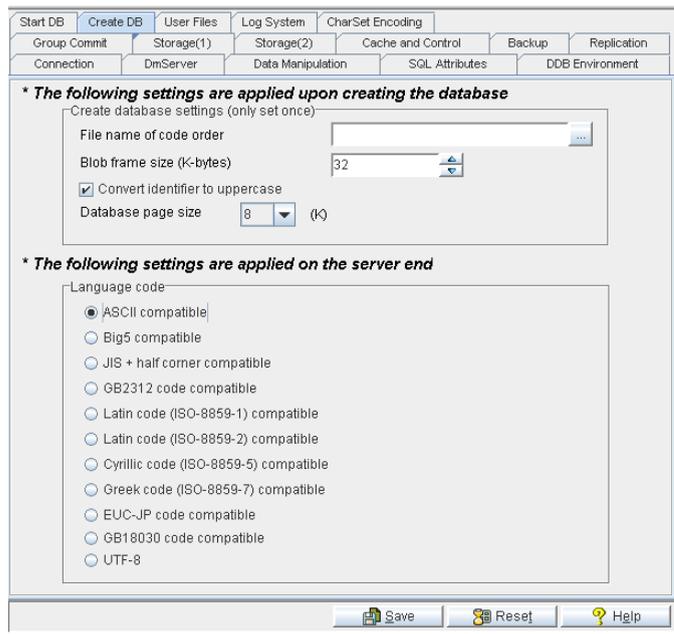


Figure 3-1 Create page of the Advanced Database Settings window

FILE NAME OF CODE ORDER

The database administrator can specify a location for the order definition file only before creating a database. The order definition file must be defined before creating a database and cannot be modified after creating a database. DBMaker uses the

order definition file to assign characters a value for comparison and sorting. The default order definition is single-byte characters followed by double-byte characters. The order definition file is used for sorting indexes alphabetically as well as for comparison operations that use character data. See the *Database Administrators Guide* or the *JConfiguration Tool Reference* for instructions on how to create an order definition file.

BLOB FRAME SIZE

The administrator can set the frame size for BLOB type data only before the creation of the database. A larger BLOB frame size means that more disk space is used each time a BLOB type data column is added, however, frames need to be added less frequently when BLOB type data is appended. If processor resources will be in demand, the BLOB frame size can be set to a larger size. If disk space is an issue and processor resources are not, the BLOB frame size can be decreased.

CONVERT IDENTIFIER TO UPPERCASE

Before a database is created, the database administrator selects whether database object identifiers will be case sensitive or not. Identifiers do not include passwords, user-defined function names, and database names. The case sensitivity selection cannot be altered for the database once the database is created.

DATABASE PAGE SIZE

A data page is the smallest unit of storage used by data files. In DBMaker 5.0, users can specify the page size. The page size can be 4 KB, 8 KB, 16 KB, or 32 KB. This setting corresponds to the **DB_PGSIK** keyword in the **dmconfig.ini** file. The default value is 8 KB.

LANGUAGE CODE

The database administrator can decide which character set the database will use for CHAR type data.

Setting the Create Database options:

1. Click the **Setup** button in the **Create Database** window.
2. Select the **Create Database** tab in the **Create Database Advanced Settings** window.

3. Enter a path or select the browse button  next to the **File Name of Code Order** field to indicate the location of the code order file.
4. Enter a BLOB frame size in bytes in the **BLOB Frame Size** field.
5. To make all database objects case insensitive select the **Convert Identifier to Uppercase** option.
6. Select the number of pages in the **Database Page Size** field.
7. To change the character set the database will use for CHAR type data, select one of the radio buttons:
 - **ASCII compatible** for ASCII code
 - **Big5 compatible** for traditional Chinese code
 - **JIS + half corner compatible** for Japanese code
 - **GB code compatible** for simplified Chinese code
 - **Latin code (ISO-8859-1) compatible** for Latin – 1 code
 - **Latin code (ISO-8859-2) compatible** for Latin – 2 code
 - **Cyrillic code (ISO-8859-5) compatible** for Cyrillic code
 - **Greek code (ISO-8859-7) compatible** for Greek code
 - **Japanese code (EUC-JP) compatible** for Japanese code
 - **Simplified Chinese (GB18030) compatible** for Simplified Chinese code
 - **UTF-8 (UTF-8)**
8. Click the **Save** button, or click the **Cancel** button to return to the **Create Database** dialog box.

Setting Storage Options

The database administrator can specify the storage location for different operating system files that make up the database. Storage options are set using two different pages in the Create database Advanced Settings window: Storage (1) and Storage (2). The Storage page exactly corresponds to the Storage page in **JConfiguration**

Tool. Please refer to the *JConfiguration Tool Reference* for more information on database file storage.

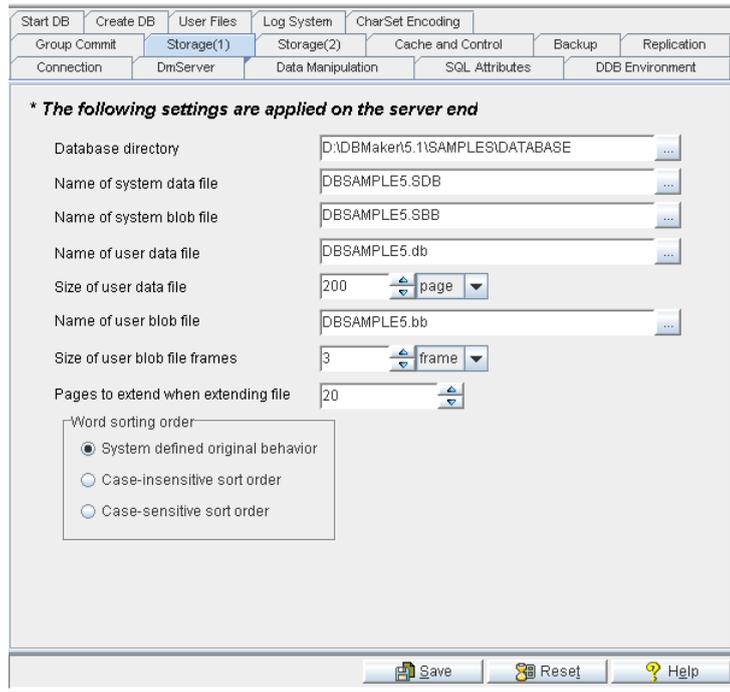


Figure 3-2 Storage (1) page of the Advanced Database Settings window

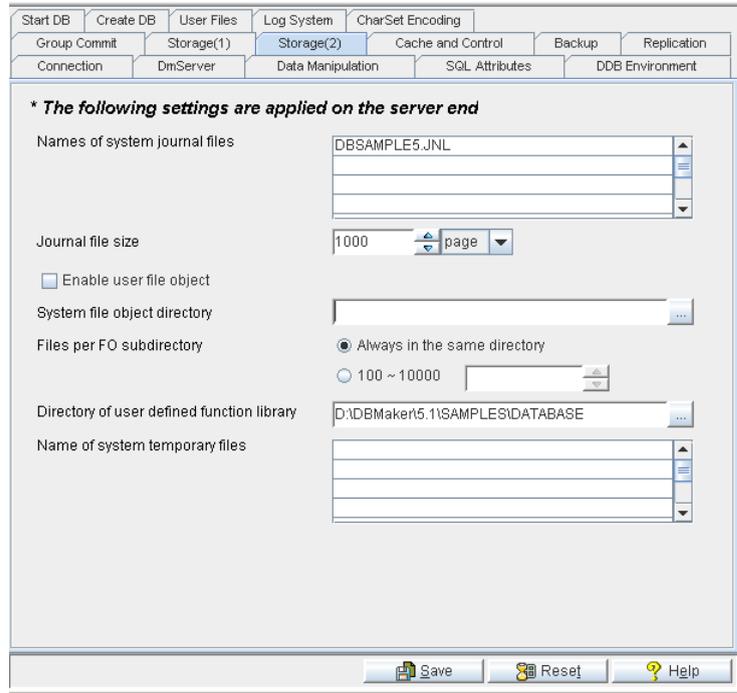


Figure 3-3 Storage (2) page of the Advanced Database Settings window

DATABASE DIRECTORY

The database administrator should have already set the database directory by following the procedure outlined in *Section 3.1*, however, a different location may be selected for the default Database Directory at any time before database startup. The Database Directory is the default location where all operating system files that relate to the database, excluding the **dmconfig.ini** file, are stored unless specified otherwise by the database administrator.

SYSTEM DATA FILE

The System Data File is an operating system file that contains database objects such as indexes, views, stored procedures, and synonyms. Objects smaller than one page in length are stored in this file. The System Data file default path is (*Database*

Directory)\(*database name*).SDB, but can be changed by the database administrator.

SYSTEM BLOB FILE

The System BLOB File is an operating system file that contains database objects such as indexes, views, stored procedures, and synonyms. Objects larger than one page in length are stored in this file. The System BLOB file default path is (*Database Directory*)\(*database name*).SBB, but can be changed by the database administrator.

USER DATA FILE

The User Data File is the operating system's file location for storing data. It contains all tables and data type records that belong to the database. The User Data file default path is (*Database Directory*)\(*database name*).DB, but can be changed by the database administrator.

SIZE OF USER DATA FILE

The database administrator may specify the initial number of file pages allocated to the User Data File. This determines the initial operating system file size of the User Data File.

USER BLOB FILE

The user BLOB File is the operating system's file location for storing BLOB type data. It contains all BLOB type records that belong to the database. The User BLOB file default path is (*Database Directory*)\(*database name*).BB, but can be changed by the database administrator.

SIZE OF USER BLOB FILE FRAMES

The database administrator may specify the initial number of BLOB frames allocated to the User BLOB File. This determines the initial operating system file size of the User BLOB File. The database administrator sets the BLOB frame size upon creation of the database. Please refer to Section 3.2, *Create Database: BLOB Frame Size* for information about altering BLOB Frame Sizes.

PAGES TO EXTEND WHEN EXTENDING FILE

When all pages in the 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 file in the event that it becomes full. If the database administrator expects that the database will grow very quickly, then a higher number should be selected to reduce the frequency of file appending. The default value is **20** pages.

WORD SORTING ORDER

Query responses can be sorted in various ways according to a set of rules. The **Word sorting order** setting tells DBMaker how to sort and present data in response to database queries. Selecting **System defined original behavior** enables the default sorting order. Selecting **Case-insensitive sort order** is necessary for obtaining query results when case-sensitivity is a consideration. This setting corresponds to the **DB_ WSORT** keyword in the **dmconfig.ini** file. The default value is *System defined original behavior*.

For more in-depth information on code and word sorting order, please refer to Chapter 4.2 in the *Database Administrator's Guide*.

JOURNAL FILES

The database administrator may specify one or more journal files for the database. If multiple journal files are created, DBMaker stores journal file entries starting with the first page of the first journal file listed, and then sequentially fills each journal file until the last file is full. When the last journal file is full, DBMaker begins recording data by writing over the old data starting with the first page of the first journal file, and so on. The size of each journal file can also be specified in the Journal File Size field.

FILE OBJECTS

The database administrator can determine if User File Objects can be used or not. User File Objects are file objects that are stored outside of DBMaker's file system. The location of System File Objects can also be specified. These are file objects stored within DBMaker's file system. DBMaker can automatically create

subdirectories within the FO directory. Each subdirectory is filled to a threshold value with new file objects. When the threshold is reached, DBMaker creates a new FO subdirectory. The value in the **Files per FO subdirectory** field indicates the threshold value. To disable subdirectories and store all file objects in the FO directory select **Always in the same directory**.

USER DEFINED FUNCTION LIBRARY

User defined functions are compiled functions stored in a dynamic link library (DLL). These user defined functions are available to users wanting to use them in DBMaker. The DLLs stored in the Directory of User Defined Function Library are accessible to DBMaker and can be used in SQL Statements or ODBC applications. The default path for the Directory of User Defined Function Library is **(DBMaker working directory)/shared/udf/**. Users can also enter a new path if a different one is desired. This can be achieved by typing the new path into the text field, or by clicking on the browse button  next to the text field. This setting corresponds to the **DB_LbDir** keyword in the **dmconfig.ini** file.

NAMES OF SYSTEM TEMPORARY FILES

System temporary files are used by DBMaker to store information about the database while the database is active. Up to eight temporary files may be specified. These files are removed when the database is shut down. If one or more filenames are written here, DBMaker uses them to store temporary information about the database. The default path for the system temporary file is *(Database Directory)*. Users can also enter a new path if a different one is desired. This can be achieved by typing the new path into the text field, or by clicking on the browse button  next to the text field.

Setting storage options:

1. Click the **Setup** button in the **Create Database** window.
2. Click the **Storage (1)** tab from the **Create Database Advanced Settings** window.
3. Enter a path into or select the browse button  next to the **Database Directory** field to indicate the location of the Database Directory

4. Enter a path into or select the browse button  next to the **Name of System Data File** field to indicate the location of the System Data File.
5. Enter a path into or select the browse button  next to the **Name of System BLOB File** field to indicate the location of the System BLOB File.
6. Enter a path into or select the browse button  next to the **Name of User Data File** field to indicate the location of the User Data File option.
7. Enter the number of pages to allocate from disk for each user data file in the **Size of User Data File** field.
8. Enter a path into or select the browse button  next to the **Name of User BLOB File** field to indicate the location of the User BLOB File.
9. Enter the number of frames to allocate from disk for each user BLOB file in the **Size of User BLOB File Frames** field.
10. Enter the number of pages or frames to append each user file with when the file is full in the **Pages to Extend when Extending File** field.
11. Select the **word sorting order** option button.
12. Click the **Storage (2)** tab from the **Create Database Advanced Settings** window.
13. Enter a full path or file name into the **Journal Files** field. More than one name may be entered. Typing just a file name places the journal files into the database directory.
14. Enter the number of pages to allocate from disk for each journal file in the **Journal File Size (pages)** field.
15. To enable the database to use external file objects, click on the **Enable User File Object** check box.
16. 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.
17. 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.
18. Enter a path into or select the browse button  next to the **Directory of User Defined Function Library** field to indicate the location of the user defined function dynamic link library.
 19. Enter up to eight full path and file names into the **Name of System Temporary File** field to indicate the location of System Temporary Files.
 20. Click the **Save** button.
 21. Click the **Cancel** button to return to the **Create Database** window.

Setting Backup Options

The database administrator can specify how DBMaker will backup data. Backups are s critical to ensure maximum data security. The Backup page exactly corresponds to the Backup page in the JConfiguration Tool. Please refer to the *JConfiguration Tool Reference* for more information on backup operations.

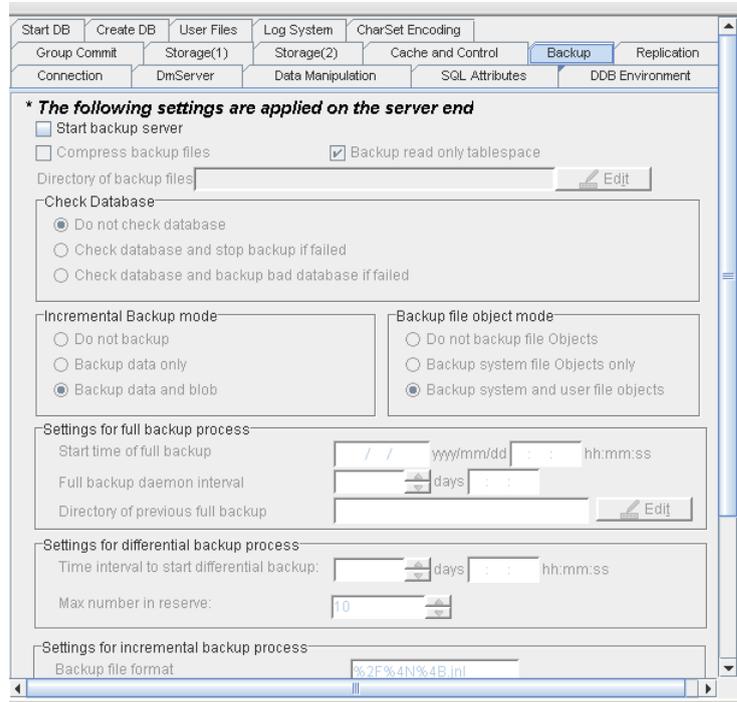


Figure 3-4 Backup page of the Create Database Advanced Settings window

START BACKUP SERVER

Start the Backup Server to enable the use of full, differential or incremental backup functions. The Backup Server must be started before the database is started.

ENABLE COMPRESS BACKUP FILES

The database administrator can choose whether to compress the backup files for full backup. When performing full backups, the system compresses the backup files when compression is enabled. This will reduce the required space for backup files.

ENABLING BACKUP READ ONLY TABLESPACE

The database administrator can select whether to backup the read only tablespace. Enabling backup of the read only tablespace permits the backup server to always backup read-only tablespace files. When this check box is disabled, the backup server will not backup the read-only tablespace files.

DIRECTORY OF BACKUP FILES

Using the **Directory of Backup Files** field the database administrator can specify the location of backup files. To guard against media failure, backup files should be stored on a disk separate from that of the current database.

CHECK DATABASE

The database administrator can specify one of three check modes to perform before execution of a full or differential backup. The three modes are:

- Do not check database
- Check database and stop backup if the database was found to be damaged
- Check database and continue with the backup operation regardless of the databases' s contidtion

INCREMENTAL BACKUP MODE

The database administrator can select a mode for the incremental backup process. Incremental backups copy only the journal files to the backup location. An incremental backup requires less time and resource to execute than a full backup, but restoring from an incremental backup requires more time. The three modes for incremental backup are:

- No Backup
- Backup Data Only
- Backup Data and BLOB

These modes do not affect the full backup process. After choosing a backup mode, be sure to enter a begin time and interval for the incremental backup daemon to

activate it. The Backup Server must be activated for incremental backup to take place.

BACKUP FILE OBJECT MODE

The settings under the **Backup File Object Mode** effect how file objects are copied during the full backup process. Selecting **Do Not Backup File Objects** disables file backup during the full backup process. Selecting **Backup System File Objects Only** results in system file objects being backed up during automatic full backups. Selecting **Backup System and User File Objects** results in both system file objects and user file objects being copied to the backup directory during automatic full backups.

SETTINGS FOR FULL BACKUP PROCESS

Enter these settings before performing a full backup. Activating the Backup Server enables a user to perform manual full backups. Full Backups require a lot of time and computing resources, however, they allow for faster restoration times. Enter a start time and interval for the full backup daemon to activate.

SETTINGS FOR DIFFERENTIAL BACKUP PROCESS

Enter these settings before performing a full backup (i.e., a differential base). Activating the Backup Server enables the user to perform manual differential backups. Differential Backups require less time and computing resources. Settings to specify include differential backup interval and the maximum number of differential backups to retain.

SETTINGS FOR INCREMENTAL BACKUP PROCESS

The settings for the incremental backup process become available only when **Start Backup Server** is enabled and the user selects Backup Data Only or Backup Data and BLOB. Settings include backup journal file format, incremental backup start time and interval, Compact backup, and the threshold for when to initiate incremental backups. These settings are needed for the Incremental Backup Process to function properly.

Setting backup options:

NOTE *The backup data location is specified by the **Directory of Backup Files** and should be on a separate disk for maximum data security.*

1. Click the **Setup** button in the **Create Database** window.
2. Click the **Backup** tab in the **Create Database Advanced Settings** window.
3. To start the backup server, select the **Start Backup Server** check box.
4. Select the **Compress backup files** check box to enable backup file compression.
5. Select the **Backup read-only tablespace** check box to include read-only tablespace files when performing full backups.
6. Enter a path in the **Directory of backup files** field or click the adjacent browse button  to specify a destination for backup files.
7. Choose a mode from the **Check Database** box. The modes are:
 - 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.
8. To enable full backups to be performed by the backup server:
 - a) Indicate a date and a time in the **Start Time of Full Backup** time fields.
 - b) Enter the number of days, hours, minutes, and seconds between each successive full backup in the **Full Backup Daemon Interval** days and time fields.
 - c) Enter a path into or select the browse button  next to the **Directory of Previous Full Backup** field to indicate a destination for the last full backup files.

9. To enable differential backups to be automatically performed by the backup server.
 - a) Enter the number of days, hours, minutes, and seconds between each successive full backup in the **Differential Backup Daemon Interval days and time** fields.
 - b) Enter a maximum value in the **Max number in reserve** field.
10. To enable incremental backups to be automatically performed by the backup server:
 - a) Activate the backup server.
 - b) Select a backup mode:
 - To select data backups only, select the **Backup Data Only** option button.
 - To backup data and BLOB files, select the **Backup Data and BLOB** option button.
 - c) Enter a format for backup journal files in the **Backup File Format** field.
 - d) Indicate a date and a time in the **Start Time of Incremental Backup** time fields.
 - e) Enter the number of days, hours, minutes, and seconds between each successive full backup in the **Time Interval to Start Incremental Backup** time fields.
 - f) To enable compact backup, click on the **Enable Compact Backup** check box.
 - g) Incremental backups can be set to automatically execute when journal files have filled to a threshold percentage.
 - 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 **% Full** field to set incremental backups to execute when any journal file is full to the value entered.
11. Click the **Save** button.
12. Click the **Cancel** button to return to the **Create Database** window.

Setting dmServer Options

The database administrator can specify the port number when creating the database by following the procedure outlined in section 3.1, or by selecting the **dmServer** tab in the create **Database Advanced Settings** window. In addition, the database administrator may also select the server address, network encryption, the idle time-out value, whether server information should be saved to a log file, and the default lock mode when creating tables. The dmServer settings can be changed upon creation of the database or before the database is started. The **dmServer** page corresponds exactly to the **dmServer** page in **JConfiguration Tool**. Please refer to Chapter 4 of the *JConfiguration Tool Reference* for more information.

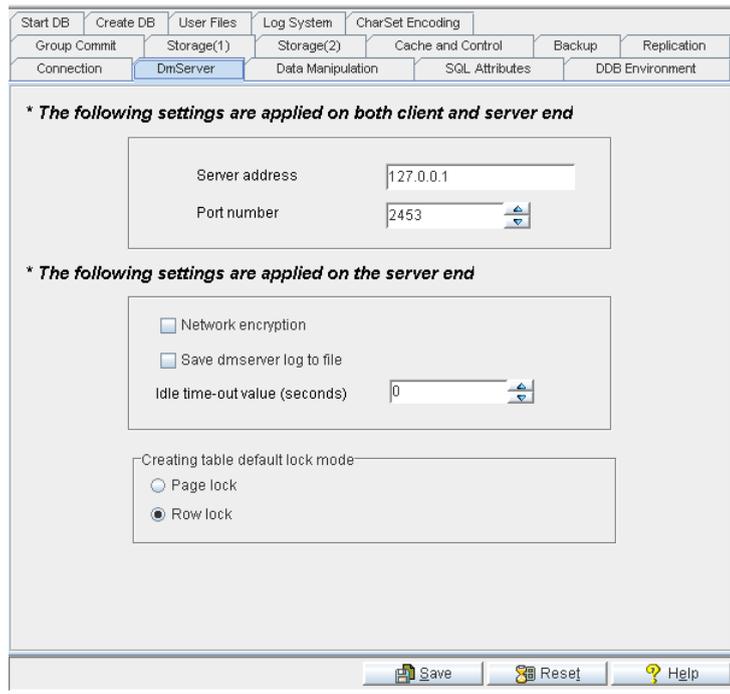


Figure 3-5 dmServer page of the Advanced Database Settings window

SERVER ADDRESS

The database administrator can alter the server IP number or host name, if DNS (Domain Name Server) has been set up properly on the client machine. This setting must be the same for the server and all clients connecting to the database.

PORT NUMBER

Server and client machines identify a multi-user database on a TCP/IP network with the port number. It must be the same for the server and all client machines connecting to the same database. The Port Number can be any integer from 1025 to 65535, and must be unique to each database on the server.

NETWORK ENCRYPTION

The database administrator can choose whether data accessed across a network is encrypted before being sent.

SAVE DMSEVER LOG TO FILE

The database administrator can supervise connections and troubleshoot connection problems by enabling Save dmServer Log to file. All dmServer commands and output is saved to a log file under the database directory with the database name and the file extension .log.

IDLE TIME OUT VALUE

The database administrator can set a time limit for users connected to the database to execute a new transaction before they are automatically disconnected. Users exceeding the Idle Time out Value without showing any activity are disconnected and all resources allocated for that user are freed.

CREATING TABLE DEFAULT LOCK MODE

Users can specify the default lock mode when creating tables. The lock mode can be either Page lock or Row lock.

- ➔ Setting DmServer options:
1. Click the **DmServer** tab in the Create Database Advanced Settings window. The **DmServer** page appears.
 2. Enter the server address in the **Server Address** field.
 3. Enter a port number in the **Port Number** field.

4. Select the **Network Encryption** check box to enable network encryption.
5. Select the **Save DmServer Log to File** check box to save the dmServer log.
6. Enter a time out value into the **Idle Time Out Value** check box.
7. Select the default lock mode for new tables by selecting the **Page lock** radio button to set page lock mode **Row Lock** radio button to set row lock mode.
8. Click the **Save** button.
9. Click the **Cancel** button to return to the **Create Database** window.

3.3 Using the Create Database Wizard

The **Create Database Wizard** helps users define important database configuration parameters that must be set before database creation. The following chart outlines the program logic for the **Create Database Wizard**. Refer to the steps in the procedure that follows the chart for a detailed description of its function.

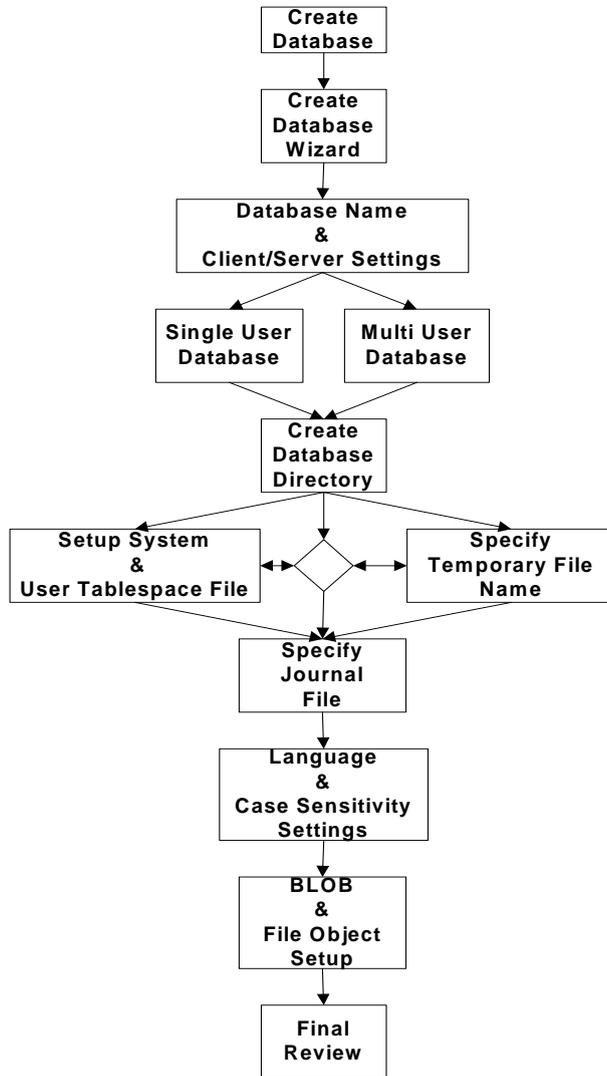


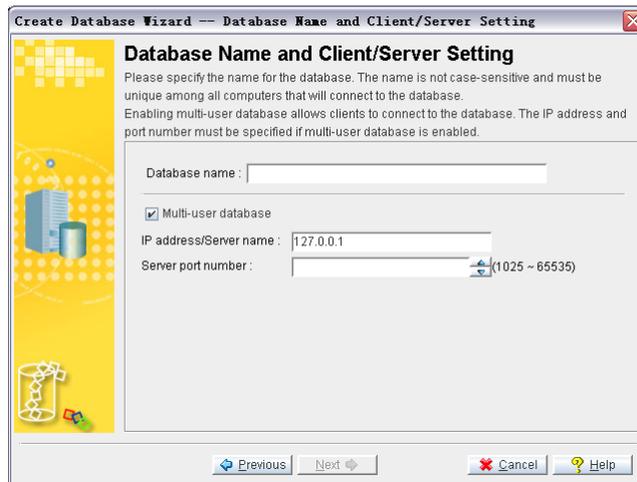
Figure 3-6 Program Logic for the Create Database Wizard

➔ **Creating a database using the Create Database Wizard**

1. Select **Create Database** from the **Wizard** drop-down menu. The **Create Database Wizard** window appears.



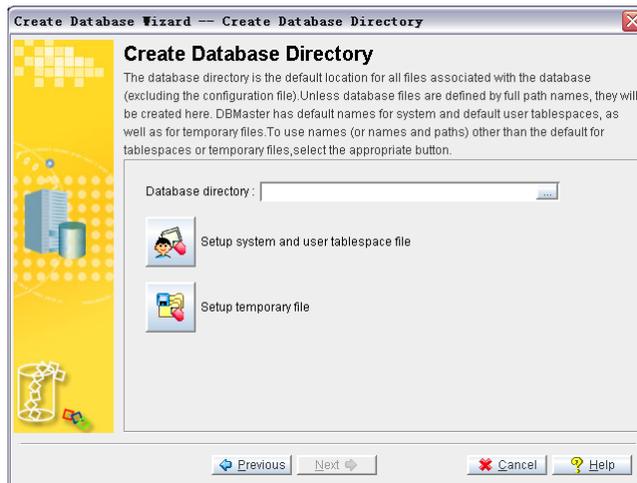
2. Click the **Next** button after reading the brief explanation of how the wizard will guide you through the process of creating a database. The **Database Name and Client/Server Setting** window appears.



