DBMaker

JConfiguration Tool Reference

CASEMaker®

CASEMaker Inc./Corporate Headquarters

1680 Civic Center Drive

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

www.casemaker.com

www.casemaker.com/support

Document No. 645049-231017/DBM41-M05262003-JCON
Publication Date: 2003-05-20

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

Trademarks

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

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

Notices

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

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

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

Contents

1 Introduction 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 3-2

Default User ID and Password 3-2
Enabling Auto-Commit 3-3
Setting the Connection Detection Interval 3-3
Application Multiple Connection Status 3-3
Setting the Connection Time-Out Value 3-3
Setting the Lock Time-Out Value 3-4

3.2 DmServer 3-5
Client/Server Settings 3-5
Server Settings 3-6
3.3 Data Manipulation 3-8
Remove Space Padding in CHAR Output 3-8
Remove Space Padding in String Concatenation 3-9
Maximum String Size 3-9
File Type Mapping Settings 3-9
Cursor Behavior Mode 3-10
Select Browse Mode 3-10
3.4 SQL Attributes 3-11
Date & Time Formats 3-11
Stored Procedures Server End Settings 3-12
Stored Procedures Client End Settings 3-12
3.5 DDB Environment 3-14
Enabling Distributed Database Mode 3-15
Remote Database Connection Time-out Value 3-15
Remote Database Lock Time-out Value 3-15
Time Interval to Recover Pending Transaction 3-15
Start Global Transaction Recovery Daemon 3-16
3.6 Group Commit 3-17
Group-Commit Threshold 3-18
Maximum Transaction Waiting Time 3-18
Maximum Number of Transactions in Wait State 3-18
3.7 Storage (1) 3-19
Database Directory 3-19
Name of System Data File 3-20
Name of System BLOB File 3-20
Name of User Data File 3-20
Number of User Data File Pages 3-21
Name of User BLOB File 3-21
Number of User BLOB Frames 3-21

Extending a File 3-22
3.8 Storage (2) 3-23
System Journal Files 3-23
User File Objects 3-24
System File Object Directory 3-24
Files per File Object Subdirectory 3-25
User-Defined Function Library 3-25
Names of System Temporary Files 3-25
3.9 Cache and Control 3-27
Cache and Shared Memory Control 3-28
Engine Control 3-29
3.10 Backup 3-32
Incremental Backup Mode Settings 3-33
Backup File Object Mode 3-33
Setting the Backup File Directory 3-33
Starting Backup Server 3-34
Settings for Full Backup Process 3-34
Settings for Incremental Backup Process 3-35
3.11 Replication 3-39
Asynchronous Table Replication Settings 3-40
Database Replication Settings 3-41
3.12 Start DB 3-45
Start Database Mode 3-46
Mail Setting of Error Report System 3-47
Enabling Multi-User Mode 3-47
Enabling Forced Start Database 3-48
Reset Asynchronous Table Replication System 3-48
Start I/O Server 3-48
Update Statistics Automatically 3-48
3.13 Create DB 3-50
File Name of Code Order 3-50
BLOB Frame Size (bytes) 3-52

Converting Identifier to Uppercase 3-53
Language Code 3-53
3.14 User Files 3-54
Creating a User-Defined File 3-54
Creating an Inverted File Text Index Storage Path 3-56
Glossary Glossary-1
Index Index-1

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 of the program features are introduced with a brief description for each function and screen images are included for most features.

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, refer to the "*ODBC Programmer's Guide*".

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

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

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

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

For more information on the 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 charges).

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

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

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

Product name and version number

Registration number

Registered customer name and address

Supplier/distributor where product was purchased

Platform and computer system configuration

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

Error message and number, if any

Any additional information deemed pertinent

1.3 Document Conventions

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

CONVENTION	DESCRIPTION
Italics	Italics indicate placeholders for information that must be supplied, such as user and table names. The word in italics should not be typed, but is replaced by the actual name. Italics also introduce new words, and are occasionally used for emphasis in text.
Boldface	Boldface indicates filenames, database names, table names, column names, user names, and other database schema objects. It is also used to emphasize menu commands in procedural steps.
KEYWORDS	All keywords used by the SQL language appear in uppercase when used in normal paragraph text.
SMALL CAPS	Small capital letters indicate keys on the keyboard. A plus sign (+) between two key names indicates to hold down the first key while pressing the second. A comma (,) between two key names indicates to release the first key before pressing the second key.
NOTE	Contains important information.
➲ **Procedure**	Indicates that procedural steps or sequential items will follow. Many tasks are described using this format to provide a logical sequence of steps for the user to follow
➲ **Example**	Examples are given to clarify descriptions, and commonly include text, as it will appear on the screen. Other forms of this convention include Prototype and Syntax.
`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 it is started. DBMaker will use default values for settings not specified by 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 4.1>DBMaker Configuration File** submenu item.

Both JServer Manager and JDBA Tool contain links to the JConfiguration Tool. The *JServer Manager User's Guide* and the *JDBA 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 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 with the TCP/IP network protocol. All of the settings for these functions can be manipulated with the JConfiguration Tool to suit a user's needs.

Performance: Cache and control settings allow a user to control how much memory is allocated for DBMaker processes.

2.1 Java Runtime Environment

All DBMaker JTools (JServer Manager, JConfiguration Tool, and JDBA Tool) use the Java Runtime Environment. To start these programs jdk (Java Development Kit (version 1.3 is currently recommended)) or a JRE (Java Runtime Environment) must be installed on the system. Windows, Linux, and Sun Solaris (SunOS) versions of the JRE are included with jdk1.3 and are automatically installed on the system when DBMaker 4.1 is installed. Versions of JRE can be exclusively downloaded from the website of distributors for other platforms.

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.

