



# DBMaker

JConfiguration Tool Reference

CASEMaker Inc./Corporate Headquarters

1680 Civic Center Drive

Santa Clara, CA 95050, U.S.A.

[www.casemaker.com](http://www.casemaker.com)

[www.casemaker.com/support](http://www.casemaker.com/support)

©Copyright 1995-2017 by CASEMaker Inc.

Document No. 645049-237064/DBM541-M02282017-JCON

Publication Date: 2017-02-28

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 ..... 1-3**
  - 1.2 Technical Support ..... 1-4**
  - 1.3 Document Conventions ..... 1-5**
- 2 JConfiguration Tool Basics ..... 2-1**
  - 2.1 Java Runtime Environment ..... 2-3**
  - 2.2 The JConfiguration Tool GUI..... 2-4**
    - Menu Bar..... 2-5
    - Tree View ..... 2-5
    - Main Console ..... 2-6
    - The Browse Button ..... 2-6
- 3 The Main Console ..... 3-1**
  - 3.1 Connection..... 3-3**
    - Default User ID and Password..... 3-3
    - Enabling Auto-Commit ..... 3-4
    - Setting the Connection Detection Interval..... 3-4
    - Application Multiple Connection Status ..... 3-4
    - Setting the Connection Time-Out Value ..... 3-5
    - Setting the Lock Time-Out Value ..... 3-5

	Enabling Net Compression .....	3-5
<b>3.2</b>	<b>DmServer .....</b>	<b>3-7</b>
	Client/Server Settings .....	3-7
	Server Settings .....	3-8
	Table Settings.....	3-9
<b>3.3</b>	<b>Data Manipulation .....</b>	<b>3-10</b>
	Remove Space Padding in CHAR Output.....	3-10
	Remove Space Padding in String Concatenation .....	3-11
	Maximum String Size .....	3-11
	File Type Mapping Settings.....	3-11
	Auto Reset the Maximum Serial Value Rollback.....	3-12
	Cursor Behavior Mode .....	3-12
	Transaction Isolation Level.....	3-12
<b>3.4</b>	<b>SQL Attributes.....</b>	<b>3-14</b>
	Date & Time Formats .....	3-14
	Stored Procedures Server End Settings.....	3-15
	Stored Procedures Client End Settings .....	3-15
	Default Float Type Setting .....	3-16
	Implicit Type Casting Mode .....	3-16
<b>3.5</b>	<b>DDB Environment .....</b>	<b>3-17</b>
	Enabling Distributed Database Mode.....	3-18
	Remote Database Connection Time-out Value.....	3-18
	Remote Database Lock Time-out Value .....	3-18
	Time Interval to Recover Pending Transaction.....	3-18
	Start Global Transaction Recovery Daemon.....	3-19
<b>3.6</b>	<b>Group Commit .....</b>	<b>3-20</b>
	Group-Commit Threshold .....	3-21
	Maximum Transaction Waiting Time .....	3-21
	Max Group Commit .....	3-21
<b>3.7</b>	<b>Storage (1).....</b>	<b>3-22</b>
	Database Directory.....	3-22
	Name of System Data File.....	3-23

Name of System BLOB File .....	3-23
Name of User Data File .....	3-23
Size of User Data File.....	3-24
Name of User BLOB File.....	3-24
Size of User BLOB File Frames .....	3-24
Extending a File .....	3-25
Auto Extend Tablespace Evenly .....	3-25
Word Sorting Order.....	3-26
Temp Tablespace Directory .....	3-26
<b>3.8 Storage (2) .....</b>	<b>3-27</b>
System Journal Files .....	3-27
User File Objects.....	3-28
System File Object Directory .....	3-29
Files per File Object Subdirectory .....	3-29
User-Defined Function Library .....	3-29
Directories of System Temporary Files .....	3-30
<b>3.9 Cache and Control .....</b>	<b>3-31</b>
Cache and Shared Memory Control.....	3-32
Engine Control .....	3-34
<b>3.10 Backup .....</b>	<b>3-36</b>
Starting Backup Server .....	3-37
Enabling Backup Files Compression .....	3-37
Enabling Backup Read-only Tablespace .....	3-37
Enabling Backup Stored Procedure .....	3-37
Setting the Backup File Directory .....	3-38
Check Database .....	3-38
Incremental Backup Mode Settings.....	3-39
Backup File Object Mode .....	3-40
Settings for the Full Backup .....	3-40
Settings for the Differential Backup .....	3-41
Settings for the Incremental Backup.....	3-42
<b>3.11 Replication.....</b>	<b>3-46</b>
Asynchronous Table Replication Settings.....	3-47

Database Replication Settings .....	3-48
<b>3.12 Start DB .....</b>	<b>3-52</b>
Start Database Mode .....	3-52
Error Report System Mail Settings .....	3-54
Enabling Multi-User Mode .....	3-54
Enabling Forced Start Database .....	3-55
Enabling Access Control List Check .....	3-55
Reset Asynchronous Table Replication System .....	3-55
Check User File .....	3-55
Start I/O Server .....	3-56
File Name of the Stopword List Definition .....	3-56
<b>3.13 Create DB .....</b>	<b>3-57</b>
File Name of Code Order .....	3-57
BLOB Frame Size (bytes) .....	3-59
Converting Identifier to Uppercase .....	3-60
Database Page Size .....	3-60
Language Code .....	3-60
<b>3.14 User Files .....</b>	<b>3-62</b>
Creating a User-Defined File .....	3-62
Creating an Inverted File Text Index Storage Path .....	3-64
<b>3.15 Log System .....</b>	<b>3-65</b>
Server Log Level .....	3-65
Error Log Level .....	3-66
Setting the Slow Statement Logging Threshold .....	3-67
Log SYSINFO .....	3-68
Setting the Maximum Log File Size .....	3-68
Setting the Maximum Number of Log Files .....	3-69
Log SQL Statements .....	3-69
Log Execution Plan .....	3-70
Log Input Parameter .....	3-70
Log Lock Information .....	3-71
Server Log Directory .....	3-71
Log ZIP .....	3-71

Log Keep Days .....	3-72
<b>3.16 CharSet Encoding .....</b>	<b>3-73</b>
Client LCode .....	3-73
Error LCode.....	3-77
<b>3.17 Statistics.....</b>	<b>3-79</b>
Start Statistics Server .....	3-79
Statistics Mode.....	3-80
Settings for the Update Statistics Process.....	3-80
Sample Ratio.....	3-81
<b>3.18 Auto Index.....</b>	<b>3-82</b>
Start Auto Index Server .....	3-82
Start Time of Auto Index .....	3-83
Setting Auto Index Interval .....	3-83
Setting Auto Create Index Threshold.....	3-83
Setting Auto Drop Index threshold .....	3-84
Setting Directory of Auto Index Log File .....	3-84
<b>3.19 Schedule Daemon.....</b>	<b>3-85</b>
Start Schedule Daemon Server .....	3-87
Number of Tasks .....	3-88
Dmschsvr Log Level.....	3-88
Setting Directory of Dmschsvr Log Files .....	3-88





# 1 Introduction

Welcome to the *JConfiguration Tool Reference*. JConfiguration Tool is a cross-platform user-friendly graphical user interface (GUI) that helps users to configure and customize DBMaker databases. Configuration parameters are used to configure DBMaker databases. JConfiguration Tool spares users the task of searching for the keywords that represent these parameters. Instead, parameters are grouped into categories dependent on how they affect the database, and each parameter is given a descriptive tag to help users easily identify and understand how the parameter affects the database.

DBMaker configuration parameters are stored as keywords in a configuration file, **dmconfig.ini**. The descriptive tags and fields in JConfiguration tool represent the keywords in the configuration file. This book shows users the relationship between the keywords and the descriptive tags. Users familiar with configuration files will find this reference helpful in better understanding the relationship between the configuration file and JConfiguration Tool. New users will find this reference useful as a bridge between the JConfiguration Tool and the *Database Administrator's Guide*, which describes most configuration parameters in terms of corresponding configuration file keywords.

This manual gives systematic instructions on how to use the JConfiguration Tool to manage the configuration settings for databases that exist in DBMaker. All program features are introduced with a brief description for each function and screen images are included for most features.

## JConfiguration Tool Reference

---

This book is intended for administrators of DBMaker databases who need to manage databases from a variety of platforms. Administrators using this manual may or may not be familiar with all of the features of DBMaker. Prior knowledge of the platform used to install DBMaker on is required and is beyond the scope of this manual. Refer to the operating system owner's manual for more details on setup and configuration settings for the platform being used.

The *JConfiguration Tool Reference* is organized according to the settings that are displayed on the tabbed pages contained in the main console of the program (see Main Console). The parameters and options contained in JConfiguration Tool main console change settings in the **dmconfig.ini** file.

## **1.1 Additional Resources**

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

- ◆ For an introduction to DBMaker's capabilities and functions, refer to the *DBMaker Tutorial*.
- ◆ For more information on designing, administering, and maintaining a DBMaker database, refer to the *Database Administrator's Guide*.
- ◆ For more information on DBMaker management, refer to the *JServer Manager User's Guide*.
- ◆ For more information on DBMaker functions, refer to the *JDBA Tool User's Guide*.
- ◆ For more information on the native ODBC API and JDBC API, refer to the *ODBC Programmer's Guide* and *JDBC Programmer's Guide*.
- ◆ For more information on the dmSQL interface tool, refer to the *dmSQL User's Guide*.
- ◆ For more information on the SQL language used in dmSQL, refer to the *SQL Command and Function Reference*.
- ◆ For more information on the ESQL/C programming, refer to the *ESQL/C User's Guide*.
- ◆ For more information on error and warning messages, refer to the *Error and Message Reference*.
- ◆ For more information on the DCI COBOL Interface, refer to the *DCI User's Guide*.

### 1.2 Technical Support

CASEMaker provides thirty days of complimentary email and phone support during the evaluation period. When software is registered, an additional thirty days of support will be included, thus extending the total support period for software to sixty days. However, CASEMaker will continue to provide email support for any bugs reported after the complimentary support or registered support has expired (free of 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](mailto: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](http://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
<i>Italics</i>	Italics indicate placeholders for information that must be supplied, such as user and table names. The word in italics should not be typed, but is replaced by the actual name. Italics also introduce new words, and are occasionally used for emphasis in text.
<b>Boldface</b>	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.
<b>NOTE</b>	Contains important information.
➤ <b>Procedure</b>	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
➤ <b>Example</b>	Examples are given to clarify descriptions, and commonly include text, as it will appear on the screen. Other forms of this convention include Prototype and Syntax.
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



# 2 JConfiguration Tool Basics

The JConfiguration Tool is used to change the settings in the **dmconfig.ini** file. All settings correspond to keywords that are described in the appendix of the *Database Administrator's Guide*. The **dmconfig.ini** file contains all information regarding a database that is required when the database is started. Settings that are changed using the JConfiguration Tool are applied to a database the next time the database is started. DBMaker uses default values for settings not specified in the **dmconfig.ini** file when the database is started. The contents of the **dmconfig.ini** file can be viewed in text format in Microsoft Windows platforms by opening the **Start>Programs>DBMaker 5.4>DBMaker Configuration File** submenu item.

Both JServer Manager and JDBC Tool contain links to the JConfiguration Tool. The *JServer Manager User's Guide* and the *JDBC Tool User's Guide* are referenced in this document whenever links between the programs exist.

The JConfiguration Tool has a wide variety of features that allow a user to customize each database. The settings in the **dmconfig.ini** file allow a database administrator to optimize the efficiency of a database by tailoring the database use to meet hardware specifications and limitations.

- ♦ **Storage:** Database source files are stored within the native operating system's root directory on the server. The name and directory path of database files can be specified according to a user's needs. Users can add new files to a full tablespace by storing them as logical files.

- ◆ **Safeguard Data:** All settings for full backup and incremental backup are user-defined.
- ◆ **Connectivity:** Databases can be replicated to remote servers, distributed across multiple servers, and accessed by multiple users using a TCP/IP network. All of the settings for these functions can be manipulated with the JConfiguration Tool to suit the user's needs.
- ◆ **Performance:** Cache and control settings allow the user to control how much memory is allocated for DBMaker processes.



### 2.1 Java Runtime Environment

DBMaker's JTools include JServer Manager, JConfiguration Tool, and the JDBC Tool. JTools are based on the Java Runtime Environment. The Java Development Kit (JDK) or a Java Runtime Environment (JRE) is required for using JTools. JRE 1.7 is automatically installed on the Windows system if DBMaker release 5.4 is installed from disk. Versions of JRE for other platforms are available for downloading from various distributors.

## 2.2 The JConfiguration Tool GUI

The JConfiguration Tool GUI consists of three main elements: the tree view, the menu bar, and the main console. The main console contains the tabbed pages where most database settings are performed.

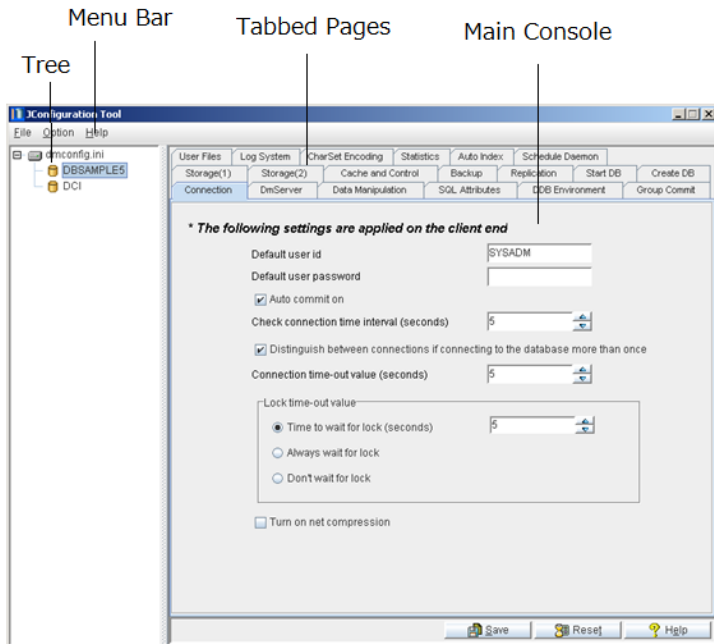


Figure 2-1 JConfiguration Tool Workspace

### Menu Bar

---

The menu bar contains the **File**, **Option**, and **Help** pull down menus. The **File** menu allows a user to exit the program, create a new section, delete a section, or save the **dmconfig.ini** file to another location. The **Options** menu lets the user choose which language the user interface is displayed in. The **Help** menu allows the user to access the help system.

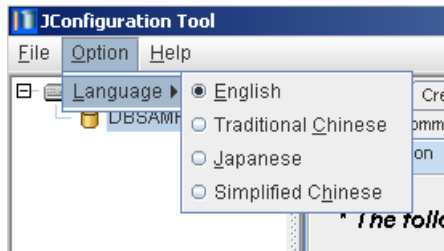


Figure 2-2 The Menu Bar: Language Preference

### Tree View

---

The tree view directory consists of the **dmconfig.ini** file root and all of the databases that are configured by it. Selecting a database from the tree gives access to the configuration of that database. Settings are only applied to the database that has been selected from the tree view.

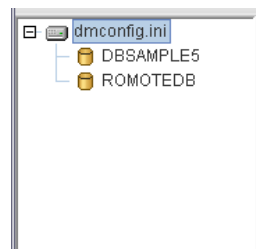


Figure 2-3 Tree Objects

## Main Console

---

The main console consists of seventeen tabbed pages. Users can change any of the settings in a category according to the tab that is selected. Each tabbed page corresponds to a chapter heading in the following sections. Settings are saved by clicking the Save button at the bottom of the console, or they can be reset to the values specified at the time of the previous save by clicking the Reset button.

## The Browse Button

---

### USING THE SELECT FILE / SELECT PATH / SAVE DIALOG BOX

The SQL Attributes, Storage (1), Storage (2), Backup, Replication, Log System, Auto Index, Start DB and Create DB tabbed pages contain settings that require the entry of a file name or a directory path. These may be entered manually or selected by using the browse button (...). Clicking the browse button opens the dialog box shown in Figure 2-4.