3. Enter a name for the database in the **Database name** field.

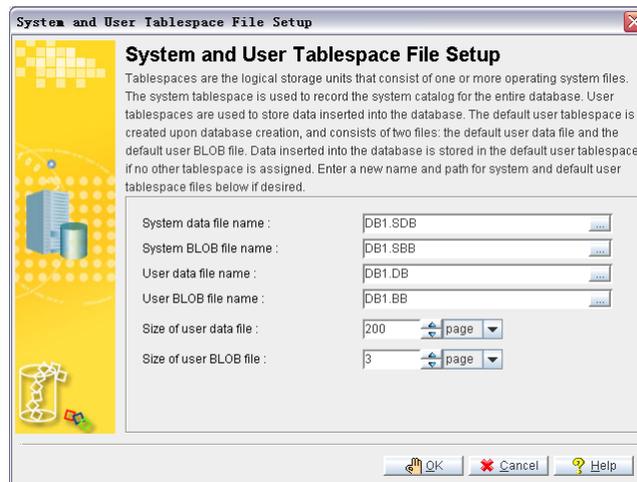
4. The user now has the choice of creating a single user or a multi-user database.
 - a) To create a single user database uncheck the **Multi user database** check box. Now continue to step 7.
 - b) To create a multi-user database check the **Multi-user database** check box. Continue to step 5.
5. Enter the IP address or host name of the server in the **Server Address** field.

NOTE *If a host name is specified, ensure that Domain Name Server is set up on the server.*
6. Enter the port number of the server in the **Server Port Number** field. This must be an integer between 1025 and 65535.
7. Click the **Next** button. The **Create Database Directory** window appears.



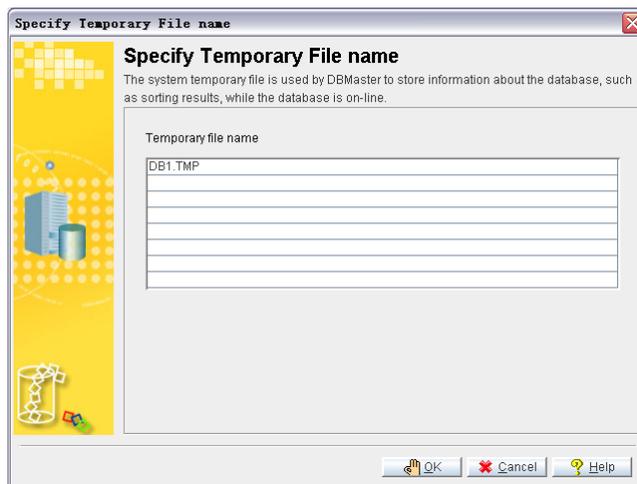
8. Enter the database directory into the **Database Directory** field, or select the browse button (...). If the browse button is selected, the **Open Directory** dialog box will appear:
9. The default database directory path is `DBMaker\5.0\bin\`. A new directory should be created for the database. To create a new database directory, click the browse button (...).
10. In the **Select file** window, navigate to the desired root for the database directory.

11. Click the **Create New Directory**  button to make a directory called **New Folder**.
12. Select the **New Folder** directory and enter the desired name for the database directory.
13. Select the new directory name from the list so that it appears in the **File name** field.
14. Click the **OK** button, the **Create Database Directory** window appears.
15. Click the **Setup system and user tablespace file** button to set the system and user tablespace files. The **System and User Tablespace File Setup** window appears.

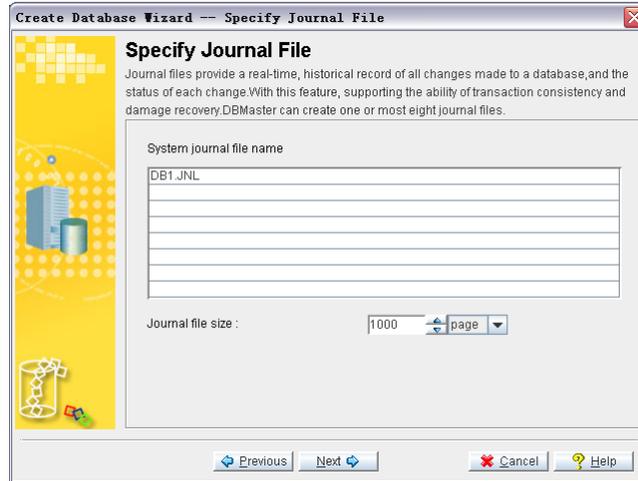


16. Enter a path or select the browse button  next to the **System data file name** field to indicate the location of the System Data File.
17. Enter a path or select the browse button  next to the **System BLOB file name** field to indicate the location of the System BLOB File.
18. Enter a path or select the browse button  next to the **User data file name** field to indicate the location of the User Data File option.
19. Enter a path or select the browse button  next to the **User BLOB file name** field to indicate the location of the User BLOB File.
20. Enter the number of pages to allocate from disk for each user data file in the **User data file pages size** field.

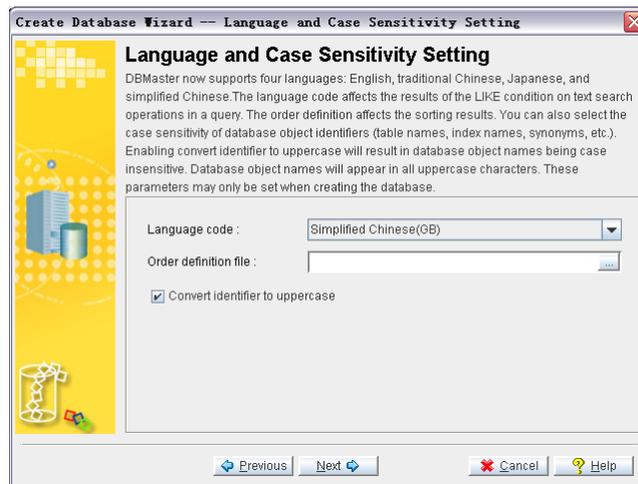
21. Enter the number of frames to allocate from disk for each user BLOB file in the **User BLOB file frames size** field.
22. Click the **OK** button. The **Create Database Directory** window reappears.
23. Click the **Setup temporary file** button to set the temporary file name. The **Specify Temporary File Name** window appears.



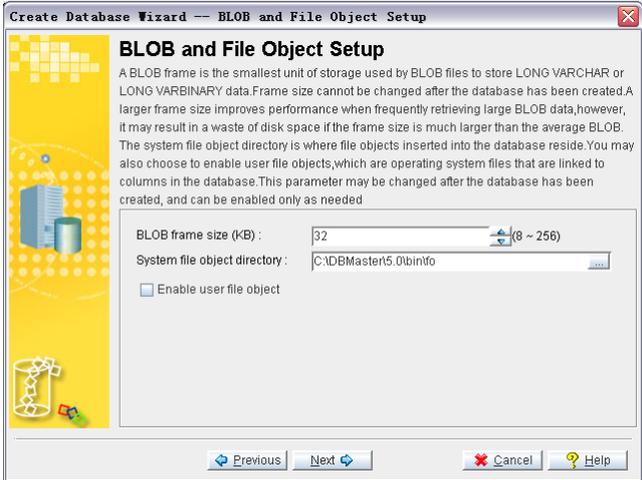
24. Double-click one of the fields to enter a name for the temporary system file.
25. Enter a name for the temporary system file and click the **OK** button. The **Create Database Directory** window reappears.
26. Click the **Next** button. The **Specify Journal File** window appears.



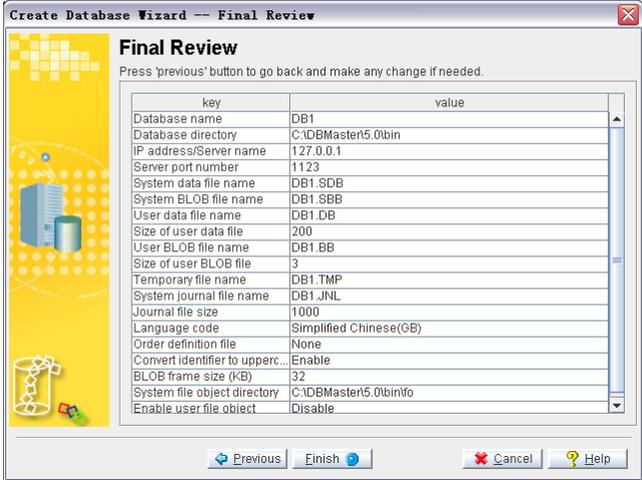
27. Double-click a field to enter the journal file name.
28. Enter the journal's filename.
29. Set the journal file size. Valid values are between 100 and 524287. The default value is *1000*.
30. Click the **Next** button. The **Language and Case Sensitivity Setting** window appears.



31. To change the character set the database will use for CHAR type data, select one of the four options from the **Language code** menu.
 - ASCII Compatible to use ASCII code
 - Big5 Compatible to use traditional Chinese code
 - JIS + Half Corner Compatible to use Japanese code
 - GB code compatible to use simplified Chinese code
 - **Latin code (ISO-8859-1)** compatible to use Latin – 1 code
 - **Latin code (ISO-8859-2)** compatible to use Latin – 1 code
 - **Cyrillic code (ISO-8859-5)** compatible to use Cyrillic code
 - **Greek code (ISO-8859-7)** compatible to use Greek code
 - **Japanese code(EUC-JP)**compatible to use Japanese code
 - **Simplified Chinese (GB18030)** compatible to use Simplified Chinese code
 - **UTF-8 (UTF-8)**
32. Enter a path or select the browse button  next to the **Order definition file** field to indicate the location of the order definition file.
33. To make all database objects case insensitive check the **Convert identifier to uppercase** check box.
34. Click the **Next** button. The **BLOB and File Object Setup** window appears.



- 35. Enter a BLOB frame size in bytes in the **BLOB frame size (KB)** field.
- 36. Enter a path or select the browse button  next to the **System file object directory** field to indicate the location of the database's system file objects.
- 37. Check the **Enable user file object** check box to allow user file objects.
- 38. Click the **Next** button. The **Final Review** window appears.



39. Verify the database settings. If any of the settings need to be changed, click **Previous** until reaching the appropriate window and make the changes. Then return to the **Final Review** window.
40. Click the **Finish** button when satisfied with the database settings.

4 Starting a Database

A database may be started with or without the **Start Database Wizard**. The **Start Database Wizard** provides an interface that explains every step of configuring a DBMaker database for startup.

Many database performance characteristics can be manipulated by changing database configuration parameters before startup. The following sections describe how to start a database with the default DBMaker settings, with your own settings, or with the **Start Database Wizard**.

4.1 Starting a Database: Basics

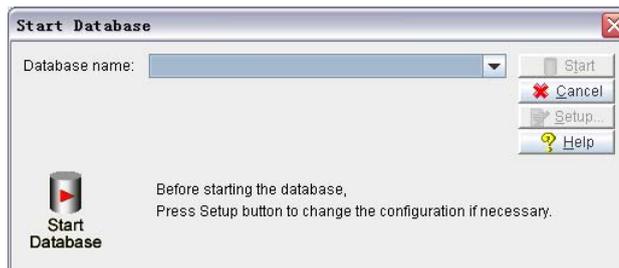
You may choose not to set any of the advanced settings if you are new to using DBMaker or your database has just been created and there is no need to adjust critical performance parameters.

➔ Starting a database:

1. To start a database, select **Start Database** from the main console. The **Start Database** window appears.

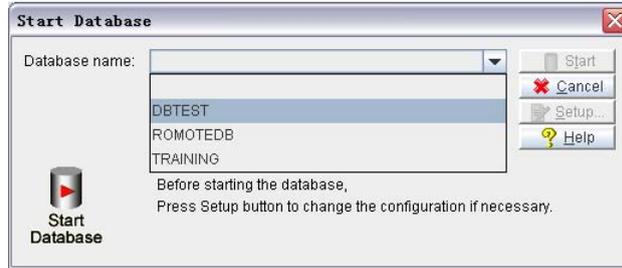


2. To use the wizard to start the database, select **Start database by wizard** and then refer to Section 4.3. Select **Start database** to avoid using the wizard, the following dialog box appears.



NOTE *You can also select a database from the Database drop-down menu.*

3. Select a database from the **Database Name** menu.



4. Choose to start the database or alter advanced database settings for the new database. To alter advanced settings, click **Setup** and refer to section 4.2. To start the new database, click **Start**. The dmServer application starts and the following message appears. Any errors are displayed in the dmServer window.



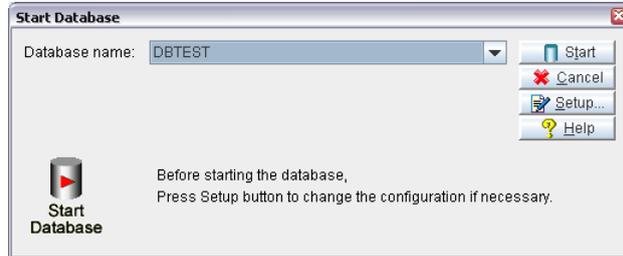
4.2 Starting a Database: Advanced Settings

Database administrators should ensure that all necessary changes to settings are made to a database before startup. Advanced settings include:

- Storage location of the database's operating system files
- Upper memory caching of data pages and system control
- Backup type, frequency, and location
- Location and frequency of table or database replication
- Database startup mode
- User defined file names
- Network connection behavior
- SQL command upper memory caching, CHAR data output padding, system File Object mapping, cursor behavior, and browse mode
- Date and time formatting for SQL statements, stored procedure settings
- Distributed database behavior
- Group commit settings

The settings in **JServer Manager** correspond exactly with the settings in the **JConfiguration Tool**. To learn more about how these settings affect the management of databases, please refer to the *JConfiguration Tool Reference*.

- ➔ Starting a database with advanced settings:
 1. Select **Start Database** from the main console, and select a database to start as outlined in section 4.1.



2. Click the Setup button. The Start Database Advanced Settings dialog box appears.
3. Enter advanced setting options (see the following procedures):
 - Setting Storage Options (Please refer to Chapter 3, *Setting Storage Options*)
 - Setting Cache and Control Options
 - Setting Backup Options
 - Setting Replication Options
 - Setting Start Options
 - Setting User File Options
 - Setting dmServer Options
 - Setting SQL Attributes
 - Setting Distributed Database (DDB) Environment Options
 - Setting Group Commit Options
4. After you have altered advanced database settings for the new database, click the **Start** button.



Setting Cache and Control Options

Cache and control settings manipulate the amount of memory used by an active database, control the maximum number of users that can access the active database, and control locks. Adjusting these settings influences database performance. For more information on performance tuning, please refer to the *Database Administrator's Guide*. The **Cache and Control** page exactly corresponds to the **Cache and Control** page in the **JConfiguration Tool**. Please refer to the *JConfiguration Tool Reference* for more information on cache and control settings.

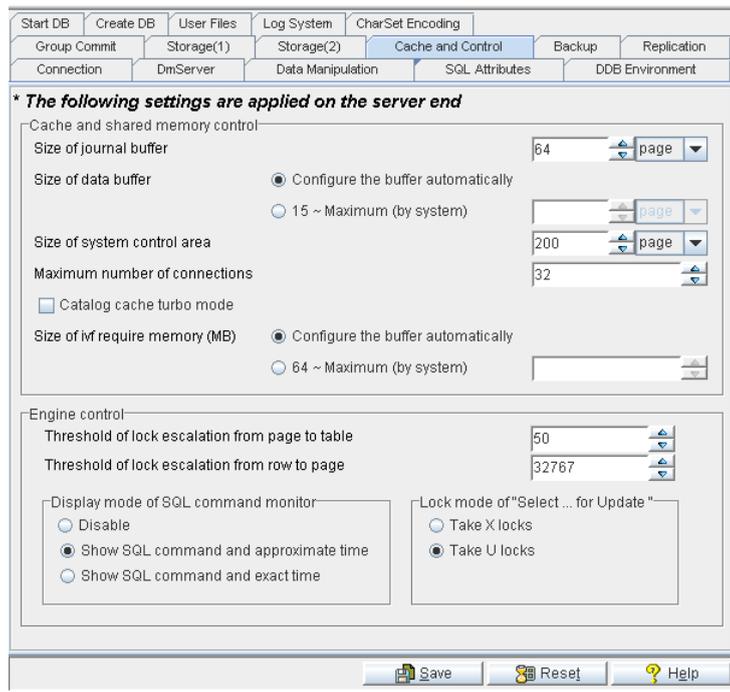


Figure 4-1 Cache and Control page of the Start Database Advanced Settings window

SIZE OF JOURNAL BUFFER

The Journal buffer stores the most recently used journal blocks as pages. It is a component of the Database Communication and Control Area (DCCA). With enough journal buffer pages, the time required to write journal blocks to disk when data is updated and the time required to read journal blocks from disk when a transaction is rolled back is reduced.

DATA BUFFER SIZE

The data buffer stores the most recently used data blocks as pages. It is a component of the Database Communication and Control Area (DCCA). DBMaker uses the most recent statistics values of schema objects to determine which data pages to keep in the buffer. Increasing or decreasing the number of data pages in the data buffer has a dramatic effect on database performance.

SYSTEM CONTROL AREA SIZE

Concurrency control blocks are stored in the System Control Area (SCA). It is a component of the Database Communication and Control Area (DCCA). Large transactions requiring many lock resources need a larger SCA to function properly. If too many resources allocated for concurrency control, the SCA can be enlarged, or the threshold of lock escalation can be decreased.

MAXIMUM NUMBER OF CONNECTIONS

The database administrator may control the maximum number of simultaneous connections that may be made to the database.

CATALOG CACHE TURBO MODE

Enabling the **Catalog Cache Turbo Mode** extends the lifetime of the catalog cache in the SCA. The default setting is *disabled*

IVF REQUIRE MEMORY SIZE

This field sets the maximum amount of memory that DBMaker reserves to cache query buffers and inverted file full text index buffers.

THRESHOLD OF LOCK ESCALATION FROM PAGE TO TABLE

When the number of locks on pages in the same table exceeds the lock escalation threshold from page to table, DBMaker automatically escalates the lock to a table

lock. When the number of locks on rows in the same table exceeds the lock escalation threshold from row to page, DBMaker automatically escalates the lock to a page lock. This number specifies the *Lock Escalation Threshold* for escalating a page lock to a table lock. When the number of locks on pages in the same table exceeds the lock escalation threshold, DBMaker automatically escalates the lock to a table lock. The default value is **60** pages.

THRESHOLD OF LOCK ESCALATION FROM ROW TO PAGE

This number specifies the *Lock Escalation Threshold* for escalating a row lock to a page lock. When the number of locks on rows in the same table exceeds the lock escalation threshold, DBMaker automatically escalates the lock to a page lock. The default value is **30** rows.

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 complete 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 a user and the time of execution, respectively. **Display Mode of SQL Command Monitor** sets the configuration for displaying these attributes. No SQL commands are show if **Disable** is selected. 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 selected. 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 selected. 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*.

LOCK MODE

This setting specifies the lock behavior of all “select ... for update” statements at the server site. By default, DBMaker takes S locks on the result set of a “select ... for update” statement. For some applications, users may want to take exclusive

locks on objects that are indicated by a “select ... for update.” Selecting the **Take X locks** radio button indicates that DBMaker will take X locks on the result set of all “select ... for update” statements.

Setting cache and control options:

1. Click on the Setup button in the Start Database window. The Start Database Advanced Settings window appears.
2. Click the Cache and Control tab in the Start Database Advanced Settings window.
3. Enter the number of journal buffer pages to allocate from memory in the Size of Journal Buffer field.
4. Enter the number of data buffer pages to allocate from memory in the Size of Data Buffer field.
5. Enter the number of pages to allocate from memory for the System Control Area in the **Size of System Control Area** field.
6. Enter the maximum number of users that can connect to the database in the Maximum number of Connections field.
7. To enable catalog cache turbo mode click on the check box next to Catalog Cache Turbo Mode.
8. Enter the number of pages at which a page lock will escalate to a table lock in the Threshold of lock Escalation form Page to Table field.
9. Enter the number of rows at which a row lock will escalate to a page lock in the Threshold of lock Escalation form Row to Page field.
10. To change the display content of the **SQL_CMD** and **TIME_OF_SQL_CMD** columns in the **SYSUSER** system table, select one of the three radio buttons.
 - Select Disable to not display SQL Commands
 - Select Show SQL command and approximate time to display the approximate time of SQL command execution
 - Select Show SQL command and exact time to display the exact time of SQL command execution
11. To change the lock mode of “select ...for update” SQL commands, select one of the two option buttons.

- Select take X locks to take exclusive locks on “select ...for update” SQL commands.
 - Select take S locks to take shared locks on “select ...for update” SQL commands.
12. Click the Save button to save all changes to the **dmconfig.ini** file.
 13. Click the Cancel button to return to the Start Database window, and click **Start** to start the database.

Setting Backup Options

The database administrator can specify how DBMaker backs up data for maximum security. The Backup page exactly corresponds to the Backup page in the **JConfiguration Tool**. Please refer to the *JConfiguration Tool Reference* for more information about backups.

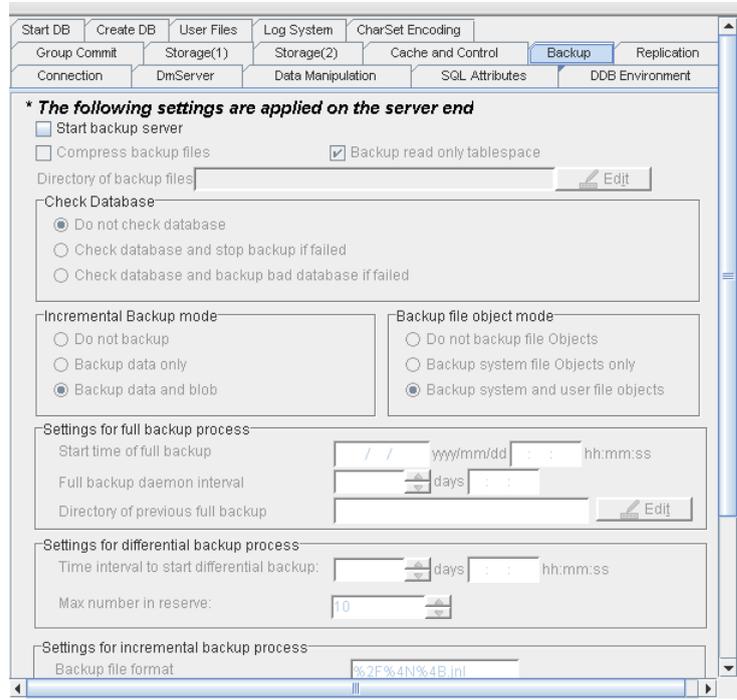


Figure 4-2 Backup page of the Start Database Advanced Settings window

START BACKUP SERVER

Start the Backup Server to enable the use of full, differential or incremental backup functions. The backup server must be started before the database is started.

ENABLE COMPRESS BACKUP FILES

The database administrator can choose whether to compress the backup files for full backups. When performing a full backup, the backup files are compress when this option is enabled. This can reduce the requirement of free space for back up files.

ENABLING BACKUP READ ONLY TABLESPACE

The database administrator can also select whether to backup read only tablespaces. Enabling backup read only tablespace permits the backup server to always backup read-only tablespace files. Disabling this check box tells the backup server to not backup the read-only tablespace files.

BACKUP FILES DIRECTORY

The database administrator can specify the location for the backup files in the **Directory of Backup Files** field. To guard against media failure, backup files should be stored on a disk separate from that of the current database.

CHECK DATABASE

The database administrator can specify whether to check databases before full and differential backups. The three check database modes are:

- Do not check database
- Check database and stop backup if the database was found to be damaged
- Check database and continue with the backup operation regardless of the databases's contidion

INCREMENTAL BACKUP MODE

The database administrator can also select a mode for the incremental backup process. Incremental backups copy only the journal files to the backup location. An incremental backup requires less time and resource to execute than a full backup, but restoring from an incremental backup requires more time. The three modes for incremental backup are:

- No Backup
- Backup Data Only
- Backup Data and BLOB

The modes do not affect the full backup process. After choosing a backup mode, be sure to enter a begin time and interval for the incremental backup. The backup server must be activated for incremental backup to take place.

BACKUP FILE OBJECT MODE

The settings under the **Backup File Object Mode** effect how file objects are copied during the full backup process. Selecting **Do Not Backup File Objects** disables file backup during the full backup process. Selecting **Backup System File Objects Only** will result in system file objects being backed up during automatic full backups. Selecting **Backup System and User File Objects** will result in both system file objects and user file objects being copied to the backup directory during automatic full backups.

SETTINGS FOR FULL BACKUP PROCESS

These settings must be entered before attempting a full backup. Activating the backup server allows users to perform manual full backup. Full backups are time consuming and resource intensive, but allow for faster restoration times. Enter a start time and interval for the full backup.

SETTINGS FOR DIFFERENTIAL BACKUP PROCESS

Enter these settings before performing a full backup (i.e., a differential base). Activating the Backup Server enables the user to perform manual differential backup. Differential backups require time and computing resources. Settings include differential backup interval and the maximum number of differential backups to retain.

SETTINGS FOR INCREMENTAL BACKUP PROCESS

The settings for the incremental backup process become available only when **Start Backup Server** is enabled and the user selects Backup Data Only or Backup Data and BLOB. Settings include backup journal file format, incremental backup start time and interval, compact backup, and the threshold to initiate incremental backups. These settings are needed for the incremental backup process to function properly.

Setting backup options:

NOTE *Specify the location for backup files using Directory of Backup Files. For maximum security, backup files should be on a separate disk.*

1. Click **Setup** in the Start Database window. The Start Database Advanced Settings window appears.
2. Click the **Backup** tab in the Start Database Advanced Settings window.
3. To start the backup server, select the **Start Backup Server** check box.
4. Select the **Compress backup files** check box to enable backup compression.
5. Click the **Backup read-only tablespace** check box to include read-only tablespace files when performing full backups.
6. Enter a path in the **Directory of backup files** field or click the adjacent browse button  **Edit** to specify a destination for backup files.
7. Choose a check mode from the **Check Database** box.
 - Select **Do not check database** option button to skip database checking before full or 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.
8. To enable full backups to be performed by the backup server,
 - a) Indicate a date and a time in the Start Time of Full Backup time fields.
 - b) Enter the number of days, hours, minutes, and seconds between each successive full backup in the Full Backup Daemon Interval time fields.
 - c) Enter a path or select the browse button  next to the Directory of Previous Full Backup field to indicate a destination for the last full backup files.

9. To enable differential backups to be automatically performed by the backup server.
 - a) Enter the number of days, hours, minutes and seconds between each successive full backup in the **Differential Backup Daemon Interval days and time** fields.
 - b) Enter a maximum value in the **Max number in reserve** field.
10. To select what types of file objects are backed up during the backup process:
 - a) Select **Do not backup file objects** to exclude file objects from backups.
 - b) Select **Backup system file objects only** to only back up system file objects.
 - c) Select **Backup system and user file objects** to back up all file objects.
11. To enable incremental backups to be automatically performed by the backup server:
 - a) Select the Backup Only option button to select data backups only, or Select the Backup Data and BLOB option button to backup data and BLOB files.
 - b) Enter a format for backup journal files in the Backup File Format field.
 - c) Indicate a date and a time in the **Begin Time of Incremental Backup time** fields.
 - d) Enter the number of days, hours, minutes, and seconds between each successive full backup in the **Time Interval to Start Incremental Backup time** fields.
 - e) Select the Enable Compact Backup check box to enable backup compression.
 - f) Incremental backups can be set to automatically execute when journal files have filled to a set percentage.
 - Select the Backup when any Journal File is Full radio button to set incremental backups to execute when any journal file is full.
 - Enter a value from 50 to 100 in the % Full field to set incremental backups to execute when any journal file is filled to the value entered.
12. Click the **Save** button.

- Click the **Cancel** button to return to the **Start Database** window, and click the **Start** button to start the database.

Setting Replication Options

Asynchronous Table Replication (ATR) periodically writes data from the distributor database to remote tables of the replicated database based according to a schedule. The Asynchronous Table Replication Daemon must be activated before the database is started for ATR to work properly. Database replication periodically writes all changes made to the primary database onto target databases, which are essentially read-only copies of the primary database located on remote servers.

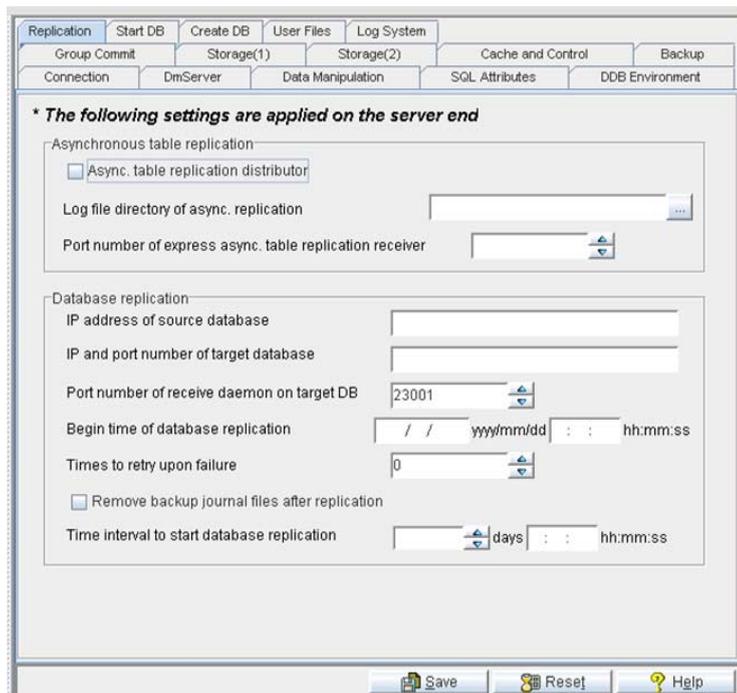


Figure 4-3 Replication page of the Start Database Advanced Settings window

ENABLING ATR DISTRIBUTOR

The schedule of the distributor daemon is dependent on the remote (i.e., destination) database and is specified using the SQL command CREATE

SCHEDULE, or using the **Create Replication Schedule Wizard** in the **JDBA Tool**. Please refer to the *SQL Command and Function Reference* for more information on SQL commands, or the *JDBA Tool User's Guide* for more information on using the **JDBA Tool**.