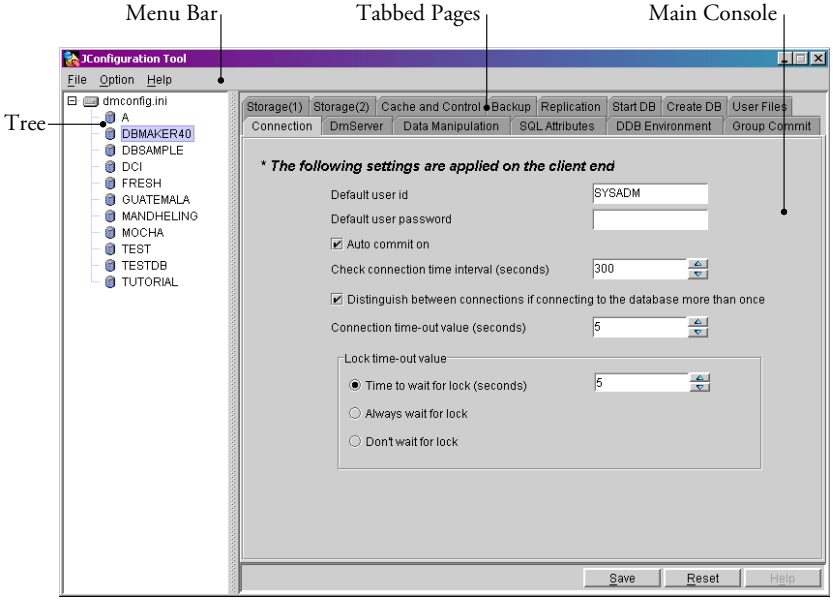


Figure 2-1 The JConfiguration Tool Workspace

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 interface is displayed in. The **Help** menu allows the user to access the help system.

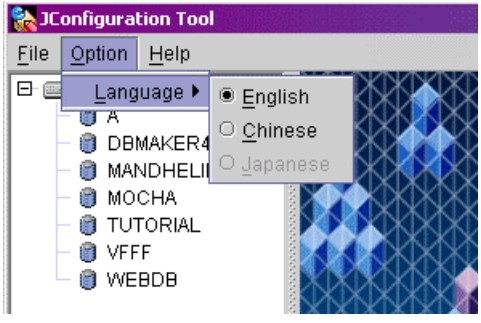


Figure 2-2 The Menu Bar: Language Preference

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.

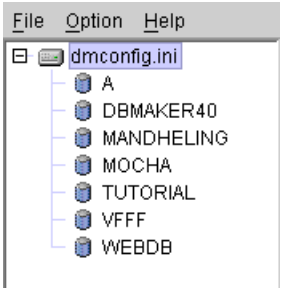


Figure 2-3 Tree Objects

Main Console

The main console consists of fourteen tabbed pages. A user 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 pressing Reset.

The Browse Button

USING THE SELECT FILE / SELECT PATH / SAVE DIALOG BOX

The SQL Attributes, Storage (1), Storage (2), Backup, Replication, 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 (...). Selecting the browse button (...) opens the following dialog box.

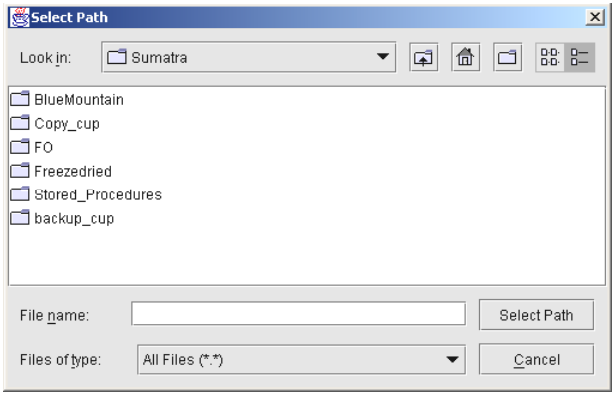


Figure 2-4 The Select Path Dialog Box

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

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

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

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

The list and detail buttons are currently inactive in the JConfiguration Tool.

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 on a directory name (down 1 level), or clicking the *up one level* button. The user can select an individual file or directory from the main window box by left clicking once on it. A user can also edit any directory or file name from the main window. To edit a file name or directory name, right click on the file or directory and make changes to the name. Pressing enter saves the new directory

or file name. Be sure that the name of the selected file or directory appears in the **File name** text 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 fourteen 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 a database. The fourteen tab page names:

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

3.1 Connection

Clicking the **Connection** tab in the main console opens the **Connection** page, which 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, and the lock time-out value set.

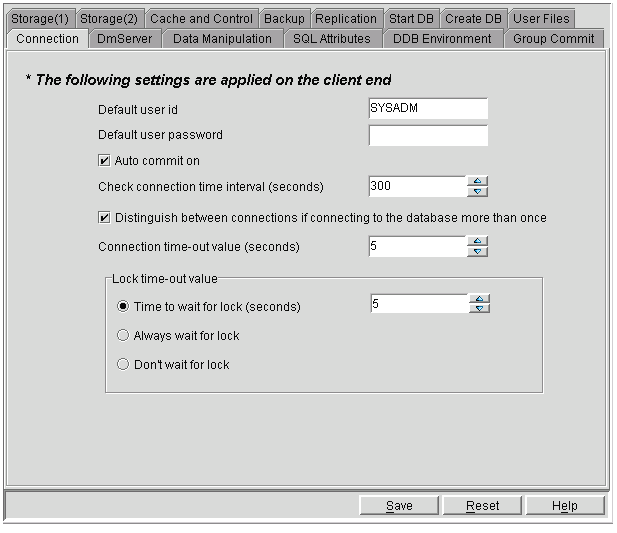


Figure 3-1 The Connection Page

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. Check 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 have been 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 in order 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. Sometimes a hardware error results in the client being cut off from the server, but DBMaker will still allocate resources for that client. DBMaker will release resources allocated for a client if the 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_DtClt** keyword in the **dmconfig.ini** file. The default setting is 300 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 if connecting to the database more than once** denotes that DBMaker will treat each duplicate connecting action 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. This setting corresponds to the **DB_DifCo** keyword in the **dmconfig.ini** file. The default setting is **enabled**.