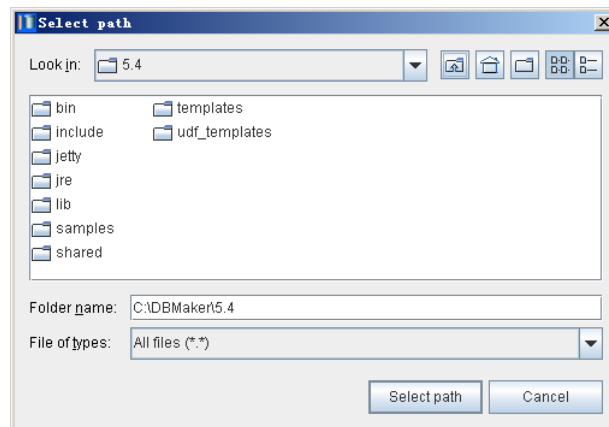


Figure 2-4 The Select Path Dialog Box

Figure 2-4 displays the **Select Path** dialog box. The **Select File** and **Save** dialog boxes differ in function, but are structurally identical. Directly underneath the title bar are the **Look in** drop down list box and five buttons:



*Up One Level* button: sets the active directory equal to the directory that is one level closer to the root directory.



*Home* button: sets the active directory equal to the user's home directory.

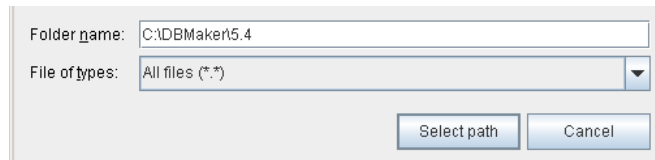


*Create New Folder* button: Creates a new folder labeled "New Folder" in the active directory.



The list and detail buttons are inactive.

The main window, which lists the files and subdirectories of the current directory, is located below the drop-down list box and buttons. A user can browse up and down the directory tree by double-clicking a directory name (down 1 level), or clicking the *up one level* button. Users can select an individual file or directory from the main window by left clicking once on the filename or directory name. A user can also edit any directory or filename from the main window. To edit a filename or directory name, right click the file or directory and make changes to the name. Click the **Save** button to save the revised filename or directory name. Be sure that the name of the selected file or directory appears in the **File name** text field.



*Figure 2-5 File Name and Files of Type text fields*

The **File name** and **Files of type** text fields are displayed below the main window. The **File name** text field should display the name of the file to be selected. A user can then select or cancel the action by clicking on the appropriate button (**Select File** or **Cancel**) located on the right.



# 3 The Main Console

The main console consists of nineteen tabbed pages that correspond to various categories of configuration parameters for the clients and server. Each of the tabbed pages is essential to the configuration, maintenance, and fine-tuning of the databases. The tab page names are:

- ◆ Connection
- ◆ DmServer
- ◆ Data Manipulation
- ◆ SQL Attributes
- ◆ DDB Environment
- ◆ Group Commit
- ◆ Storage (1)
- ◆ Storage (2)
- ◆ Cache and Control
- ◆ Backup
- ◆ Replication
- ◆ Start DB
- ◆ Create DB
- ◆ User Files

- ◆ Log System
- ◆ CharSet Encoding
- ◆ Statistics
- ◆ Auto Index
- ◆ Schedule Daemon



### 3.1 Connection

Clicking the **Connection** tab in the main console opens the **Connection** page. This page displays connection parameters and options set by clients. On this page, the User ID can be changed, the password set, Auto-Commit enabled, client connection status checked by time interval, single-user multiple connections detected, connection time-out value set, lock time-out value set, and net compression enabled.

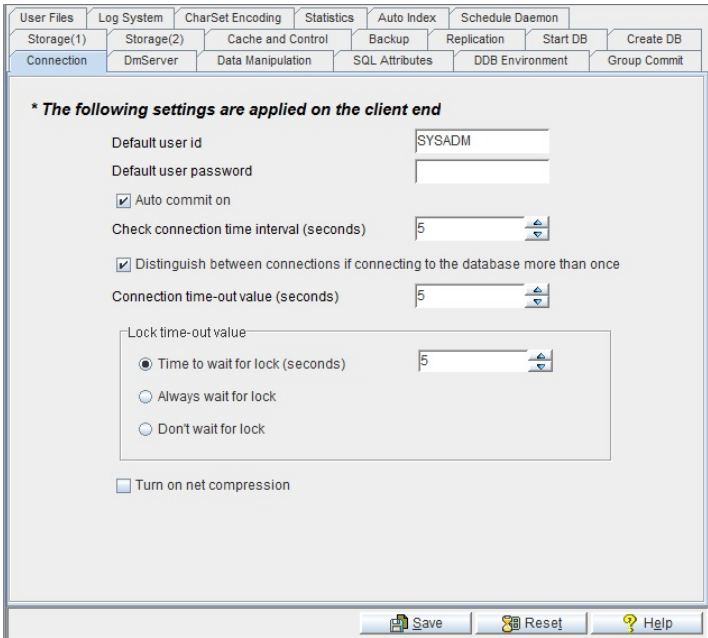


Figure 3-1 Connection Page

### Default User ID and Password

Typing a new default user login ID into the text field changes the user ID that appears when logging on to the database. The password can be likewise modified. Verify that the user has been created for the database before setting it as the **Default User ID**.

Refer to the *JDBA Tool User's Guide* for instructions on the creation of users and groups. These settings correspond to the `DB_UsrId` and `DB_PasWd` keywords in the `dmconfig.ini` file for user ID and password, respectively.

### **Enabling Auto-Commit**

---

SQL transactions are automatically committed after they are executed when **Auto Commit On** is enabled. When **Auto Commit On** is disabled, a `COMMIT TRANSACTION` command must be stated after the SQL command has been executed for it to be committed to the database. This setting corresponds to the `DB_AtCmt` keyword in the `dmconfig.ini` file. The default value is *on*.

### **Setting the Connection Detection Interval**

---

The **Check Connection Time Interval** field value is the time interval in seconds between attempts at client detection. Hardware or other errors can cause connectivity problems between a client and the server; however, DBMaker still allocates resources for that client. It releases allocated resources when the client connection is no longer detected. Setting this value to zero disables automatic client detection. A user may manually enter a value into the combo box, or increase or decrease the value by clicking the arrow buttons to the right. This setting corresponds to the `DB_DtCl` keyword in the `dmconfig.ini` file. The default setting is *0* seconds.

### **Application Multiple Connection Status**

---

This setting specifies the connection behavior attributed to an application that connects to the same database more than once. Enabling **Distinguish connections** allows DBMaker to treat each duplicate connection as a separate connection. Disabling the check box indicates DBMaker to merge all duplicate connecting actions into one connection. This setting is used while connecting to a database.

### Setting the Connection Time-Out Value

---

This value specifies the connection time-out value, in seconds, when a client is trying to connect to the database server. If a database has not been started or the server IP address is wrong, users must wait until the connection times out. Users can increase or decrease the waiting time by adjusting the time-out value. This parameter is set from the client side. A user may manually enter a value into the combo box, or increase or decrease the value by clicking the arrow buttons to the right. This setting corresponds to the `DB_CTimO` keyword in the `dmconfig.ini` file. The default setting is 5 seconds.

### Setting the Lock Time-Out Value

---

The integer in the combo box specifies the lock time-out value in seconds. When the user needs to acquire a lock on a database object, such as a table or a tuple, and that object has already been allocated to another transaction, the user must wait until the object is released. DBMaker waits for the object until a lock is acquired on the object or until the waiting time expires, at which time a lock time-out error message is returned. The waiting time can be shortened if the user prefers not to wait too long. Set the combo box value to -1 or enable **Always Wait for Lock** to disable the time-out. This causes DBMaker to continue waiting until the lock is released. Setting the combo box value to 0 or enabling **Don't Wait for Lock** reduces the user's wait time to zero. Each connection has its own `dmconfig.ini`, so each user can set an individual lock time-out value before connection to the database. This setting corresponds to the `DB_LTimO` keyword in the `dmconfig.ini` file. The default lock time-out value is 5 seconds.

### Enabling Net Compression

---

When enabled **Net Compression** compresses data deliver between servers and clients. Data is compressed before sent from the server and then decompressed as it is received by the client thereby reducing the amount of data transmitted and the result is

## JConfiguration Tool Reference

---

improved performance. This setting corresponds to the `DB_NetZc` keyword in the `dmconfig.ini` file. The default setting is *disabled*.

## 3.2 DmServer

Clicking the **DmServer** tab opens the **DmServer** page. This page displays various network settings related to server connectivity and management. On this page the server address and port number can be set for both the client and server. Enabling network encryption, saving a dmServer log file, and the idle time-out value can only be set for the server. Users can also select the default lock mode used when tables are created.

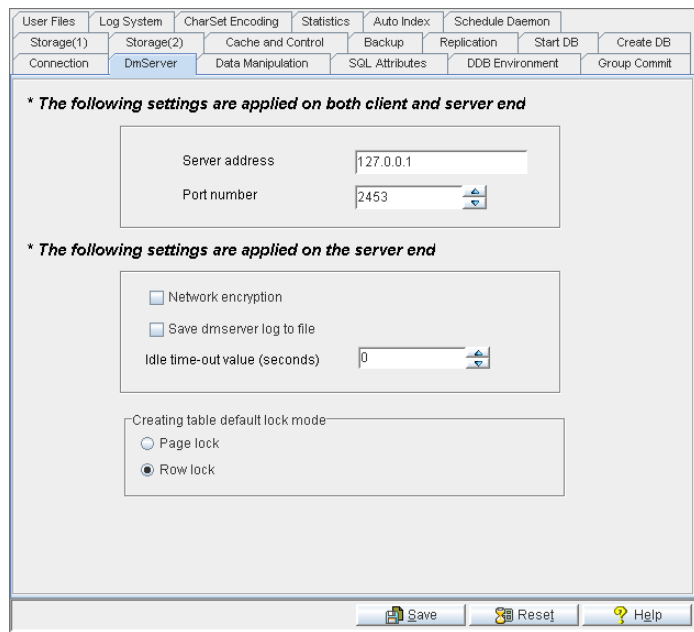


Figure 3-2 DmServer Page

### Client/Server Settings

Client/Server settings must be set on both clients and the server, and includes the IP address and port number of the server machine.

### SERVER ADDRESS

The **Server Address** field must contain a string that specifies the TCP/IP address of the server machine (i.e., server computer) or the host name of the server machine. If Domain Name Server (DNS) has been set up properly on the client machine, you can specify the domain name in the field. The **Server Address** is required at connection time on all client and server machines. If this address is not correct, the connection fails. If you want to run your application in single user database mode, you must leave the **Server Address** field blank. Please contact your local network administrator or a TCP/IP networking manual for more in-depth information. This setting corresponds to the **DB\_SvAdr** keyword in the **dmconfig.ini** file.

### PORT NUMBER

The **Port Number** field should contain an integer that specifies the TCP/IP port number to which the database server is attached. For a specific database, this number must exactly match on all clients and the server, or the connection will fail. This setting corresponds to the **DB\_PtNum** keyword in the **dmconfig.ini** file. The **Port Number** can be any integer value from 1,025 to 65,535.

## Server Settings

---

Server settings only need to be specified on the server machine. The server settings include encryption of network traffic, whether or not to record the dmServer log, and the amount of time to wait before disconnecting idle connections.

### NETWORK ENCRYPTION

This setting specifies if DBMaker should turn network encryption on or off. All network data between DBMaker server and all clients will be encrypted when network encryption is enabled. The encryption technique used by DBMaker consists of a mix of DES and RSA. This setting corresponds to the **DB\_NetEc** keyword in the **dmconfig.ini** file. The default setting is *disabled*.

### SAVING DMSEVER LOG TO FILE

Enabling the **Save dmServer Log to File** setting saves all dmServer command line text to a log file. This function is only available in Windows versions of DBMaker. DBMaker saves it as *<windows directory>\<database name>.log*, where *<windows directory>* is the active directory used by the windows operating system, and saves it in ASCII file format. Enabling this feature allows the database administrator to supervise connections and troubleshoot connection problems. This setting corresponds to the **DB\_SvLog** keyword in the **dmconfig.ini** file. The default value is *disabled*.

### IDLE TIME-OUT VALUE

This field specifies the idle time-out interval in seconds. DBMaker server automatically disconnects connections that are idle for longer than the idle timeout interval. Any database operation resets the idle timeout interval. This feature forces all idle connections to release all database resources, including buffers, pages, locks, and memory. Set the value to 0 to disable this feature; DBMaker keeps all idle connections open. This value must be greater than the value in the **Check Connection Time Interval** field on the **Connection** page. Please refer to *Setting the Connection Detection Interval* in section 3.1 for additional information. If not, DBMaker automatically resets value in the **Check Connection Time Interval** field on the **Connection** page to this value. This setting corresponds to the **DB\_ITimO** keyword in the **dmconfig.ini** file. The default value is *0*.

### Table Settings

---

Users can specify the lock mode that will be used as the default when creating tables. The lock mode can be either **Page lock** or **Row lock**.

## 3.3 Data Manipulation

Selecting the **Data Manipulation** tab opens the **Data Manipulation** page. This page displays settings used to manage upper memory allocation for certain DBMaker client functions. The parameters and options include removal of space padding for CHAR and string concatenation, maximum string size, file type mapping, cursor behavior, and browse mode.

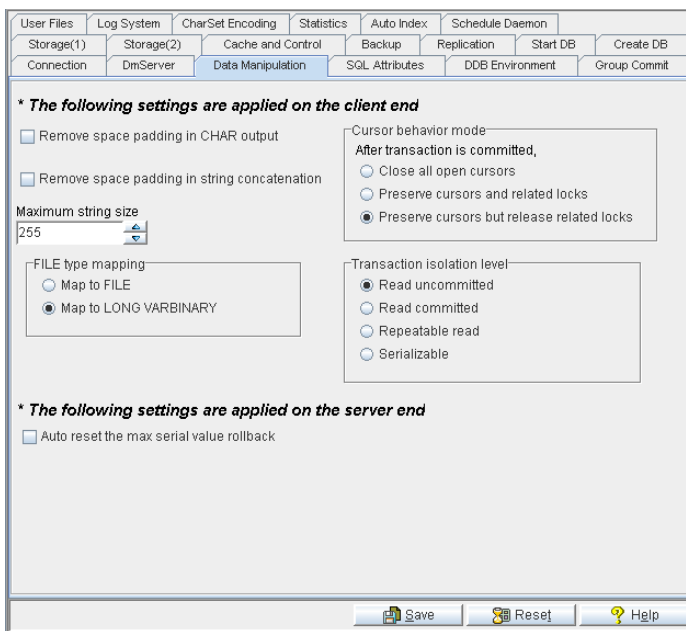


Figure 3-3 Data Manipulation Page

### Remove Space Padding in CHAR Output

This setting specifies whether the space padding for CHAR type data is removed or not. Space padding for CHAR type data in a result set is retained when this feature is disabled. The space padding for all CHAR type data is removed before copying it to



the user buffer if this feature is enabled. It allows user applications to get fixed length CHAR data that excludes the trailing space padding generated in DBMS during data insertion. This setting corresponds to the `DB_RmPad` keyword in the `dmconfig.ini` file. The default setting is *disabled*.

### Remove Space Padding in String Concatenation

---

This setting specifies if the space padding is removed before applying the string concatenation operator (`||`). Disabling this feature allows space padding for fixed length CHAR type data to be kept before applying a string concatenation operator. Enabling this feature indicates the space padding is removed before applying the string concatenation operator. This feature is set on both the client and server. If this feature is not set on the client, then the setting used is the same as what is set on the server. This setting corresponds to the `DB_StrOP` keyword in the `dmconfig.ini` file. The default setting is *disabled*.

### Maximum String Size

---

This field is where the string length for STRING type data is set. STRING type data is used only by user-defined functions (UDF). A UDF can only return data of a fixed size; this setting limits the length of STRING data so clients can avoid receiving an excessively long string. The number can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right. This setting corresponds to the `DB_StrSz` keyword in the `dmconfig.ini` file. The default maximum string length is *255*.

### File Type Mapping Settings

---

This setting specifies what type non-user-defined FILE type data is mapped to. Development tools, such as Borland's Delphi or Microsoft Visual Basic, do not recognize FILE type data because it is not defined by ODBC. To allow these tools to access data of FILE type, select **Map to LONG VARBINARY** Columns defined as FILE type data are then stored as LONG VARBINARY. File type data is then stored as an internal file object in the *FO* directory, and can be accessed by other programs

through the DBMaker engine. This setting corresponds to the `DB_Fotyp` keyword in the `dmconfig.ini` file. The default setting is *Map to LONG VARBINARY*.

### **Auto Reset the Maximum Serial Value Rollback**

---

This setting is used to auto reset serial number to the first value when it hit maximum serial value. Providing an integer value for the optional `START` parameter when defining a `SERIAL` column can specify the first value in a number sequence, or the `START` parameter omitted to use the default value of 1. Disabling this feature will prompt error message when serial value hit max. This setting corresponds to the `DB_RstSn` keyword in the `dmconfig.ini` file. The default setting is *0*.