SETTING THE PORT NUMBER OF THE EXPRESS ATR RECEIVER

Creating an express asynchronous table replication requires this setting. Express ATR uses a different method from ODBC calls to update target tables. All source and destination tables that share the same tables must have the same Express ATR port number. Only databases that use the DBMaker engine can use this feature. For more information, please refer to the *DBMaker Database Administrators Guide*.

SETTING THE ATR LOG FILE DIRECTORY

Asynchronous table replication error logs are associated with only the source database. By default, these are located under the source database directory. Users should not manually remove the replication log files. The default path for ATR Log files is **(database name)/(TRPLOG)** and is automatically created by DBMaker. The total length of the ATR log file directory path must not exceed 256 characters in length.

IP ADDRESS OF SOURCE DATABASE

This field, used for database replication, specifies the port number of the RP_RECV daemon at the slave database. It must be different from the **port number** specified on the dmServer page of the slave database's **dmconfig.ini** file and the same as the port number specified by the **IP and Port Number of target database** field of the primary database. This field should be blank on the primary database server. There is no default value.

TARGET DATABASE IP AND PORT NUMBERA

Entering the **Port Number of Receive Daemon on Target DB** specifies the location of slave databases with respect to the primary database. These numbers are specified on the primary database side of the database replication. It should correspond to the values in the **Server Address** and **Port Number of Receive Daemon on Target DB** fields specified in each target (i.e., slave) databases

dmconfig.ini file. DBMaker supports up to 256 slave databases for each primary database. The format should follow this syntax:

```
address[:port number] {, address[:port number]}
```

Where the address is the IP number or host name of each slave (i.e., target) database, and the port number is the replication receive daemon port number specified in the **dmconfig.ini** file of each slave database. The default port number is **23001**. Use a comma or blank space to separate each slave database's information.

➤ Example

A typical representation of IP address and port numbers of slave databases in the **dmconfig.ini** file:

```
192.168.9.222:5100, Server2:5101, Server3
```

There are three slave databases. One is 192.168.9.222 with port number 5100, another is Server2 with port number 5101, and the other is Server3 with default port number 23001.

RECEIVE DAEMON PORT NUMBER FOR TARGET DATABASES

This field is used only if the database is a slave database. It specifies the port number of the Replication Receive Daemon at the slave database. It must be different from the Port Number set in the dmServer page that is used by the slave database and the same as the port number specified in the **Port Number of Receive Daemon on Target DB** field in the primary database. The default value is **23001**.

SETTING THE REPLICATION START TIME

This value specifies the time when database replications begin. JConfiguration Tool automatically enters values into the first two spaces of the **yyyy** field and the first space of the **hh** field when values are entered into other fields. This setting corresponds to the **RP_BTime** keyword in the **dmconfig.ini** file. The default value is the starting time of the primary database.

TIMES TO RETRY ON FAILURE

During database replication a network failure may occur. In this case, DBMaker attempts to reconnect to the remote databases. This value specifies how many times DBMaker will try to reconnect.

REMOVE BACKUP JOURNAL FILES AFTER REPLICATION

Enabling **Remove Backup Journal Files after Replication** tells DBMaker to delete journal backup files after sending them to remote databases during database replication.

SETTING THE REPLICATION DAEMON INTERVAL

The value in the **Time Interval to Start Database Replication** combo box specifies the time interval, in days, when database replication occurs. Next to the combo box is a field for time input, which specifies the time interval in hours, minutes, and seconds. The total time interval is determined by adding the two values together. For example, inputting 1 into the **days** field and inputting 12:00:00 into the **hh:mm:ss** field causes an incremental backup to execute every day and a half.

Setting replication options:

1. Click **Setup** in the **Start Database** window. The **Start Database Advanced Settings** window appears.
2. Click the **Replication** tab in the **Start Database Advanced Settings** window.
3. To enable asynchronous table replication (ATR):
 - a) Enable the check box next to **Async. Table Replication Distributor**.
 - b) If using express ATR, enter a value into the **Port Number of Express Async. Table Replication Receiver**.
 - c) Enter a path into or select the browse button  next to the **Log File Directory of Async. Replication** field to indicate a location for the ATR log file.
4. To enable database replication:
 - a) Enter the **IP Address of Source Database**.
 - b) If you are modifying settings for the primary (i.e., source) database, enter the **IP and Port Number of Target Database**.

- c) If you are modifying settings for the target (i.e., slave) database, enter the **Port Number of Receive Daemon on Target DB**.
 - d) Enter a date and time into the **Begin Time of Database Replication** time fields.
 - e) Enter a value into the **Times to Retry upon Failure** field
 - f) If desired, enable **Remove Backup Journal Files after Replication**
 - g) Enter the number of days, hours, minutes, and seconds between each successive database replication in the **Time Interval to Start Database Replication** time fields.
5. Click **Save**.
 6. Click **Cancel** to return to the Start Database window, and click **Start** to start the database.

Setting Start Database Options

There are six start-up modes for starting a database using **JServer Manager**.

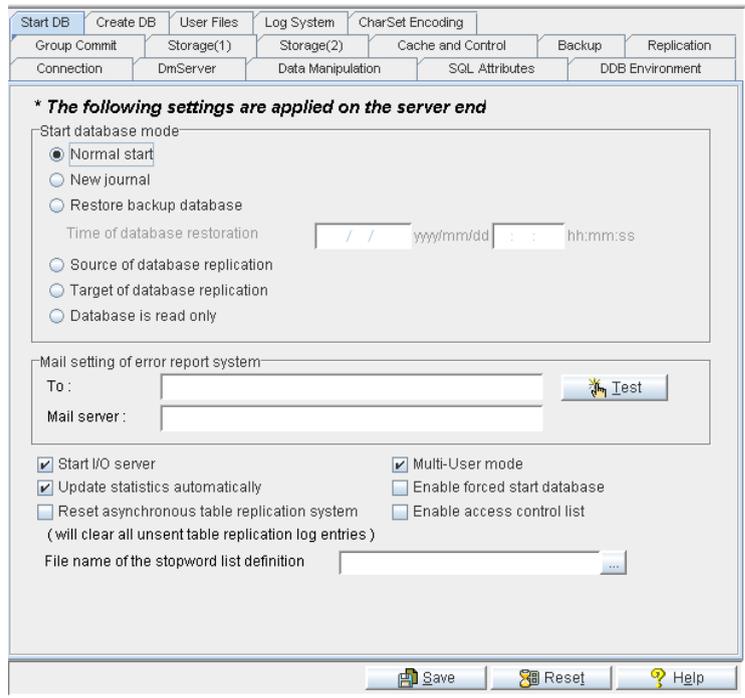


Figure 4-4 Start Database page of the Start Database Advanced Settings window

NORMAL START MODE

This mode starts up a system normally. If the database crashed last time, DBMaker performs crash recovery automatically to bring the database into a consistent and stable state.

NEW JOURNAL MODE

This mode starts up a system normally, but creates new journal files from names specified in the **Names of System Journal Files** on the **Storage** page. All old records are overwritten if the previous journal file names are kept. This setting must be selected if the user wants to change the journal file size, add more journal files, or change the journal file name. We recommended performing an incremental, differential or full backup before selecting this option

RESTORE BACKUP DATABASE

This mode uses the backed up database files, including the journal file, to start the database. DBMaker uses the incremental backup files to rollover the operations up to the time specified in the fields indicated by the **Time of Database Restoration** option. When this option is selected, Server Manager prompts the user to copy backup files to the database and then prompts the user for each incremental file to use for rolling the database over upon startup. If no value is specified or the date specified is later than the time of the last incremental backup, the Time of Database is restored to its default value.

SOURCE OF DATABASE REPLICATION

This mode is used for database replication. Starting up a database with this mode makes it a primary (source) database. Database replication may be performed manually or according to a schedule, and is set using the JDBA Tool. Please refer to the *JDBA Tool User's Guide* for more information on database replication.

TARGET OF DATABASE REPLICATION

This mode is used for database replication. Starting up a system with this mode makes it a slave database. The database is opened as read-only and must have the proper settings for database replication. Database replication may be performed manually or according to a schedule, and is set using the **JDBA Tool**. Please refer to the *JDBA Tool User's Guide* for more information on database replication.

DATABASE IS READ-ONLY

This mode starts up a system normally, but the database is read-only, providing only read privilege to users. Starting a primary database in read-only mode prevents users from modifying it.

ERROR REPORT SYSTEM MAIL SETTINGS

DBMaker creates an error report log containing all of the errors that occur during normal operation. These errors may not interfere in the performance of the database and may not be noticeable to the database administrator unless the error log is regularly checked. Alternatly, an e-mail address and SMTP server can be specified for relaying the error information. DBMaker automatically sends the error messages to the recipient as soon as they occur.

START I/O SERVER

The primary purpose of the I/O server daemon is to manage data page buffers. The checkpoint daemon periodically clears the data page buffers and writes data to disk. Enabling the I/O server tells whether DBMaker should turn the I/O and checkpoint daemon on or off. The default setting is *enabled*.

MULTI USER MODE

To allow more than one user to access a database, enable Multi-user Mode. Client machines will be able to connect to the server over a network.

UPDATE STATISTICS AUTOMATICALLY

DBMaker records, in the system tables, statistical information about database objects. Periodically the database needs to be read and statistics recalculated and written to the system tables to ensure efficient database operation. Enabling **Update statistics automatically** ensures that DBMaker automatically and periodically recalculates database statistics according to its internal scheduling daemon.

ENABLING FORCED START DATABASE

This mode forces a database to start even if a serious error occurs while starting the database (e.g., a crash).

RESET ASYNCHRONOUS TABLE REPLICATION SYSTEM

Reset Asynchronous Table Replication System clears all unsent table replication log entries upon startup.

Setting database startup options:

1. Click the Setup button in the Start Database window.
2. Click the Start Database tab in the Start Database Advanced Settings window.
3. Select a database startup mode by clicking on one of the following option buttons available in the Start Database Mode field.
 - Normal Start
 - New Journal

- Restore Backup Database (Enter the restoration date and time in the Time of Database Restoration time fields)
 - Source of Database Replication
 - Target of Database Replication
 - Database is Read Only
4. To enable the database to have only one connection, disable the Multi-User Mode option.
 5. To allow the database to start even if there are errors, enable the Enable Forced Start Database option.
 6. To clear all unsent table replication log entries upon startup enable the Reset Asynchronous Table Replication System option.
 7. Click Save.
 8. Click Cancel to return to the Start Database window, and click **Start** to start the database.

ENABLING ACCESS CONTROL LIST CHECK

This check box database administrators (DBAs) to block user access. DBAs can allow all users from any IP address to connect to the database. In addition, DBAs can block some users come, from any network address, from connecting to the database.

STOPWORD LIST DEFINITION FILE NAME

Users can define the list of stopwords according their needs. Users can enter the path of the stopword list definition file into the text field, or click the browse button (⋯) next to the text field to choose the directory containing the stopword list definition file. If users do not specify any configuration, DBMaker loads a default stopword containing a pre-defined file name based on LCODE.

Setting User File Options

DBMaker allows users to create new data files or BLOB files and add them to a tablespace when the original tablespace is full. Users specify a logical file name

(i.e., without a full path) when creating a file. However, users can map the logical file name to a physical file name that is used by the operating system.

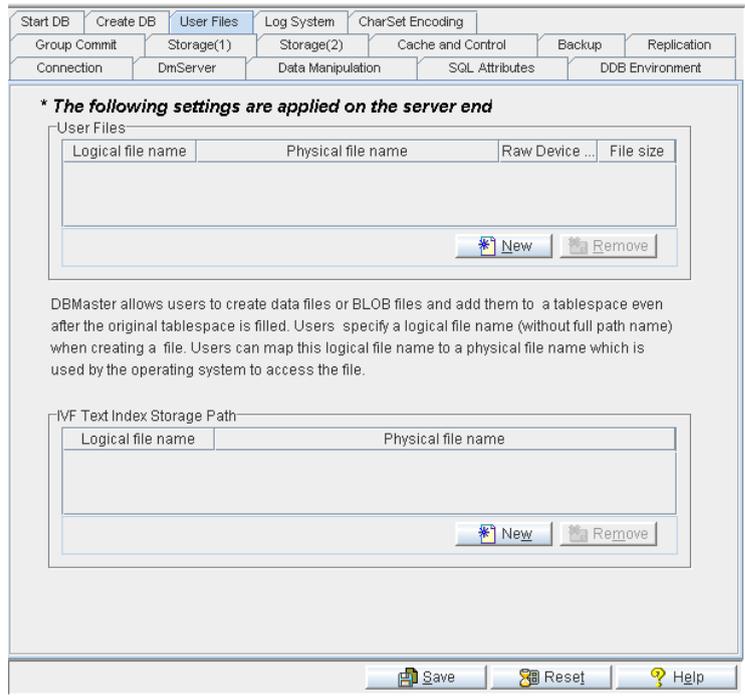
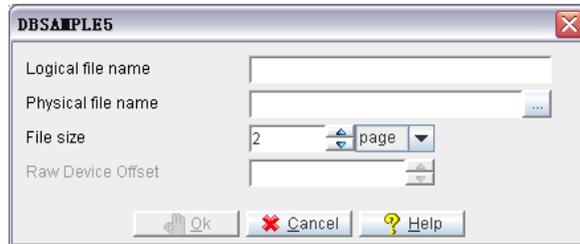


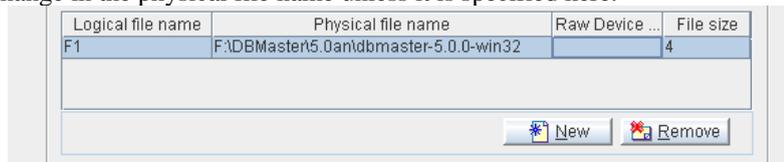
Figure 4-5 User Files page of the Start Database Advanced Settings window

Managing files in tablespaces:

1. Click the **Setup** button in the Start Database window.
2. Click the **User Files** tab in the Start Database Advanced Settings window.
3. To create a user-defined file click the **New** button at the bottom of the user files field. A dialog box appears displaying the name of the database:



4. Enter the logical file name in the **Logical file name** field (this is the path referenced by SQL commands and by DBMaker to reference your data).
5. Enter a Physical File Name by entering a new path into the **Physical File Name** field, or by clicking the browse button  next to the **Physical File Name** field. This is the operating system path. The maximum number of characters for the Physical File Name path is 79.
6. Enter a value into the **Page size** box. This indicates the number of data pages allocated for the User-Defined File. The valid range of values for page size is from 2 to 524,287.
7. Click the **OK** button.
8. A new user defined file appears in the User Files field. The logical file name, physical file name, and page size can be changed after the user-defined file is created. Double-click the text to edit it. This is useful if data files are moved within the operating system. DBMaker will not recognize a change in the physical file name unless it is specified here.



9. To delete a user-defined file click Remove. DBMaker will no longer recognize the logical file name.
10. Click Save.
11. Click Cancel to return to the Create Database window, and click **Start** to start the database.

Setting DmServer Options

DmServer settings may be changed before a database is started. The settings correspond exactly to the **DmServer** page in the JConfiguration Tool. Please refer to Chapter 4 of the *JConfiguration Tool Reference* for more information

Figure 4-6 dmServer page of the Start Database Advanced Settings window

SERVER ADDRESS

The database administrator can alter the server IP number or host name in the event that DNS is not configured properly over the network. This setting must be the same for the server and all clients connecting to the database.

PORT NUMBER

Server and client machines identify a multi-user database on a TCP/IP network using the port number. It must be the same for the database server and all client machines connecting to the same database. The port number can be any integer from 1025 to 65535, and must be unique to the database being accessed.

NETWORK ENCRYPTION

The database administrator can choose whether data accessed across a network is encrypted before being sent.

SAVE DMSERVER LOG TO FILE

The database administrator can supervise connections and troubleshoot connection problems by enabling **Save DmServer Log to file**. All DmServer commands and output is saved to a log file under the database directory with the database name and file extension .log.

IDLE TIME OUT VALUE

The database administrator can set a time limit whereby users connected to the database must execute a new transaction within the time limit or they are automatically disconnected. Users exceeding the Idle Time out Value without showing any activity are automatically disconnected. In addition, all processor resources allocated for that user are released.

CREATING TABLE DEFAULT LOCK MODE

Users can specify the default lock mode when creating tables. The lock mode can be:

Page lock or Row lock.

➤ Setting DmServer options:

1. Click the DmServer tab in the Create Database Advanced Settings window. The DmServer page appears.
2. Enter the server address in the Server Address field.
3. Enter a port number in the Port Number field.
4. Select the Network Encryption check box to enable network encryption.
5. Select the Save dmServer Log to File check box to save the dmServer log.
6. Enter a time out value into the Idle Time Out Value check box.
7. Select the creating table default lock mode:
 - Select the **Page lock** option button to set page lock mode.
 - Select the **Row Lock** option button to set row lock mode.

8. Click Save.
9. Click Cancel to return to the Create Database window, and click **Start** to start the database.

Setting SQL Attributes

SQL date and time input and output formats can be modified, as well as the location of stored procedure related files.

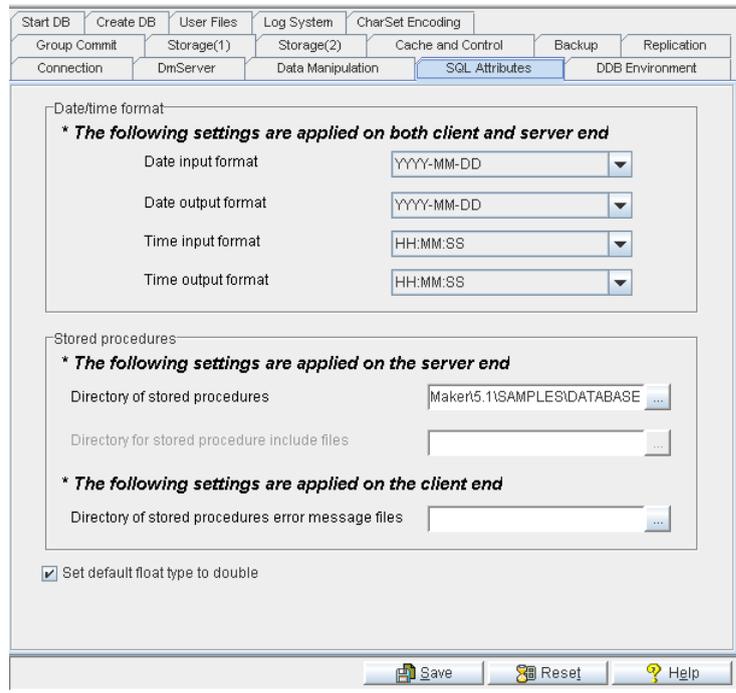


Figure 4-7 SQL Attributes page of the Start Database Advanced Settings window

DATE / TIME FORMAT

Date and time input and output formats for SQL statements can be selected from the available formats located in the drop-down menus.

DIRECTORY OF STORED PROCEDURES

The directory shown here specifies the path stored procedure files are placed in. The stored procedure files include the dynamic linking library files and temporary files generated during stored procedure creation.

DIRECTORY OF STORED PROCEDURE ERROR MESSAGE FILES

The directory shown here specifies the path stored procedure log files are placed in. The stored procedure log files include the error log files sent from the database server while creating stored procedures and the trace log file for stored procedure execution.

DIRECTORY OF STORED PROCEDURE INCLUDE FILES

The **Directory of Stored Procedures** specifies the path stored procedure include files are placed in. It is used when the user needs extra include files for stored procedures. This keyword is only useful in non-Windows operating systems.

SET DEFAULT FLOAT TYPE IS DOUBLE TYPE

Choose the **Set default float type is double type** check box specifies that the FLOAT column will be same as the DOUBLE type, which uses 8 bytes as internal storage and value range. And the type name is DOUBLE. Otherwise, the FLOAT type will be the REAL type, which uses 4 bytes as internal storage, and the type name is REAL. This setting corresponds to the **DB_FLtDb** keyword in the **dmconfig.ini** file. The default FLOAT type is **DOUBLE**.

➤ Setting SQL Attributes:

1. Click Setup in the Start Database window.
2. Click the SQL Attributes tab in the Start Database Advanced Settings window.
3. Select date and time input and output formats from the available drop down menus.
4. Enter a path into or select the browse button  next to the Directory of Stored Procedures field to indicate a storage location for stored procedures.
5. Enter a path into or select the browse button  next to the Directory of Stored Procedure Error Message Files field to indicate a location for the stored procedure error message files.

6. Enter a path into or select the browse button  next to the Directory of Stored Procedure field to indicate a location for stored procedure include files.
7. Enable the **Set default float type is double type** option.
8. Click Save.
9. Click Cancel to return to the Create Database window, and click **Start** to start the database.

Setting Distributed Database Environment Options

You can set DDB environment options that let your database handle queries in a distributed database environment.

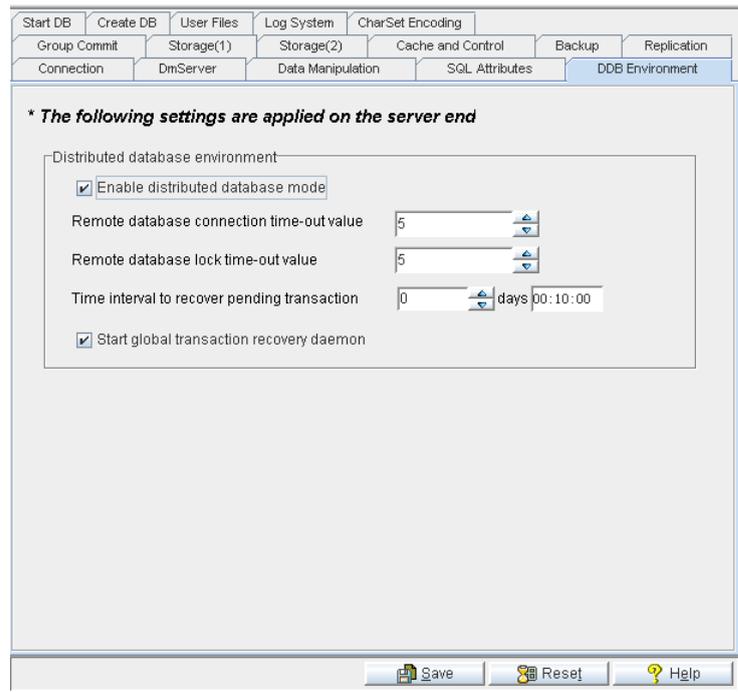


Figure 4-8 DDB Environment page of the Start Database Advanced Settings window

ENABLE DISTRIBUTED DATABASE MODE

This setting enables the database to be used in distributed mode. It must be enabled for synchronous table replication to work. Synchronous table replication is set using the JDBC Tool. For more information on distributed data, synchronous table replication, and coordinator and participant databases, please refer to the *JDBC Tool User's Guide* or the *Database Administrator's Guide*.

REMOTE DATABASE CONNECTION TIME-OUT

The value in the combo box specifies the time, in seconds, that a coordinator database waits when reestablishing a connection to a participant database.

REMOTE DATABASE LOCK TIME-OUT

The number in the combo box specifies the time, in seconds, that the coordinator database waits when reestablish a lock on the requested data in a participant database.

TIME INTERVAL TO RECOVER PENDING TRANSACTION

This value represents the time interval at which the global transaction recovery daemon is activated.

GLOBAL TRANSACTION RECOVERY DAEMON

This setting activates an automatic recovery mechanism. The mechanism checks whether a database has any problems with pending transactions, and then recovers them. This feature prevents loss of data when a network failure occurs or if there is an error at the participant database. The number in the combo box specifies the time interval in days from which the global transaction recovery daemon is activated.

Setting distributed database options:

1. Click Setup in the Start Database window.
2. Click the DDB Environment tab in the Start Database Advanced Settings window.
3. Click on the Enable Distributed Database Mode check box. The options below are highlighted.
4. Enter the number of seconds to wait for a connection to be established before returning an error in the Remote Database Connection Time-out value field.

5. Enter the number of seconds to wait when trying to establish a lock on a remote database before returning an error in the Remote Database Lock Time-out Value field.
6. Enable the Start Global Transaction Recovery Daemon option.
7. Enter the number of days, hours, minutes, and seconds between every activation of the global transaction recovery daemon in the Time Interval to Recover Pending Transaction time fields.
8. Click Save.
9. Click Cancel to return to the Create Database window, and click **Start** to start the database.

Setting Group Commit Options

The group-commit function improves transaction processing by synchronizing journal file operations. DBMaker collects as many transactions as possible over a given interval and then commits them simultaneously to improve on-line transaction processing performance. This method is suitable for large numbers of short transactions running simultaneously.

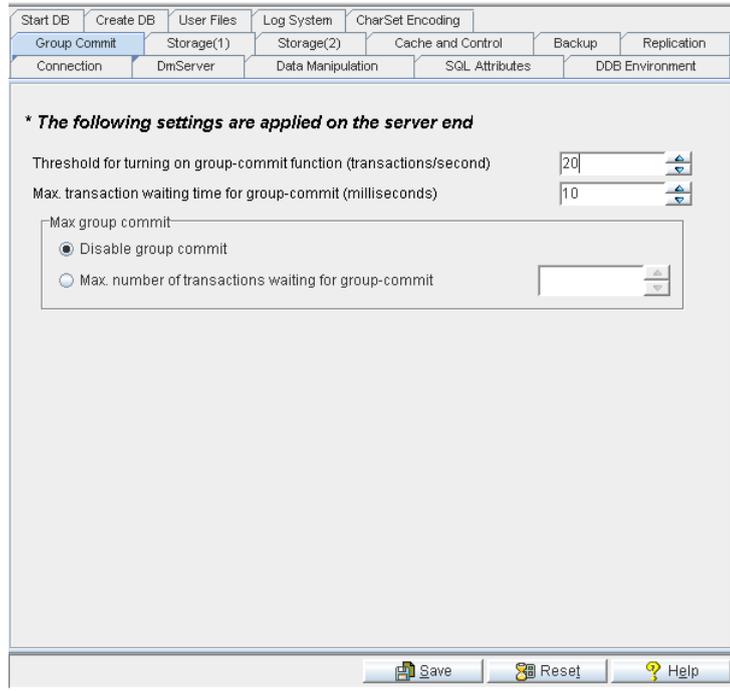


Figure 4-9 I/O Server page of the Start Database Advanced Settings window

GROUP COMMIT THRESHOLD

The group-commit function is only active when a certain threshold number of transactions per second are exceeded. The value in the **Threshold for turning on group-commit function (transactions / second)** field determines the activation threshold.

MAXIMUM TRANSACTION WAITING TIME

Each transaction will wait for a specified interval before automatically being committed, independent on the number of other transactions waiting for a group-commit. The value in the **max transaction waiting time for group-commit (milliseconds)** field indicates to the database the longest time interval any one transaction should wait. The default value is **30** milliseconds.

COMPLETELY DISABLE GROUP COMMIT

Selecting this check box disables the group commit function.

MAXIMUM NUMBER OF TRANSACTIONS IN WAIT STATE

Transactions waiting for a group commit will wait until the maximum transaction waiting time before the group-commit. However, if a certain number of transactions waiting for a group-commit are reached, then the group-commit will occur regardless of the waiting time. This number of transactions is specified in the **Max. number of transactions waiting for group-commit** field. The default value is *0*.

Setting group commit options:

1. Click **Setup** in the Start Database window.
2. Click the **Group Commit** tab in the Start Database Advanced Settings window.
3. Enter a value in the **Threshold for turning on group-commit function (transactions / second)** field.
4. Enter a value in the **max transaction waiting time for group-commit (milliseconds)** field.
5. Select **Completely disable group commit** option button or enter a value in the **Max. number of transactions waiting for group-commit** field.
6. Click **Save**.
7. Click **Cancel** to return to the Create Database window, and click **Start** to start the database.

4.3 Using the Start Database Wizard

The Start Database Wizard helps users define important database configuration parameters that must be set before a database is started. The Start Database Wizard follows a systematic process, however, selecting different parameters results in a different procedure.

The following chart outlines the program logic for the Start Database Wizard. Refer to the separate steps in the procedure that follow the chart for a detailed description of its function.