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 set the

time-out value to change the waiting time. 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 been already been allocated to another transaction, the user must wait until the object is released. DBMaker will wait 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 will be returned. The waiting time can be shortened if the user prefers not to wait too long. To disable the time-out, set the combo box value to –1 or enable **Always Wait For Lock**. This will cause DBMaker to continue waiting until the lock is released. The combo box value can be set to 0 or **Don't Wait For Lock** is enabled to indicate that the user does not want to wait at all. 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.

3.2 DmServer

Clicking on the **DmServer** tab opens the **DmServer** page, which displays various network settings related to server connectivity and management. On this page the server address and port number are set for both the client and server. Enabling network encryption, saving a dmServer log file, and the idle time-out value are only set for the server.

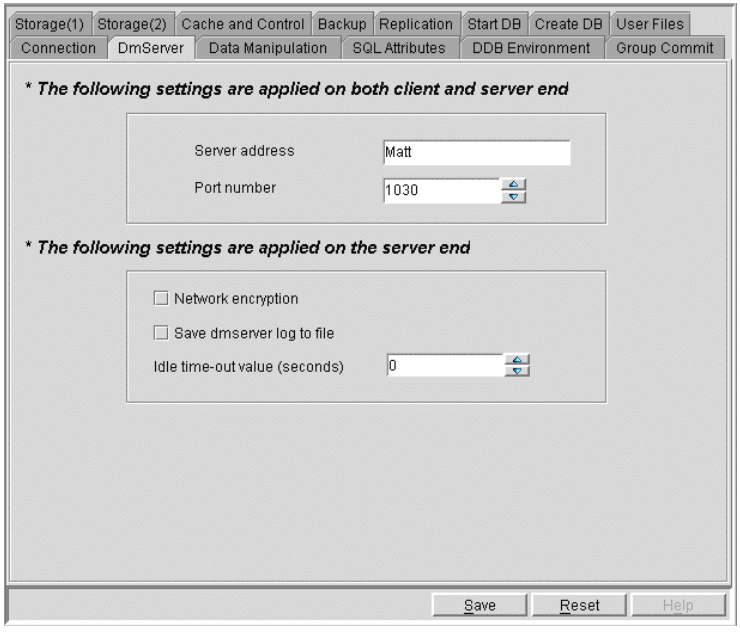


Figure 3-2 the DmServer Page

Client/Server Settings

Client/Server settings must be set on both client and server machines, 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 or the host name of that machine. If DNS (Domain Name Server) 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 will fail. If you want to run your application in single user database mode, you must leave the **Server Address** field blank. Please see your network administrator or any manual about TCP/IP networking 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. This number must exactly match on all client and server machines for a database, 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 from 1025 to 65535.

Server Settings

Server settings only need to be specified on the server machine, and include encryption of network traffic, whether or not to save the dmServer log to file, and the amount of time to wait for an idle connection before disconnecting it.

NETWORK ENCRYPTION

This setting specifies if DBMaker should turn network encryption on or off. If network encryption is turned on, all network data between DBMaker server and all clients will be encrypted. 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 DMSERVER LOG TO FILE

Enabling the **Save dmServer Log to File** setting saves all dmServer command line text to 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 will automatically disconnect any connection that has not had any database operations after 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 will keep all idle connections open. This value must be greater than the value in the **Check Connection Time Interval** field on the **Connection** page (Refer to *Setting the Connection Detection Interval* in section 3.1). If not, DBMaker will automatically reset this value to be greater than the **Check Connection Time Interval**. This setting corresponds to the **DB_ITimO** keyword in the **dmconfig.ini** file. The default value is 0.

3.3 Data Manipulation

Selecting the **Data Manipulation** tab opens the **Data Manipulation** page, which displays settings used to manage upper memory allocation for certain DBMaker's functions on the client. 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.

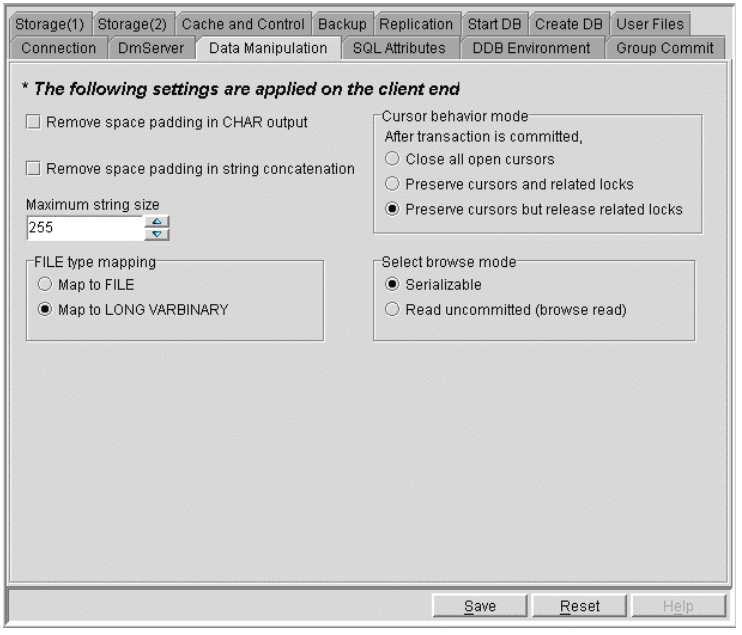


Figure 3-3 The Data Manipulation Page

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 will be 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 at both the client end and server end. If this feature is not set at the client end, then the setting is the same as it is at the server end. 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 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 will then be 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** .

Cursor Behavior Mode

This setting specifies the behavior of the cursor after the end of a transaction. Selecting **close all open cursors** indicates all cursors that are still open will be closed after any transaction is 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**.