### **Cursor Behavior Mode**

---

This setting specifies the behavior of the cursor at the completion of a transaction. Selecting **close all open cursors** indicates all cursors that remain open are closed after transactions are committed. Selecting **preserve cursors and related locks** indicates all still open cursors will be kept open after a transaction is committed, and all locks will be preserved; any exclusive locks will become shared. Selecting **preserve cursors but release related locks** indicates all still open cursors will be kept open after a transaction is committed, but all related locks will be released after the end of the transaction. In all cases, the cursor will be closed if any transaction is aborted. This setting corresponds to the `DB_CBMod` keyword in the `dmconfig.ini` file. The default setting is *preserve cursors but release related locks*.

### **Transaction Isolation Level**

---

This setting specifies the lock behavior of a select statement. Selecting `Read uncommitted (dirty read)` will not lock the result set of the `SELECT` statement. Selecting `Read committed` will lock the result set with `S` lock in the duration of the `SELECT` statement. After the statement completing, all the `S` locks that locked by the `SELECT` statement will be released. Selecting `Repeatable read` denotes DBMaker will lock the result set with `S` lock and hold the locks until the transaction completed. Selecting `Serializable` denotes DBMaker will lock the predicate of the `SELECT`

statement with S lock (also known as range lock) and hold the locks until the transaction completed. This setting corresponds to the `DB_IsoLv` keyword in the `dmconfig.ini` file. The default setting is *Read uncommitted*.

## 3.4 SQL Attributes

Selecting the SQL Attributes tab opens the SQL Attributes page, which displays the available options. The settings on this page include separate date and time input and output formats for the clients and server, separate stored procedure directory options for both the clients and server, as well as the default FLOAT type.

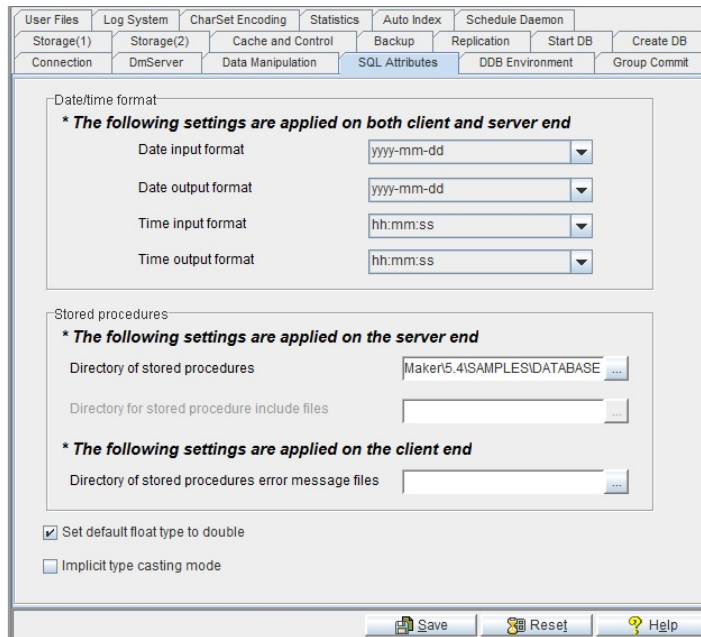


Figure 3-4 SQL Attributes Page

### Date & Time Formats

Date and time input/output formats for SQL statements can be selected from the available formats in the drop down menus. These settings correspond to the DB\_DaiFm, DB\_DaoFm, DB\_TmiFm, and DB\_TmoFm keywords in the dmconfig.ini file. The default value for DB\_DaiFm is none (all input methods

accepted). The default value for `DB_DaoFm` is `yyyy:mm:dd`. The default value for `DB_TmiFm` is none (all input methods accepted). The default value for `DB_TmoFm` is `hh:mm:ss`. Please refer to *Appendix B* in the *ODBC Programmer's Guide* for more information.

## Stored Procedures Server End Settings

---

### STORED PROCEDURES DIRECTORY

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. The user can identify a new location for the stored procedure file directory by entering a directory path in the field or by selecting a path using the browse button (⋮). The default path for the stored procedure directory is `<Database Directory>\`. This setting corresponds to the `DB_SPDir` keyword in the `dmconfig.ini` file.


### DIRECTORY FOR STORED PROCEDURE INCLUDE FILES

The **Directory for Stored Procedure Include Files** specifies the path where stored procedure include files are placed. It is used when the user needs extra include files for stored procedures. This keyword is only useful in non-Windows operating systems. The user can identify a new location for the stored procedure include file directory by entering a directory path in the field or by selecting a path using the browse button (⋮). The default path for the stored procedure include file directory is `<Database Directory>\`. This setting corresponds to the `DB_SPInc` keyword in the `dmconfig.ini` file.

## Stored Procedures Client End Settings

---

The directory shown in the **Directory of Stored Procedures Error Message Files** field specifies the path stored procedure log files are placed in. The stored procedure log files include the error log files that are sent from the database server while creating stored procedure, and the trace log file for stored procedure execution. The user can identify a new location for the stored procedure log file directory by entering a

directory path in the field or by selecting a path using the browse button () . The default path for the stored procedure log directory is *<Working Directory>\*. This setting corresponds to the `DB_SPLog` keyword in the `dmconfig.ini` file.

### **Default Float Type Setting**

---

This setting specifies the byte value used for internal storage of the FLOAT type. Selecting the **Set default float type is double type** check box means that the FLOAT column will use the same 8 byte storage and type name as the DOUBLE type. The FLOAT type and name will be the REAL type when the **Set default float type is double type** check box is not selected and will use 4 bytes of internal storage. This setting corresponds to the `DB_FltDb` keyword in the `dmconfig.ini` file. The default FLOAT type is *DOUBLE*.

### **Implicit Type Casting Mode**

---

Selecting the **Implicit type casting mode** check box means that the DBMaker automatically converts a value from one datatype to another. It mainly contains conversion between numeric data and character data. A numeric data and a character data both include data of multiple types. A numeric data's type can be integer (int, serial), smallint, bigint, bigserial, float, double, and decimal. A character data's type can be char, varchar, nchar and nvarchar. This setting corresponds to the `DB_ItcMd` keyword in the `dmconfig.ini` file. The default setting is *disabled*.

### 3.5 DDB Environment

Selecting the DDB (distributed database) Environment tab opens the DDB Environment page, which displays the available distributed database environment options. The parameters and options available on this page include enabling distributed database mode, remote database connection time-out value, remote database lock time-out value, time interval to recover a pending transaction, and starting the global transaction recovery daemon. The settings are related to distributed databases and are only applied to the server.

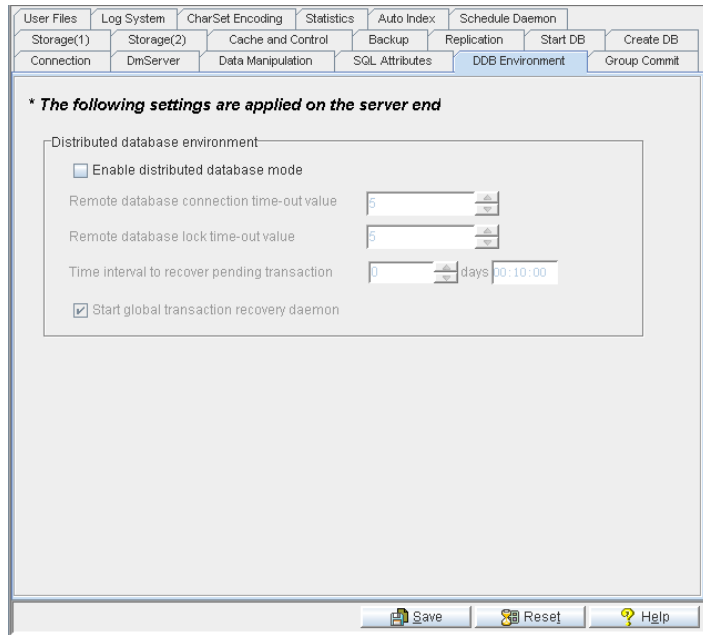


Figure 3-5 DDB Environment Page

### **Enabling Distributed Database Mode**

---

Clicking on the **Enable Distributed Database Mode** check box will enable the database to be used in distributed mode. This setting must be enabled for synchronous table replication to work. This setting corresponds to the `DD_DDBMd` keyword in the `dmconfig.ini` file. The default setting is *disabled*. The following settings apply to the setup of a distributed database. For more in-depth information regarding distributed databases and relevant settings, refer to Section 16 in the *Database Administrator's Guide*.

### **Remote Database Connection Time-out Value**

---

The number in the combo box specifies the time in seconds that the coordinator database should wait when trying to establish a connection to a participant database. The number can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right. This setting corresponds to the `DD_CTimO` keyword in the `dmconfig.ini` file. The default value is *5 seconds*.

### **Remote Database Lock Time-out Value**

---

The number in the combo box specifies the time in seconds that the coordinator database should wait when trying to establish a lock on the requested data in a participant database. The number can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right. This setting corresponds to the `DD_LTimO` keyword in the `dmconfig.ini` file. The default value is *5 seconds*.

### **Time Interval to Recover Pending Transaction**

---

The number in the combo box specifies the time interval at which the global transaction recovery daemon (refer to *Start Global Transaction Recovery Daemon* in this section) is activated 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 the global transaction



recovery daemon to be activated every day and a half. The number of days can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right; the number of hours, minutes, and seconds are manually entered. This setting corresponds to the **DD\_GTI** keyword in the **dmconfig.ini** file. The default value is *10 minutes*.

### **Start Global Transaction Recovery Daemon**

---

Clicking the **Start Global Transaction Recovery Daemon** check box 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 there is an error at the participant database. This setting corresponds to the **DD\_GTSvr** keyword in the **dmconfig.ini** file. The default value is *enabled*.

## 3.6 Group Commit

Selecting the **Group Commit** tab opens the **Group Commit** page, which displays the settings for: threshold, maximum wait time, completely disabled, and maximum number of transactions. 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. It is suitable for large numbers of short transactions running simultaneously. The settings on this page are applied only to the server.

The screenshot shows the 'Group Commit' configuration page within the JConfiguration Tool. The page has a tabbed interface at the top with the following tabs: User Files, Log System, CharSet Encoding, Statistics, Auto Index, Schedule Daemon, Storage(1), Storage(2), Cache and Control, Backup, Replication, Start DB, Create DB, Connection, DmServer, Data Manipulation, SQL Attributes, DDB Environment, and Group Commit. The 'Group Commit' tab is selected. Below the tabs, there is a note: '\* The following settings are applied on the server end'. The settings are: 'Threshold for turning on group-commit function (transactions/second)' with a value of 20; 'Max. transaction waiting time for group-commit (milliseconds)' with a value of 10; and a section for 'Max. group commit' with two radio buttons: 'Disable group commit' (selected) and 'Max. number of transactions waiting for group-commit' (unselected). At the bottom of the page, there are three buttons: Save, Reset, and Help.

Figure 3-6 Group Commit Page

### Group-Commit Threshold

---

The group-commit function will only become 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 this activation threshold. This setting corresponds to the `DB_GcChk` keyword in the `dmconfig.ini` file. The default value is *20 transactions per second*.

### 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)** indicates to the database the longest time interval any one transaction should wait. This setting corresponds to the `DB_GcWtm` keyword in the `dmconfig.ini` file. The default value is *10 milliseconds*.

### Max Group Commit

---

The maximum number of transactions before a group-commit can be specified or this feature can be disabled.

### COMPLETELY DISABLE GROUP COMMIT

Selecting **Completely disable group commit** disables the group commit function.

### MAXIMUM NUMBER OF TRANSACTIONS IN WAIT STATE

Transactions waiting for a group-commit 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 occurs regardless of the waiting time. This number of transactions is specified in the **Max. number of transactions waiting for group-commit** field. This setting corresponds to the `DB_GcMxw` keyword in the `dmconfig.ini` file. The default value is *0*.

## 3.7 Storage (1)

Selecting the **Storage (1)** tab opens the **Storage (1)** page, which displays the available directory storage and file name options. The settings include the database directory, name of the system data file, name of the system BLOB file, name of the user data file, name of the user Blob file, number of the user BLOB file frames, number of pages to extend a full BLOB file when using autoextend tablespaces, and the word sorting order. The settings on this page are only applied to the server.

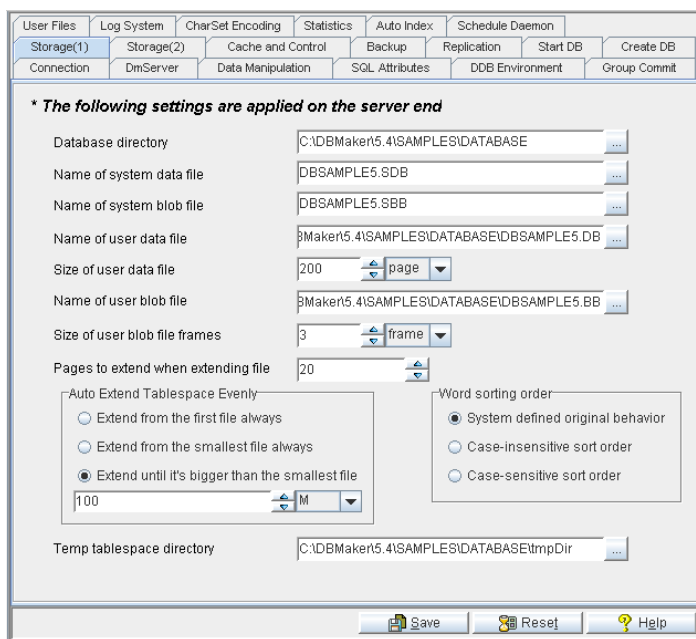


Figure 3-7 Storage (1) Page

### Database Directory

The directory shown here is the default directory for system and user files that belong to the database. Unless specified otherwise, all files created for the database will reside

under this directory. The user can identify a new location for the default database directory by entering a directory path in the field or by selecting a path using the browse button (⋮). When a new database is created, DBMaker will assign a default value of (DBMaker Installation Directory) \5.4\Bin for the database directory. Users creating a new database should change this directory or create a new directory for the new database. This setting corresponds to the `DB_DbDir` keyword in the `dmconfig.ini` file.

### Name of System Data File

---

The user can select a file to be the system data file. The system data file is where database objects such as indexes, views, stored procedures, and synonyms are stored. Objects under one data page in length are stored in this file. This setting corresponds to the `DB_DbFil` keyword in the `dmconfig.ini` file.

### Name of System BLOB File

---

The user can select a file to be the system BLOB file. The system BLOB file is where database objects over one data page in length are stored. This setting corresponds to the `DB_BbFil` keyword in the `dmconfig.ini` file.

### Name of User Data File

---

This is the operating system's file location for storing data. It contains all tables and data type records that belong to the database. Upon creation of a database, JConfiguration Tool automatically assigns a default name that is the same as the database name with the file extension `.DB`. The default directory path is the *Database Directory* path listed above. Users can also enter a new name if a different one is desired. This can be achieved by typing the new name into the text field, or by clicking on the browse button (⋮) next to the text field. Users can select another directory for the user data file if the default directory may not have enough space for all the data to grow into. This can be achieved by typing the new directory path into the text field, or by clicking on the browse button (⋮) next to the text field. This setting corresponds to the `DB_UsrDb` keyword in the `dmconfig.ini` file.

### Size of User Data File

---

Under the **Name of the user Data File** field is the **Size of User Data File** field. This is used to control the size of the User Data File upon its creation. There are three unit options for Data file size: Page, M (megabytes), and G (gigabytes). Please note that the actual size will be 1 page less than the specified value when using M or G. A higher page number or larger size allocates more space in this file and allows for the insertion of more record data before the file is automatically extended (refer to *Extending a File* in this section), a lower page number or a smaller size allocates less space. This number can be manually entered or adjusted using the arrows to the right. The default value is **200 pages**.

### Name of User BLOB File

---

This is the operating system's file location for storing BLOB data. It contains all BLOB type records that belong to the database. Upon creation of a database, JConfiguration Tool automatically assigns a default name that is the same as the database name with the file extension .BB. The default directory path is the *Database Directory* path listed above. Users can also enter a new name if a different one is desired. This can be achieved by typing the new name into the text field, or by clicking on the browse button (⋮) next to the text field. Users can select another directory for the user BLOB file if the default directory may not have enough space for all the data to grow into. This can be achieved by typing the new directory path into the text field, or by clicking on the browse button (⋮) next to the text field. This setting corresponds to the **DB\_UsrBb** keyword in the **dmconfig.ini** file.