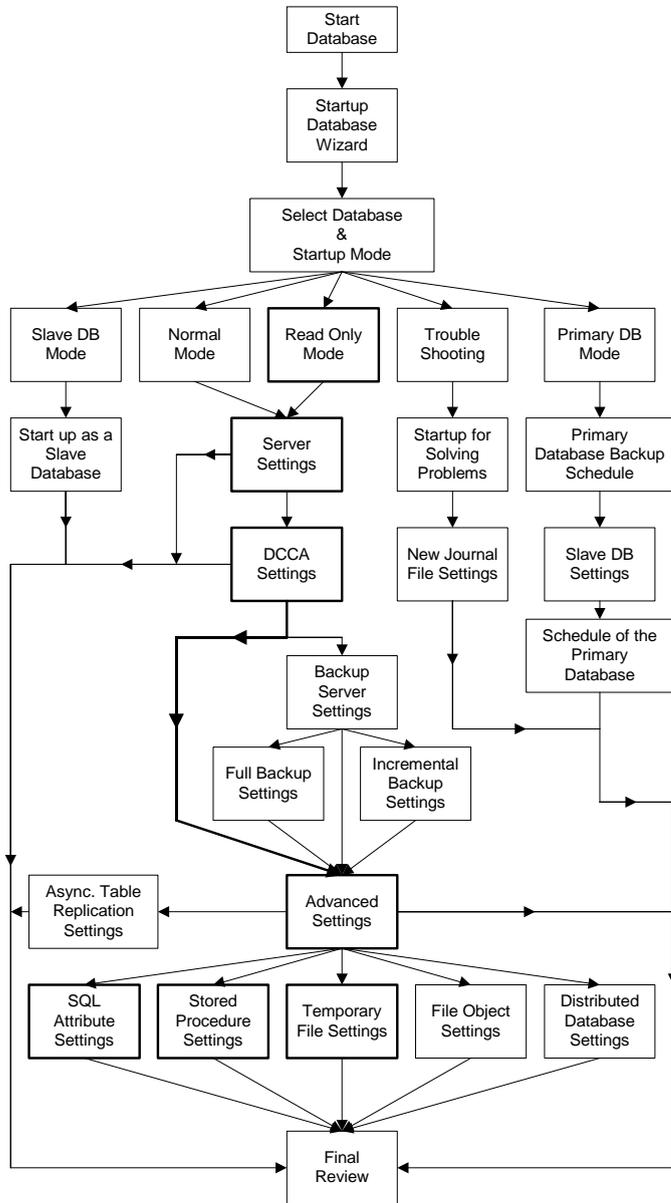
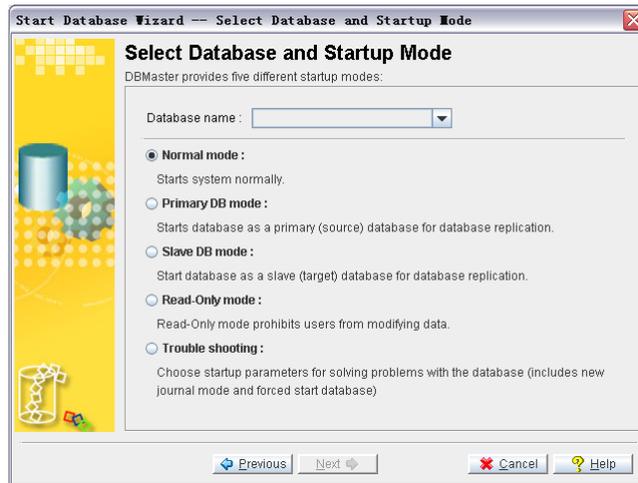


Figure 4-10 Program Logic for the Start Database Wizard

- Starting a database using the Start Database Wizard
 - Select **Start Database** from the **Wizard** item in the menu bar. The **Welcome to Start Database Wizard** window appears.
 - Click the **Next** button. The **Select Database and Startup Mode** window appears. The following choices are available:



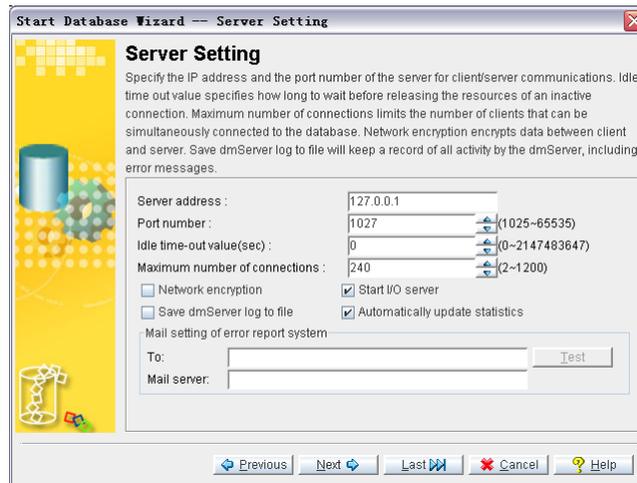
- Normal Mode: Refer to Normal or Read Only Mode Startup if this option is chosen.
 - Primary DB Mode: Refer to Primary DB Mode Startup if this option is chosen.
 - Slave DB Mode: Refer to Slave DB Mode Startup if this option is chosen.
 - Read Only Mode: Refer to Normal or Read Only Mode Startup if this option is chosen.
 - Trouble Shooting: Refer to Trouble Shooting Startup if this option is chosen.
- After choosing one of the startup modes, click the **Next** button. Refer to the following section according to the startup mode selection.

Normal or Read Only Mode Startup

Starting the database in normal mode means that the database is not the source or target for database replication, and is intended for normal use. The start database wizard allows advanced settings to be changed before database startup.

Read-only mode starts the database normally, except that no data may be modified.

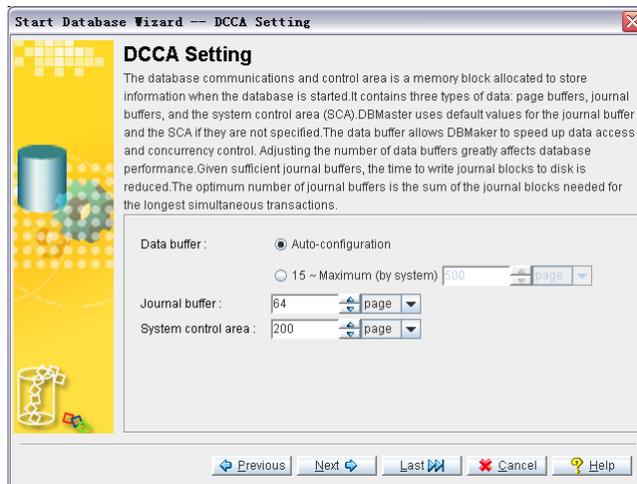
- Starting a database in normal or read-only mode
 1. Start the database in **Normal mode** or **Read Only mode** from the **Select Database and Startup Mode** window.
 2. The **Server Setting** window appears.



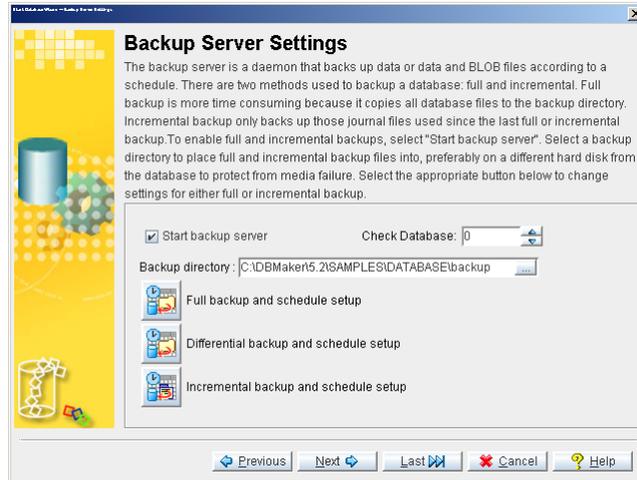
NOTE At any point during the wizard, click **Last** to skip to the last step of the wizard (the **Final Review** window). Refer to step 4 in this procedure.

3. Enter the IP address or host name of the server in the **Server Address** field.
4. Enter the port number of the server in the **Server Port Number** field. This must be an integer between 1025 and 65535.
5. Enter the idle time out value of the server in the **Idle time-out value** field.
6. Enter a value for the maximum number of connections to the database in the **Maximum number of connections** field.

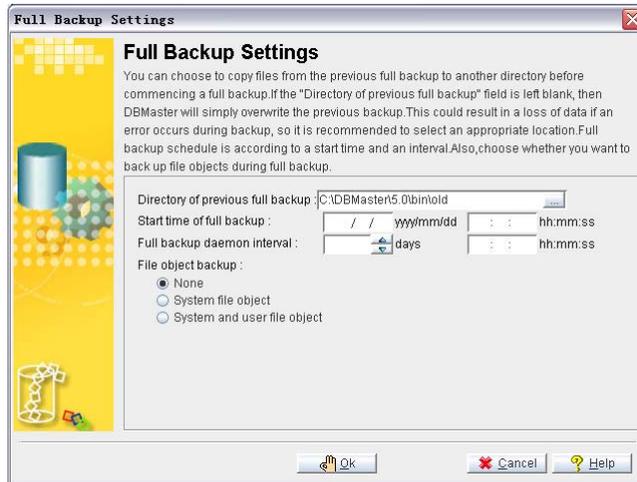
7. Check the **Network encryption** check box to allow for network encryption.
8. Check the **Save dmServer log to file** check box to save the dmServer log to file.
9. Check the **Start I/O server** check box to start the I/O server for the database.
10. Check the **Automatically update statistics** check box to have the database's statistics updated automatically.
11. **Mail setting of error report system.**
12. Click the **Next** button. The **DCCA Setting** window appears.



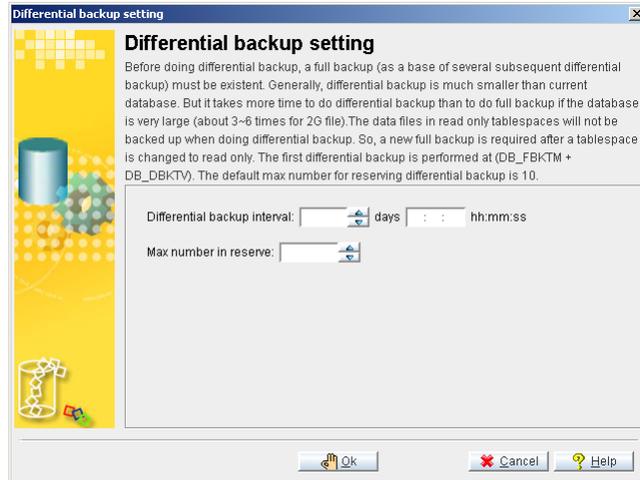
13. Enter the number of data buffer pages to allocate from memory in the **Data buffer (page)** field or select **Auto-configuration** to have DBMaker specify the value.
14. Enter the number of journal buffer pages to allocate from memory in the **Journal buffer (page)** field.
15. Enter the number of pages to allocate from memory for the System Control Area in the **System control area (page)** field.
16. Click the **Next** button. The **Backup Server Settings** window appears.



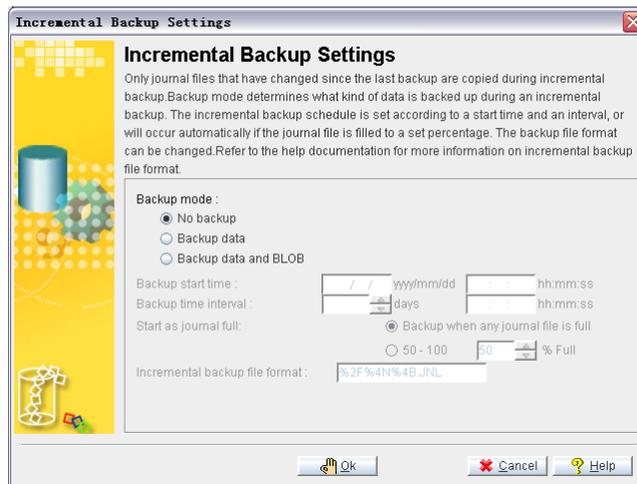
17. To enable backups, check the **Start backup server** check box. The **Backup directory** field, the **check database** field, the **Full backup and schedule setup**, the **differential backup and schedule setup** and the **Incremental backup and schedule setup** buttons become active.
18. Enter the port number of the server in the **Check Database** field. This must be 0,1 or 2.
19. Enter a path into or select the browse button  next to the **Backup directory** field for the location of the backup data.
20. Click the **Full backup and schedule setup** button to setup the full backup settings. The **Full Backup Settings** window appears.



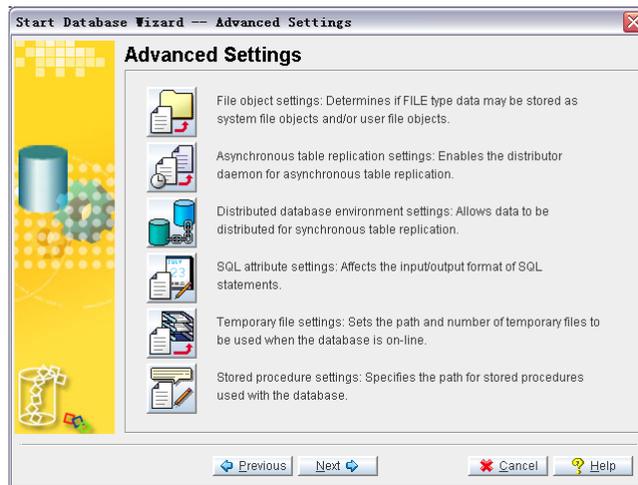
21. Enter a path into or select the browse button  next to the **Directory of previous full backup** field for the location of the data from a previous backup. If this field is left blank, DBMaker will overwrite the previous backup.
22. Enter a start time for the full backup in the **Start time of full backup** field.
23. Enter the interval for the full backup replication daemon to update in the **Full backup daemon interval** field.
24. Select whether system or user file objects are backed up.
25. Click the **OK** button. The **Backup Server Settings** window reappears.
26. Click the **Differential backup and schedule setup** button to setup the differential backup settings. The **Differential Backup Settings** window appears.



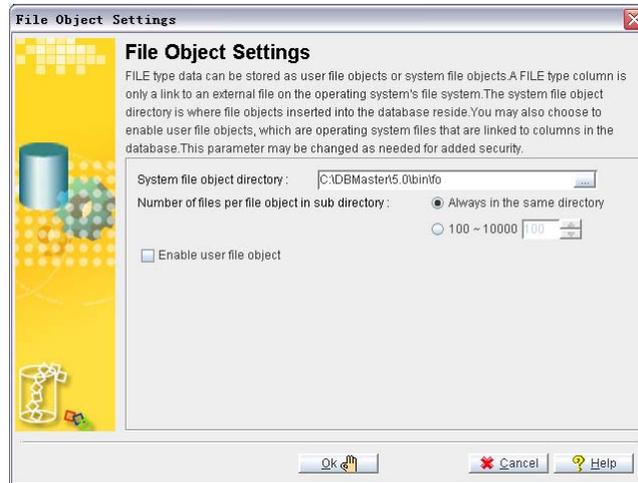
27. Enter the interval for the **differential backup** replication daemon to update in the **Differential backup daemon interval** field.
28. Enter a number in the **Max number in reserve** field.
29. Click the **OK** button. The **Backup Server Settings** window reappears.
30. Click the **Incremental backup and schedule setup** button to setup the full backup settings. The **Incremental Backup Settings** window appears.



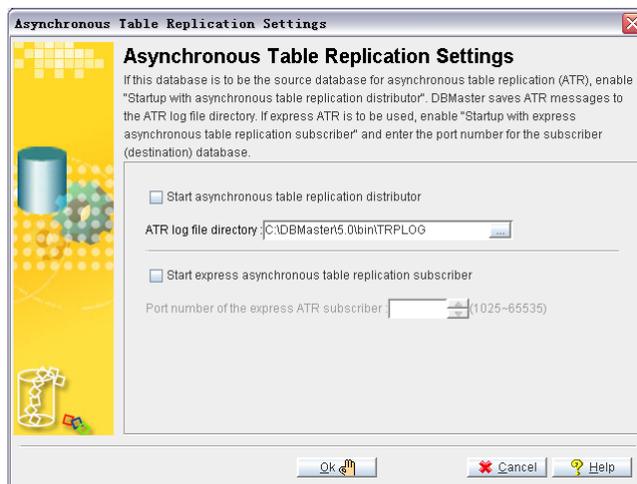
31. To enable incremental backups check the **Backup data** or the **Backup data and BLOB** check box.
32. Enter a start time for the backup in the **Backup start time** field.
33. Enter the interval for the incremental backup replication daemon to update in the **Backup time interval** field.
34. Enter a value for the journal files to be backed up in the **Start incremental backup when journal file __% Full**, or check the **Backup when any journal file is full** check box.
35. Enter a file format for the backup data in the **Incremental backup file format** field.
36. Click the **OK** button. The **Backup Server Settings** window reappears.
37. Click the **Next** button. The **Advanced Settings** window appears.



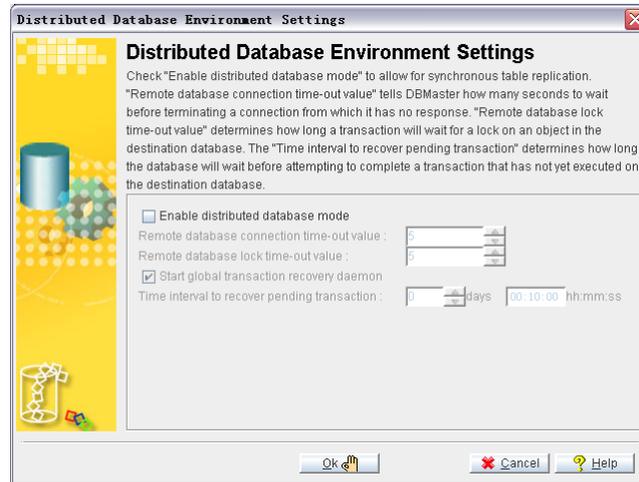
38. To set file object options click the File object settings button. The **File Object Settings** window appears.



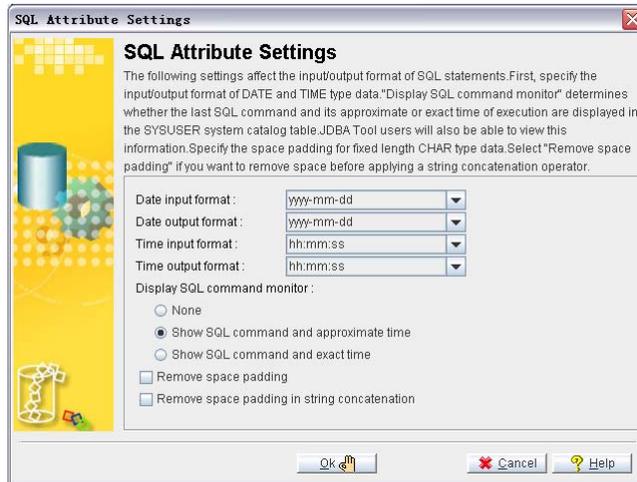
- a) Enter a path into or select the browse button  next to the **System file object directory** field for the location of the system file objects.
 - b) Enter a value for the number of files per file object in the file object directory in the **Number of files per file object in sub directory** or check the **Always in the same directory** check box.
 - c) Check the **Enable user file object** check box to enable the use of user file objects.
 - d) Click the **OK** button. The **Advanced Settings** window reappears.
39. To set up asynchronous table replication settings click the Asynchronous table settings button. The **Asynchronous Table Settings** window appears.



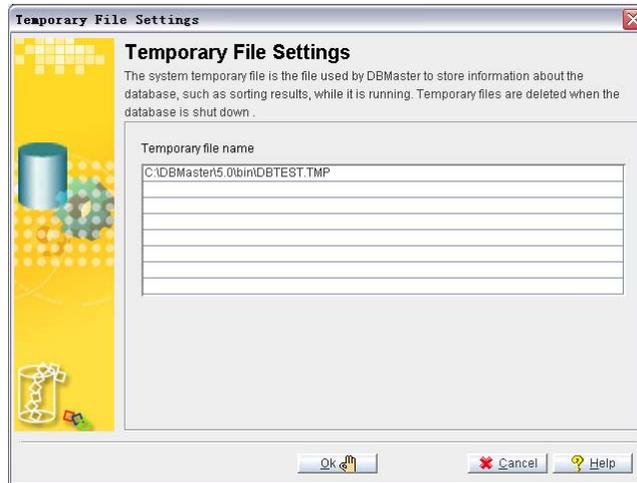
- a) Check the **Start asynchronous table replication distributor** check box to start asynchronous table replication.
 - b) Enter a path into or select the browse button  next to the **ATR log file directory** field for the location of the ATR log files.
 - c) Check the **Start express asynchronous table replication subscriber** check box to start express asynchronous table replication. The **Port number of the express ATR subscriber** field becomes active.
 - d) Enter a port number in the **Port number of the express ATR subscriber** field.
 - e) Click the **OK** button. The **Advanced Settings** window reappears.
40. To set the distributed database settings click the **Distributed database environment settings** button. The **Distributed database environment settings** window appears.



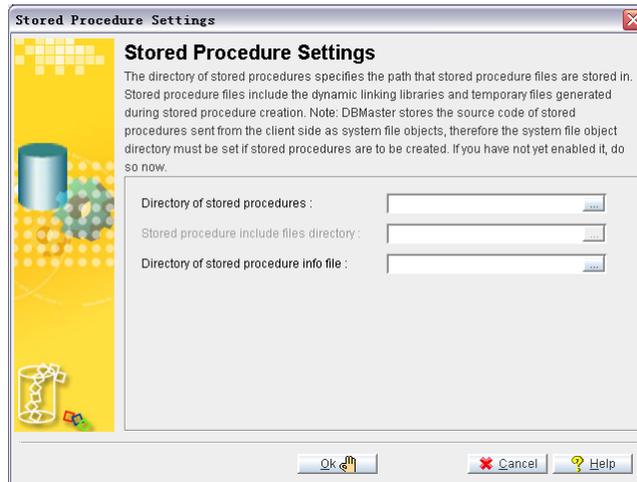
41. To enable DBMaker to function as a distributed database check the **Enable distributed database mode** check box. The remaining options in the window become active.
 - a) Set a time-out value in the Remote database connection time-out value field.
 - b) Set a lock time-out value in the Remote database lock time-out field.
 - c) Check the Start global transaction recovery daemon check box to start the daemon.
 - d) Set a time interval to recover pending transactions in the Time interval to recover pending transactions fields.
 - e) Click the OK button. The Advanced Settings window reappears.
42. To set SQL attribute settings click the **SQL attribute settings** button. The **SQL Attributes Settings** window appears.



- a) Select an input date format from the **Date input format** drop down list.
 - b) Select an output date format from the **Date output format** drop down list.
 - c) Select an input time format from the **Time input format** drop down list.
 - d) Select an output time format from the **Time output format** drop down list.
 - e) Select the whether the last SQL command and its time of execution are displayed in the system catalog table from the **Display SQL command monitor** check boxes.
 - f) Select **Remove space padding for CHAR data** to remove space padding before a CHAR type data.
 - g) Select **Remove space padding in string concatenation** to remove space padding before a string concatenation.
43. Click the **OK** button. The **Advanced Settings** window reappears.
 44. To set temporary file settings click the **Temporary file settings** button. The Temporary File Settings window appears.

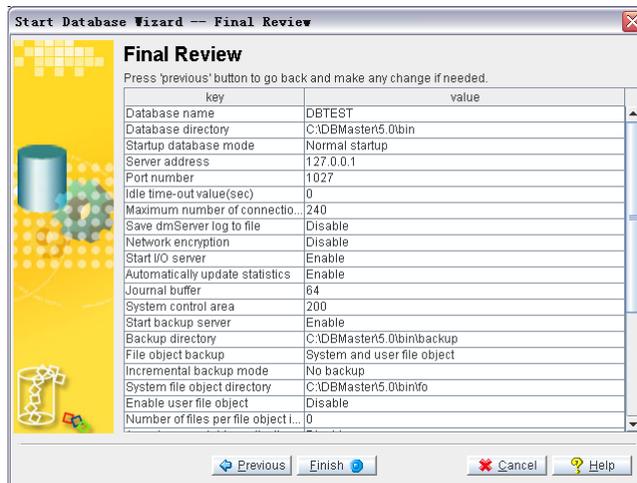


- a) Double-click a field to enter the temporary file name.
 - b) Enter the temporary file's name.
 - c) Click the **OK** button. The **Advanced Settings** window reappears.
45. To set stored procedure options click the **Stored procedure settings** button. The **Stored Procedure Settings** window appears.



- a) Enter a path into or select the browse button  next to the **Directory of stored procedures** field for the location of stored procedures.

- b) Enter a path into or select the browse button  next to the **Stored procedures include files directory** field for the location of the ATR log files.
 - c) Enter a path into or select the browse button  next to the **Directory of stored procedure info file** field for the location of information files for stored procedures.
 - d) Click the **OK** button. The **Advanced Settings** window reappears.
46. Click the **Next** button. The **Final Review** window appears.



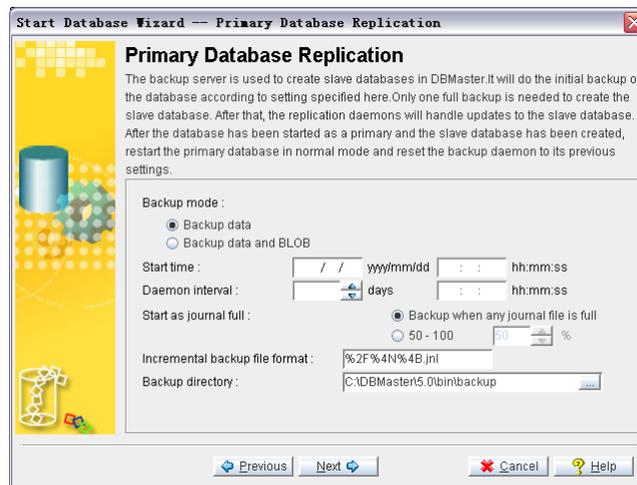
47. Check over the database settings. If any of the settings need to be changed, click the **Previous** button to the appropriate window and make the changes. Then return to the **Final Review** window.
48. Click the **Finish** button when satisfied with the database settings.

Primary DB Mode Startup

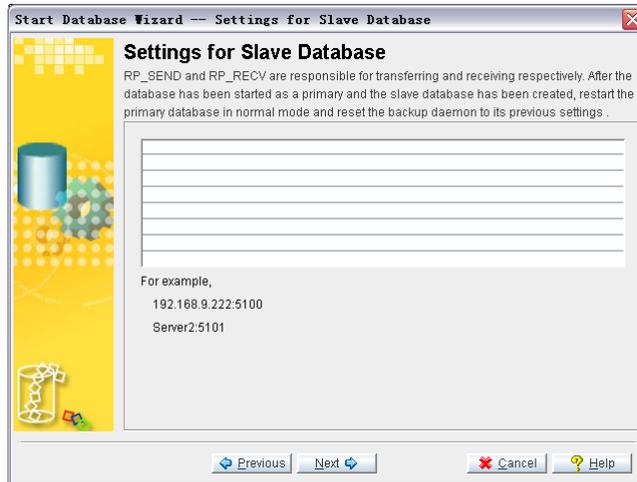
The **Database Startup Wizard** simplifies slave database creation. A slave database is initially created from the primary database by choosing Primary DB Mode startup. The Database Startup wizard backs up the databases to another location and in the process creates the slave database. After the slave database is created, the primary database should be restarted in normal mode. The primary

database is still referred to as a primary database, but it does not need to be started in Primary DB mode.

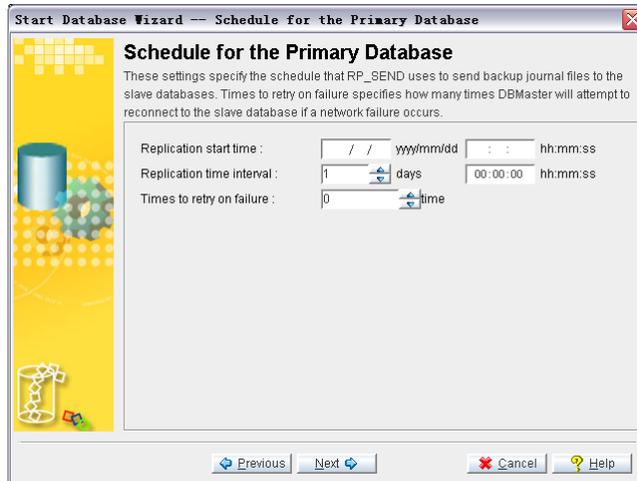
- ➔ Starting a database in primary DB mode:
 1. Start the database in **Primary DB Mode** from the **Select Database and Startup Mode** window.
 2. The **Primary Database Replication** window appears. Set the backup database options.



3. Select the **Backup data** check box to backup standard data only or select the **Backup data and BLOB** check box to backup standard and BLOB data to the slave database.
4. Set the backup database's start time in the **Start time** fields.
5. Set the replication daemons' update schedule in the **Daemon interval** fields.
6. Select the **Backup when any journal file is full** check box to have the journal files backed up when any of the journal files are full, or set the percentage of a journal file being full before it is backed up.
7. Set the backup file format in the **Incremental backup file format** field.
8. Enter a path into or select the browse button  next to the **Backup directory** field for the location of journal files sent by the backup database.
9. Click the **Next** button. The **Settings for Slave Database** window appears.

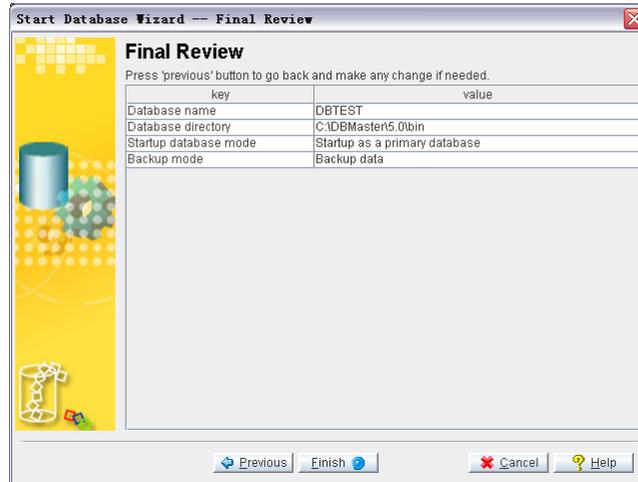


10. Set the port number and IP address for the client database.
11. Click the **Next** button. The **Schedule for Primary Database** window appears.



12. Enter a start time for the schedule in the **Replication start time** fields.
13. Set the time interval between replications in the **Replication time interval** fields.

14. Set the number of times that the primary database will attempt to reconnect to the slave database in the **Times to retry on failure** field.
15. Click the **Next** button. The **Final Review** window appears.