Select Browse Mode

This keyword specifies the lock behavior of a select statement. Selecting **Serializable** denotes DBMaker will take an S lock on the result set of a SELECT statement. Selecting **Read uncommitted (browse read)** will not lock the result set of the SELECT statement. This setting corresponds to the **DB_Brows** keyword in the **dmconfig.ini** file. The default setting is **Serializable**.

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, as well as separate stored procedure directory options for both the clients and server.

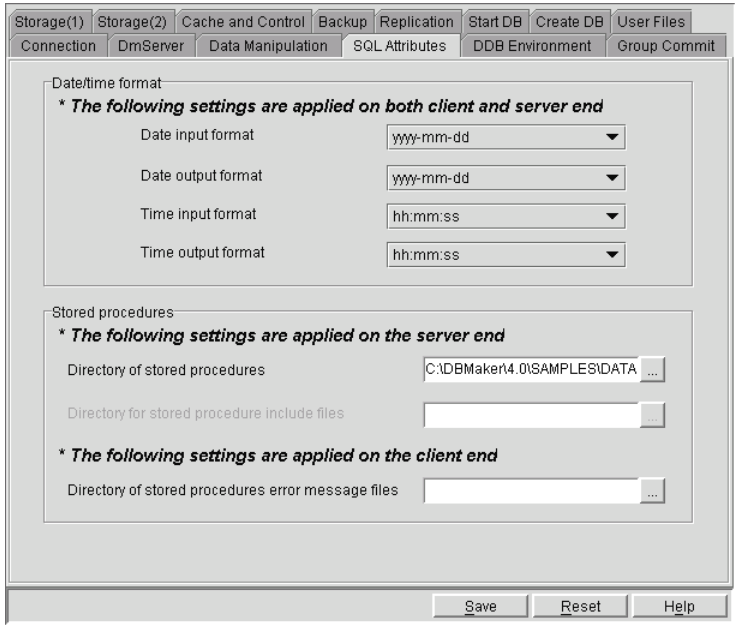


Figure 3-4 The SQL Attributes Page

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 the *ODBC Programmer's Guide — Appendix B* for more in-depth 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 that stored procedure include files are placed in. It is used when the user needs extra include files for stored procedures. This keyword is only useful in non-Windows operating systems. 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.

3.5 DDB Environment

Selecting the **DDB Environment** (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.

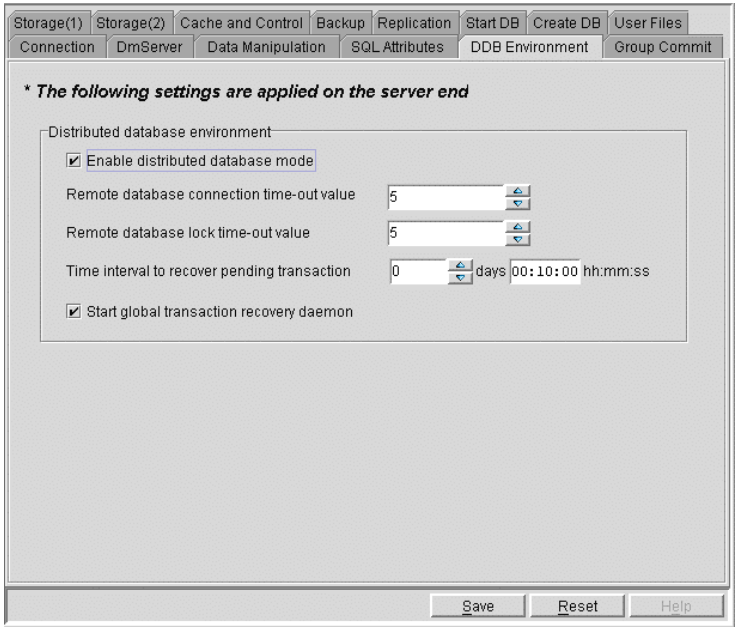


Figure 3-5 The DDB Environment Page

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 17.3 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_GTItv** 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; threshold, maximum wait time, and maximum number of transactions settings. 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.

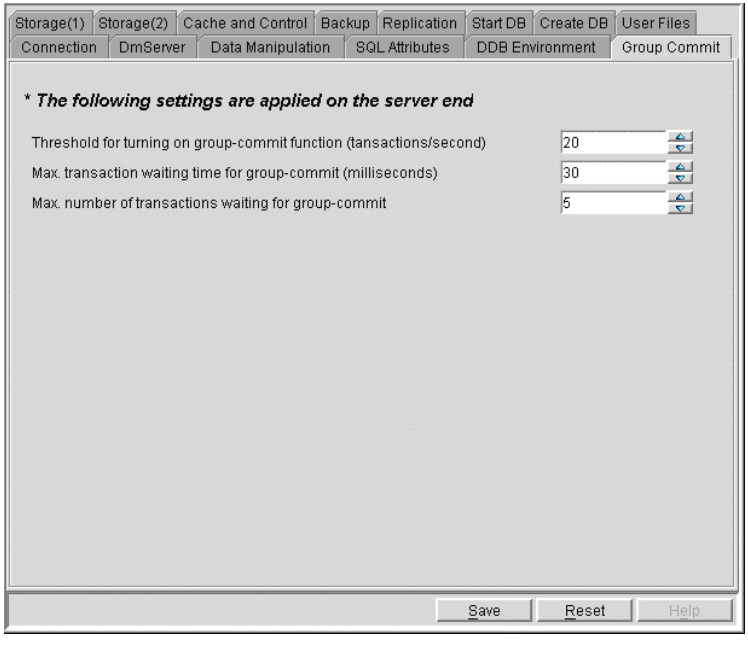


Figure 3-6 The Group Commit Page

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 30 milliseconds.

Maximum Number of Transactions in Wait State

Transactions waiting for a group-commit will wait until the maximum transaction waiting time before the group-commit. However, if a certain number of transactions waiting for a group-commit are reached, then the group-commit will occur regardless of the waiting time. This number of transactions is specified in the **Max. number of transactions waiting for group-commit** field. This setting corresponds to the **DB_GcMxw** keyword in the **dmconfig.ini** file. The default value is five transactions.