### Size of User BLOB File Frames

---

Under the **Name of User BLOB File** field is the **Size of User BLOB File Frames** combo box. This is used to control the size of frames that initially exist in the User BLOB File upon its creation. There are three unit options for BLOB file size: Page, M (megabytes) and G (gigabytes). The actual size is 1 frame less than the specified value when using M or G. A greater frame number or larger size allocates more space in this file and allows for the insertion of more BLOB data before the file is automatically

extended. A lower frame number or smaller size allocates less space. Manually enter and adjust these values using the arrows. The default value is **3 frames**. Please refer to *Extending a File* in this section for additional information.

### **Extending a File**

---

When all pages in the data file or BLOB file are full, DBMaker automatically extends the number of pages or frames in the file to allow the database to grow. The **Number of Pages to Extend While Extending File** setting tells DBMaker how many pages or frames to add to the full file in the event that it is full. If the database administrator expects the database will grow quickly, then a higher number is prudent to reduce frequently appending the file. Manually enter and adjust this number using the arrows. This setting corresponds to the **DB\_ExtNp** keyword in the **dmconfig.ini** file. The default value is **20 pages**.

### **Auto Extend Tablespace Evenly**

---

When a tablespace is full, DBMaker automatically extends the number of pages or frames in the file to allow the tablespace to grow. The **Extend from the first file always** setting tells DBMaker automatically expand the tablespace from the first file always, it can obtain good performance, but all files in the tablespace are not be expanded evenly. The **Extend from the smallest file always** setting keeps all files in a tablespace in even expansion, but performance may be bad because all rows of a table may be scattered into all files in turn. The **Extend until it's bigger than the smallest file** setting expands from the smallest file firstly, after that, continue to expand this file and does not switch to the second smallest file until the current file's size is bigger than the sum of the second smallest file's size and the value of **DB\_ExtHd**. In this way both performance and balance of file size can be taken into account. The **Extend until it's bigger than the smallest file setting** can be altered by inputting a value in the field or by using the arrows to the right of the field. There are three unit options for page size: Page, M (megabytes), and G (gigabytes). This setting corresponds to the **DB\_ExtHd** keyword in the **dmconfig.ini** file. The default value is **100M**.

### 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 information refer to Chapter 4.2 in the *Database Administrator's Guide*.

### Temp Tablespace Directory

---

The temporary tablespace (TMPTABLESPACE) is only used to store external temp tables (ETT). The temporary tablespace also is an auto-extend tablespace. It has exactly two files: **DB\_TMPDB** for saving data and **DB\_TMPBB** for saving blob. Default TMPTABLESPACE directory is **DB\_DbDir\tmpDir**.



## 3.8 Storage (2)

Selecting the **Storage (2)** tab opens the **Storage (2)** page, which displays the available file storage options. The settings on this page include names and paths for journal files, journal file size (in pages), enabling/disabling user file objects, the system file object directory, maximum number of file objects per/subdirectory, directory of user-defined function library files, and directories of system temporary files. The settings on this page are only applied to the server.

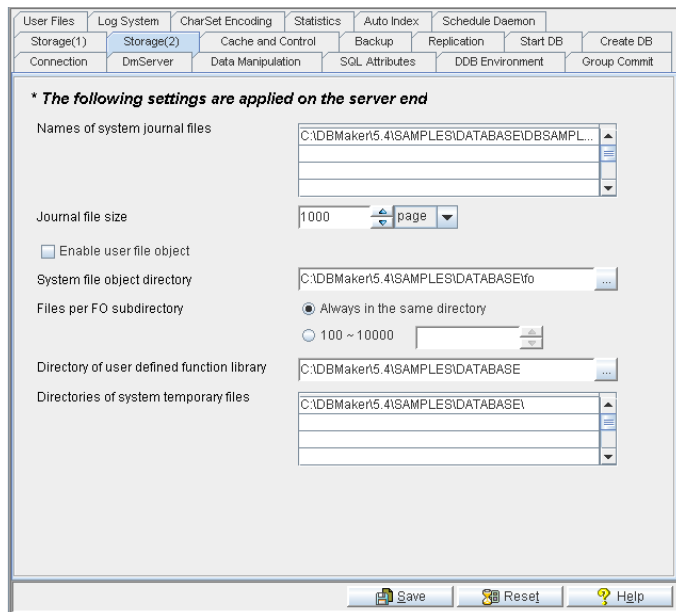


Figure 3-8 Storage (2) page

### System Journal Files

System journal files record the result of a transaction on a set of data. The journal file essentially is a record of all changes made to data in the database. The **Names of**

**System Journal Files** lists the journal files currently used by the database. Up to eight journal files can be specified. If more than one journal file is listed, journal entries are entered into the first journal file until every journal file block is full, then journal entries are entered into the first blocks of the next journal file in the list. Journal entries are recorded in a sequential fashion until all journal files are full. After all journal files are full, journal blocks are overwritten starting with the first blocks of the first journal file listed. Upon creation of the database, JConfiguration Tool gives the journal file a default name that is the same as the database name with the file extension `.jnl`. The default directory path is the `<Database Directory>` path. Users can also enter a new name if a different one is desired. This can be achieved by typing the new name into the text field. Users can select another disk for the journal file to improve data transfer efficiency. This can be achieved by typing the new full directory and file path into the text field. This setting corresponds to the `DB_JnFil` keyword in the `dmconfig.ini` file.

The **Journal File Size** field is under the **Names of System Journal Files** field. There are three unit options for page size: Page, M (megabytes), and G (gigabytes). Please note that the actual size will be 1 page less than the specified value when using M or G. This setting corresponds to the `DB_JnlSz` keyword in the `dmconfig.ini` file.

## User File Objects

---

Users can create a column of the *file object* data type. This creates a column with records that are direct references to files. The referenced files can be user file objects or system file objects. User file objects are stored on the DBMaker server.

If user files are deleted or moved, DBMaker will not recognize them and an error will be returned when it attempts to access the non-existent files. Only when the **Enable User File Object** check box is selected the database will be able to create user file objects. The default setting is *disabled*. This setting corresponds to the `DB_UsrFo` keyword in the `dmconfig.ini` file.

### System File Object Directory

---

System file objects are referenced files stored in the system file object directory. System file objects are file type data that is inserted into a file type column. File type data inserted this way is automatically saved in the **System File Object Directory**. System file objects are internal to DBMaker. When a database created, the default path for the system file object directory is *<Database Directory>|FO*. 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 click the browse button (⋮) next to the text field. Changes to the file object path only apply when starting the database. This setting corresponds to the **DB\_FoDir** keyword in the **dmconfig.ini** file.

### Files per File Object Subdirectory

---


DBMaker supports the automatic creation of subdirectories within the FO directory. Each subdirectory is filled, to a threshold value, with new file objects. When the threshold value is reached, DBMaker creates a new FO subdirectory. The number in the **Files per FO subdirectory** field specifies this threshold value. To store all file objects in the FO directory select **Always in the same directory**. This setting corresponds to the **DB\_FoSub** keyword in the **dmconfig.ini** file. The default setting is *Always in the same directory*.

### User-Defined Function Library

---

User-defined functions are compiled functions stored in a dynamic link library (DLL) and access by the user from within 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 the browse button (⋮) next to the text field. This setting corresponds to the **DB\_LbDir** keyword in the **dmconfig.ini** file.

### Directories 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 storage directories of the system temporary files may be specified. These files are removed when the database is shut down. If one or more file paths 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 the browse button () next to the text field. This setting corresponds to the **DB\_IttDir** keyword in the **dmconfig.ini** file.

### 3.9 Cache and Control

Selecting the **Cache and Control** tab opens the **Cache and Control** page. This page displays options available for performance tuning when using the database communication and control area (DCCA). Cache and control settings manipulate the amount of upper memory used by an active database, control the maximum number of users that can access an active database and control locks. The journal buffer, data buffer and system control area are all components of the database communication and control area (DCCA). Adjusting these settings influences database performance. The settings on this page are only applied to the server.

For more information on performance tuning, please refer to Chapter 18 in the *Database Administrator's Guide*.

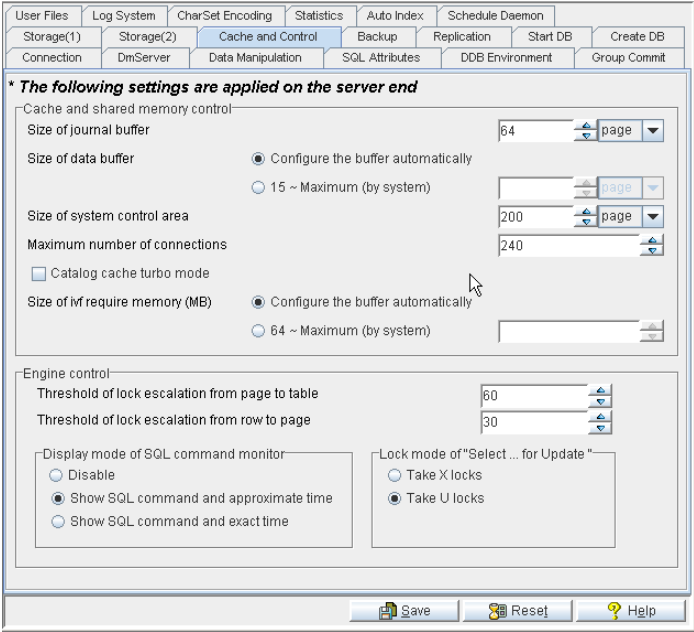


Figure 3-9 Cache and Control Page

### Cache and Shared Memory Control

---

#### JOURNAL BUFFER

The journal buffer stores the most recently used journal blocks as pages. The time required to write journal blocks to disk when data is updated and to read journal blocks from disk when a transaction is rolled-back is reduced when sufficient journal buffer pages are available. The **Size of Journal Buffer** setting can be altered by inputting a value in the field or by using the up and down arrows to the right of the field. There are three unit options for page size: Page, M (megabytes), and G (gigabytes). Please note that the actual size will be 1 page less than the specified value when using M or G. This setting corresponds to the **DB\_NJnlB** keyword in the **dmconfig.ini** file. The default value is **64 pages**.

Please refer to the section *Tuning Journal Buffers* in Chapter 18 of the *Database Administrator's Guide* for more information.

#### DATA BUFFER

The data buffer stores the most recently used data blocks as pages in memory. DBMaker uses the most recently used algorithm to determine which data pages are buffered. The data buffer can be manually set to a specific value or DBMaker can configure it automatically. To have DBMaker set the number of page buffers, select **Configure the buffer automatically**. The **Size of Data Buffer** setting can be altered by inputting a value in the field or by using the up and down arrows to the right of the field. There are three unit options for page size: Page, M (megabytes), and G (gigabytes). Please note that the actual size will be 1 page less than the specified value when using M or G. Increasing or decreasing the size of the data buffer dramatically affects database performance. This setting corresponds to the **DB\_NBufs** keyword in the **dmconfig.ini** file. The default value is **Configure the buffer automatically**.

Refer to the section *Tuning Page Buffer Cache* in Chapter 18 of the *Database Administrator's Guide* for more in-depth information on optimizing this value.

### SYSTEM CONTROL AREA

Concurrency control blocks are stored in the System Control Area (SCA). Large transactions requiring many lock resources need a larger SCA to function properly. If too many resources are being allocated for concurrency control, the SCA can be enlarged, or the threshold of lock escalation can be decreased. Please refer to *Engine Control* later in this section for additional information. The **Size of System Control Area setting** can be altered by inputting a value in the field or by using the arrows to the right of the field. There are three unit options for page size: Page, M (megabytes), and G (gigabytes). Please note that the actual size will be 1 page less than the specified value when using M or G. This setting corresponds to the **DB\_ScaSz** keyword in the **dmconfig.ini** file. The default value is **200 pages**. Please refer to the section *Tuning the System Control Area* in Chapter 18 of the *Database Administrator's Guide* for more information on optimizing this value.

### MAXIMUM NUMBER OF CONNECTIONS

This setting controls the maximum number simultaneous database connections. The **Maximum Number of Connections** setting can be altered by inputting a value in the field or by using the up and down arrows to the right of the field. This setting corresponds to the **DB\_MaxCo** keyword in the **dmconfig.ini** file. The default value is **240**.

### CATALOG CACHE TURBO MODE

Enabling the **Catalog Cache Turbo Mode** extends the lifetime of the catalog cache in the SCA. Please refer to *Tuning the Catalog Cache* in Chapter 18 of the *Database Administrator's Guide* for more in-depth information. This setting corresponds to the **DB\_Turbo** keyword in the **dmconfig.ini** file. The default setting is *disabled*.

### SIZE OF IVF REQUIRED MEMORY

This keyword is used to specify the size of the cache used for inverted file text index storage in kilobytes. This setting corresponds to the **DB\_IFMem** keyword in the **dmconfig.ini**. The default value is *Configure the buffer automatically*.

### Engine Control

---

#### THRESHOLD OF LOCK ESCALATION FROM PAGE TO TABLE

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 page locks to a table lock. This setting corresponds to the `DB_LetPT` keyword in the `dmconfig.ini` file. 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 row locks to a page lock. This setting corresponds to the `DB_LetRP` keyword in the `dmconfig.ini` file. The default value is *30 rows*.

#### DISPLAY MODE

**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 **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 reduces database performance. This setting corresponds to the `DB_SQLSt` keyword in the `dmconfig.ini` file. 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 denotes DBMaker will take X locks on the result set of all “select ... for update” statements. This setting corresponds to the **DB\_ForUX** keyword in the **dmconfig.ini** file.

## 3.10 Backup

Selecting the **Backup** tab opens the **Backup** page. This page displays the options available for incremental, differential and full backups. The settings on this page determine the actions of the backup server daemon and specify the location to store backup files. The settings on this page are only applied to the server.

For more information on backup modes and procedures, please refer to Chapter 15 in the *Database Administrator's Guide*.

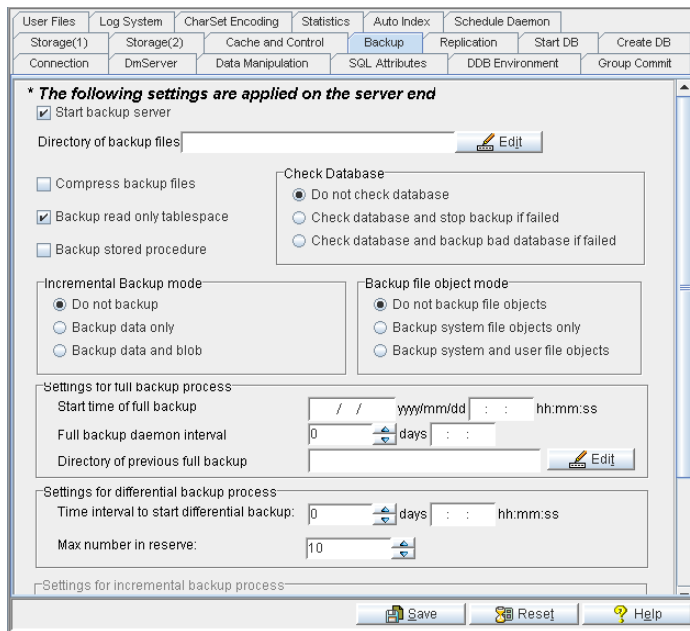


Figure 3-10 Backup Page

### Starting Backup Server

---

Selecting the **Start Backup Server** check box activates the backup server daemon. All data, system, and journal files are periodically copied directly to the backup directory when the backup server daemon is activated. Users can also access Backup Server settings in the **Settings for Full Backup Process**, **Settings for Differential Backup Process** and **Settings for Incremental Backup Process** fields. These settings must be configured properly for the backup process to execute. This setting corresponds to the **DB\_BkSvr** keyword in the **dmconfig.ini** file. The default setting is *disabled*.

### Enabling Backup Files Compression

---

Selecting the **Compress backup files** check box activates backup file compression. The storage needed for full-backups of database files is reduced using this option. Backup Server and JServer Manager utilize this function when full-backup files are compressed. This setting corresponds to the **DB\_BkZip** keyword in the **dmconfig.ini** file. The default setting is *disabled*.

### Enabling Backup Read-only Tablespace

---

Selecting the **Backup read only tablespace** check box allows the backup process to backup read-only tablespace files. When the check box is disabled the read-only tablespace files are not backed up. This setting corresponds to the **DB\_BkRTs** keyword in the **dmconfig.ini** file. The default setting is *enabled*.