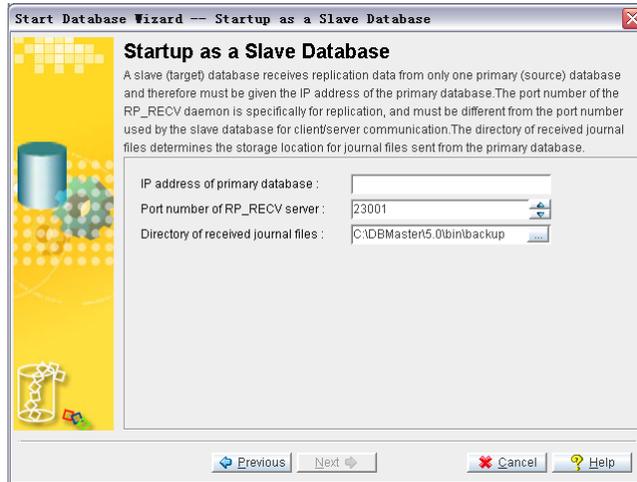
16. Check over the database settings. If any of the settings need to be changed, click the **Previous** button to the appropriate window and make the changes. Then return to the **Final Review** window.
17. Click the **Finish** button when satisfied with the database settings.

Slave DB Mode Startup

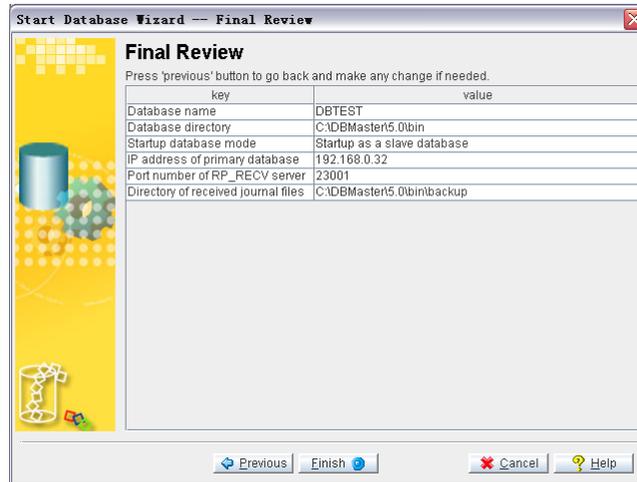
Slave DB Mode startup is used when the database is the target of database replication. A slave database must be a replica of the source database, and is initially created by backing up a database to a remote location. After creating the slave database, it must be started as a slave database to enable it to receive updates from the primary database. Slave databases are read-only. If a database created normally is started as a slave database, it starts as a read-only database and an error occurs when the database attempts to update tables from the source database.

- ➔ Starting a database in slave database mode:
 1. Start the database in **Slave Database Mode** from the **Select Database and Startup Mode** window.

2. The **Startup as a Slave Database** window appears.



3. Enter the IP address of the server in the **IP address of primary database** field.
4. Enter the port number of the RP_RECV daemon in the **Port number of the RP_RECV server** field.
5. Enter a path into or select the browse button  next to the **Directory of received journal files** field for the location of journal files sent by the primary database.
6. Click the **Next** button. The **Final Review** window appears.

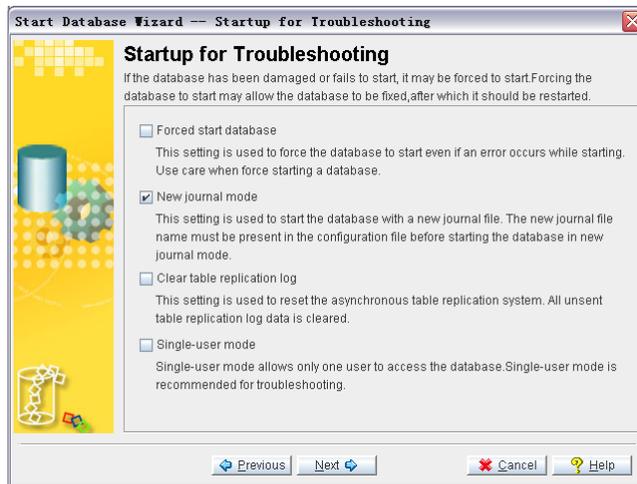


7. Check over the database settings. If any of the settings need to be changed, click the **Previous** button to the appropriate window and make the changes. Then return to the **Final Review** window.
8. Click the **Finish** button when satisfied with the database settings.

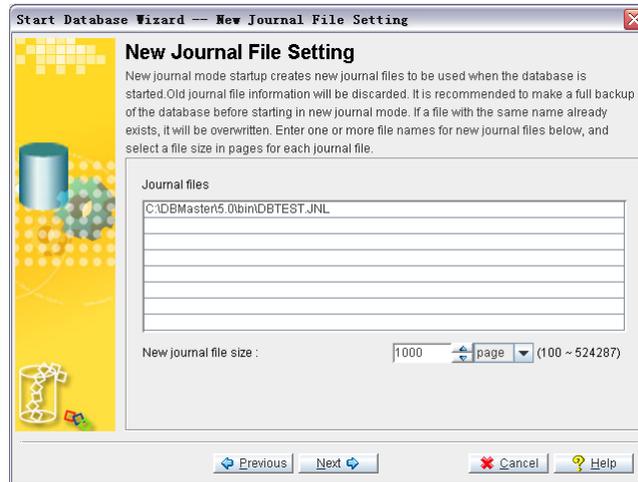
Trouble Shooting Startup

Trouble Shooting Startup is useful when you are having trouble starting a database, and can sometimes be used to diagnose and fix problems within the database

- Starting a database in trouble shooting mode:
 1. Start the database in **Trouble Shooting** mode from the **Select Database and Startup Mode** window.
 2. The **Startup for Trouble Shooting** window appears.



3. To force the database to startup in single user mode select the **Single user mode** check box.
4. To force start the database in normal mode, select the **Forced start database** check box.
5. Check the **Clear table replication log** check box to clear the asynchronous table replication log.
6. To start the database with a new journal file, select the **New journal mode** check box.
7. Click the **Next** button. If the **New journal mode** check box is checked, go to step 7. If the New journal mode check box is not checked, go to step 10.
8. The **New Journal File Setting** window appears.



9. Enter the file name and path of the new journal files in the **Journal files** field.
10. Enter the size of the journal files in the **New journal file size (pages)** field. Valid values are 100 to 524287.
11. Click the **Next** button. The **Final Review** window appears.



12. Check over the database settings. If any of the settings need to be changed, click the **Previous** button to the appropriate window and make the changes. Then return to the **Final Review** window.

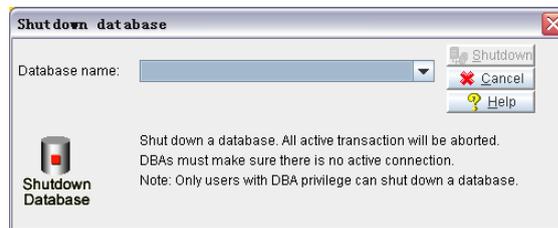
13. Click the **Finish** button when satisfied with the database settings.

5 Shutting Down a Database

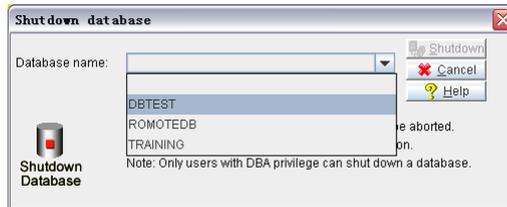
JServer Manager can be used to shut down a database. All existing connections to a database are terminated when the database is shut down. Any user connected to that database at the time of database shutdown is notified of the shutdown via an error message. You should be sure that all users are disconnected and that there are no pending distributed transactions to the database before shutting it down. The status of pending transactions and connections may be checked using the monitor feature in the JDBA Tool. Please refer to the *JDBA User's Guide* for more information.

Shutting down a database

1. To shut down a database, select Shut down database from the main console or from the Database drop-down menu. The **Shutdown Database** window appears.



- To select a database, click on the arrow on the Database Name field. A menu listing all databases on the server appears.



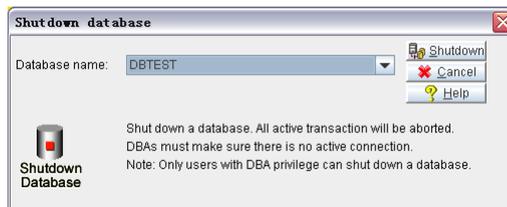
- Select the database to be stopped (i.e., shutdown) from the Database Name menu. The **Login** dialog box appears prompting the user to log on to the database.



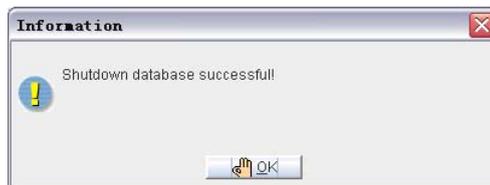
- Enter your user ID in the User ID field.

NOTE *You must have DBA privilege to shut down the database.*

- Click the OK button. The Shut down database window appears. The name of the database that is to be shut down appears in the Database Name field.



- Click the **Shutdown** button to shut down the database. All connections to the database are terminated.



- Click the OK button.

6 Using an NT Service

Installing an NT service allows a database server to start automatically the next time the user logs into Windows NT. Installing the database server as an NT service allows the database administrator to log out of Windows NT while the database server continues to operate (i.e., the process remains running). Therefore, all connections between the user and client are preserved. If a database is not installed as a Windows NT, then all connections are terminated when the database administrator logs out of Windows.

To install a database as an NT service, you must log on to Windows NT with a user account that has administrator security privileges or with the Administrator account itself. Additionally, you must have SYSADM security privileges for the database.

6.1 Adding an NT Service

You can add new NT services for a database by selecting the database and choosing a start mode. The following start mode options are available.

- **Auto Started:** In most cases, select this start mode when installing the service. This allows the database server to start immediately after the NT system boots-up the next time you enter the system.
- **Demand Started:** In some instances you might want to select this as the start mode when installing the service. For example, if the database crashed and the database server must be force started. If demand started is selected, the user must manually start the service from within Windows NT, Windows 2000, or Windows XP. In this case, no users should connect to the server until it is properly started as this could potentially damage the database.

➔ Adding an NT service to a database:

1. Select NT Service from the Database drop down menu. The NT Service dialog box appears.



2. Click the Install button. The Add NT Service dialog box appears.



3. Select a database from the Database to be started as NT service menu.
4. Select the database to start an NT Service for from the Database to be started as NT service menu.
5. To start the database in Auto Started mode, select the Auto Started radio button.
6. To start the database in Demand Started mode, select the Demand Started radio button.
7. Click the OK button. The NT Service window appears. The database where the service is installed is displayed in the Installed Services field.



6.2 Viewing Installed NT Services

You can view all services installed for the database. This function is useful for tracking the different services that DBMaker creates within Windows NT, Windows 2000, or Windows XP.

- ➔ Viewing installed NT services:
1. Select **NT Service** from the Database drop down menu. The NT Service dialog box appears. Installed services appear in the Installed Services list.



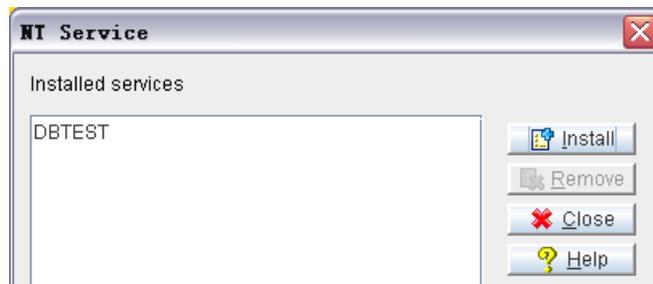
NOTE *You can also view installed NT services by selecting the Administrative Tools\Services icon in Windows 2000 or the Services icon in Windows NT.*

6.3 Removing NT Services

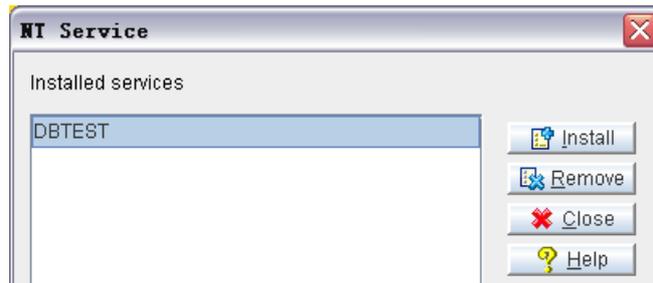
If an NT service is no longer required for a database, you can remove it from the database.

➔ Removing NT service for a database

1. Select NT Service from the Database drop down menu. The NT Service window appears.



2. Select a database from the Installed Services list box.

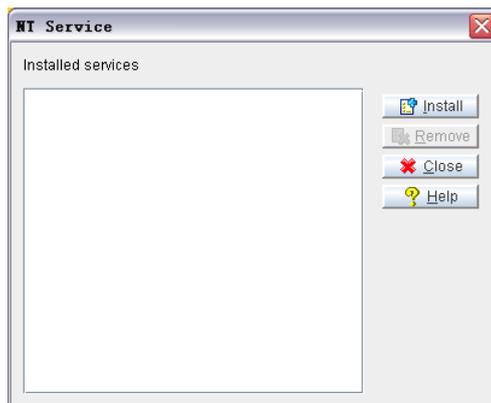


3. Click the Remove button. The Delete Database Service dialog box appears to confirm your choice.



4. Click the OK button to remove the database service. Selecting **OK** removes the database from NT service. The removed database no longer appears in the NT **Service** window.

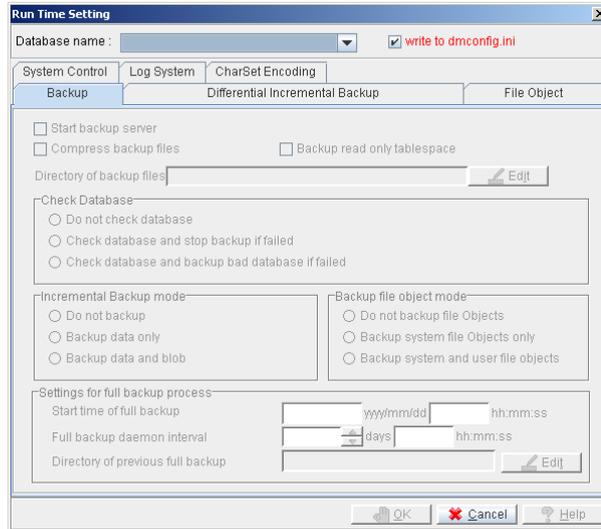
NOTE *If the selected database still appears on the list, close the NT Service window and reopen it. The Installed Services list is refreshed and will no longer show the removed database.*



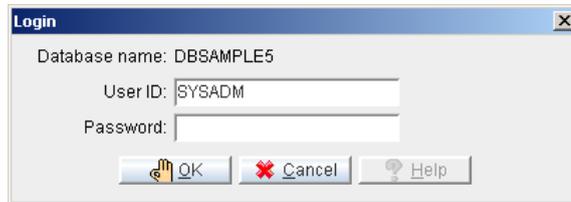
7 **Altering Run Time Settings**

Certain database configuration parameters may be changed while the database is on-line (i.e., started or running). These settings include backup settings (full, differential and incremental backup frequency and location), file object settings (enable user file objects, system file object location), system control settings, log system settings and character set encoding.

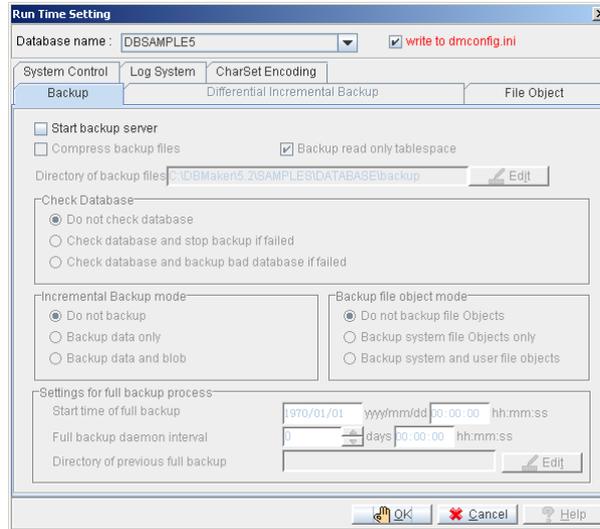
- Selecting a database for run-time setting changes:
 1. Select Run Time Setting from the database drop down menu.
 2. The Run Time Setting window appears.



3. Select a database from the Database Name menu.
4. The Login dialog box appears.



5. Click the OK button. The database you logged into appears in the Database Name field of the Run Time Setting dialog box.



6. To use the updated settings in the next session, make sure that the **write to dmconfig.ini** check box is selected.
7. 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.*

8. Set the following options:
 - Backup Settings
 - Differential Incremental Backup Setting
 - File Object Settings
 - System Control
 - Log System
 - CharSet Encoding

7.1 Changing Backup Settings

Backup settings that may be altered during run-time include the following: check database, incremental backup mode, file object backup mode, backup location, deactivation of backup BLOB, and deactivation of backup data, and full backup process setting. During runtime, backup modes may only be changed to a state where less data is backed up (except in the case of file objects). This may be useful, for example, if the database administrator wants to free up resources by changing the backup mode from “backup data and BLOB” to “backup data”

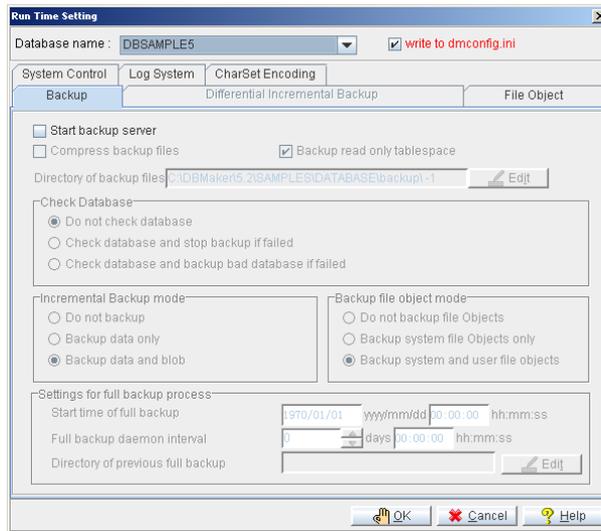


Figure 7-1 Backup page of the Run-Time Settings window

Start Backup Server

Enabling the **Start Backup Server** check box activates the Backup Server demaon. So user can access setting in the **Backup** tab and **Differential Incremental Backup** tab except the **incremental backup mode** box, user can

dynamic alter the backup server setting during run-time. Otherwise, all components are disabled about **Backup** tab and **Differential Incremental Backup** tab.

Compress Backup files

Selecting the **Compress Backup files** activates the backup file compression. And ensures backup files will be compressed by the Backup Server when performing a full backup.

Backup Read only tablespace

Backup server can backup the read-only tablespace files when performing a full backup. Select this check box will backup the read-only tablespace files. Clearing this check box will not backup the read-only tablespace files. Please note, a full backup must be performed after setting a tablespace to read-only tablespace. If you don't backup the latest files, serious errors may result when restoring the full backup database containing the read-only tablespace.

Directory of backup files

The backup directory is the directory where the backup server stores all backup (i.e., journal) files. You should create a backup directory on a different disk from where the database files reside to prevent a loss of both the database and the backup files in the event of a media error. The default path for backup files is the database directory */backup* and is automatically created by DBMaker. The total length of the backup directory path must not exceed 255 characters.

Check Database

You can choose whether check database before full backup and differential backup.

CHECK MODE	DESCRIPTION
Do not check database	Not check database before full backup and

CHECK MODE	DESCRIPTION
	differential backup.
Check database and stop backup if failed.	Check database before full backup and differential backup. Backup server will write error message and stop this backup if find the database has been damaged.
Check database and backup bad database if failed.	Check database before full backup and differential backup. Backup server will write error message and continue to back up bad database to BKDIR/BADDB directory if find the database has been damaged. It will be done only once to back up bad database when backup server first find the damaged database. After then, if checking database is ok, backup server will remove bad backup resided in BKDIR/BADDB directory and continue normal backup, otherwise, only write error message and stop this backup.

Incremental backup files always follow the last full backup or differential backup. So, incremental backup file will be placed into BKDIR/BADDB after finding database has damaged, and they are switched back into BKDIR after database is ok.

Incremental Backup Mode

You can choose to turn off the incremental backup mode, or turn off only the BLOB backup 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 then make 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, the 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.

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.

Settings for full backup process

START TIME OF FULL BACKUP

To set the time in which the first full backup will begin for the database, enter the date in the yyyy/mm/dd field, as well as the time in the hh:mm:ss field. Reenter the numbers if they appear incorrectly the first time.

FULL BACKUP DAEMON INTERVAL

The number in the combo box specifies the time interval at which the Full Backup occurs in days. Next to the combo box is a field for time input, which specifies the time interval in hours, minutes, and seconds. The total time interval is determined by adding the two values together, so inputting 1 into the **days** field and inputting 12:00:00 into the **hh:mm:ss** field would cause a full backup to be made every day and a half.

DIRECTORY OF PREVIOUS FULL BACKUP

This Directory of previous full backup specifies the directories where the backup server puts the previous version of full backup files.

➔ Changing backup settings:

1. Select Start backup server check box.
2. To enable compress backup files, click on the **Compress** backup files check box.
3. To enable backup the read-only tablespace files when performing a full backup, click on the **Backup read-only tablespace** check box.
4. Enter a path into or select the browse button  **Edit** next to the **Directory of backup files** field to indicate a destination for the backup files.
5. Select a check mode from the **Check Database** box.
 - To not check database before full backup and differential backup, click the **Do not check database** option button.
 - To check database before full backup and differential backup, and backup server will write error message and stop this backup if find the database has been damaged. Click the **Check database and stop backup if failed** option button.
 - To check database before full backup and differential backup, Backup server will write error message and continue to back up bad database to BKDIR/BADDB directory if find the database has been damaged. Click the **Check database and backup bad database if failed.** option button.

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, click the **Do not backup** option button.
7. Select a file object backup mode from the **file object Backup mode** box.
 - To not backup file objects, select the **Do not backup file objects** option button.
 - To backup system file objects, click the **Backup system file objects only** option button.
 - To backup system and user file objects, click the **Backup system and user file objects** option button.
8. Setting a full backup process from the **Settings for full backup process** area.
 - Indicate a date and a time at which full backups are to begin in the **Start time of full backup** fields.
 - Enter the number of days, hours, minutes, and seconds between each successive full backup in the **Full backup daemon interval** time fields.
 - Enter a path into or select the browse button  **Edit** next to the **Directory of previous full backup** field to indicate a destination for the backup server puts the previous version of full backup files.
9. 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.

7.2 Changing Differential and incremental Backup Settings

The differential incremental backup settings page allows user setting for differential backup process and incremental backup process.

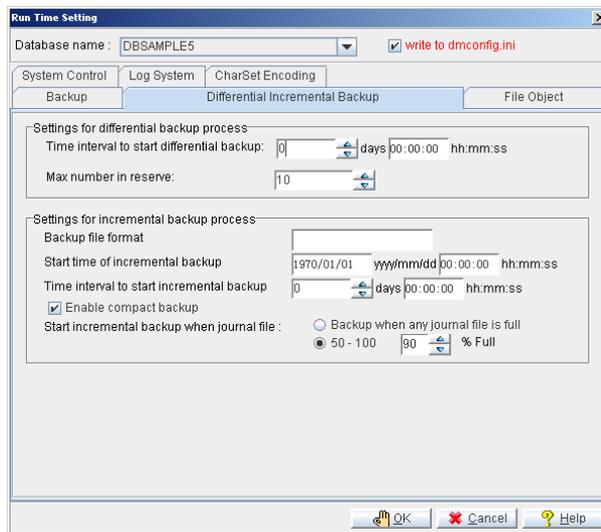


Figure 7-2 Differential Incremental Backup Settings window

Settings for differential backup process

TIME INTERVAL TO START DIFFERENTIAL BACKUP

The number in the combo box specifies the time interval at which the Differential Backup occurs in days. Next to the combo box is a field for time input, which specifies the time interval in hours, minutes, and seconds. The total time interval is determined by adding the two values together, so inputting 1 into the **days** field and inputting 12:00:00 into the **hh:mm:ss** field would cause a differential backup to be made every day and a half.

MAX NUMBER IN RESERVE

The number in the box specifies the max number of differential backup after a full backup (differential base). Backup server will remove the oldest differential backup if the number of differential backup after a full backup exceeds the specified max number. The default value is 10. Value 0 denotes not to use this function, in other words, the number of differential backup is unlimited.

Settings for incremental backup process

BACKUP FILE FORMAT

The input box specifies the format Backup Server used to name 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

To set the time in which the first incremental backup will begin for the database, enter the date in the yyyy/mm/dd field, as well as the time in the hh:mm:ss field. Reenter the numbers if they appear incorrectly the first time.

TIME INTERVAL TO START INCREMENTAL BACKUP

The number in the combo box specifies the time interval at which the Incremental Backup occurs in days. Next to the combo box is a field for time input, which specifies the time interval in hours, minutes, and seconds. The total time interval is determined by adding the two values together, so inputting 1 into the **days** field and inputting 12:00:00 into the **hh:mm:ss** field would cause a differential backup to be made every day and a half.

ENABLE COMPACT BACKUP

Enable Compact Backup Mode ensures that only needed journal files are copied by the Backup Server when performing an incremental backup. As a result, backup storage space is saved. However, restoring a database may take more time.

STARTING INCREMENTAL BACKUPS VIA A JOURNAL TRIGGER

The user may want DBMaker to create an incremental backup before the journal file is completely full. The journal trigger value specifies the percentage a journal file must fill before the Backup Server performs an on-line incremental backup. You can specify a percentage between 50 and 100 for the journal trigger value. You can combine the journal trigger value with the backup schedule to backup your database on a regular schedule.

- ➔ Changing differential and incremental backup settings:
1. Setting 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 max value in the **Max number in reserve** fields.
 2. Setting a incremental backup process from the **Settings for incremental backup process** area.
 - Enter a format Backup Server used to name incremental backup journal files.
 - Indicate a date and a time at which incremental backups are to begin in the **Start time of incremental backup** fields.
 - 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 backup, click on the **Enable Compact Backup** check box.
 4. Incremental backups can be set to automatically execute when journal files have filled to a set percentage. 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 completely filled.

Altering Run Time Settings

- Enter a value from 50 to 100 in the 50 – 100 % field to set incremental backups to execute when any journal file is full to the value entered.

7.3 Changing File Object Settings

Selecting the **File Object** tab in the **Run Time Settings** window allows the user to enable external user file objects and change the storage location of system file objects. DBMaker can automatically create subdirectories within the FO directory. Each subdirectory is filled with new file objects up to a threshold value. When the threshold is reached, DBMaker creates a new FO subdirectory.

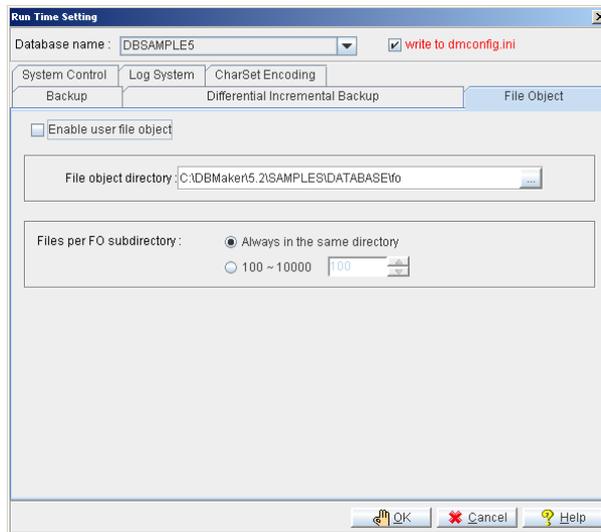


Figure 7-3 File Object page of the Run Time Settings window

- 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 a threshold value, between 100 and 10000, for the number of file objects 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 the OK button from the bottom of the Run Time Settings window.

7.4 Changing System Control Settings

The system control tab allows settings to be changed that directly affect database performance. It also allows enabling of distributed mode during runtime and allows for the number of pages to extend a file when an autoextend tablespace is full.

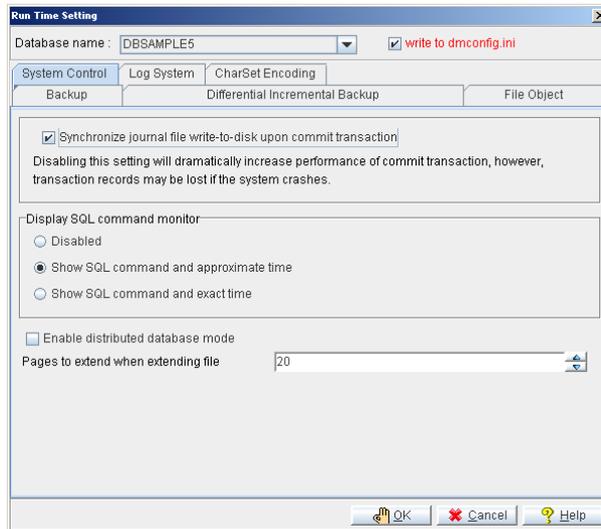


Figure 7-4 System Control page of the Run Time Settings window

Journal File Synchronization

Under normal operating conditions, when a transaction is committed, DBMaker initiates a two-step process for the journal blocks involved in the transaction. The first step flushes the relevant journal blocks in the operating system's file system, or the operating system's cache. The second step preserves the transaction by forcing the operating system to write the journal blocks to disk. In the event of a system crash, DBMaker's two-step process protects the journal blocks.

Performance of the commit transaction process improves significantly when the second step of this process is omitted. The journal blocks, in this case, remain in

the operating system's cache until the operating system writes them to disk. However, if the system crashes at this point, the records of any transactions that were still in the operating system's cache are 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 are displayed. No SQL commands are shown if Disabled is selected. The most recent SQL command executed by the user and its approximate time of execution are shown if Show SQL command and approximate time is clicked. The most recent SQL command executed by the user and its exact time of execution are shown if Show SQL command and exact time is clicked. Displaying the exact time of execution uses more CPU resources and slows down the database. The default mode is *Show SQL command and approximate time*.