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, number of user data file pages, name of the user BLOB file, number of the user BLOB file frames, and the number of pages to extend a full BLOB file when using autoextend tablespaces. The settings on this page are only applied to the server.

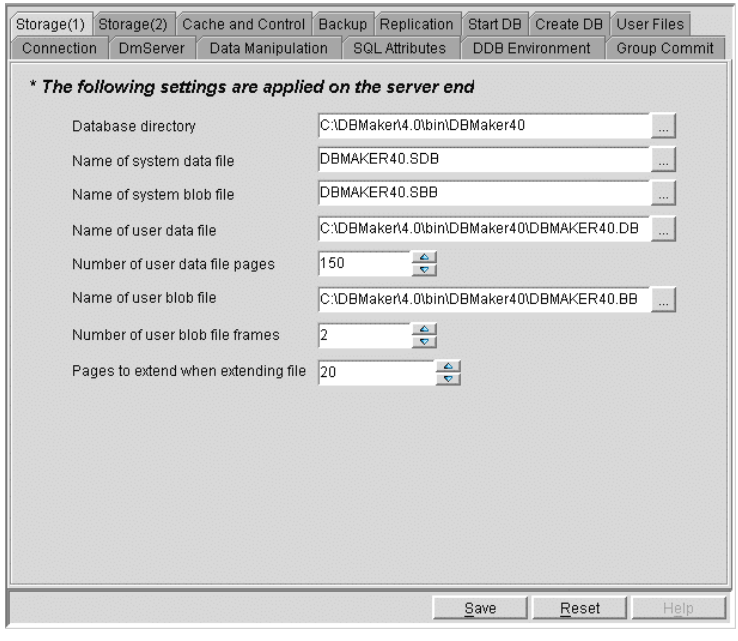


Figure 3-7 The Storage (1) Page

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)\4.1\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 4KB (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 4KB (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.

Number of User Data File Pages

Under the **User Data File** field is the **Number of User Data File Pages** field. This is used to control the number of pages that exist in the User Data File upon its creation. A higher number 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 number allocates less space. This number can be manually entered or adjusted using the arrows to the right. The default value is 150 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.

Number of User BLOB Frames

Under the **Name of User BLOB File** field is the **Number of User BLOB File Frames** combo box. This is used to control the number of frames that initially exist in the User BLOB File upon its creation. A higher number allocates more space in this file and allows for the insertion of more BLOB data before the file is automatically extended (refer to *Extending a File* in this section), a lower number allocates less space. This number can be manually entered or adjusted using the arrows to the right. The default value is two pages.

Extending a File

When all pages in the data file or BLOB file are full, DBMaker can automatically extend the number of pages or frames in the file to allow the database to grow. The **Number of Pages to Extend While Extending File** setting tells DBMaker how many pages or frames to add to the full file in the event that it is full. If the database administrator expects that the database will grow very quickly, then a higher number should be picked to lessen the frequency at which the file is appended. This number can be manually entered or adjusted using the arrows to the right. This setting corresponds to the **DB_ExtNp** keyword in the **dmconfig.ini** file. The default value is 20 pages.

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 the names of system temporary files. The settings on this page are only applied to the server.

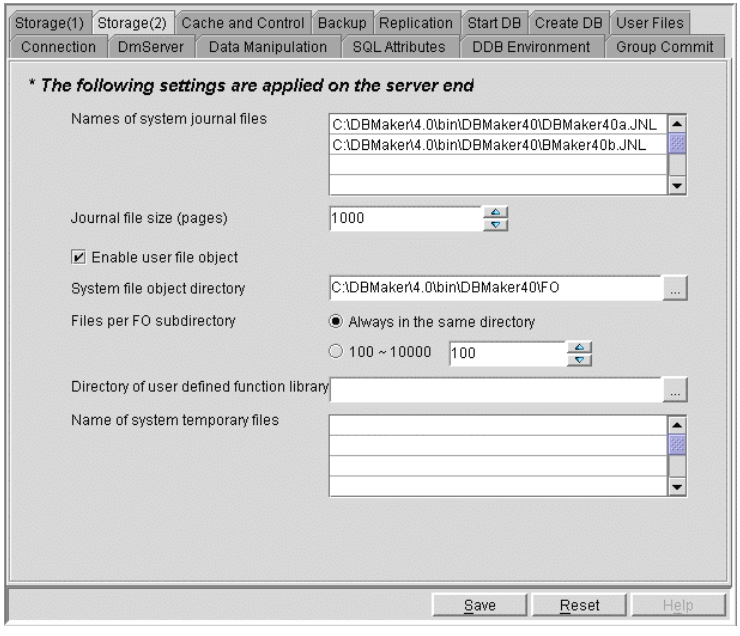


Figure 3-8 The Storage (2) page

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** (**in pages**) is set in the field under the **Names of System Journal Files** field. 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 by clicking on *The Browse Button* () next to the text field. Any changes to the file object path will apply upon starting the database. This setting corresponds to the **DB_FoDir** keyword in the **dmconfig.ini** file.

Files per File Object Subdirectory

DBMaker gives you the option to automatically create subdirectories within the FO directory. Each subdirectory is filled to a threshold value with new file objects. When the threshold is reached, DBMaker creates a new FO subdirectory. The number in the **Files per FO subdirectory** field indicates 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) that the user wants to be able to use in DBMaker. The DLLs stored in the Directory of User Defined Function Library are accessible to DBMaker and can be used in SQL statements or ODBC applications. The default path for the Directory of User Defined Function Library is<*DBMaker working directory*>**/shared/udf/**. Users can also enter a new path if a different one is desired. This can be achieved by typing the new path into the text field, or by clicking on *The Browse Button* () next to the text field. This setting corresponds to the **DB_LbDir** keyword in the **dmconfig.ini** file.