### Enabling Backup Stored Procedure

---

Selecting the **Backup stored procedure** check box allows the backup process to backup stored procedures. When the check box is disabled the stored procedures are not backed up. This setting corresponds to the **DB\_BkSPm** keyword in the **dmconfig.ini** file. The default setting is *disabled*.

### Setting the Backup File Directory

---

The **Directory of Backup Files** field shows the directory where Backup Server saves files. You should create a backup directory on a disk different from that where the database files reside to prevent loss of both the database and the backup files in the event of a media error. Multiple backup paths can be specified. The default path for backup files is *<Database Directory>\Backup* and is automatically created by DBMaker. The total length of any backup directory path must not exceed 256 characters. Users should enter a new path for the backup file directory by typing the new path into the text field, or by clicking the *Edit* button next to the text field. This setting corresponds to the **DB\_BkDir** keyword in the **dmconfig.ini** file. Please consider the following items when setting multiple backup paths.

- ◆ When a database system backs up files, it will try to store files in the paths one by one for each file. For example, if you want to store a file to backup directory DIR\_1 but DIR\_1 does not have enough space to store the file, then DBMaker tries to store it to backup DIR\_2, DIR\_3, and so on. Files are directed to backup paths by file size. For example, consider two directories called dir1 and dir2. The file sizes are as follows: db\_bkdir = dir1 33 dir2 44. Backup files larger than 132 KB (i.e.,  $33 \times 4$  if the page size is set to 4 KB) are backed up into dir2.
- ◆ Only the first backup directory can accept slave site backup files
- ◆ File objects must be backed up in the first backup directory
- ◆ The maximum number of backup paths that can be specified are 32

**NOTE** *The backup directory file size is calculated as follows: The file size number multiplied by the value of DB\_PgSiz from dmconfig.ini file For example setting DB\_PgSiz to 4 K and the file size to 44 means the physical file size is  $44 \times 4$  KB = 176 KB.*

### Check Database

---

The **Check Database** field specifies whether the database is checked before full and differential backups. Selecting the **Do not check database** mode means the database is not checked before full and differential backups. Selecting **Check database and stop**

**backup if failed** causes a database check before full and differential backups. Additionally, Backup Server writes any error message and stops the backup if find the database is damaged. Selecting **Check database and backup bad database if failed** causes a database check before full and differential backups. Backup Server writes any error message and if the database is damaged, continues to back up the bad database to *BKDIR\BADDB* directory. Damaged database are only backed up once directly after Back Server first finds the damaged database. During the next backup, if the database is found to be undamaged, Backup Server removes the damaged backup residing in *BKDIR\BADDB* directory and continues with the normal backup process. If the database is still damaged an error is generated and the backup operating terminates. This setting corresponds to the keyword `DB_BkChk` found in the `dmconfig.ini` file. The default is *Do not check database*.

### Incremental Backup Mode Settings

---

The settings in the **Incremental Backup Mode** field specify the incremental backup mode for a database. The Backup Server must be started for modes other than **Do not backup** function. Selecting **Do not backup** enables **NON-BACKUP** mode. This mode causes the journal file's oldest records to be overwritten when the journal file is full. Selecting **Backup Data Only** enables **BACKUP-DATA** mode. This mode allows for a full recovery from an instance failure, and full recovery of non-BLOB data in a media failure. Selecting **Backup Data and BLOB** enables **BACKUP-DATA-AND-BLOB** mode. This mode allows for full recovery of all data. These settings apply only to the operation of journal files, and therefore to the operation of the incremental backup process. Selecting **Backup Data Only** or **Backup Data and BLOB** allows the user to access settings in the **Settings for Incremental Backup Process** field. These settings must be configured properly for the incremental backup process to work. Please see for additional information. For more in-depth information on backup modes, please refer to Chapter 15 of the *Database Administrator's Guide*. This setting corresponds to the `DB_BMode` keyword in the `dmconfig.ini` file. The default mode is *Do not backup*.

### Backup File Object Mode

---

The settings under the **Backup File Object Mode** effect how file objects are copied during a full backup process. Selecting **Do Not Backup File Objects** disables file backup during the full backup process. Selecting **Backup System File Objects Only** causes system file objects being backed up during automatic full backups. Selecting **Backup System and User File Objects** causes both system file objects and user file objects being copied to the backup directory during automatic full backups. This setting corresponds to the **DB\_BkFoM** keyword in the **dmconfig.ini** file. The default mode is *Do not backup file objects*.

### Settings for the Full Backup

---

The following settings configure the full backup process and must be set for the backup daemon to function properly.

#### SETTING THE START TIME FOR FULL BACKUPS

Specify when the first full backup will begin for the database by entering a date in the **yyyy/mm/dd** field and a time in the **hh:mm:ss** field. Reenter the numbers if they appear incorrectly the first time. This setting corresponds to the **DB\_FBkTm** keyword in the **dmconfig.ini** file. There is no default value.

**NOTE** *Full backups are processed only when the full backup start time is set.*

#### SETTING THE FULL BACKUP DAEMON INTERVAL

The number in the combo box specifies, in days, the time interval at which the Full Backup Daemon is activated. 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 in **days** and inputting 12:00:00 in **hh:mm:ss** causes the Full Backup Daemon to be activated every day and a half. The number of days can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right; the number of hours, minutes, and seconds are manually entered. This setting corresponds to the **DB\_FBkTv** keyword in the **dmconfig.ini** file.

### SPECIFYING THE PREVIOUS FULL BACKUP DIRECTORY

All old backup data residing in the backup directory at the time of backup is rewritten to the **Directory of Previous Full Backup**. This previous full backup directory corresponds to the **DB\_BkOdr** keyword. Old backup information already in the **Directory of Previous Full Backup** will be overwritten in the instance of a full backup unless the directory name is changed or the old backup file names are changed. As with the **Directory of Backup Files**, the **Directory of Previous Full Backup** should reside on a disk or system separate from the *Database Directory* to ensure that data can be recovered in the event of a media failure. The total length of any backup directory path must not exceed 256 characters in length. Users must enter a path for the backup file directory by typing the path into the text field, or by clicking on *Edit* button beside the text field. DBMaker will not copy the previous full backup files if no **Directory of Previous Full Backup** is specified. This setting corresponds to the **DB\_BkDir** keyword in the **dmconfig.ini** file.

When setting multiple backup paths you should keep the following in mind:

- ◆ When database system backs up files, it will try to store files in the paths one by one for each file. For example, if you want to store a file to backup directory DIR\_1 but DIR\_1 does not have enough space to store the file, then DBMaker will try to store it to backup DIR\_2 or DIR\_3 and so on. If all backup directories are full, an error message will be returned stating that all backup directories are full.
- ◆ Only one backup directory at a time is used to backup files in the slave site.
- ◆ File objects must be backed up in the first backup directory.

The maximum number of backup paths that can be specified is **32**.

### Settings for the Differential Backup

---

The following settings are available only when **Start Backup Server** is enabled. A full backup, in this case also called the Differential Base, must exist before setting up the differential backup process.

### SETTING THE DIFFERENTIAL BACKUP DAEMON INTERVAL

The number in the combo box specifies the time interval at which the Differential Backup occurs in number of days. Next to the combo box, there is a field for time input, which specifies the time interval in hours, minutes, and seconds. The total time interval is determined by adding the two values together, so inputting 1 into the **days** field and inputting 12:00:00 into the **hh:mm:ss** field would cause an incremental backup to occur every day and a half. The number of days can be manually entered into the combo box, or increased or decreased by clicking the up and down arrows to the right of the field. The number of hours, minutes, and seconds can only be manually entered. This setting corresponds to the **DB\_DbKtv** keyword in the **dmconfig.ini** file.

### MAX NUMBER IN RESERVE

The number in the box specifies the maximum number of differential backup to be retained after a full backup (i.e., differential base). Backup Server removes the oldest differential backup when the number of differential backups after a full backup exceeds the maximum number value. This setting corresponds to the keyword **DB\_DbKmx** found in the **dmconfig.ini** file.

## Settings for the Incremental Backup

---

The following settings become available only when **Start Backup Server** has been enabled and the user has selected **Backup Data Only** or **Backup Data and BLOB**. The following settings are needed for the Incremental Backup Process to function properly.

### SETTING THE JOURNAL BACKUP FILE FORMAT

The backup filename format allows you to specify the format Backup Server will use to name incremental backup files. The backup filename format may include both text constants and must include format sequences (escape sequences) that represent special character strings.



An incremental backup file name must consist of at least three special character strings: the full backup id, the database name, and the backup identification number. Backup Server assigns a full backup ID when naming incremental files in a backup sequence. When restoring a database, DBMaker uses the full backup ID to correctly recreate the backup sequence. The database name correctly identifies the database to which an incremental backup file belongs. The backup identification number identifies the relative position of the incremental backup file in the backup sequence.

Format sequences have three parts: the escape character, the length value, and the format character. Valid format sequences are:

**%[x]F** — The full backup ID. The variable *x* may have values 1 through 4 where the values represent the following formats:

1: full backup id shown as YYYYMMDD, e.g., 20010917

2: full backup id shown as MMDD, e.g., 0917

3: full backup id shown as MMDDhhmm, e.g., 09171305

4: full backup id shown as DDhhmmss, e.g., 17130558

**%[n]B** — The backup identification number

**%[n]N** — The name of the database associated with the journal file

### ➤ Example

```
DB_BkFrm = %3F%n.%B
```

If the database name is test1, the incremental backup files will be named *09171305test1.1.jnl*, *09171305test1.2.jnl*...

For more in-depth information, refer to the section *Setting the Backup Filename Format* in Chapter 15 of the *Database Administrator's Guide*. This setting corresponds to the **DB\_BkFrm** keyword in the **dmconfig.ini** file. The default file name format is *%2F%4N%4B.jnl*.

### SETTING THE START TIME FOR INCREMENTAL BACKUPS

To set the time at which the first incremental backup will start being processed for the database, enter the date in the `yyyy/mm/dd` field, and the time in the `hh:mm:ss` field. Reenter the numbers if they appear incorrectly the first time. This setting corresponds to the `DB_BkTim` keyword in the `dmconfig.ini` file. There is no default value.

### SETTING THE INCREMENTAL BACKUP DAEMON INTERVAL

The number in the combo box specifies the time interval at which the Incremental Backup occurs in number of days. Next to the combo box, there is a field for time input, which specifies the time interval in hours, minutes, and seconds. The total time interval is determined by adding the two values together, so inputting 1 into the `days` field and inputting 12:00:00 into the `hh:mm:ss` field would cause an incremental backup to occur every day and a half. The number of days can be manually entered into the combo box, or increased or decreased by clicking the up and down arrows to the right of the field. The number of hours, minutes, and seconds can only be manually entered. This setting corresponds to the `DB_BkItv` keyword in the `dmconfig.ini` file.

### ENABLING COMPACT BACKUP

Checking the **Enable Compact Backup** checkbox specifies whether Backup Server will backup entire journal files or only full journal blocks when it performs an online incremental backup. If the **Enable Compact Backup** checkbox is enabled, then the Backup server will only backup journal blocks not previously backed up. Not every journal block contains data needed to restore a database, so Backup Server will only copy the necessary journal blocks when it performs a backup. This allows the user to save storage space on the backup media, but it also means that restoring a database may take more time. This setting corresponds to the `DB_BkCmp` keyword in the `dmconfig.ini` file. The default value is *enabled*.

### INCREMENTAL BACKUP INITIATION VALUE

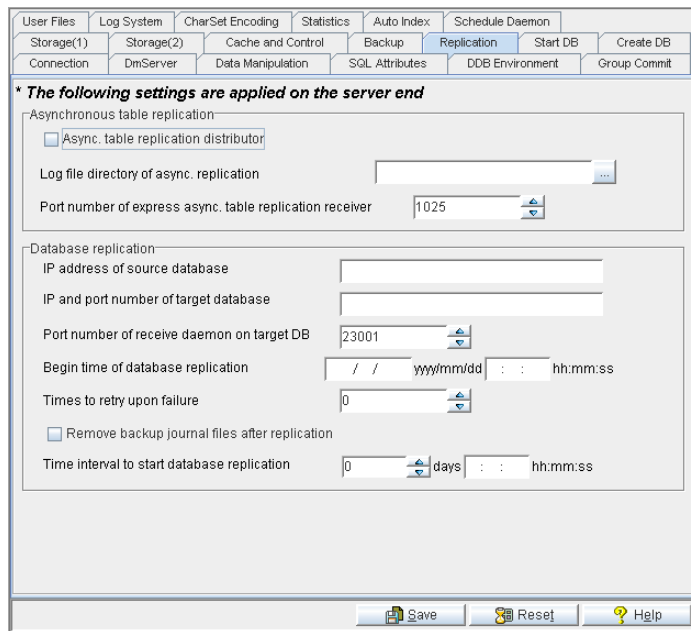
The user may want to allow DBMaker to create an incremental backup before the journal file is completely full. The journal trigger value specifies the percentage a

journal file must fill before Backup Server will perform an online incremental backup. The journal trigger value and the backup schedule can be combined to backup a database on a regular schedule and when journal files fill to a specified percentage.

Selecting **Backup when any Journal File is Full** sets the Incremental Backup Process to activate any time the journal file is full. Selecting **50-100** allows the user to input a value into the % **full** combo box. Users can manually input a value between 50% and 100% or use the up and down arrows to the right of the field to adjust this value. The journal file will be backed up any time it is filled to the percentage specified. This setting corresponds to the **DB\_BkFul** keyword in the **dmconfig.ini** file. The default value is *90*.

## 3.11 Replication

Selecting the **Replication** tab opens the **Replication** page, which displays the available table replication and database replication options. The settings on this page are only applied to the server. Table replication allows data to be shared across databases. Changes are made by clients to the source database and then transferred to destination databases by one of two methods. The first method is synchronous replication, whereby changes to the source table are simultaneously made to the destination table. Distributed database mode must be on for synchronous table replication to work. The second method is asynchronous replication; a schedule is first defined for the distributor daemon, then SQL statements extracted from the transaction log periodically update the destination table.



*Figure 3-11 Replication Page*

### Asynchronous Table Replication Settings

---

Asynchronous table replication (ATR) periodically writes data from the distributor database to destination tables based on a schedule. The schedule may be set using JDBA Tool or manually executed in dmSQL with SQL commands. To set the schedule using JDBA Tool, please refer to the *JDBA Tool User's Guide*. The schedule SQL command sequence consists of four parameters: the destination database to replicate to, the start time of the first replication, the time interval at which replication takes place, and a user password. The user password must correspond to a user that has privilege to insert, delete, and update records on the remote database.

#### ➤ Example

How a schedule is created for the remote database DESTDB.

```
dmSQL > CREATE SCHEDULE FOR REPLICATION TO destdb
        BEGIN AT 2000/1/1 00:00:00
        EVERY 12:00:00
        IDENTIFIED BY User Password;
```

For the above example, all destination tables on the DESTDB database are updated starting at midnight, New Year's Eve. Destination tables are subsequently updated every day at midnight and noon. Note that a source database can have multiple schedules on different databases, and can write to databases that use different database engines. Any time the source database has a change written to a source table, the change is recorded in the replication log file directory as one of many replication log files. When the distributor daemon is activated, it translates these files into ODBC calls that are then sent to the destination databases. Using ODBC calls to update destination tables allows ATR to be performed onto any type of database that supports ODBC. This is known as Heterogeneous ATR. Replication log files that have been sent to the distributor daemon are deleted after use, and information about them recorded in the ATRP.LOG file in the *Database Directory*. Errors returned by the distributor or subscriber daemons on the destination database are recorded in the ATRERROR.LOG file in the *Database Directory*. For more in-depth information on ATR, please refer to the *Database Administrator's Guide*.


### ENABLING ATR DISTRIBUTOR

Clicking the check box next to **Asynchronous Table Replication Distributor** activates the distributor daemon. The schedule of the distributor daemon is dependent on the remote (destination) database and is specified using the SQL CREATE SCHEDULE command, specified above. This setting corresponds to the **DB\_AtrMd** keyword in the **dmconfig.ini** file. The default value is *disabled*.

### SETTING THE PORT NUMBER OF THE EXPRESS ATR RECEIVER

This setting is used only when the database is the destination of express asynchronous table replication. Express ATR's method for updating target tables is different from that used by ODBC. Setting the port number denotes this database will start an express ATR receiver daemon that binds on this port. If the port number is unspecified, the express ATR daemon is not started. For more in-depth information, please refer to Section 17.3 in the *Database Administrator's Guide*. This setting corresponds to the **DB\_EtrPt** keyword in the **dmconfig.ini** file. There is no default value.