Enable Distributed Database Mode

This setting enables the database for use in distributed mode. It must be enabled for synchronous table replication to function. Synchronous table replication is set using the JDBA Tool. For more information on distributed data, synchronous table replication, and coordinator and participant databases, please refer to the *JDBA Tool User's Guide* and 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 continue 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. If the database administrator expects that the database will grow very quickly, then a large number reduces the frequency of file appending. 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.

7.5 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 here are only applied to the server.

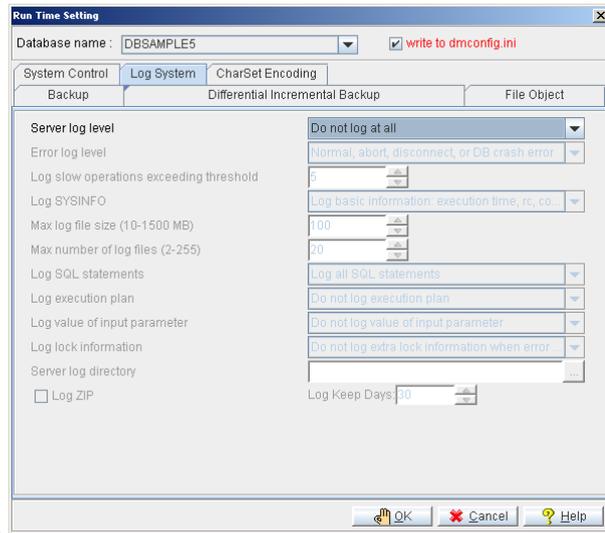
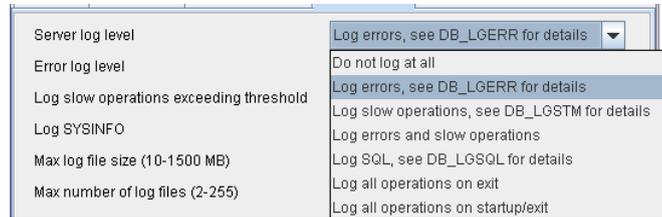


Figure 7-5 System Control page of the Run Time Settings window

Server Log Level

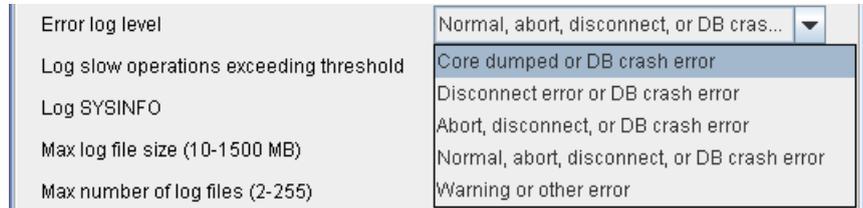
Users specify when and what operations are logged according to set the Server log level. DBMaker provides seven log levels as follows:



- **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 and 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 are logged according to the five levels described here:



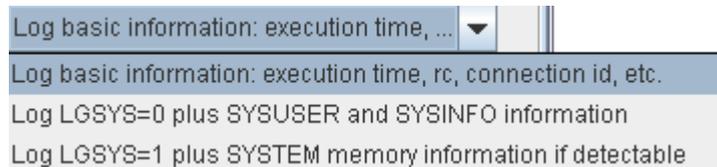
- **Core dumped or DB crash error:** logs all errors that occur having error codes > 30000
- **Disconnect error or DB crash error:** logs all errors that occur having error codes > 20000
- **Abort, disconnect or DB crash error:** logs all errors that occur having error codes > 10000
- **Normal, abort, disconnect or DB crash error:** logs all errors that occur having error codes > 100

Logging Slow Statements

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

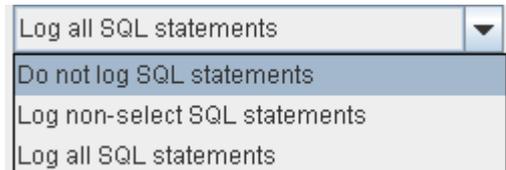
This setting specifies, in megabytes, the maximum file size for log and text files. DBMaker stores the log as DBNAME_1.LOG. When the log file size reaches the specified value, subsequent logs are generated as follows: DBSAMPLE5_2.LOG, DBSAMPLE5_3.LOG, ..., DBSAMPLE5_n.LOG, where n is the value specified by DB_LGFNO or when DB_LGFNO is unspecified the default value of 20 is used. Log files are overwritten 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_LGF SZ, 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 Statements

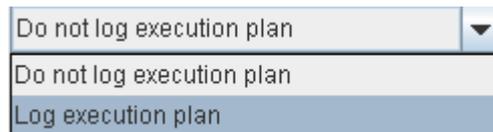
Users can specify what system information will be logged using **Log SQL** when server logging is enabled for **Log SQL, detail in DB_LGSQL (DB_LGSRV = 4)**. There are three SQL logging settings for recording different levels of information detail:



- **Do not log SQL statement**
- **Log any non-select SQL statement**
- **Log all SQL statement**

Log Execution Plan

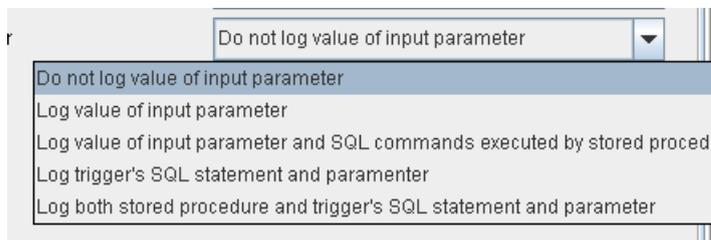
This setting specifies whether to log the execution plan for select, update, and delete statements. There are two logging options:



- **Do not log execution plan**
- **Log execution plan**

Log Input Parameter

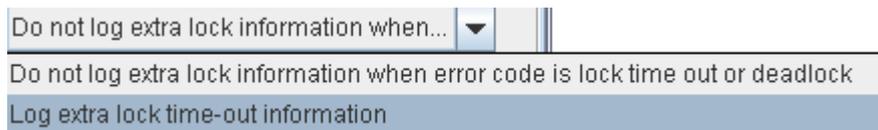
User can specify whether to log the input parameter's value. DBMaker provides five logging options each offering a different amount of information detail:



- **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:



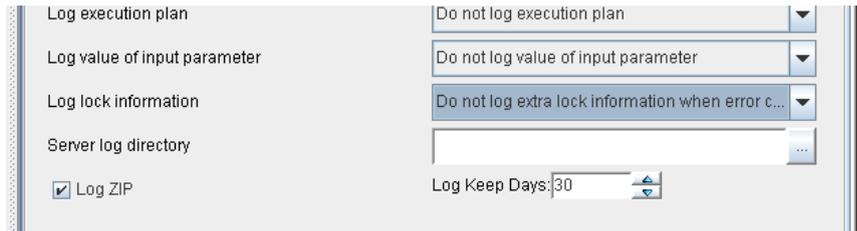
- **Do not log extra lock information when error code is lock time out or deadlock**
- **Log extra lock time out information**

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.

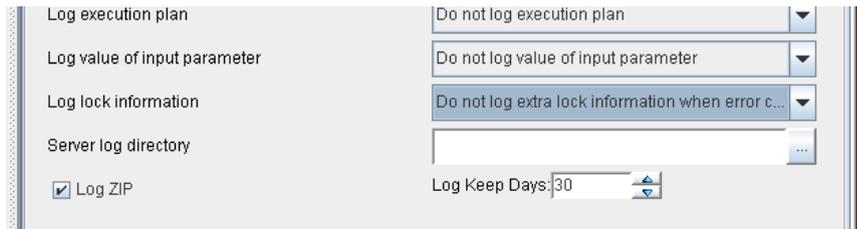
Log ZIP

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 won't 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, disconnections, 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 error codes below 30,000
 - Select **Disconnect error or DB crash error** to log error codes below 20,000
 - Select **Abort, disconnect, or DB crash error** to log error codes below 10,000
 - Select **Normal, abort, disconnect, or DB crash error** to log the error codes below 100
 - Select **Warning or other error** to log error codes over 0 (i.e., all errors)
 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, connection ID, etc.** to log basic system information.

- select **Log all the information as the above setting (DB_LGSYS=0) and SYSUSER and SYSINFO information** to log SYSUSER, SYSINFO and 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 browse 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 to retain log files

7.6 Changing Character Set Encoding

Selecting the **CharSet Encoding** tab in the **Run Time Settings** window allows the user in client to set the database language code and error message code themselves.

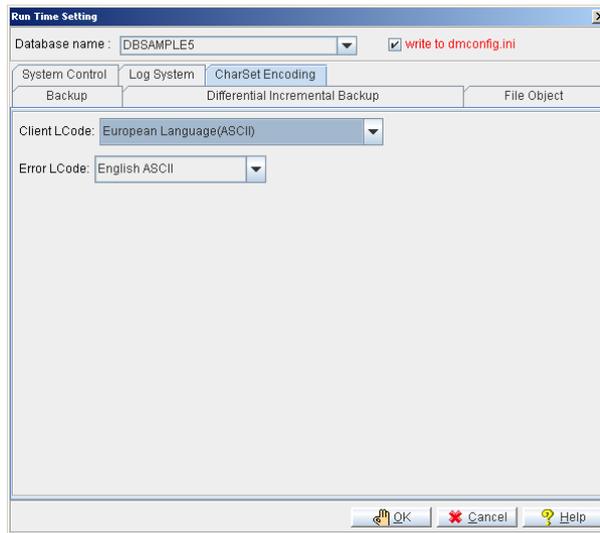


Figure 7-6 Character Set Encoding Settings window

Client LCCode

When use multilingual database and the LCODE of the Database Server is set to 10 (i.e., it is a UTF-8 database), client side can choose different language code to connect to the UTF-8 database server. 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 databases, 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 character set settings:
1. Select a client lcode.
 2. Select an error message lcode.

8 Renaming a Database

Databases may be renamed by using the Rename Database wizard. This easy-to-use tool simplifies the renaming of a database. When using Windows you must update the ODBC data source if you wish to connect to the database via ODBC.

- ➔ Renaming a database:
1. Click the **Wizards** option from the menu bar. A pop up menu appears.
 2. Select **Rename Database** from the list. The **Welcome to the Rename Database Wizard** window appears.
 3. Click the **Next** button. The **Choose a Database** window appears.

Rename Database Wizard - Choose a Database

Choose a Database
Choose the database you want to rename and select a new name for it.

Old database name :

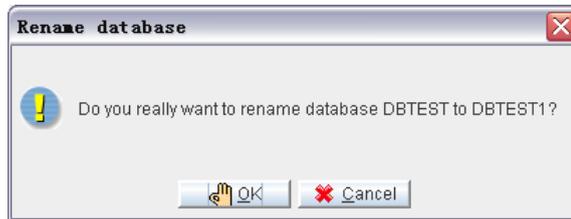
New database name :

User name :

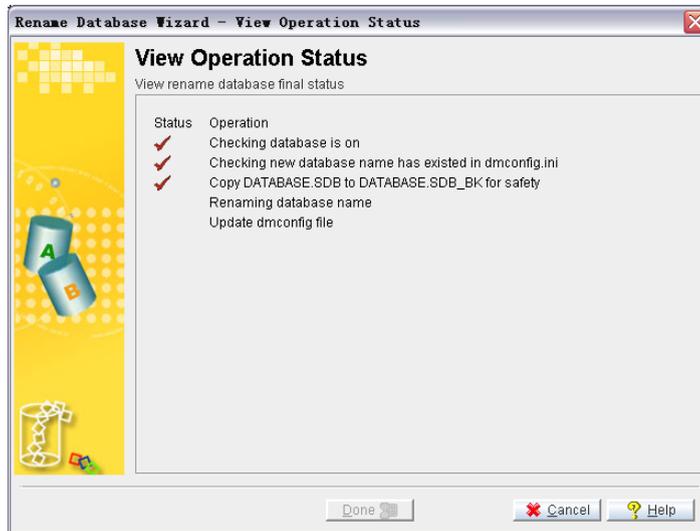
Password :

Previous Next Cancel Help

4. Select a database from the **Old database name** menu.
5. Enter a name in the **New database name** field.
6. Enter the system administrator's password in the **User name** field. SYSADM is in the field as the default name.
7. Enter the system administrator's password.
8. Click the **Next** button. The **Rename database** dialog box appears.



9. Click the **OK** button. The **View Operation Status** window and the **Information** dialog box appear.



10. Click the **OK** button in the **Information** dialog box. The dialog disappears.
11. Click the **Done** button in the **View Operation Status** window. The **JServer Manager** main window reappears.

9 Diagnosing a Database

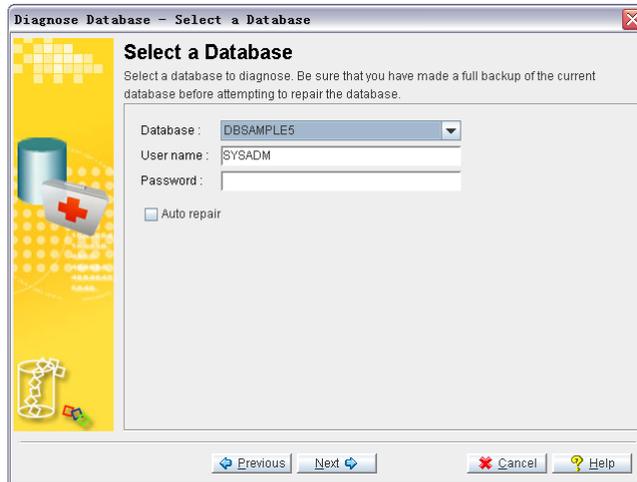
This section details database diagnostics and repairing. This is primarily performed using the **Diagnose Database Wizard**. The wizard is a very intuitive tool that guides the user through the process of repairing a database.

➔ Diagnosing and repair a database:

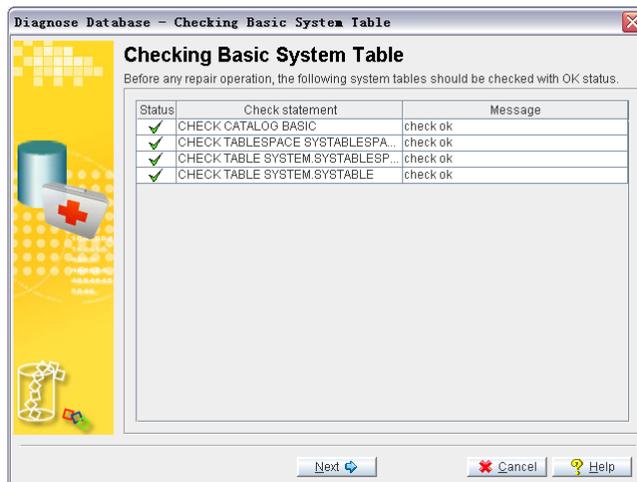
1. Select **Diagnose Database** from the main console or the **Wizards** menu. The **Welcome to the Database Repair Wizard** window appears.



2. Click the **Next** button. The **Choose a Database** window appears.



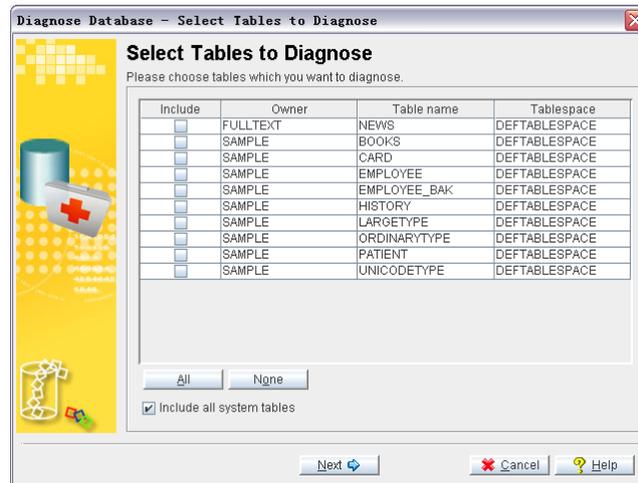
3. Select a database from the **Database** menu.
4. Enter the system administrator's name in the **User name** field.
5. Enter the system administrator's password in the **Password** field.
6. To enable automatic database repair check the **Repair automatically** check box.
7. Click the **Next** button. The **Check Basic System Table** window appears.



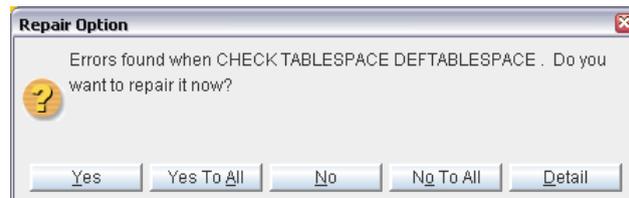
8. A check mark appears in the **Status** field for each system table that is functioning normally. An X appears in the **Status** field if there is a problem. A message also appears in the corresponding **Message** field.

NOTE *If any check operation fails in this step, the wizard will not be able to continue.*

9. Click the **Next** button. The **Select Tables to Diagnose** window appears.

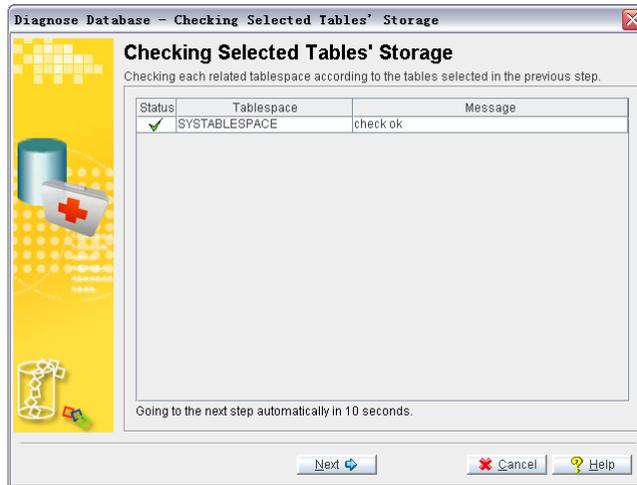


10. Select a table to diagnose by checking the **include** box in the appropriate row. To select all tables click the **All** button.
11. Click the **Next** button.
12. The **Checking Selected Tables' Storage** window appears, displaying the tablespaces containing the selected table. If errors are found, the **Repair Option** dialog box appears.

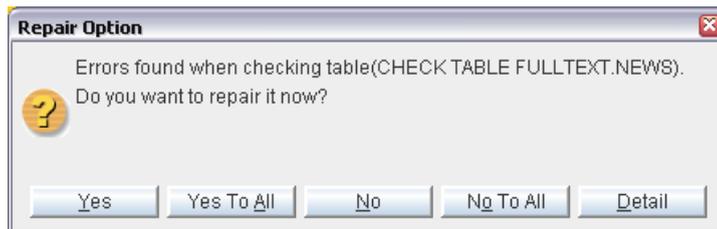


13. Select **Yes** to repair the tablespace in question. Select **Yes To All** to repair all tablespaces. Select **No** to skip repairing the tablespace in question. Select **No To All** to skip repairing all tablespaces. Select **Detail** to see the error

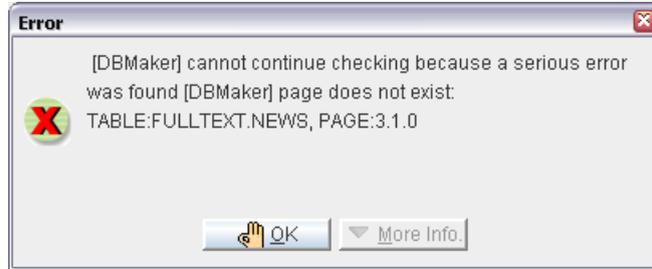
message. After all tablespaces are repaired or skipped, the **Checking Selected Tables' Storage** window appears.



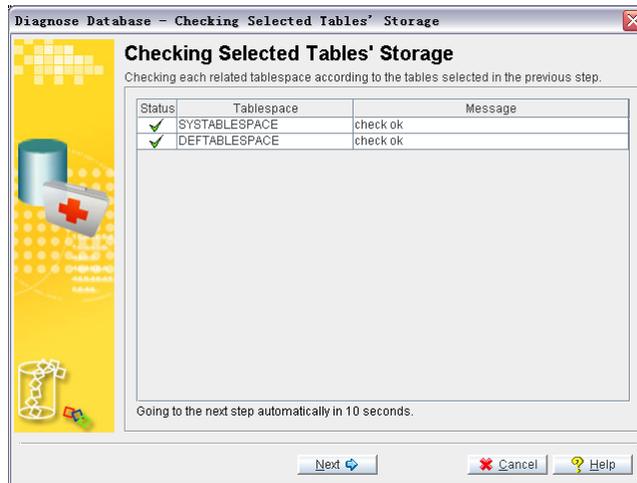
14. Click **Next** or wait 10 seconds, the wizard will automatically go to the next step.
15. DBMaker checks the remaining system tables and selected user tables. If any errors are found, the **Repair Option** dialog box appears.



16. Select **Yes** to repair the table in question. Select **Yes To All** to repair all tables. Select **No** to skip repairing the table in question. Select **No To All** to skip repairing all tables. Select **Detail** to see the error message. Selecting **Detail** opens the **Error** dialog box.

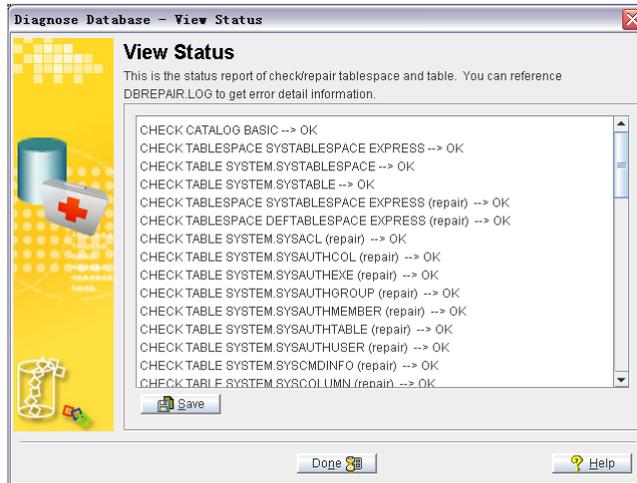


17. Click **OK** to continue checking tables. After all tables have been checked the **Checking Selected Tables** window appears.

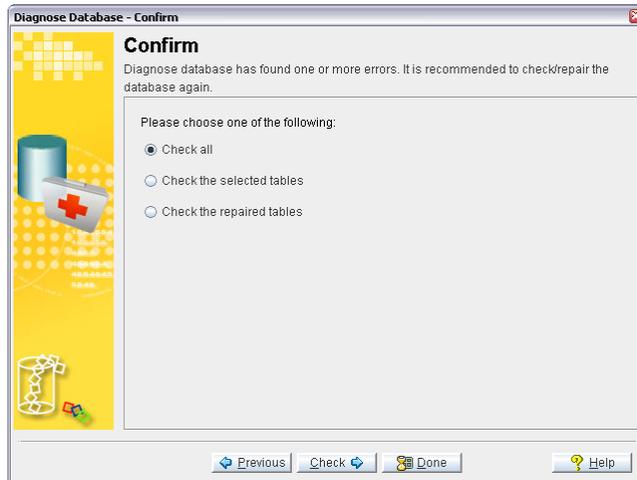


18. The **Checking Selected Tables** window displays the status of the tables checked. Click the **Next** button to continue. The **View Status** window appears.

NOTE *If errors were found and not repaired in a tablespace, then tables belonging to the broken tablespace cannot be repaired.*



19. The **View Status** window displays a report of the error check. If any repairs were performed, the **Next** button appears, otherwise the **Done** button is displayed. Click the **Next** button to continue, or **Done** to complete the wizard
20. The **Confirm** window appears.



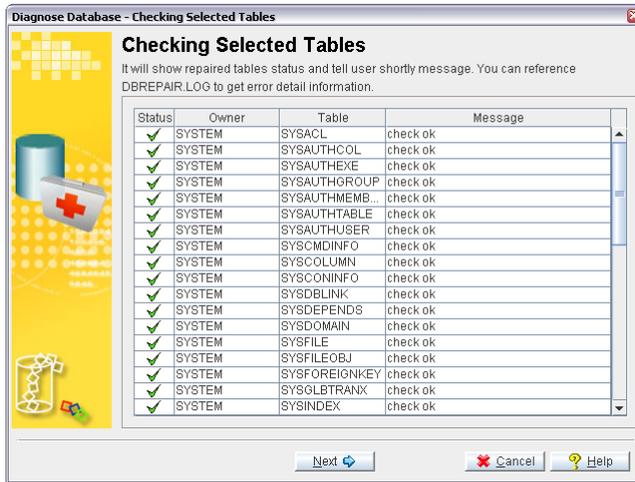
21. The **Confirm** window allows the database to be rechecked to ensure that repairs were effective. Select **Check all** to recheck the entire database. Select **Check the selected tables** to recheck only the tables initially

selected. Select **Check the repaired tables** to recheck the tables that contained errors.

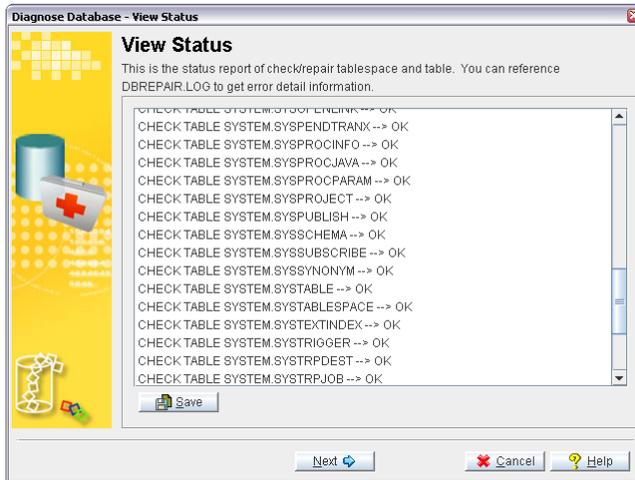
22. The **Diagnose Database Wizard** opens to the **Checking Selected Tables' Storage** window and checks tablespaces according to the choice made in the Confirm window.



23. Click the **Next** button or wait 10 seconds, the wizard automatically continues to the next step. DBMaker rechecks the selected tables.



24. Click Next. The View Status window appears a final time.

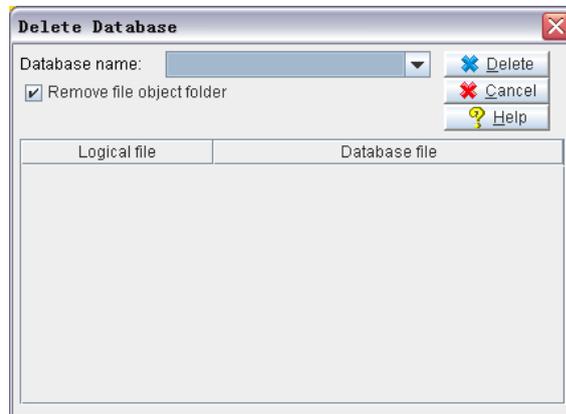


10 Deleting a Database

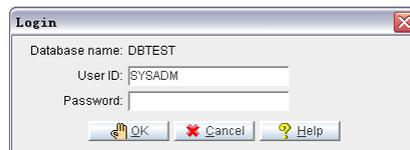
Once you no longer require a database, you can delete it using **JServer Manager**.

➤ Deleting a database

1. Select **Delete Database** from the main console. The **Delete Database** window appears.

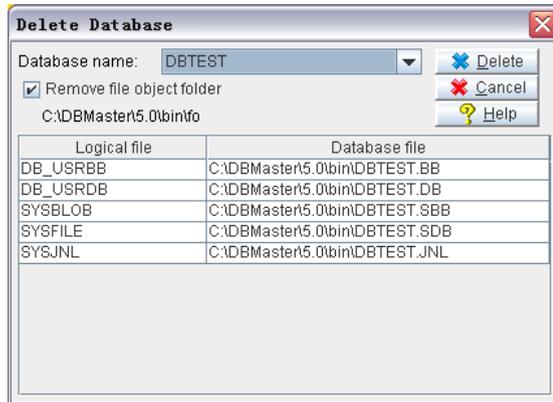


2. Select a database from the **Database Name** menu.
3. The **Login** window appears.



4. Enter **SYSADM** in the **User Name** field.

5. Click the **OK** button. The **Delete Database** window appears. All operating system files associated with the database are listed.



6. Click **Delete** to remove all of these files as well as information related to the database from the **dmconfig.ini** file. The **Delete Database** dialog box appears.



7. Click **OK**. The **Delete Database** dialog box will display, confirming your selection.
8. Click **OK**.

11 Backing up a Database

DBMaker allows the use of different options for backing up a database. In addition to backup daemons periodically, executing incremental, differential and full backups, on demand backups for online or offline databases are supported at all times.

Online full backups increase the load on a system's processor and storage devices. Please consider scheduling online full backups when client resource demands are relatively low.

The following backup methods are provided by DBMaker.

- **Online Full Backup by Backup Server:** JServer Manager automatically backs up the database and file objects to the location specified in the dmconfig.ini file. All data in the database and all file objects are copied to the backup location. This backup method requires that the database be started and the Backup Server activated. Clients may remain connected to the database during online backups.
- **Online Differential Backup by Backup Server:** All data that has changed since the latest full backup is copied to the backup location. Differential backup must depend on a full backup. This backup method requires that the database be started and the Backup Server activated.
- **Online Incremental Backup by Backup Server:** All journal blocks are copied to the incremental backup location as specified in the dmconfig.ini file.

Incremental backups can be performed while the database is online and clients may remain connected to the database.