Names of System Temporary Files

System temporary files are used by DBMaker to store information about the database while the database is active. Up to eight temporary files may be specified. These files are removed when the database is shut down. If one or more filenames are written

here, DBMaker will use them to store temporary information about the database. The default path for the system temporary file is <*Database Directory*>\. Users can also enter new paths if different ones are desired. This can be achieved by typing the new paths into the text fields. This setting corresponds to the **DB_TpFil** keyword in the **dmconfig.ini** file.

3.9 Cache and Control

Selecting the **Cache and Control** tab opens the **Cache and Control** page, which displays the available communication and control area (DCCA) options used for performance tuning. Cache and control settings are used to 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 in-depth information on performance tuning, refer to Chapter 15 in the *Database Administrator's Guide*.

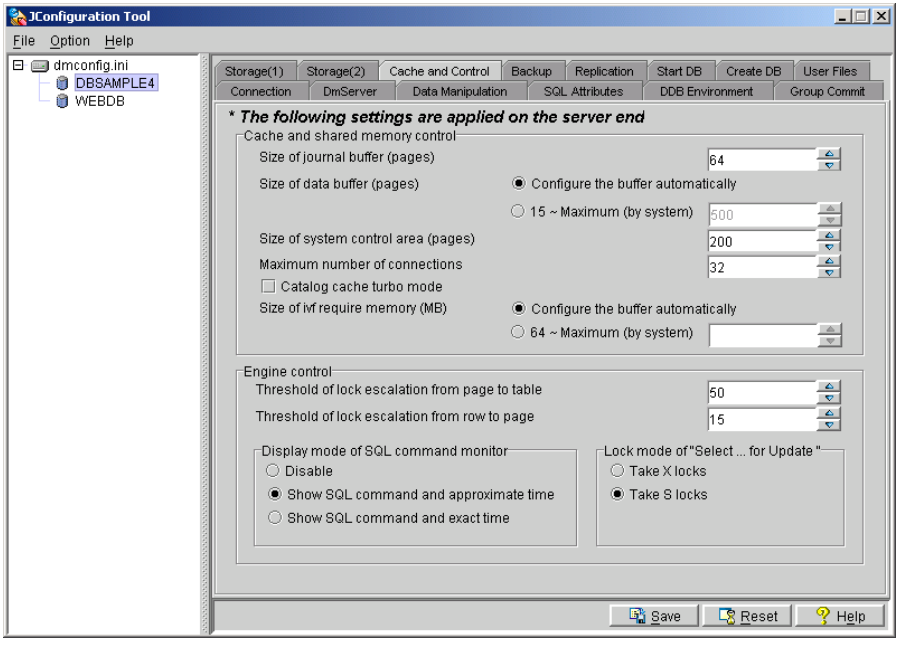


Figure 3-9 The Cache and Control Page

Cache and Shared Memory Control

JOURNAL BUFFER

The journal buffer stores the most recently used journal blocks as pages. When enough journal buffer pages are available, 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. The **Size of Journal Buffer (pages)** 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. Refer to the section "*Tuning Journal Buffers"* in Chapter 17 of the *Database Administrator's Guide* for in-depth information on optimizing this value. This setting corresponds to the **DB_NJnlB** keyword in the **dmconfig.ini** file. The default value is 64 pages.

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 to keep in the buffer. The data buffer can be manually set to a specific value or DBMaker can automatically configure it. To have DBMaker set the number of page buffers, select **Configure the buffer automatically**. The **Size of Data Buffer (pages)** 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. Increasing or decreasing the number of data pages in the data buffer has a dramatic effect on database performance. Refer to the section "*Tuning Page Buffer Cache"* in Chapter 17 of the *Database Administrator's Guide* for more in-depth information on optimizing this value. This setting corresponds to the **DB_NBufs** keyword in the **dmconfig.ini** file. The default value is **Configure the buffer automatically**.

SYSTEM CONTROL AREA

The System Control Area (SCA) is where concurrency control blocks are stored. Large transactions requiring many lock resources will 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 (refer to *Engine*

Control later in this section). The **Size of System Control Area (pages)** 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. Refer to the section "*Tuning the System Control Area"* in Chapter 17 of the *Database Administrator's Guide* for more in-depth information on optimizing this value. This setting corresponds to the **DB_ScaSz** keyword in the **dmconfig.ini** file. The default value is 200 pages.

MAXIMUM NUMBER OF CONNECTIONS

This setting controls the maximum number of connections that can simultaneously access a database. 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 32.

CATALOG CACHE TURBO MODE

Enabling the **Catalog Cache Turbo Mode** extends the lifetime of the catalog cache in the SCA. Refer to the section "*Tuning the Catalog Cache"* in Chapter 17 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 determine 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 will automatically escalate the lock to a table lock. This setting

corresponds to the **DB_LetPT** keyword in the **dmconfig.ini** file. The default value is 50 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 will automatically escalate the lock to a page lock. This setting corresponds to the **DB_LetRP** keyword in the **dmconfig.ini** file. The default value is 15 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 clicked. The most recent SQL command executed by the user and its approximate time of execution are shown if **Show SQL command and approximate time** is clicked. The most recent SQL command executed by the user and its exact time of execution are shown if **Show SQL command and exact time** is clicked. Displaying the exact time of execution uses more CPU resources and slows down the database. 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, which displays the available incremental backup and full backup options. 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 in-depth information on backup modes and procedures, refer to Chapter 14 in the *Database Administrator's Guide*.