### SETTING THE ATR LOG FILE DIRECTORY

The **Log File Directory of Async. Replication** field shows the directory where DBMaker saves the replication log files for asynchronous table replication. ATR error logs are only associated with the source database. By default they are located under the source database directory. The replication log files are binary and users should not manually remove them. The default path for ATR Log files is *<Database Directory>\ATRLOG* and is automatically created by DBMaker. The total length of the ATR log file directory path must not exceed 256 characters. Users should enter a new path for the ATR log file directory by typing the new path into the text field, or by clicking the browse button (  ) next to the text field. This setting corresponds to the **RP\_LgDir** keyword in the **dmconfig.ini** file.

## Database Replication Settings

---

Database replication results in the creation of a primary database and a slave database. The Database Replication Daemon periodically updates the slave database by copying

the backup journal files from the primary database. The Backup Server must be activated and backup server mode must be set to *Backup data only* or *Backup data and BLOB* for database replication to function properly.

If access to a network is limited or bandwidth problems result in slow network connectivity, slave databases may be created on local area servers to allow fast read-only access to local clients. The following settings apply to the creation and subsequent operation of replicated databases. Backing up the database to another location initially creates the slave database. Database replication can be executed afterwards to synchronize the slave database with the source database. Database replication can be executed manually or through the database replication daemon. The IP address and port number must be specified.

For more information please refer to section 17 in the *Database Administrator's Guide*.

### IP ADDRESS OF SOURCE DATABASE

This field, used for database replication, specifies the port number of 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 Databases** field of the primary database. This setting corresponds to the RP\_Privy keyword in the **dmConfig.ini** file. There is no default value.

### IP AND PORT NUMBER OF TARGET DATABASES

Entering the **Port Numbers 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, and should correspond to the values in the **Port Number of Receive Daemon on Target Database** field specified in each target (i.e., slave) database as defined in the **dmconfig.ini** file. DBMaker supports up to 256 slave databases for each primary database. The format should follow the following syntax.

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

The information for each slave database can be separated by a comma or blank space. The default port number is *23001*.

### ➤ Example

```
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. This setting corresponds to the **RP\_SlAdr** keyword in the **dmconfig.ini** file. There is no default value.

### PORT NUMBER OF RECEIVE DAEMON ON TARGET DATABASE

This field is used only in the 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 Numbers of Receive Daemons on Target DB** field in the primary database. Please refer to section 3.2 on *Port Numbers* for additional information. This setting corresponds to the **RP\_PtNum** keyword in the **dmconfig.ini** file. The default value is *23001*.

### SETTING THE REPLICATION START TIME

To set the time at which the first database replication will occur, enter the date in the **yyyy/mm/dd** field, and the time in the **hh:mm:ss** field. Reenter the numbers if they appear incorrectly the first time. This setting corresponds to the **RP\_BTime** keyword in the **dmconfig.ini** file.

### TIMES TO RETRY ON FAILURE

Enter a value in the combo box to specify how many times DBMaker attempts to connect to remote databases during database replication after a network failure. This setting corresponds to the **RP\_ReTry** keyword in the **dmconfig.ini** file. The default value is *0*.

### REMOVE BACKUP JOURNAL FILES AFTER REPLICATION

Enabling **Remove Backup Journal Files after Replication** specifies that DBMaker clears journal backup files after sending them to remote databases during database



replication. This setting corresponds to the **RP\_Clear** keyword in the **dmconfig.ini** file. The default value is *disabled*.

### SETTING THE REPLICATION DAEMON INTERVAL

The number in the **Time interval to Start Database Replication** combo box specifies, in days, the time interval at which database replication occurs. The time input field combo box adjacent to the **Time interval to Start Database Replication** combo box specifies, in hours, minutes, and seconds, the time interval. The total time interval is determined by adding the two values. For example, inputting 1 for **days** and inputting 12:00:00 for **hh:mm:ss** executes an incremental backup every day and a half. The number of days can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right; the number of hours, minutes, and seconds are manually entered. This setting corresponds to the **RP\_Iterv** keyword in the **dmconfig.ini** file.

## 3.12 Start DB

Selecting the **Start DB** (Database) tab opens the **Start DB** page. This page displays options that are available when starting a database. The settings on this page include start database modes, mail setting of error report system, starting the I/O server, multi-user mode, checking user file, force starting a database, resetting asynchronous table replication, and accessing control list. The settings on this page are only applied to the server.

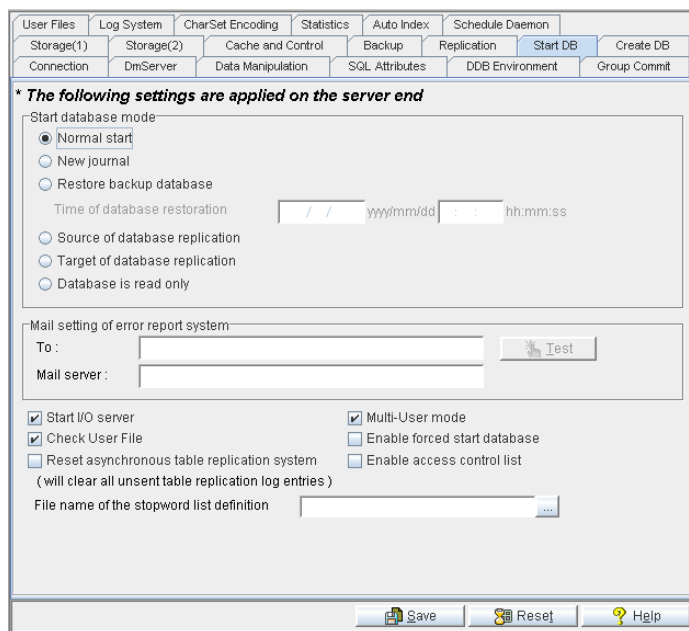


Figure 3-12 Start DB (database) Page

### Start Database Mode

There are six database start-up modes. These modes correspond to values of the `DB_SMode` keyword in the `dmconfig.ini` file. The default value is *Normal Start*.

Please refer to section 4.3 in the *Database Administrator's Guide* for more in-depth information on database startup modes.

### **NORMAL START**

Normal start starts up the database system according to the configuration parameters. An automated crash recovery is performed to bring the database to a consistent and stable state if DBMaker detects that the database experienced a crash since it was most recently started.

### **NEW JOURNAL**

Starts up a system normally, but creates a new journal file whose name is specified in the **Names of System Journal Files** on the **Storage** page. All old records are overwritten if the previous journal file names are used. 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. An incremental or full backup is recommended before selecting this option.

### **RESTORE DATABASE BACKUP**

Uses the backed up database files, including the journal file, to start the database. DBMaker uses the incremental backup files to roll over the operations up to the point in time specified in the fields indicated by **Time of Database Restoration**. 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 needed to roll 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 Restoration** reverts to its default value.

### **SOURCE OF DATABASE REPLICATION**

This mode is used for database replication. Starting up a system with this mode makes it a primary (source) database. For more information on database replication, refer to the section on *Database Replication Settings*.

### TARGET OF DATABASE REPLICATION

This mode is used for database replication. Starting up a system with this mode makes it a slave (i.e., destination or target) database. For more information on database replication, refer to the section *Database Replication Settings*.

### DATABASE IS READ ONLY

Starts up a system normally, but the database is read-only and only provides read privilege to all users. Starting a primary database in read-only mode prohibits users from making any modifications.

## Error Report System Mail Settings

---

DBMaker creates an error report log of 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. Alternatively, when a destination e-mail address and SMTP server are specified DBMaker will automatically e-mail error messages to the recipient as soon as they occur. Up to eight destination addresses may be specified in the **To:** field, each separated by commas. The destination e-mail address corresponds to the **DB\_ERMRv** keyword in the **dmconfig.ini** file. The SMTP server for relaying mail is specified in the **Mail Server:** field, and corresponds to the **DB\_ERMSv** keyword. The destination e-mail address setting is empty by default; therefore, the feature is *disabled* by default.

## Enabling Multi-User Mode

---

Enabling **Multi-User Mode** permits more than one user to access a database. Disabling the check box limits database access to a single user. This setting corresponds to the **DB\_UMode** keyword in the **dmconfig.ini** file. The default setting is *multi-user*.

### Enabling Forced Start Database

---

Enabling the **Enable Forced Start Database** check box forces databases to start even when errors occur during the startup process. The default setting is *disabled*. This setting corresponds to the **DB\_ForcS** keyword in the **dmconfig.ini** file.

**NOTE** *DBMaker server may not be able to recover some data from an instance failure if forced database startup is attempted in new journal start mode.*

### Enabling Access Control List Check

---

Enabling the check box next to **Enable Access Control List Check** permits the DBA to allow or block some users come from any network address. Disabled the check box allows all user from any IP to connect the database. The default setting is *disabled*. This setting corresponds to the **DB\_StACL** keyword in the **dmconfig.ini** file.

### Reset Asynchronous Table Replication System

---

Enabling the checkbox next to **Reset Asynchronous Table Replication System** clears all unsent table replication log entries upon startup. For more information on asynchronous table replication, refer to the section on *Asynchronous Table Replication Settings*. This setting corresponds to the **RP\_Reset** keyword in the **dmconfig.ini** file. The default setting is *disabled*.

### Check User File

---

DBMaker allows users to check their files when the database starts in the way of warm start. Selecting the **Check User File** check box enables this function, dmserver checks all users' files to make sure they are still on the disk when the database starts in the way of warm start, if not, DBMaker gives a warning message to notice a user with DBA authority or higher to avoid operating the files which already have been moved. This warning message is recorded into the log file *DMEVENT.LOG*. This setting corresponds to the **DB\_ChkFl** keyword in the **dmconfig.ini** file. The default setting is *enabled*. Please note that this function does not support single-user mode.


### Start I/O Server

---

Selecting the **Start I/O server** check box enables the I/O server daemon. This corresponds to the `DB_IOSvr` keyword in the `dmconfig.ini` file. Enabling the I/O server tells whether DBMaker should turn the I/O and checkpoint daemon on or off. 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. The default setting is *enabled*.

### File Name of the Stopword List Definition

---

Users can define the list of stopwords according to their specific need. 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 of the stopword list definition file. If a stopword list definition file is not specified then DBMaker loads a default stopword containing pre-defined file name based on LCODE. This setting corresponds to the `DB_StpWd` keyword in the `dmconfig.ini` file.

### 3.13 Create DB

Selecting the **Create DB** (database) tab opens the **Create DB** page. This page displays the available options used to create a database. The options include code order file name, BLOB file size (bytes), uppercase identifier conversion, and the choice of the language code. The settings on this page are applied to the server when a database is created.

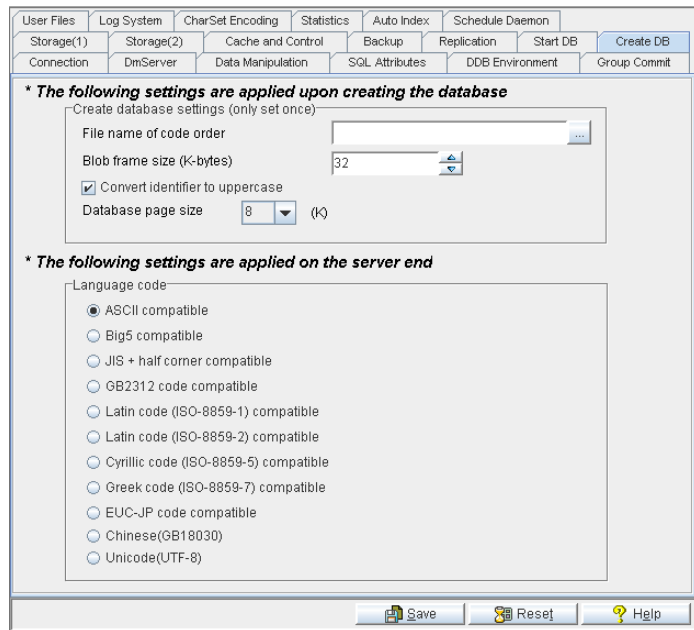


Figure 3-13 Create DB (database) Page

#### File Name of Code Order

The **File Name of Code Order** field shows the name of the order definition file that is placed in the `shared\codeorder` subdirectory located in the DBMaker installation directory. The order definition file is a text file that affects sorting results in DBMaker.

It is used for sorting within indexes and for comparison operators in SQL WHERE statements. Any text editor can be used to create the order definition file.

The order definition file arranges the sequence of valid characters. The recommended naming convention is `codename_ordertype.ord`, where `codename` is the name of the language code and `ordertype` is the type of ordering, e.g., `big5_stroke.ord`.

### ➤ Example

A simple order definition file:

```
Comment: write relevant information here.