- **Online Full Backup Interactively:** The destination of backup files may be specified. File objects are not backed up.
- **Online Full Backup Interactively to Tape:** All data in the database is copied to one or more tapes. File objects are not backed up.
- **Online Incremental Backup Interactively:** All journal blocks are copied to an incremental backup location that may be specified at the time of the incremental backup. Incremental backups can be performed while the database is online and clients may remain connected to the database.
- **Online Incremental Backup Interactively to Current:** The data in the database, from the point of the most recent backup to the end of the current journal file, is backed up. The advantage of performing an incremental backup to current is that you can better protect the database against crashes.
- **Offline Full Backup:** All data in the database is copied to the backup location. The database must be offline before a full backup is performed. File objects are not backed up.
- **Offline Full Backup to Tape:** All data in the database is copied to one or more tapes. File objects are not backed up.

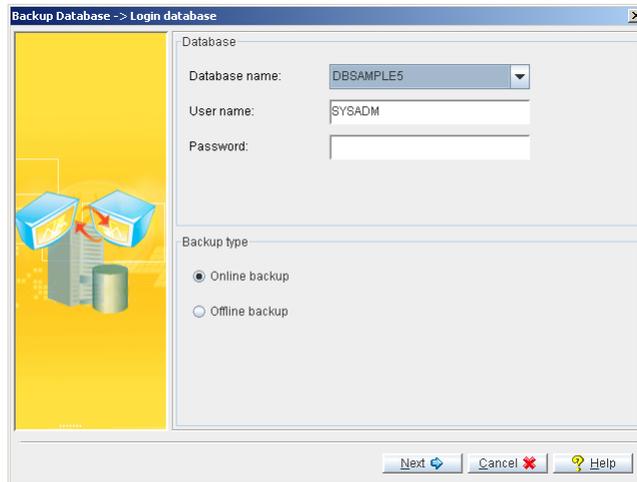
11.1 On-line Full Backup by Backup Server

A full backup may be performed quickly and easily while the database is started using On-line Full Backup by Backup Server. Online full backups performed this way are copied to the location specified in the dmconfig.ini file. The backup directory should be located on a disk separate from the disk the database is stored on to prevent loss of data in the event of media failure.

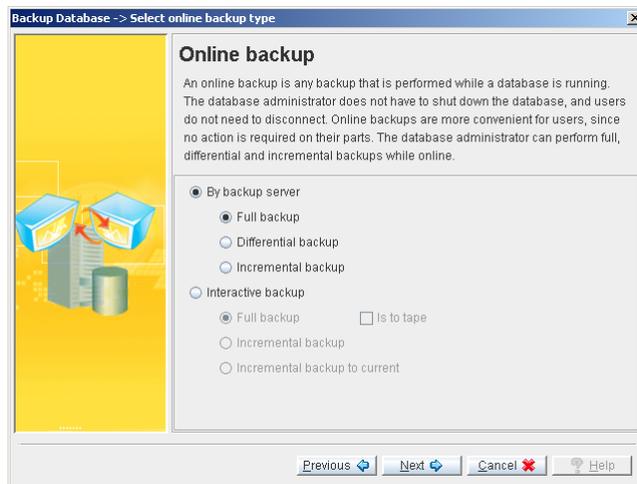
Online full backup by backup server can be performed on a remote server, and may be used to perform file object backup, however, on-line full backup interactively is not capable of these functions.

Be sure that the backup server is activated before using this backup method. If an error message “backup server doesn’t exist” appears, shut the database down and activate the backup server when restarting. Please refer to sections 4.2 and 4.3 for additional instructions on starting the backup server.

- ➔ Performing an online full backup by backup server:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu. Enter a user name and password. The user name must be a user with DBA authority or higher.
 3. Select the **Online backup** option.



4. Click the **Next** button. The **Select online backup type** window opens. If backup server is on, backup function will use backup server mode to backup database first, otherwise use interactive backup mode to backup database. Click the **Full Backup** button under the **by backup server** button.



5. Click the **Next** button open the **Online full backup by backup server** window.



6. Click the **Finish** button. The **OnLine backup by backup server** message box appears while the backup is in process.



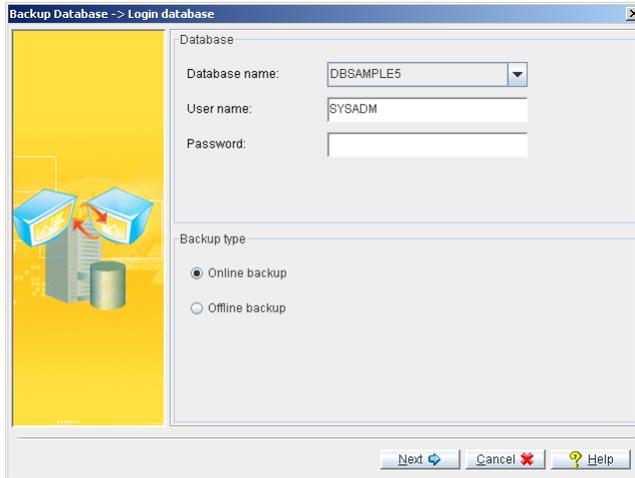
7. The **Online backup by backup server** message box is replaced by a confirmation dialog box when the backup is complete. Previous backup files are overwritten. If a directory for previous full backup files is specified, then the old backup files will be copied to the directory of previous backup files. Use the **Backup** page of the **Start Database Advanced Settings** window to specify the directory for previous backup files.

11.2 On-line Differential Backup by Backup Server

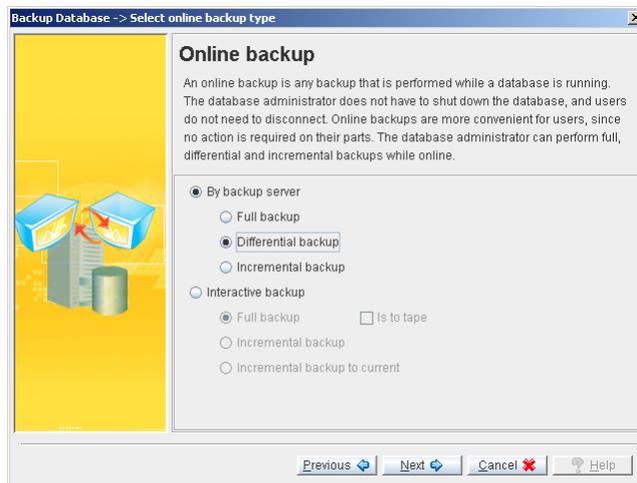
Full backup copy all data files. This is a time-consuming and space-consuming operation. In order to avoid this disadvantage, we need a new type backup, differential backup. You can perform a differential backup operation for a database that is already started.

A differential backup is based on the latest full backup of the data and records only the data that has changed since the most recent full backup. User can make more frequent backup because differential database backups are smaller than full backups. Making frequent backups decreases user's risk of losing data.

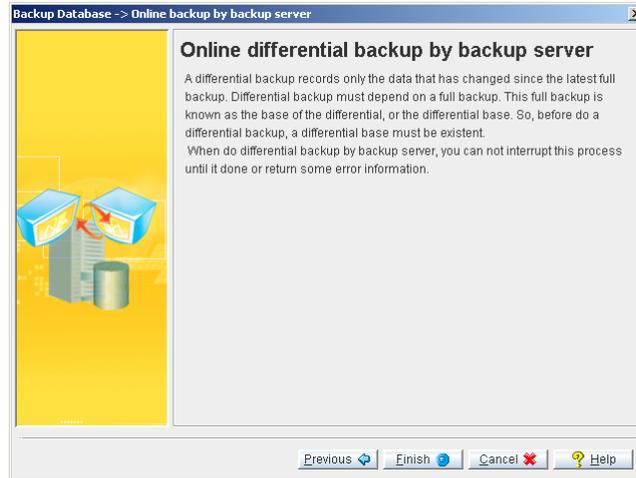
- Performing an online differential backup by backup server:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu. Enter a user name and password. The user name must be a user with DBA authority or higher.
 3. Select the **Online backup** option.



4. Click the **Next** button. The **Select online backup type** window opens. Select the **Differential backup** button under the **by backup server** button.



5. Click the **Next** button open the **Online differential backup by backup server** window.



6. Click the **Finish** button. The **OnLine backup by backup server** message box appears while the backup is in process.



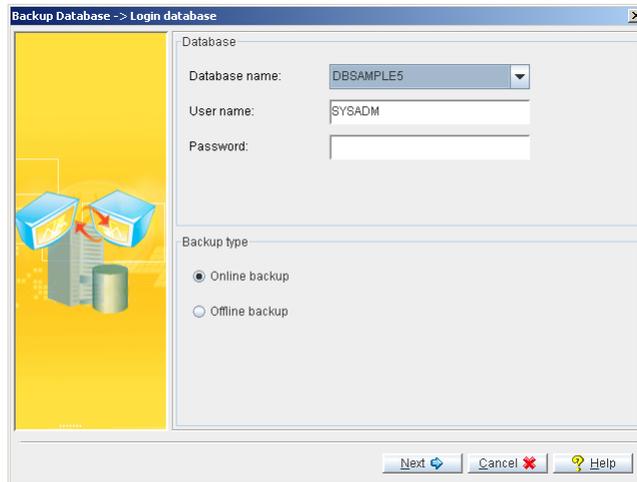
7. The **Online backup by backup server** message box is replaced by a confirmation dialog box when the backup is complete.

11.3 On-line Incremental Backup by Backup Server

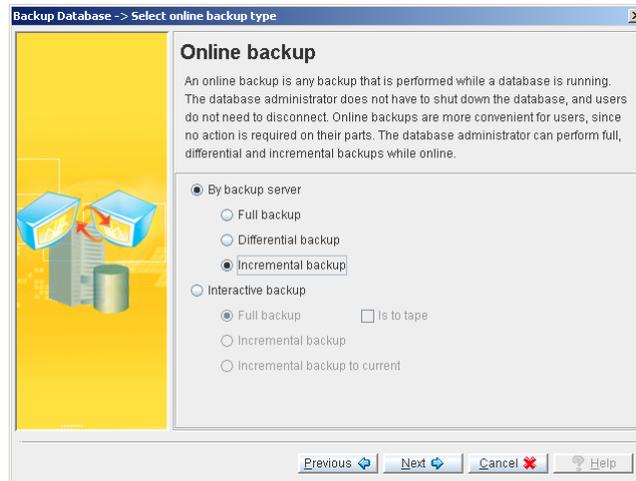
Incremental backups differ from full backups in that they only copy full journal files to the backup destination directory. To allow a database to recover its files, it is necessary to perform a full backup before an incremental backup.

An incremental backup may be performed quickly and easily while the database is started using **On-line Incremental Backup by Backup Server**. Incremental backups performed by this method are copied to the location specified in the **dmconfig.ini** file. The backup directory should be located on a disk separate from the disk the database is stored on to prevent loss of data in the case of media failure.

- ➔ Performing an On-line incremental backup by backup server:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu. Enter a user name and password. The user name must be a user with DBA authority or higher.
 3. Select the **Online backup** option.

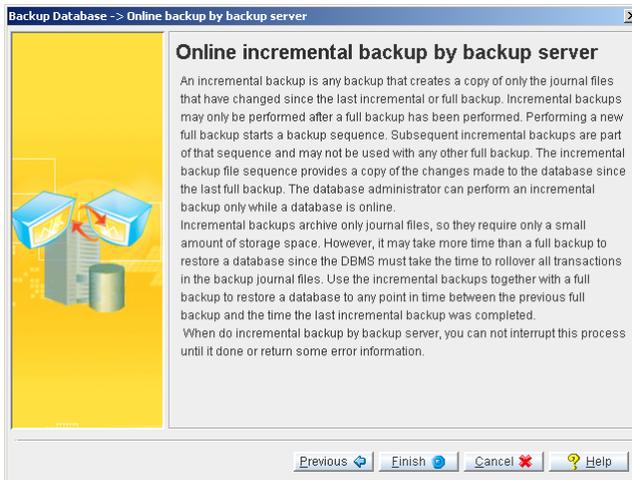


4. Click the **Next** button. The **Select online backup type** window opens. Select the **Incremental backup** button under the **by backup server** button.



NOTE *Be sure that the backup server is activated before using this backup method. If an error message “backup server doesn’t exist” appears, shut down the database and activate the backup server when restarting. For instructions on starting the backup server, please refer to sections 4.2 and 4.3.*

5. Click the **Next** button open the **Online backup by backup server** window.



6. Click the **Finish** button. The **OnLine incremental backup by backup server** message box appears while the backup is in process.



7. The **Online backup by backup server** message box is replaced by a confirmation dialog box when the backup is complete.

11.4 On-line Full Backup Interactively

You can perform a backup operation for a database that is already started. Performing an online backup involves specifying a location for the backup files. You should choose a backup directory location on a separate disk to minimize the risk of loss of data in the event of media failure. You can also change the following incremental backup settings when making an online full backup.

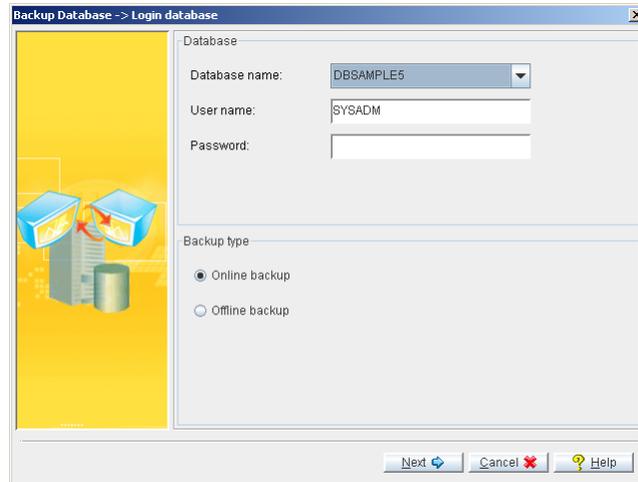
BACKUP MODE	DESCRIPTION
Turn off backup mode	Disables the incremental backup daemon. When backup mode is disabled, journal files are not backed up.
Backup data	All data is written to the journal but the incremental backup daemon only backs up non-BLOB data in the journal files.
Backup data and BLOB	All data is written to the journal and the incremental backup daemon backs up all journal files.
Write to dmconfig.ini	Saves changes in the incremental backup mode to the dmconfig.ini file. New settings take effect the next time the database is started.

Incremental backups should be used as supplements full backups.

- Performing an online full backup interactively:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu, Enter a user name and password. The user name must be a user with DBA authority or higher.

NOTE *Any user with DBA security privilege may backup a database.*

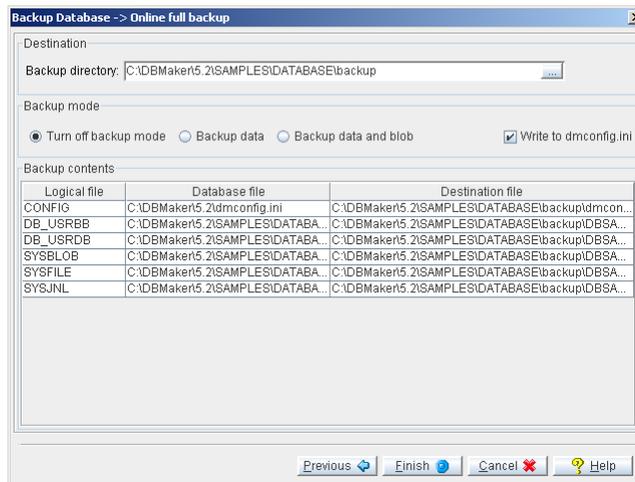
3. Select the **Online backup** option.



- Click the **Next** button. The **Select online backup type** window opens. If backup server is off, backup function will use interactive backup mode to backup database. Select the **Full backup** button under the **Interactive** backup button.



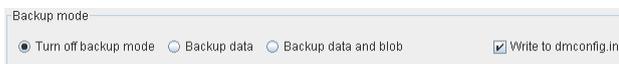
- Click the **Next** button open the **Online full backup** window with the list of operating system files to be backed up. The destination file location is the default backup directory specified in the **dmconfig.ini** file.



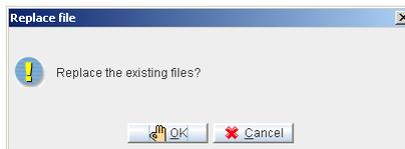
- To select a new path for the backup directory or specify a location by entering it in the **Backup Directory** field, please refer to chapter 3.1 for more information about how to create a new backup directory:

NOTE *The database administrator may specify another location for the default backup directory with the JConfiguration Tool. Please see the JConfiguration Tool Reference for more information.*

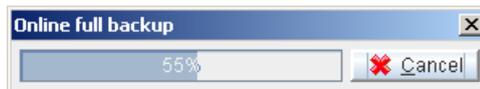
- Change the incremental backup settings according to your needs as shown:



- To turn off backup mode, click the **Turn off backup mode** radio button.
 - To turn on data backup mode, click the **Backup data** radio button.
 - To turn on data and blob backup mode, click the **Backup data and blob** radio button.
- Click the **Finish** button to save all files to the backup directory, the **Replace file** window appears. If files with the same name already exist in the backup directory, they are overwritten. If a directory for previous full backup files is specified then the old backup files will be copied to the directory of previous backup file. Please specify the directory for previous backup files from the **Backup** page of the **Start Database Advanced Settings** window.



- Click the **OK** button. The **OnLine full backup** message box appears while the backup is in process.



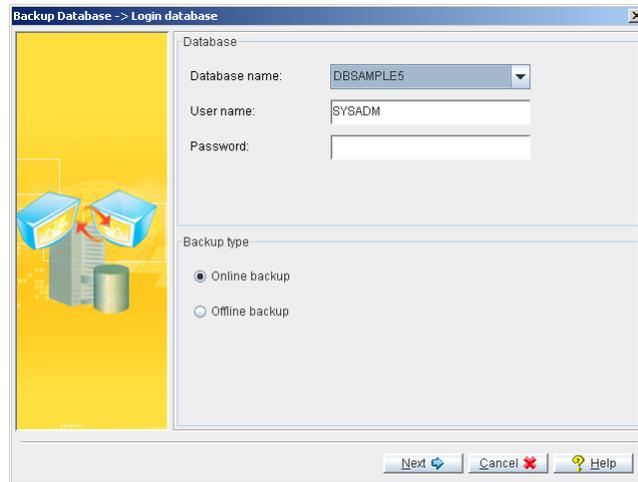
- The **Online full backup** message box is replaced by a confirmation dialog box when the backup is complete. Previous backup files are overwritten.

11.5 On-line Full Backup Interactively to Tape

When a database is started, you can perform an online full backup of your database files to one or more tape devices. You can also change the following incremental backup settings when making an online full backup.

BACKUP MODE	DESCRIPTION
Turn off backup mode	Disables the incremental backup daemon. When backup mode is disabled, journal files are not backed up.
Backup data	All data is written to the journal but the incremental backup daemon only backs up non-BLOB data in the journal files.
Backup data and BLOB	All data is written to the journal and the incremental backup daemon backs up all journal files.
Write to dmconfig.ini	Saves changes in the incremental backup mode to the dmconfig.ini file. New settings take effect the next time the database is started.

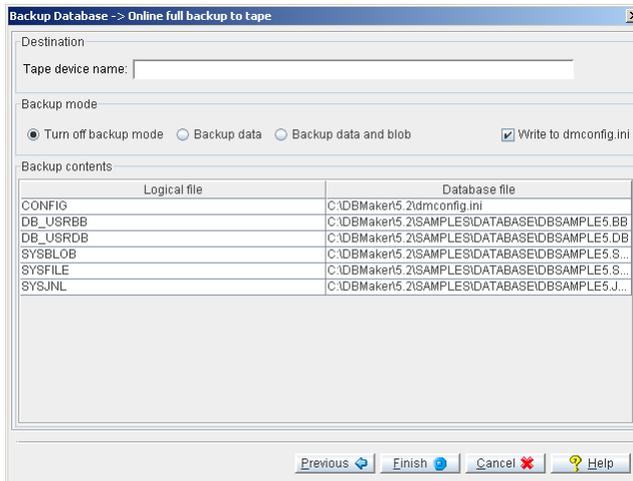
- Performing a full backup Interactively to tape:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu. Enter a user name and password. The user name must be a user with DBA authority or higher.
 3. Select the **Online backup** option.



4. Click the **Next** button. The **Select online backup type** window opens. Select the **Full backup** button under the **Interactive** backup button and enable the **Is to tap** check box.



5. Click the **Next** button open the **Online full backup to tape** window with the list of operating system files to be backed up.



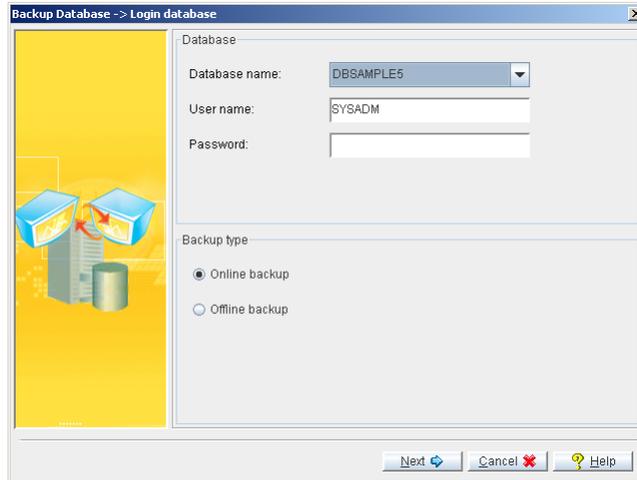
6. Enter the path of the tape device in the **Tape Device Name** field. DBMaker supports multiple tape backups. You can choose to use one or several tapes according to the tape size and the size of the backup files.
7. Change the incremental backup settings as shown:

- To turn off backup mode, select the **Turn off backup mode** radio button.
 - To turn on data backup mode, click the **Backup data** radio button.
 - To turn on data and blob backup mode, select the **Backup data and blob** radio button.
8. Click the **Finish** button. The database begins copying to the tape.

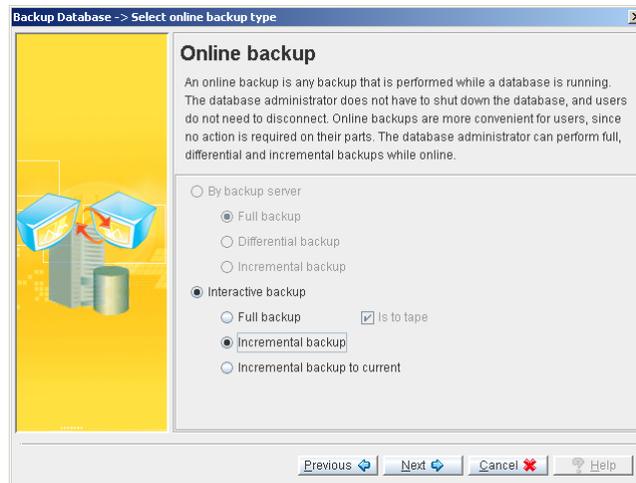
11.6 On-line Incremental Backup Interactively

Incremental backups differ from full backups in that they only copy full journal files to the backup destination directory. To allow a database to recover its files, it is necessary to perform a full backup before an incremental backup. The incremental backup daemon can be set to automatically copy journal files when they become full to the set capacity. In this way, it handles all the journal files itself and makes sure that the required data is backed up. It is possible to change the destination file location for manually performed incremental backups, however, this is not recommended. Backed up journal files are stored in the location specified on the Backup page of the Advanced Settings windows. Ideally the backed up journal files are stored in the same directory as the full backup. If you have not started the backup server, shut the database down and restart it with this setting enabled. For more information, please refer to Start Backup Server in section 4.2 and to the *Database Administrators Guide*.

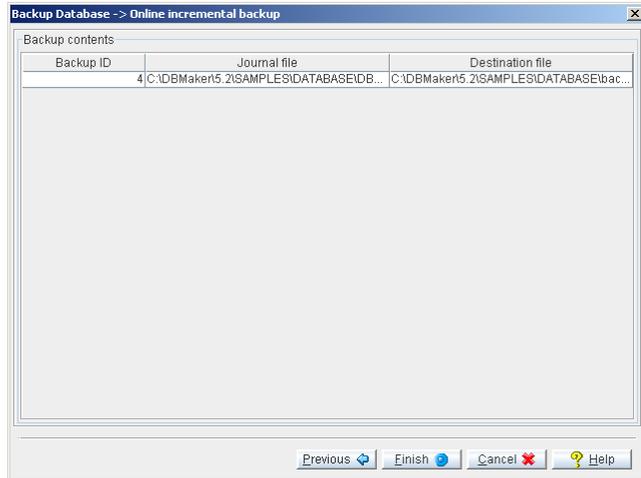
- Performing an On-line incremental backup:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu. Enter a user name and password. The user name must be a user with DBA authority or higher.
 3. Select the **Online backup** option.



4. Click the **Next** button. The **Select online backup type** window opens. Select the **Incremental backup** button under the **Interactive backup** button.



5. Click the **Next** button open the **Online incremental backup** window.



6. Click the **Finish** button. The **Online incremental backup** message box appears while the backup is in process.

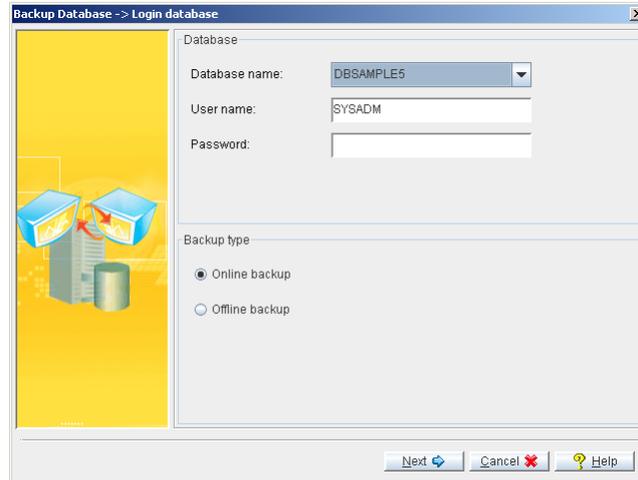


7. The **Online incremental backup** message box is replaced by a confirmation dialog box when the backup is complete.

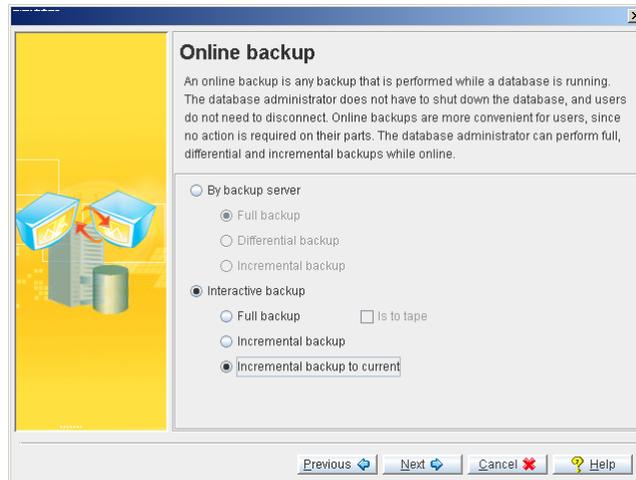
11.7 On-line Incremental Backup Interactively to Current

The **On-line Incremental Backup Interactively to Current** option copies all journal files to the backup directory, including the journal file that is currently in use. Incremental backups differ from full backups in that they only copy full journal files to the backup destination directory. To allow your database to recover its files, perform a full backup before an incremental backup. The incremental backup daemon can be set to automatically copy journal files when they fill to capacity. In this way, it handles all the journal files itself and makes sure that the required data is backed up. It is possible to change the destination file location for manually performed incremental backups, however, this is not recommended. Backup journal files are stored in a location as specified on the Backup page of the Advanced Settings windows, and ideally are stored in the same directory as the full backup. For more information, please refer to Start Backup Server and to the *Database Administrators Reference*.

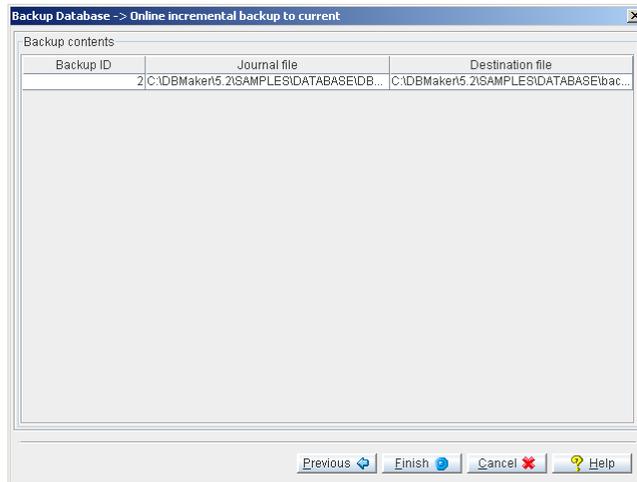
- Performing an incremental backup of current journal files:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu. Enter a user name and password. The user name must be a user with DBA authority or higher.
 3. Select the **Online backup** option.



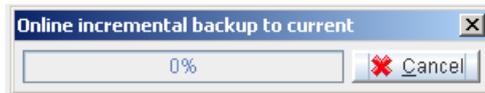
4. Click the **Next** button. The **Select online backup type** window opens. Select the **Incremental backup to current** button under the Interactive backup button.



5. Click the **Next** button open the **Online incremental backup to current** window.



6. Click the **Finish** button. The **Online incremental backup to current** message box appears while the backup is in process.