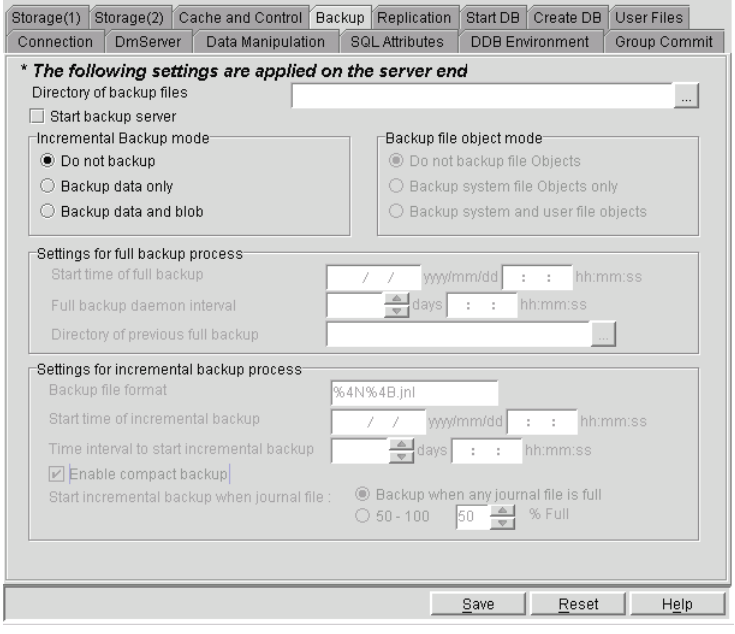


Figure 3-10 The Backup Page

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 in order for modes other than **Do not backup** to function. Selecting **Do not backup** enables NON-BACKUP mode, which causes the journal file's oldest records to be overwritten when the journal file is full. Selecting **Backup Data Only** enables BACKUP-DATA mode, which 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, which 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** will allow 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 (see *Settings for Incremental Backup Process*). For more in-depth information on backup modes, refer to Chapter 14 of the *Database Administrator's Guide.* This setting corresponds to the **DB_BMode** keyword in the **dmconfig.ini** file. The default mode is ***No Backup***.

Backup File Object Mode

The settings under the **Backup File Object Mode** effect how file objects are copied during the full backup process. Selecting **Do Not Backup File Objects** disables file backup during the full backup process. Selecting **Backup System File Objects Only** will result in system file objects being backed up during automatic full backups. Selecting **Backup System and User File Objects** will result in both system file objects and user file objects being copied to the backup directory during automatic full backups. This setting corresponds to the **DB_BkFoM** keyword in the **dmconfig.ini** file. The default mode is *No Backup*.

Setting the Backup File Directory

The **Directory of Backup Files** field shows the directory where the backup server puts all full backup files and incremental backup (journal) files. You should create a

backup directory on a different disk from the database files to prevent the loss of both the database and the backup files in the event of a media error. The default path for backup files is (*Database Directory*)**\Backup** and is automatically created by DBMaker. The total length of the backup directory path must not exceed 256 characters in length. Users should enter a new path for the backup file directory by typing the new path into the text field, or by clicking on *The Browse Button* () next to the text field. This setting corresponds to the **DB_BkDir** keyword in the **dmconfig.ini** file.

Starting Backup Server

Enabling the **Start Backup Server** check box activates the full 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 also can access settings in the **Settings for Full Backup Process** field. These settings must be configured properly for the full backup process to work. The default setting is *disabled*. This setting corresponds to the **DB_BkSvr** keyword in the **dmconfig.ini** file.

Settings for Full Backup Process

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

To set the time at which the first full 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; JConfiguration Tool automatically enters values into the first two spaces of the **yyyy** field and the first space of the **hh** field if values are entered into other fields. This setting corresponds to the **DB_FBkTm** keyword in the **dmconfig.ini** file. There is no default value.

NOTE *The full backup is processed only if the full backup start time is set.*

SETTING THE FULL BACKUP DAEMON INTERVAL

The number in the combo box specifies the time interval at which the Full Backup Daemon (See *Starting Backup Server*) 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 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_BkOld** 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 the 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 *The Browse Button* () next to 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.

Settings for Incremental Backup Process

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 1-4 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 the journal file belongs to.

➲ **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 14 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. JConfiguration Tool automatically enters values into the first two spaces of the **yyyy** field and the first space of the **hh** field if values are entered into other fields. 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 ***full***.

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.

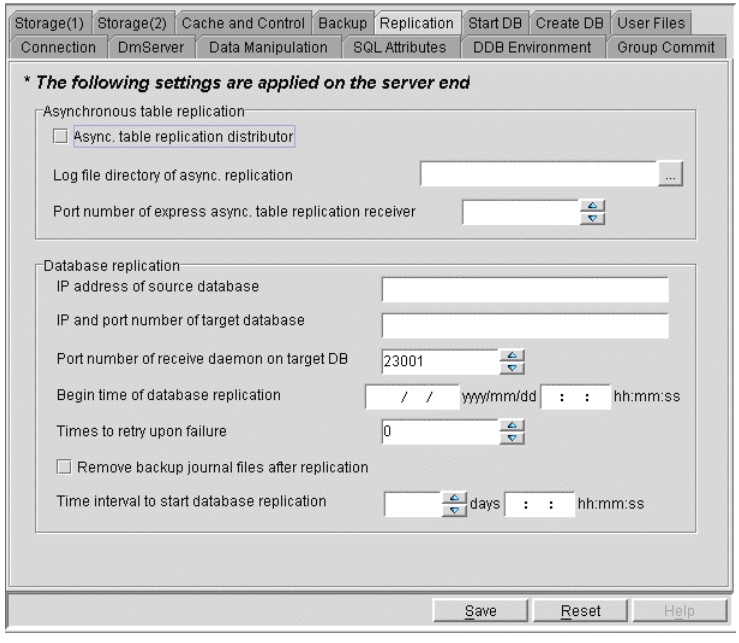


Figure 3-11 The Replication Page

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 uses a different method from ODBC calls to update target tables. Setting the port number denotes this database will start an express ATR receiver daemon that binds on this port. If the port number is not specified, the express ATR daemon will not be started. For more in-depth information, refer to Section 18.5 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 the DBMaker puts replication log files for asynchronous table replication. ATR error logs are associated with only the source database, and by default 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*)**\TRPLOG** and is automatically created by DBMaker. The total length of the ATR log file directory path must not exceed 256 characters in length. Users should enter a new path for the ATR log file directory by typing the new path into the text field, or by clicking on *The Browse Button* () next to the text field. This setting corresponds to the **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 on

and backup server mode must be set to *Backup data only* or *Backup data and BLOB* in order for database replication to function properly.