[BEGIN]      // begin to arrange the character sequence

c           // ASCII 0x63
0x62       // Character 'b'
a           // ASCII 0x61

[SINGLE]     // Single-Byte Character Default Order

[DOUBLE]    // Double-Byte Character Default Order

0xA440     // a Chinese character
0xA441     // a Chinese character
0xA442     // a Chinese character
```

All lines before the keyword "[BEGIN]" are considered comments. All of the words that appear after "/" or "/"\* are also comments. After the "[BEGIN]" keyword, each line represents one character. The character definition should occupy the first position of the line and be followed by at least one space or a new line character. Characters in the order definition file are listed from lowest to the greatest value. In the example just above, character "c" is less than "b" and "b" is less than "a".

If a text editor cannot edit some characters, they can be represented with a hexadecimal value. The character "a" can be represented by an "a" or the corresponding code value (0x61).


The creator of the sort order may only be interested in some characters and let others be sorted by default, i.e. binary. "[SINGLE]" and "[DOUBLE]" keywords can be used



to represent the single character set and double character set, which are not specified in the definition file. If the keyword "[SINGLE]" is not added to the order definition file, single-byte characters that are not specified will be before all other characters in the definition file. If the keyword "[DOUBLE]" is not added to the order definition file, double-byte characters that are not specified will be after all other characters in the definition file.

DBMaker uses default settings if errors are found in the definition file. For example, if "[BEGIN]" is lost, DBMaker will use the default sorting for all characters. If the same character appears more than once, the first instance is processed and all others are ignored. After creating a new database, carefully check if the behavior of the sort order is correct. The default sorting sequence is according to the binary sequence of the language set being used.

In a distributed database environment, all databases should use the same sort order definition file. If copying or moving the entire database to another machine, remember to also copy the sort order definition file.

The order definition file is only used upon creation of the database. Changing this field for an existing database has no effect. Users can enter a new path for the order definition file by typing a new path into the text field, or by clicking the browse button (  ) next to the text field. This setting corresponds to the **DB\_Order** keyword in the **dmconfig.ini** file.

### **BLOB Frame Size (bytes)**

---

The value in the combo box specifies the size of each BLOB frame in kilobytes. This setting is used when the database is created. Changing this value for an existing database has no effect.

*default value:*     32 (KB )

*valid range:*       8 through 256 (KB)

The number in the combo box can be manually entered or adjusted using the arrows to the right. This setting corresponds to the **DB\_BfrSz** keyword in the **dmconfig.ini** file.

### Converting Identifier to Uppercase

---

The check box next to **Convert Identifier to Uppercase** specifies the case sensitivity of all object identifiers in the database. Identifiers include names for databases, synonyms, tables, tablespaces, columns, user and group names, and triggers. Regardless of this setting, user passwords and user-defined functions are always case sensitive and database names are always case insensitive. Disabling the check box indicates that all of the database's identifiers are case sensitive. Enabling the check box indicates all identifiers in database are case insensitive; under this mode, all identifiers are converted to upper case when they are defined. This setting is used when the database is created; changing this setting for an existing database has no effect. This setting corresponds to the `DB_IDCap` keyword in the `dmconfig.ini` file. The default value is *enabled*.

### Database Page Size

---

DBMaker data files are stored as data pages. A data page is DBMaker's smallest unit of storage. In DBMaker 5.4, users can specify a page size of 4 KB, 8 KB, 16 KB, or 32 KB. This setting corresponds to the `DB_PgSiz` keyword in the `dmconfig.ini` file. The default value is **8 KB**.

### Language Code

---

The language code for data tables is determined at the time of the database creation by selecting one of the following options in the **Language Code** field. It cannot be changed later. The language code affects the result of LIKE operations in a query. The language options include ASCII compatible (English), BIG5 compatible (Traditional Chinese), JIS + Half Corner compatible (Japanese (Shift-JIS + Half Corner)), GB2312 code compatible (Simplified Chinese), Latin code (ISO-8859-1) compatible, Latin code (ISO-8859-2) compatible, Cyrillic code (ISO-8859-5) compatible, Greek code (ISO-8859-7) compatible, Japanese code (EUC-JP) compatible, Simplified Chinese (GB18030) compatible or UTF-8. For more in-depth information, please refer to the *SQL Command and Function Reference*. This value is required when the database is started. Once a database has been created, there is no way to change the language code

unless the database is recreated; changing this setting for an existing database has no effect. This setting corresponds to the `DB_LCode` keyword in the `dmconfig.ini` file. The default value is *ASCII Compatible*.

## 3.14 User Files

Selecting the **User Files** tab opens the **User Files** page. This page displays the **User Files** dialog box. The options on this page are only applied to the server.

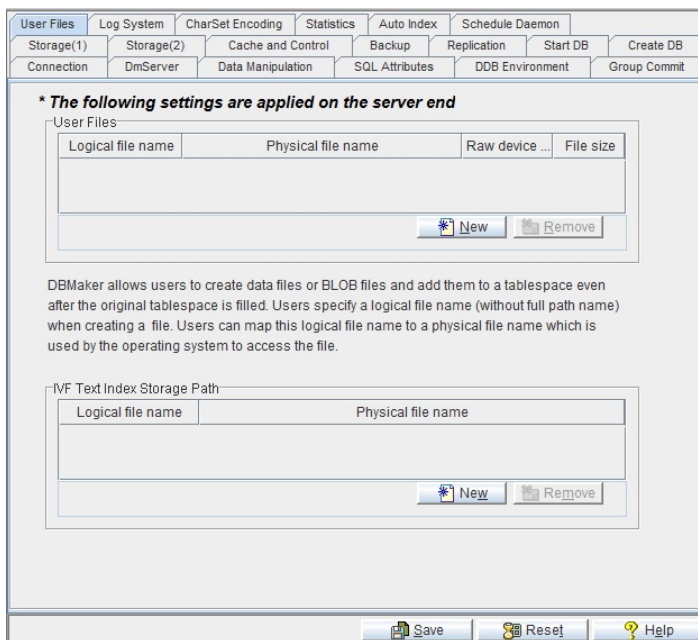


Figure 3-14 User Files Page

### Creating a User-Defined File

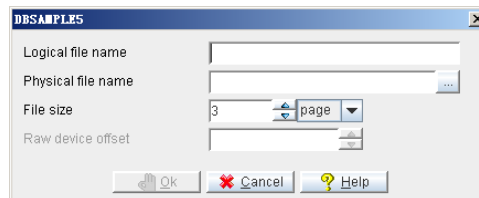
DBMaker allows users to create data files or BLOB files and add them to a tablespace even when the original tablespace is filled. A logical file name can be specified (without a full path name being specified) when creating a file. The logical file name can later be mapped to a physical file name used by the operating system to access the file. This setting corresponds to the **User-defined filename** keyword in the **dmconfig.ini** file.

When creating multiple files on a raw device please keep in mind:

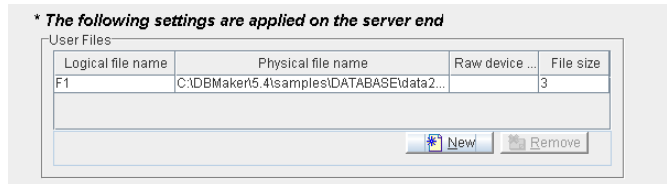
- ◆ Tablespaces cannot be set to autoextend when multiple files are set in a single raw device
- ◆ When a raw device is initially set and files creates on it, the raw device offset and the file's page number cannot be modified
- ◆ The total size of all files in a single raw device is restricted to less than 2 GB

### ➔ To create a user-defined file:

1. To create a user-defined file, click the **New** button at the bottom of the **User Files** field. The following dialog box appears:



2. Enter the logical file name in the **Logical file name** field (this is the path referenced by SQL commands and by DBMaker to reference the data).
3. Enter a Physical File Name by entering a new path into the **Physical file name** field, or click the browse button (...). This is the operating system path. The maximum length for the physical file name path is 256 characters.
4. If you are running in a UNIX environment set the file offset in the **Raw Device Offset** field.
5. Enter a value into the **File size** box. This indicates the number of data pages allocated for the user-defined file. There are three unit options for page size: Page, M (megabytes), and G (gigabytes). Please note that the actual size will be 1 page less than the specified value when using M or G. The range of values for the number of data pages is between 2 and 524,287 bytes.
6. Click **OK**. A new user-defined file appears in the **User Files** field. The **Logical File Name**, **Physical File name**, and number of pages can be changed after the user-defined file has been created. Double-clicking the text allows the user to edit it. This is useful if data files are moved within the operating system directory. DBMaker does not recognize a change in the physical file name unless it is specified here.



7. To delete a user-defined file click **Remove**. DBMaker no longer recognizes the logical file name.

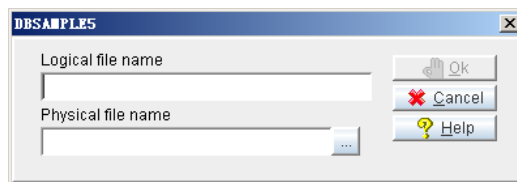
### Creating an Inverted File Text Index Storage Path

---

DBMaker provides the Inverted File (IVF) text index function for improved text index query performance on larger columns. In order to create an IVF text, however, you must first specify the file in which the IVF index will be created.

#### ➔ To specify a logical and physical file name for an IVF text index storage:

1. To specify a path for an IVF text index, click **New** at the bottom of the **IVF Text Index Storage Path** field. The following dialog box appears:



2. Enter the logical file name in the **Logical file name** field (this is the path referenced by SQL commands and by DBMaker to reference the data).
3. Enter a Physical File Name by entering a new path into the **Physical file name** field, or click the browse button (...). This is the operating system path. The maximum length for the Physical File Name path is 256 characters.
4. Click **OK**. A new IVF text index storage path appears in the **IVF Text Index Storage Path**. The **Logical File Name** and **Physical File name** can be changed after the IVF text index storage path has been specified. Double-clicking the text allows the user to edit it. This is useful if an IVF file is moved within the operating system directory. DBMaker does not recognize a change in the physical file name unless it is specified here.
5. To remove an IVF text index storage path click **Remove**. DBMaker no longer recognizes the logical file name.

### 3.15 Log System

Selecting the Log System tab opens the Log System page, which displays the settings for the Log system. The options on this page are only applied to the server.

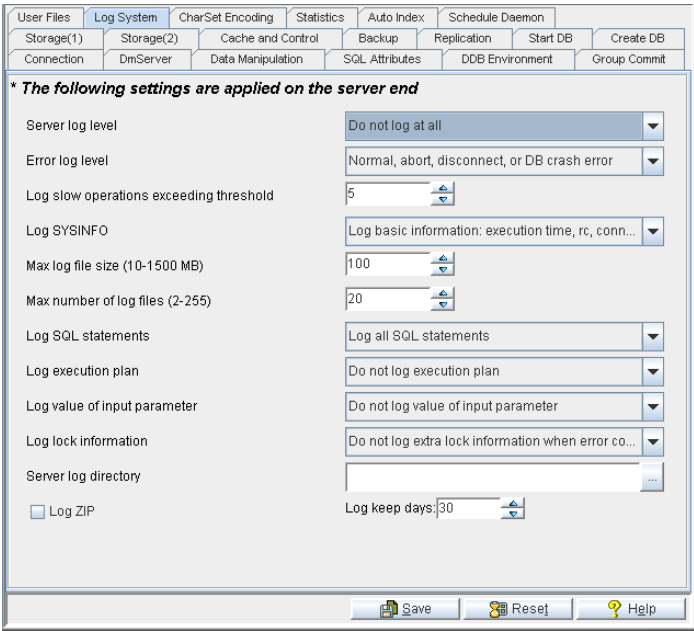
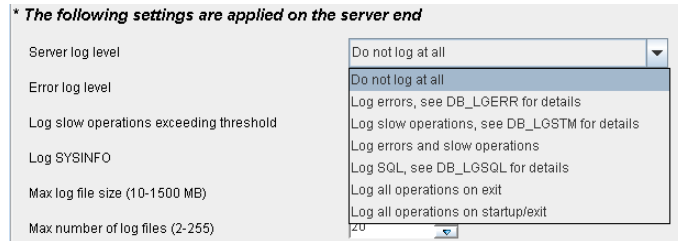


Figure 3-15 Log System Page

### Server Log Level

DBMaker provides extensive logging capabilities. Users can specify the level of detail and how often information is logged according to the following seven error logging levels:



- ◆ **Do not log at all:** disables the log system
- ◆ **Log error, see DB\_LGERR for details:** logs errors and the default errors according to the logging level specified in **Error log level**
- ◆ **Log slow operation, see DB\_LGSTM for details:** 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, see DB\_LGSQL for details:** logs connects, disconnects, commits, rollbacks, SQL command executions, errors, and slow operations
- ◆ **Log all operation on exit:** logs all operations during the exit process
- ◆ **Log all operation on startup/exit:** logs all operations during the enter and process and the exit process

This setting is specified by the **DB\_LgSvr** keyword in the **dmconfig.ini** file. The default value is *Do not log at all*.

## Error Log Level

---

Users can specify the level of detail that will be recorded when server logging is enabled as specified by the **Server log level** setting including log error. The **Error log level** indicates what kind of errors will be logged according to the five levels described here:



\* The following settings are applied on the server end

Server log level	Do not log at all
Error log level	Normal, abort, disconnect, or DB crash error
Log slow operations exceeding threshold	Core dumped or DB crash error
Log SYSINFO	Disconnect error or DB crash error
Max log file size (10-1500 MB)	Abort, disconnect, or DB crash error
Max number of log files (2-255)	Normal, abort, disconnect, or DB crash error
	Warning or other error
	20

- ◆ **Core dumped or DB crash error:** logs all errors that occur having error codes > 30000
- ◆ **Disconnect error or DB crash error:** logs all errors that occur having error codes > 20000
- ◆ **Abort, disconnect or DB crash error:** logs all errors that occur having error codes > 10000
- ◆ **Normal, abort, disconnect or DB crash error:** logs all errors that occur having error codes > 100
- ◆ **Warning or other error:** logs all errors (i.e., all errors that occur having error codes > 0)

This setting corresponds to the `DB_LgErr` keyword in the `dmconfig.ini` file. The default level is *Normal, abort, disconnect or DB crash error*.

### Setting the Slow Statement Logging Threshold

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.

This setting corresponds to the `DB_LgSTm` keyword in the `dmconfig.ini` file. The default value is 5 seconds.

## 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:

\* The following settings are applied on the server end

Server log level	Do not log at all
Error log level	Normal, abort, disconnect, or DB crash error
Log slow operations exceeding threshold	5
Log SYSINFO	Log basic information: execution time, rc, conn...
Max log file size (10-1500 MB)	
Max number of log files (2-255)	

- Log basic information: execution time, rc, connection id, etc.
- Log above information plus SYSUSER and SYSINFO information
- Log above information plus SYSTEM memory information if detectable

- ◆ **Log basic information: execution time, rc, connection id, etc**
- ◆ **Log above information plus SYSUSER and SYSINFO information**
- ◆ **Log above information plus SYSTEM memory information if detectable**

This setting corresponds to the **DB\_LgSys** keyword in the `dmconfig.ini` file. The default setting is *Log basic information: execution time, rc, connection id, etc.*

## Setting the 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: `DBNAME_2.LOG`, `DBNAME_3.LOG`, ..., `DBNAME_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.

This setting corresponds to the **DB\_LgFSz** keyword in the `dmconfig.ini` file. The default setting is **100 MB**.

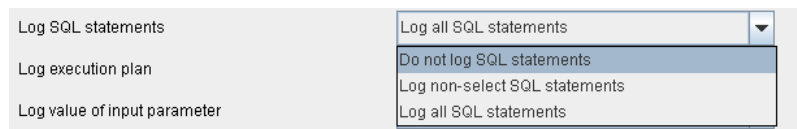
### Setting the Maximum Number of Log Files

This setting specifies the number of the log files that are generated before the log rolls-over (i.e., the first log file is overwritten with new log information). DBMaker stores the log as DBNAME\_1.LOG. When the log file size reaches the default value of 100 MB or the size specified by `DB_LgFSz`, the log information is logged to a subsequent log file as follows: DBSAMPLE4\_2.LOG, DBSAMPLE4\_3.LOG, ..., DBSAMPLE4\_n.LOG where n is the number of log files. The log information will be written to next log file until either the default of 20 log files have been generated or the number of log files generated is equal to the value specified by this setting. At this point, the log rolls-over. In other words, log files are erased and recreated with the new log information beginning with log file DBNAME\_1.LOG. Users can enter a value into the field or adjust the existing value using the up and down arrows located to the right of the field.