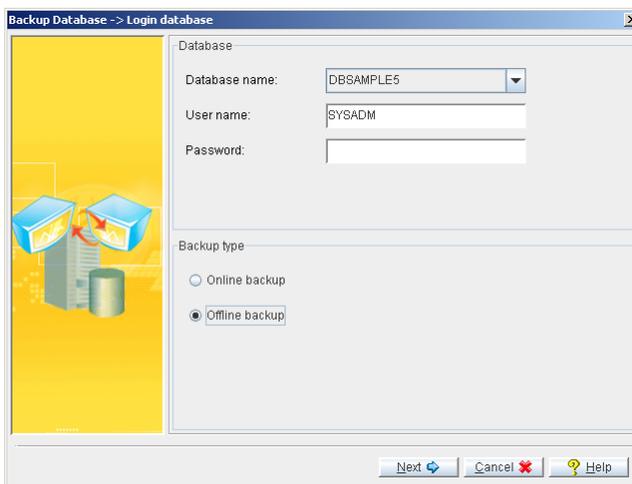


7. The **Online incremental backup to current** message box is replaced by a confirmation dialog box when the backup is complete.

11.8 Off-line Full Backup

You can perform backups for databases that are not yet started. These are called offline backups. Performing an offline backup involves specifying a location for the backup files. You should choose a backup directory location on a separate disk to minimize the risk of loss of data in the event of a media failure.

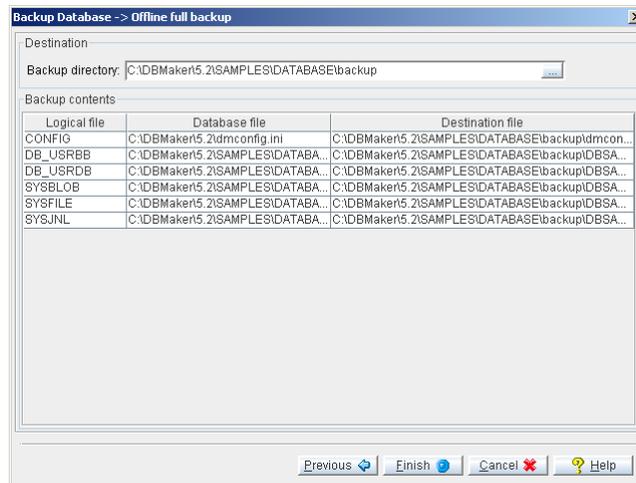
- Performing an Offline full backup:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu. Enter a user name and password. The user name must be a user with DBA authority or higher.
 3. Select the **Offline backup** option.



4. Click the **Next** button. The **Select offline backup** window opens. Select the **Full backup** option.



5. Click the **Next** button. The **offline full backup** window appears with a list of operating system files to be backed up. The destination file location is the default backup directory specified in the **dmconfig.ini** file.



6. To select a new path for the backup directory or enter a new valid path in the **Backup Directory** field. Please refer to chapter 3.1 for more information about how to create a new backup directory.
7. Click the **Finish** button to save all files in the backup directory. The **Offline full backup** message box appears while the backup is in process.



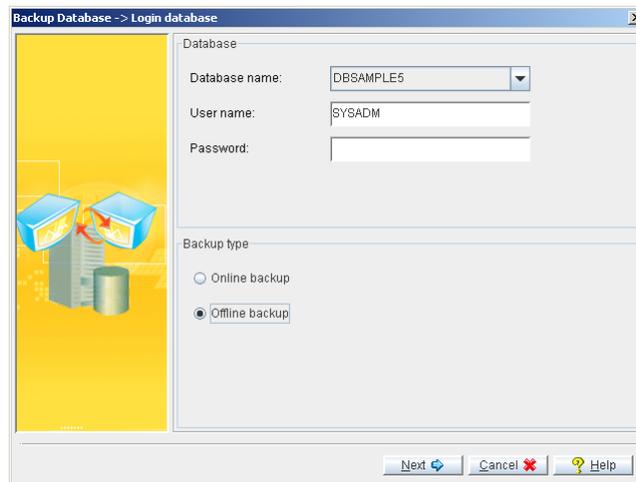
8. The **Offline full backup** message box is replaced by a confirmation dialog box when the backup is complete.

NOTE *If the files already exist in the backup directory, the Database Administrator may choose to overwrite them.*

11.9 Off-line Full Backup to Tape

As well as backing up your database to another file location, you can back it up to tape. Performing an offline backup involves specifying a location for the backup files. Performing backups to tape minimizes the risk of loss of data due to a media failure.

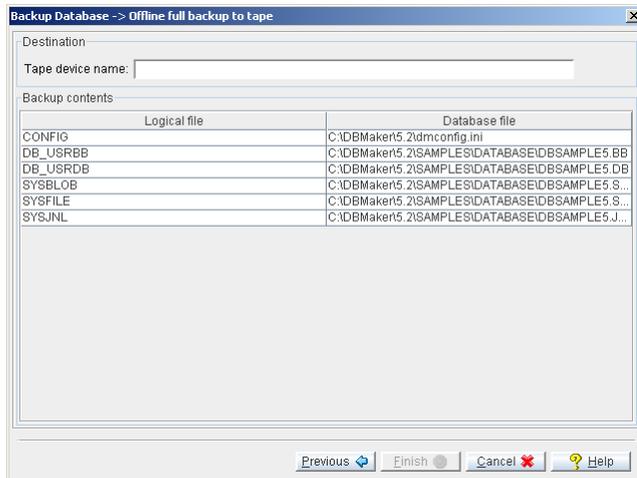
- Performing an offline full backup to tape:
 1. Select **Backup Database** from the main console or the **Database** menu. The **Login database** window appears.
 2. Select a database from the **Database Name** menu. Enter a user name and password. The user name must be a user with DBA authority or higher.
 3. Select the **Offline backup** option.



4. Click the **Next** button. The **Select offline backup** window opens. Select the **Full backup** option and enable the **Is to tape** check box.



5. Click the **Next** button. The **offline full backup to tape** window appears with a list of operating system files to be backed up. The destination file location is the default backup directory specified in the **dmconfig.ini** file.



6. Enter the device name in the Tape Device Name field. DBMaker supports multiple tape backup. You can choose use one or more tapes according to the tape size and the size of the backup files.
7. Click the **Finish** button to execute the full backup.

12 Managing Log Files

DBMaker keeps a set of log files that record different aspects of each database's history. In **JServer Manager**, you can view the log files associated with a database as well as save them to a location. If you wish to delete log files to manage the size of your database, you can clear them from the database.

The log file includes a date and time stamp with a character string. You can view logs that record the following database functions:

Error Log (dmerror.log): The error log is a record of all critical error messages returned by DBMaker that are associated with the database.

Warning log (dmevent.log): The warning log is a record of all the warning information come from the daemon.

Asynchronous Table Replication Log (atrp.log): This log is a record of all ODBC commands made to target databases by the distributor daemon. The Asynchronous Table Replication log is stored in the source database and contains information about all target databases of the source database.

Asynchronous Table Replication Error Log (atrperror.log): This log file records the errors returned from target databases upon execution of ODBC commands by the distributor daemon. The Asynchronous Table Replication Error log is stored in the source database and contains information about all target databases of the source database.

RP.LOG (rp.log): The replication log records backup journal file updates made from the source database to slave databases. The replication log file is stored in the

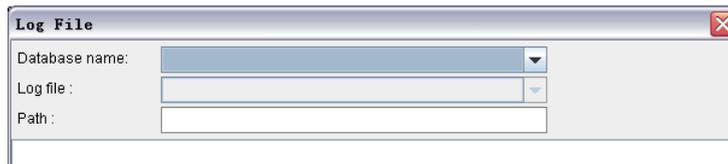
source database and contains information about all slave databases of the source database.

Backup History Log (dmbackup.his): The backup history log records the time and date of all backup functions applied to the database. This includes full, differential and incremental backups.

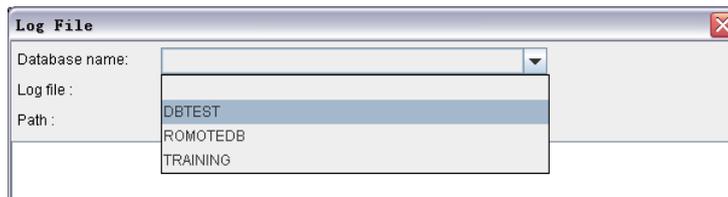
NOTE *Offline full backup is separated from dmbackup.his. it is recorded with offBackup.his.*

➔ Viewing the log files:

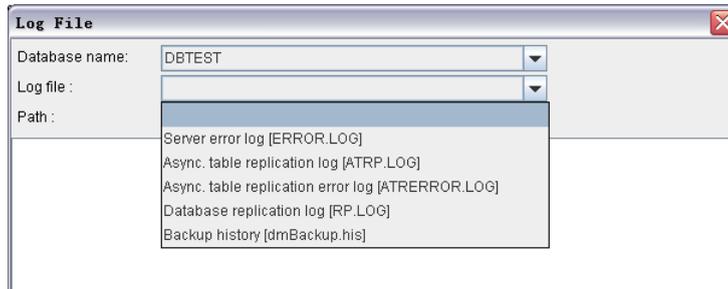
1. Select Log File from the Database drop down menu to open the Log File window.



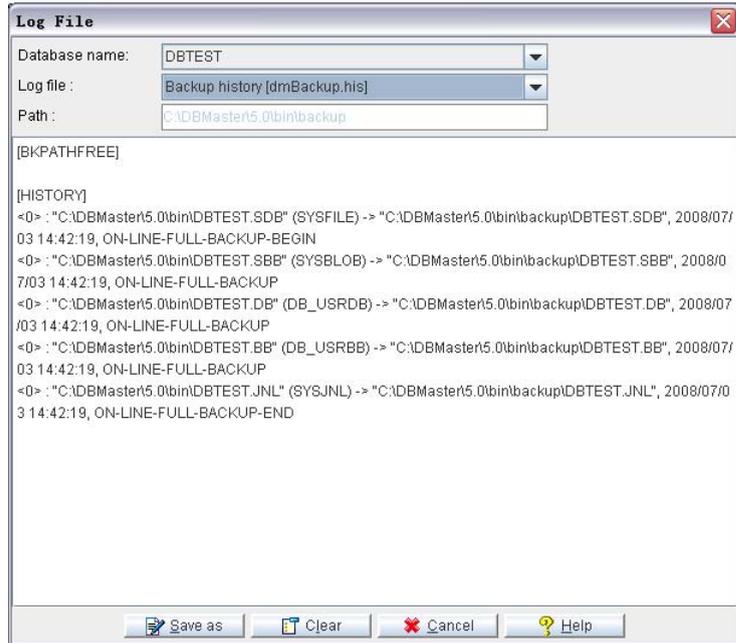
2. Select a database from the Database Name menu.



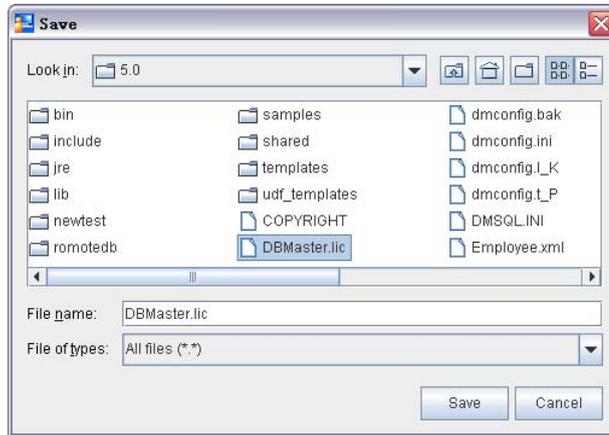
3. Select a log file to view from the Log File menu.



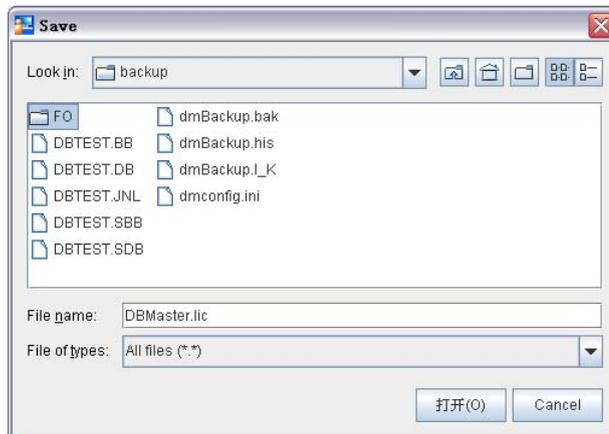
- To browse through the records, use the scroll bar. The DBA may also choose to save the log file to a different location (this does not move the file, only copies it), or clear the contents of the log file by selecting Save or Clear from the bottom of the **Log File** window.



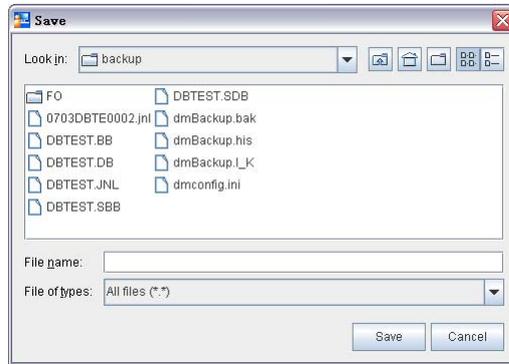
- Saving log files to a different location:
 - Select a log file.
 - To save the log file to a different location, click the Save As button from the bottom of the Log File window. The Save window appears.



3. To preserve the file name, select the file name of the file being viewed from the list of files.



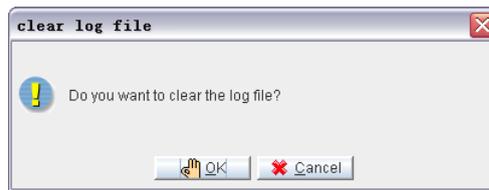
4. The database administrator can also choose to enter a new name in the File Name field.
5. To select a new directory or a new disk, click the *Up One Level* button  button until the desired path's root is available.
6. To select another disk location for the log file, choose a location from the Look In menu.



7. Select the desired path down the directory tree by double clicking on the folder icon next to the directory name, a list of files contained in the directory appear.
8. To overwrite an existing file,
 - a) Select a file in the directory.
 - b) Click the Save As button. The file is overwritten.
9. To rename a log file:
 - a) Enter a name in the File Name field.
 - b) Select Save to save the log file in the current directory.

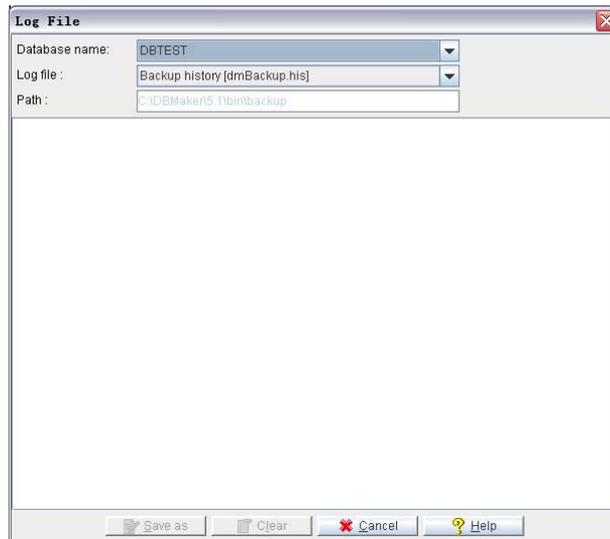
NOTE *Log files are necessary for proper database function. Saving log files to different locations or clearing necessary log files interferes with database restoration. If you want to clear your log files to manage the size of your database, you must first back up your log files.*

- ➔ Clearing log files:
1. Click the Clear button at the bottom of the Log File window. JServer Manager prompts to confirm this action.



NOTE *Once cleared, log files cannot be restored.*

2. Click the OK button to clear the log file. The contents of the log file is deleted.



13 Restoring a Database

The database administrator may find it necessary to restore data from backup files if a database suffers from an unrecoverable error. A database restoration process can be performed from disk or from tape. Restoring the database from tape recreates the database as it existed at the time of the most recent full backup or differential backup as a baseline. If you restore the database from disk, you can further restore the database up to the execution time of the last incremental backup.

NOTE *For more information on database restoration, please refer to the Database Administrators Guide.*

The following subsections provide details and examples for both of the database restoration methods.

13.1 Restoring a Database from Disk

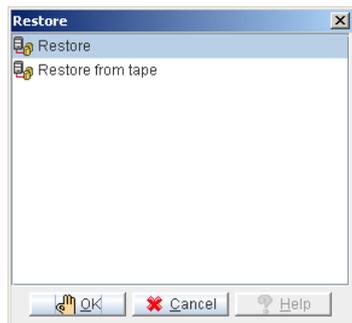
You can restore a database that has been backed up to another disk location on your computer or on the network.

NOTE *To restore a database you must have DBA privileges or higher.*

➔ Restoring a database from disk:

1. Select Restore Database from the main console.

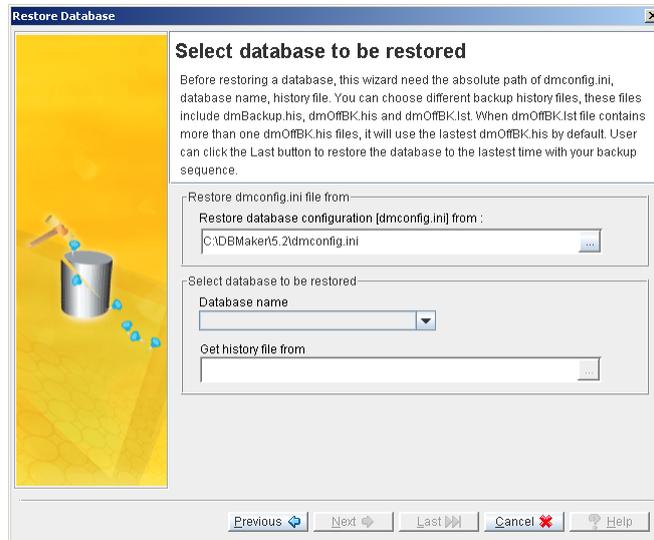
NOTE *You can also select Restore Database from the Database menu.*



2. Make sure that Restore is selected from the Restore window.
3. Click the OK button. The Restore Database window appears.



4. Click the Next button. The Restore Wizard window appears.

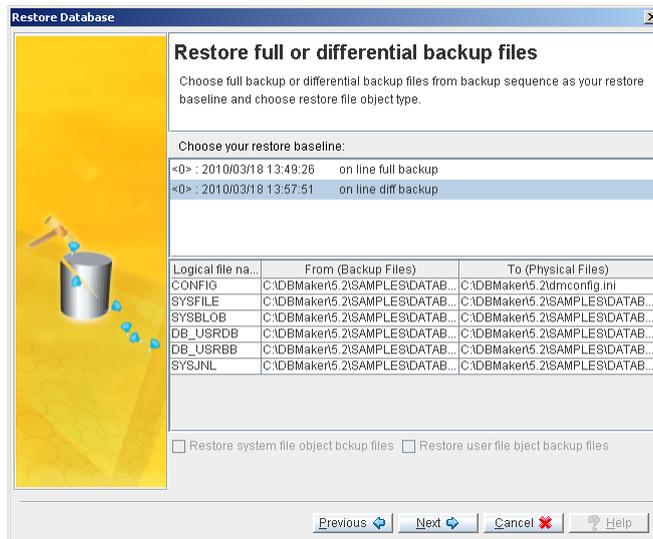


5. Click the field under **Database Name** and select a database from the drop-down list of databases. Only databases available on the server appear in the drop-down list.

6. Select a database. The Restore Database Configuration [dmconfig.ini] from field shows the default location of the configuration file. If you moved the **dmconfig.ini** file to a new location, then enter the new path or click on the browse button  in the Restore database configuration [dmconfig.ini] from field.
7. If you moved the backup history file to a new location, then enter the new path or click on the browse button  in the Get History File From field. There are three kinds of backup history file, includes dmBackup.his, dmOffBk.lst and dmOffBk.his. If you assigned the dmBackup.his file, the restoration process is for the online backup sequence, and continue step 8. If you assigned the dmOffBk.lst and dmOffBk.his history files, the restoration process is for the offline backup sequence, and go to step 16.

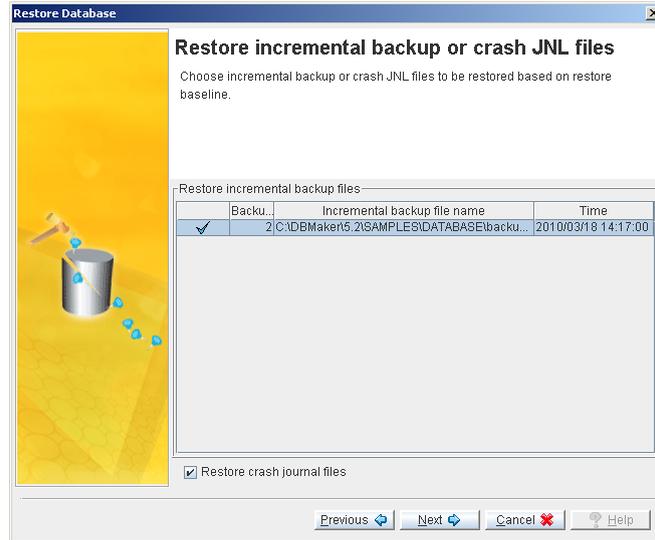
NOTE *After the database to be restored has been selected, the location of the history log file appears in the **Get History File From** field. The default restoration process is for online backup sequence.*

8. Click the **Next** button. The **Restore full or differential backup files** page appears. Select a backup that contains the files you wish to restore from the top list. The logical file names, backup files, and physical files are displayed in the bottom list.

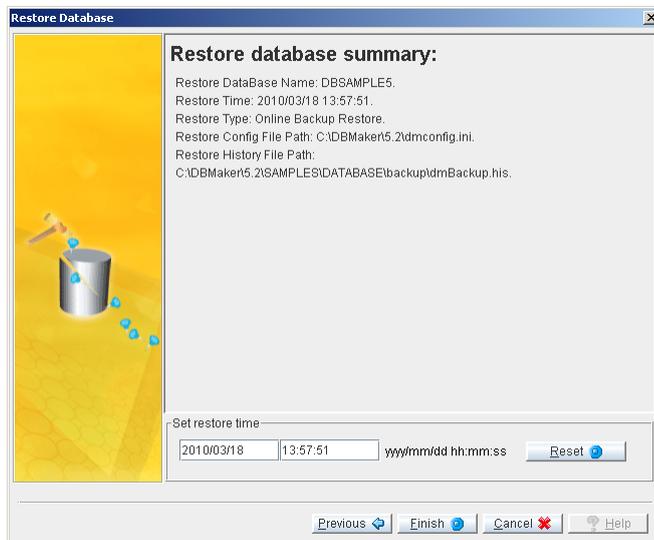


NOTE *Select whether to restore system file objects or both system and user file objects by checking the appropriate check boxes.*

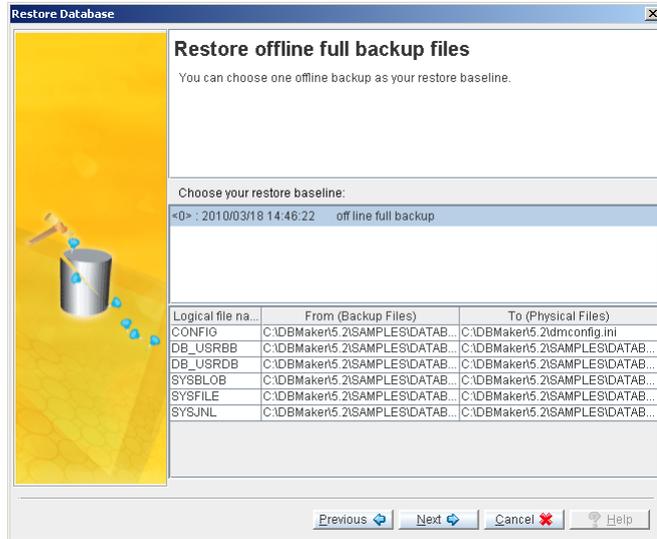
9. Click the Next button. The Restore Incremental Backup Files page appears.



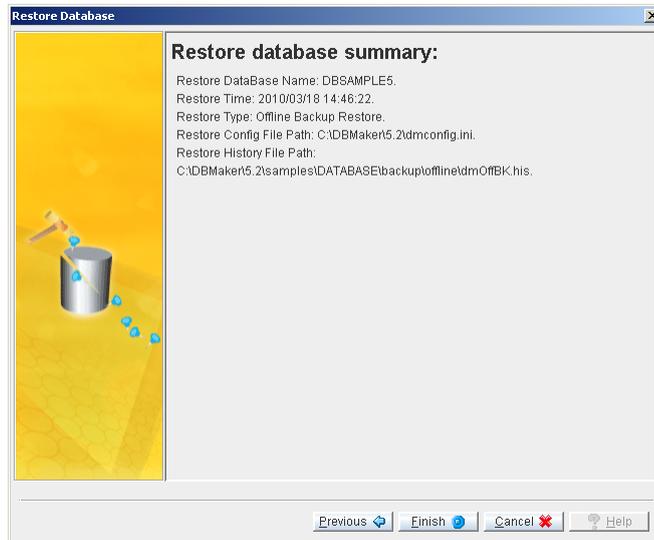
10. The Incremental Backup File Name field shows all the incremental backup files. If you need to change the incremental backup file path, enter a path in the Incremental Backup File Name field. You can edit the file path but must not skip restoration of any of the files.
11. If you need to restore journal files created after the last incremental backup, select the Restore Crash Journal Files check box. Clearing this check box prevents restoring the journal files after the last incremental backup.
12. Click the **Next** button, the **Restore Database Summary** window appears, from this window, user can easily browse the online restoration informations. User also have a chance to change the RTime value, this value is efficient immediately when user click the **Reset** button.



13. Now continue to step 7. Enter the new path or click on the browse button  in the Get History File From field to select a dmOffBK.lst or dmOffBK.his history file.
14. Click the **Next** button. The **Offline full backup files** page appears. Select a backup that contains the files you wish to restore from the top list. The logical file names, backup files and physical files are displayed in the bottom list.



15. Click the **Next** button. The **Restore Database Summary** window appears, from this window, user can easily browse the offline restoration informations. User can click **Previous** button to change this setting.



16. If all the information is satisfying, click the **Finish** button to accomplish the restoration.

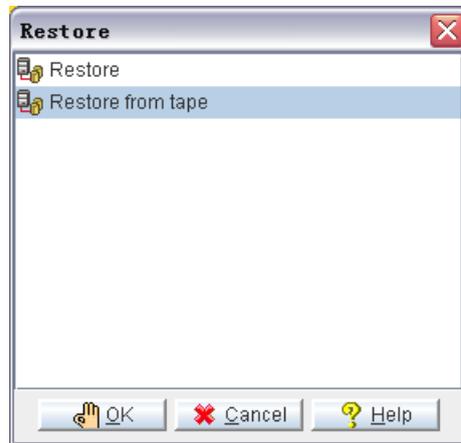
NOTE *The current journal files that were copied into a temporary directory are used to restore the database to its condition just prior to shutdown.*

13.2 Restoring a Database from Tape

You can restore a database that has been backed up to a tape device. You may choose to restore the backup history log and **dmconfig.ini** file from tape. Restoring the backup history logs from tape overwrites the current history log. All records of differential and incremental backups made since the last full restoration to tape are lost. It is not possible to restore the database to a more recent status using the incremental files after the backup history log is copied from tape. To restore a database, you must have DBA authority or higher.

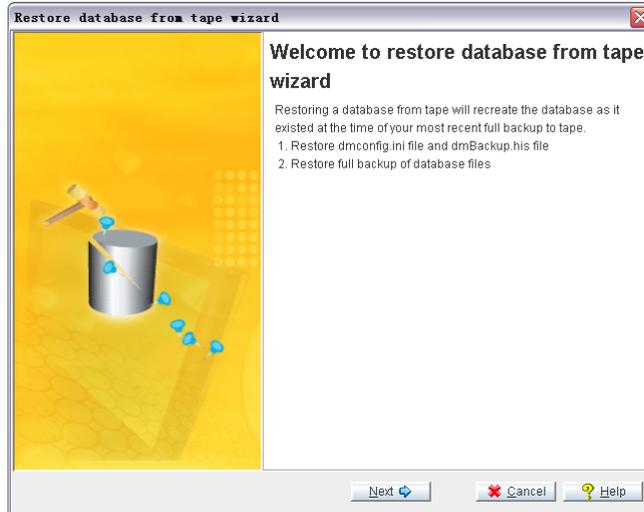
➤ Restoring a database from tape:

1. Select Restore Database from the main console. The Restore window appears.



NOTE *You can also select Restore Database from the drop-down menu. The Restore window opens.*

2. Select Restore from Tape from the Restore window and click the OK button. The Restore Database from Tape Wizard starts.



3. Click the Next button. The Restore a Database page appears.
4. Enter the name of the tape device in the Tape Device Name field.
5. Insert the tape from which the database is to be restored.



6. To get the **dmconfig.ini** file from tape, make sure that the Get dmconfig.ini from Tape check box is selected.

7. To get the **dmconfig.ini** file from disk, clear the check box.
8. To copy the **dmconfig.ini** file to different location, enter it in the Copy dmconfig.ini File to field.
9. To copy the backup history file from tape, make sure that the Copy Backup History from Tape check box is selected.
10. To copy the Backup History file from another source, clear the Copy Backup History from Tape check box.
11. Click **Next**. The name of the database and a list of the files in the database appear. The logical files should map to physical destination locations.
12. Click the **Finish** button to complete the database restoration process.

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. A table in a destination database that receives data from a source table.

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 (i.e., remote) databases.

Exclusive Lock (i.e., 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 (i.e., horizontal partition)

A fragment is the replication of a given range of data tuples.

Journal Blocks

Internal data measurement unit of 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 selected for replication.

Publication

A data set on the source table available for replication.

Remote Database

A database located on another server different from that being accessed by the client.

Remote Tables

Tables on a database located on a server different from the one connected to the client.

Replication Domain

The replication fragment (i.e., horizontal partition) and projection (i.e., vertical partition) together are called a replication domain. The range of a table's data that is replicated.

Select Lock (i.e., 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 in the source database where the replicated data originates.

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).

Target Database

A database that receives data from a database replication or table replication.