This setting corresponds to the `DB_LgFN0` keyword in the `dmconfig.ini` file. The default setting is *20*.

### 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_LgSvr = 4`). There are three SQL logging settings for recording different levels of information detail:



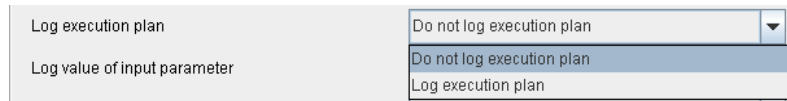
- ◆ Do not log SQL statements
- ◆ Log non-select SQL statements
- ◆ Log all SQL statements

This setting corresponds to the `DB_LgSQL` keyword in the `dmconfig.ini` file. The default setting is *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:



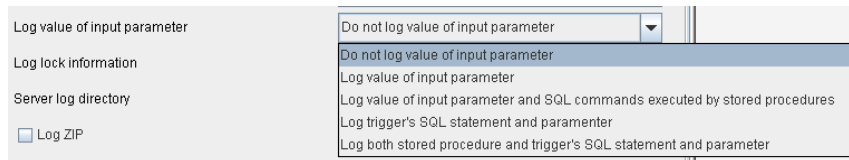
- ◆ Do not log execution plan
- ◆ Log execution plan

This setting corresponds to the `DB_LgPln` keyword in the `dmconfig.ini` file. The default setting is *Do not 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:



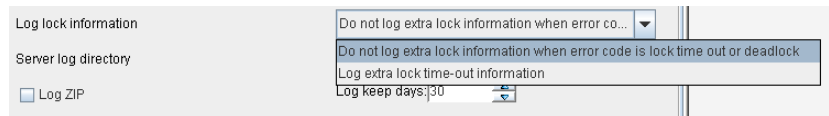
- ◆ Do not log value of input parameter
- ◆ Log value of input parameter
- ◆ Log value of input parameter and SQL commands executed by stored procedures
- ◆ Log trigger's SQL statement and parameter
- ◆ Log both stored procedure and trigger's SQL statement and parameter

This setting corresponds to the `DB_LgPar` keyword in the `dmconfig.ini` file. The default setting is *Do not log value of input 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

This setting corresponds to the `DB_LgLck` keyword in the `dmconfig.ini` file. The default setting is *Do not log extra lock information when error code is lock time out or deadlock*.

### Server Log Directory

---

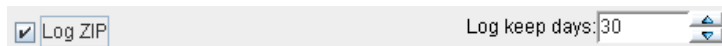
The directory path of the log files can be specified according to a user's needs. It may be entered manually or selected by using the browse button (⋮). Selecting the browse button (⋮) opens a dialog box.

This setting corresponds to the `DB_LgDir` keyword in the `dmconfig.ini` file. The default setting is *DB\_DbDir\LgDir*.

### 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.

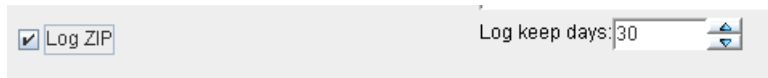


This setting corresponds to the `DB_LgZip` keyword in the `dmconfig.ini` file. The default setting is `0`.

### 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.



This setting corresponds to the `DB_LgDay` keyword in the `dmconfig.ini` file. The **Log Keep Days** can be any integer value from 0 to 365. The default setting is `30`.

### 3.16 CharSet Encoding

Selecting the CharSet Encoding tab opens the Character Set Encoding page. This page displays the language code options and error message code options set by clients. On this page, client side users can set the language code for themselves and it can be different from the language code in Server side.

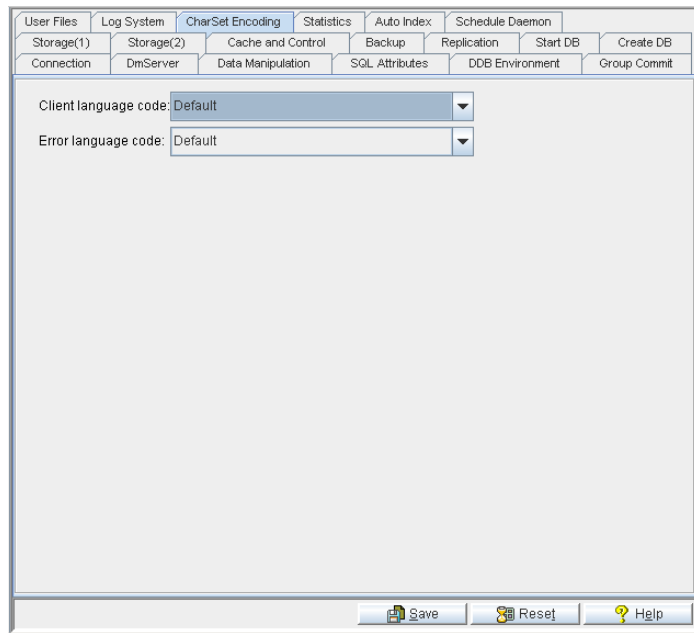
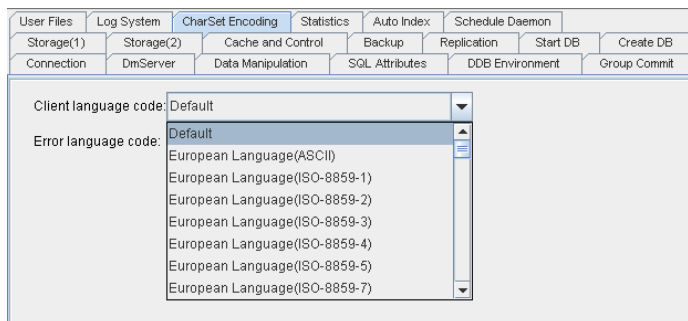


Figure 3-16 CharSet Encoding Page

#### Client LCode

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 language codes that client side can set 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)

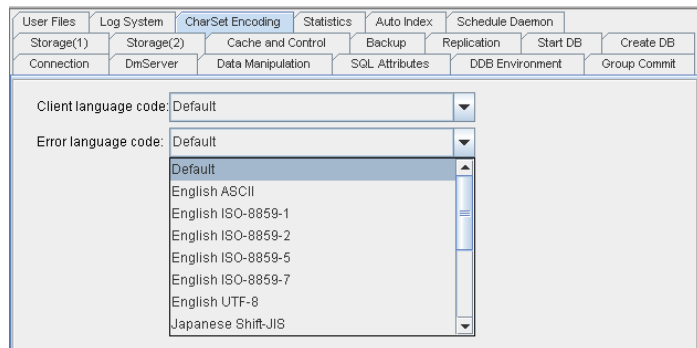
- ◆ Thai (MACTHAI)
- ◆ Laotian (MULELAO-1)
- ◆ Laotian (CP1133)
- ◆ Vietnamese (VISCII)
- ◆ Vietnamese (TCVN)
- ◆ Vietnamese (CP1258)
- ◆ Unicode (UTF-8)

This setting corresponds to the `DB_CliLCODE` keyword in the `dmconfig.ini` file. The default setting is *DEFAULT* condition if user did not do any set action, then system will automatically choose proper code to ensure the client can connect to server successfully.

## Error LCode

---

In multilingual database, client side can set their own output locale codes of error message. They can be:



- ◆ Deafault
- ◆ English ASCII
- ◆ English ISO-8859-1

- ◆ English ISO-8859-2
- ◆ English ISO-8859-5
- ◆ English ISO-8859-7
- ◆ English UTF-8
- ◆ Japanese Shift-JIS
- ◆ Japanese EUC
- ◆ Japanese UTF-8
- ◆ Simplified Chinese GBK
- ◆ Simplified Chinese GB18030
- ◆ Simplified Chinese UTF-8
- ◆ Traditional Chinese Big5
- ◆ Traditional Chinese UTF-8

This setting corresponds to the `DB_ErrLCODE` keyword in the `dmconfig.ini` file. The default setting is *DEFAULT* status if user did not do any set action. System will automatically choose the proper code for error message.

### 3.17 Statistics

Selecting the **Statistics** tab opens the **Statistics** page. This page displays the update statistics setting option. The options on this page are only applied to the server.

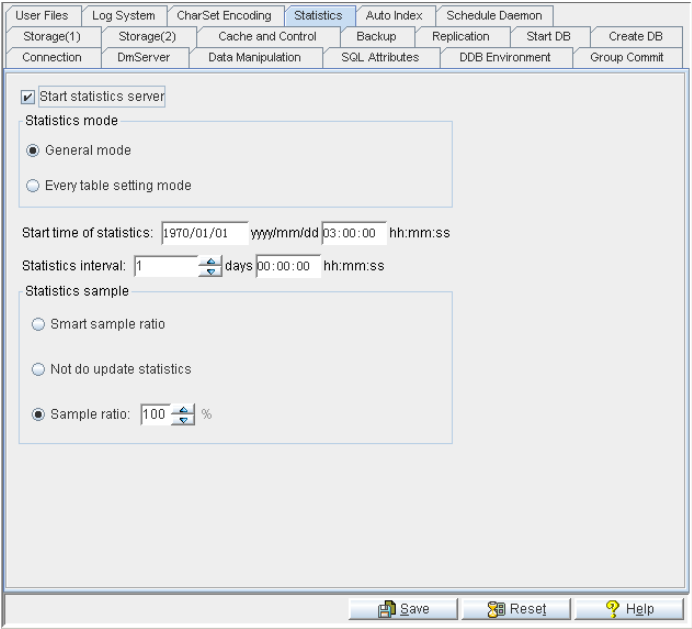


Figure 3-17 Statistics Page

#### Start Statistics Server

DBMaker keeps statistical information about database objects in the system tables. Periodically the database needs to be scanned for recalculating the statistics. The updated statistics are then written to the system tables to ensure efficient database operation. Enabling **Update statistics** ensures that DBMaker automatically and periodically recalculates database statistics according to an internal scheduling daemon.

Selecting the **Start statistics server** check box enables the auto update statistics server daemon. This keyword corresponds to the **DB\_StSvr** keyword in the **dmconfig.ini** file. The default setting is *enabled*.

### **Statistics Mode**

---

The settings in the **Statistics Mode** field specify the Updated Statistics daemon mode for a database. Selecting **general mode** enables startup update statistics daemon in general mode and the sample ratio will be decided by value of the keyword **DB\_StsSp** in **dmconfig.ini**. Selecting **every table setting mode** enables startup update statistics daemon in every table setting mode, that is to say, the sample ratio will be decided by every table's mode and sample ratio. Users can set every table's mode and sample ratio with the command "UPDATE STATISTICS SET".

### **Settings for the Update Statistics Process**

---

#### **SETTING THE START TIME OF STATISTICS**

Specify when the first update statistics will begin for the database by entering a date in the **yyyy/mm/dd** field and a time in the **hh:mm:ss** field. Reenter the numbers if they appear incorrectly the first time. This setting corresponds to the **DB\_StsTm** keyword in the **dmconfig.ini** file. Default value is *1970-01-01 03:00:00*.

#### **SETTING THE STATISTICS DAEMON INTERVAL**

The number in the combo box specifies, in days, the time interval at which the Update Statistics Daemon is activated. 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 in **days** and inputting 12:00:00 in **hh:mm:ss** causes the Update Statistics Daemon to be activated every day and a half. The number of days can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right; the number of hours, minutes, and seconds are manually entered. This setting corresponds to the **DB\_StsTv** keyword in the **dmconfig.ini** file.

### Sample Ratio

---

Users can set update statistics sample ratio in **Statistics sample** field of **Statistics** page. This setting corresponds to the **DB\_StsSp** keyword in the **dmconfig.ini** file. **Smart sample ratio** indicates the database intelligently obtain sample ratio; **Not do update statistics** indicates the database doesn't update statistics value. Additional, users can set sample ratio to a number between 1 and 100.

## 3.18 Auto Index

Selecting the **Auto Index** tab opens the **Auto Index** page. This page displays the settings for Auto Index. The options on this page are only applied to the server.

The screenshot shows the 'Auto Index' configuration page. At the top, there are several tabs: 'User Files', 'Log System', 'CharSet Encoding', 'Statistics', 'Auto Index' (selected), and 'Schedule Daemon'. Below these are sub-tabs: 'Storage(1)', 'Storage(2)', 'Cache and Control', 'Backup', 'Replication', 'Start DB', 'Create DB', 'Connection', 'DmServer', 'Data Manipulation', 'SQL Attributes', 'DDB Environment', and 'Group Commit'. The main content area is titled '\* The following settings are applied on the server end'. It contains the following settings:

- Start auto index server
- Start time of auto index: 1970/01/01 yyyy/mm/dd 00:00:00 hh:mm:ss
- Auto index interval: 0 days 01:00:00 hh:mm:ss
- Auto create index threshold: 1
- Auto drop index threshold: 30
- Directory of auto index log file: [empty field with browse button]

At the bottom of the window, there are three buttons: 'Save', 'Reset', and 'Help'.

Figure 3-18 Auto Index Page

### Start Auto Index Server

Auto indexes can be created automatically by the auto index daemon. Meanwhile, auto indexes can be dropped by the auto index daemon automatically if the index can merged with other indexes or it is not be used exceed the number of specified dropped days. The maximum column number of an auto index is 16. Creating an auto index requires U lock when set *COMMIT ON*.



DBMaker also supports “*SET LOADAUTOINDEX ON/OFF*” syntax and ODBC function *SQLSetConnectOption* for user to decide whether load auto index or not. For more information on ODBC functions, please refer to the *ODBC Programmer's Guide*.

Selecting the **Start auto index server** check box enables the auto index daemon. This setting corresponds to the **DB\_IdxSv** keyword in the **dmconfig.ini** file. The default setting is *disabled*.

### Start Time of Auto Index

---

Specify the first time of the auto index daemon to perform auto index by entering a date in the **yyyy/mm/dd** field and a time in the **hh:mm:ss** field. Reenter the numbers if they appear incorrectly the first time. This setting corresponds to the **DB\_IdxTm** keyword in the **dmconfig.ini** file. Default value is *1970-01-01 00:00:00*.

### Setting Auto Index Interval

---

The number in the combo box specifies, in days, the time interval at which the auto index daemon is activated. 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 in **days** and inputting 12:00:00 in **hh:mm:ss** causes the auto index daemon to be activated every day and a half. The number of days can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right; the number of hours, minutes, and seconds are manually entered. This setting corresponds to the **DB\_IdxTv** keyword in the **dmconfig.ini** file.

### Setting Auto Create Index Threshold

---

This number specifies the threshold value for auto creating an auto index by auto index daemon. If the number of scan log (these logs with the same table id, table version and the column id list) reaches or exceeds this value, auto index daemon will create an auto index according to the serial log. This number can be manually entered

into the combo box, or increased or decreased by clicking the arrows to the right. This setting corresponds to the **DB\_IdxLn** keyword in the **dmconfig.ini** file. With the database running, users can set the value of **IDXLN** by calling **setSystemOption()**. The default value is *1*.

### **Setting Auto Drop Index threshold**

---

This number specifies auto drop index threshold for auto index daemon, if an index not be used reaches or exceeds *n* days will be dropped by auto index daemon. This number can be manually entered into the combo box, or increased or decreased by clicking the arrows to the right. This setting corresponds to the **DB\_IdxDp** keyword in the **dmconfig.ini** file. With the database running, users can set the value of **IDXDP** by calling **setSystemOption()**. The default value is *30*.

### **Setting Directory of Auto Index Log File**

---

This auto index's directory is used to save log files. The default path is the database installation directory. This setting corresponds to the **DB\_IdxLg** keyword in the **dmconfig.ini** file.

## 3.19 Schedule Daemon

Selecting the **Schedule Daemon** tab opens the **Schedule Daemon** page. This page displays schedule daemon setting options. The settings on this page determine the actions of the **dmschsvr** server daemon. The options on this page are only applied to the server.

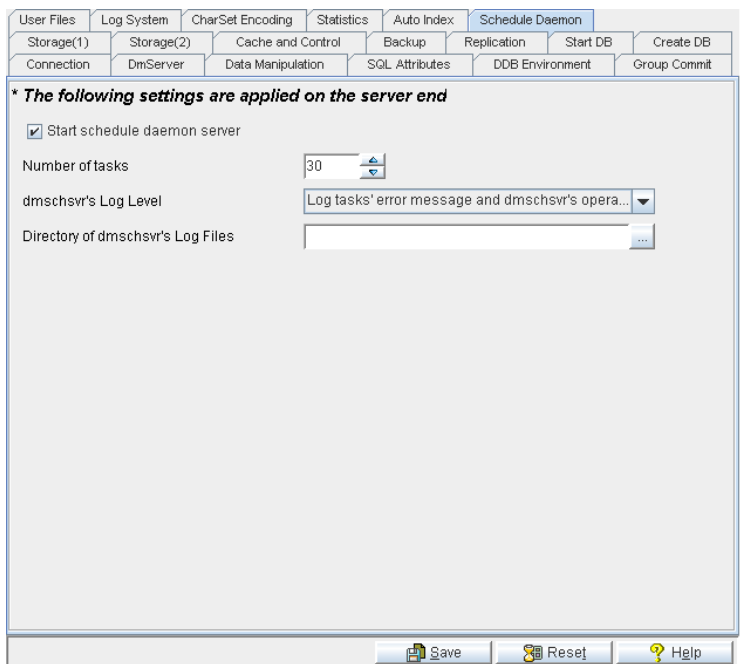


Figure 3-19 Schedule Daemon Page

### Start Schedule Daemon Server

Selecting the **Start schedule daemon server** check box enables the **dmschsvr** server daemon. This corresponds to the **DB\_SchSv** keyword in the **dmconfig.ini** file. Enabling the schedule daemon server tells whether DBMaker should turn the **dmschsvr** daemon and checkpoint daemon on or off. The primary purpose of the

**dmschsvr** daemon is to execute a SQL statement, a procedure, or an executable program, all of that are designed by users in advance. Users can create schedules for routine tasks and **dmschsvr** will automatically execute these tasks, cutting a large amount of routine maintenance. The default setting is *disabled*. For more information on schedule, please refer to *Section 13* in the *Database Administrator's Guide*.

### Number of Tasks

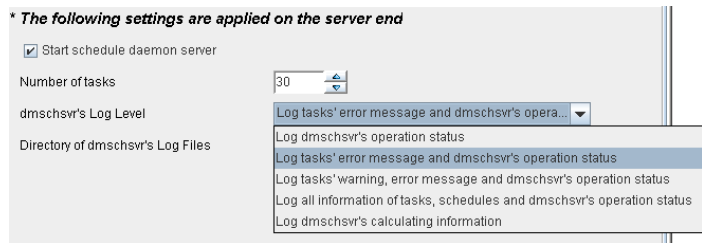
---

The **Number of tasks** field value is used to specify the number of task can be aroused by **dmschsvr** at the same time. This setting corresponds to the **DB\_TskNo** keyword in the **dmconfig.ini** file. The default value is *30*.

### Dmschsvr Log Level

---

Users can specify what schedule daemon log information will be output according to the following five output levels. This setting corresponds to the **DB\_SchLgLev** keyword in the **dmconfig.ini** file. The default value is *1*.



- ◆ **Log dmschsvr's operation status:** Value of **DB\_SchLgLev** should be set to 0, and only log **dmschsvr's** operation status
- ◆ **Log task's error message and dmachsvr's operation status:** Value of **DB\_SchLgLev** should be set to 1, and log tasks' error message and **dmschsvr's** operation status
- ◆ **Log tasks' warning, error message and dmschsvr's operation status:** Value of **DB\_SchLgLev** should be set to 2, and log tasks' warning, error message and **dmschsvr's** operation status

- ♦ **Log all information of tasks, schedules and dmschsvr's operation status:** Value of `DB_SchLgLev` should be set to 3, and log all information of tasks, schedules and `dmschsvr`'s operation status
- ♦ **Log dmschsvr's calculating information:** Value of `DB_SchLgLev` should be set to 4, and log `dmschsvr`'s calculating information, so that a user can monitor whether `dmschsvr` is running normally

### **Setting Directory of Dmschsvr Log Files**

---

This `dmschsvr`'s directory is used to save schedule daemon log files. If a user doesn't set this directory, these log files will be stored in the directory which is specified by `DB_DbDir`. This setting corresponds to the `DB_SchLgDir` keyword in the `dmconfig.ini` file.