If access to a network is limited or bandwidth problems lead to a slow network, 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 functioning of replicated databases. Backing up the database to another location initially creates the slave database. Database replication can be executed afterwards to update the slave database so that it matches 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 in-depth information on distributed databases settings, refer to section 18.2 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_Primy** 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 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 (slave) databases **dmconfig.ini** file. DBMaker will support up to 8 slave databases for each primary database. The format should follow the following syntax.

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

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

➲ **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 (refer to the section *"Port Number*" in section 3.2) 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. 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. JConfiguration Tool automatically enters values into the first two spaces of the **yyyy** field and the first space of the **hh** field if values are entered into other fields. This setting corresponds to the **RP_BTime** keyword in the **dmconfig.ini** file.

TIMES TO RETRY ON FAILURE

Enter a number in the combo box to specify how many times DBMaker will try 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 should clear 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 the time interval at which database replication occurs in days. Next to the combo box is a field for time input, which specifies the time interval in hours, minutes and seconds. The total time interval is determined by adding the two values together, so inputting 1 into the **days** field and inputting 12:00:00 into the **hh:mm:ss** field would cause an incremental backup to be made every day and a half. 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, which displays the available options used when starting a database. The settings on this page include start database modes, error reporting email settings, multi-user mode, forced starting a database, resetting asynchronous table replication, starting I/O server, and updating statistics automatically. The settings on this page are only applied to the server.

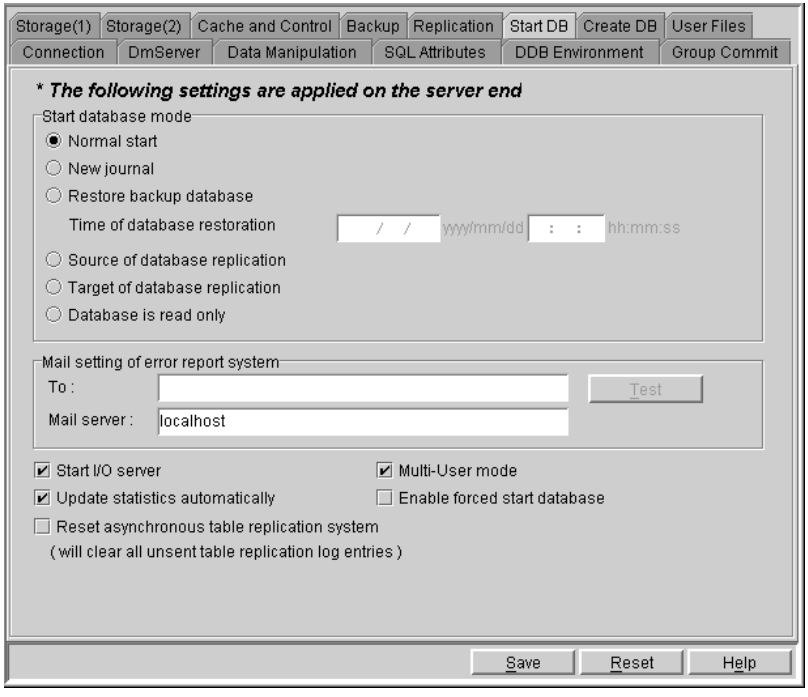


Figure 3-12 The Start DB (database) Page

Start Database Mode

There are 6 start-up modes. These modes correspond to values of the **DB_SMode** keyword in the **dmconfig.ini** file. The default value is *Normal Start.* For more in-depth information on database startup modes, refer to section 5.4 in the *Database Administrator's Guide*.

NORMAL START

Normal start starts up the database system according to the configuration parameters. If the database crashed last time, DBMaker will perform crash recovery automatically to bring the database to a consistent and stable state.

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 will be overwritten if the previous journal file names are kept. This setting must be selected if the user wants to change the journal file size, add more journal files, or change the journal file name. It is recommended to do an incremental or full backup before selecting this option.

RESTORE DATABASE BACKUP

Uses the backed up database files (including the journal file) to start the database. DBMaker will use 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 will prompt the user to copy backup files to the database and then prompt the user for each incremental file that is to be used 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** will revert 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 (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.

Mail Setting of Error Report System

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, a destination e-mail address and SMTP server for relaying the mail may be specified and DBMaker will automatically send 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 destination e-mail address setting is empty by default; therefore, the feature is disabled by default. The SMTP server for relaying mail is specified in the **Mail Server:** field, and corresponds to the **DB_ErMSv** keyword..

Enabling Multi-User Mode

Enabling **Multi-User Mode** permits more than one user to access a database. Disabling the check box allows only one user to access the database. The default

setting is *multi-user*. This setting corresponds to the **DB_UMode** keyword in the **dmconfig.ini** file.

Enabling Forced Start Database

Enabling the check box next to **Enable Forced Start Database** forces a database to start even if an error occurs while starting. 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.*

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*. The default setting is *disabled*. This setting corresponds to the **RP_Reset** keyword in the **dmconfig.ini** file.

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 default setting is *enabled*. 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.

Update Statistics Automatically

DBMaker keeps statistical information about database objects in the system tables. Periodically the database needs to be read and statistics recalculated and written to the system tables to ensure efficient database operation. Enabling **Update statistics**

automatically ensures that DBMaker periodically recalculates database statistics according to an internal scheduling daemon. This keyword corresponds to the **DB_StSvr** keyword in the **dmconfig.ini** file. The default setting is *enabled*.

3.13 Create DB

Selecting the **Create DB** (database) tab opens the **Create DB** page, which 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.

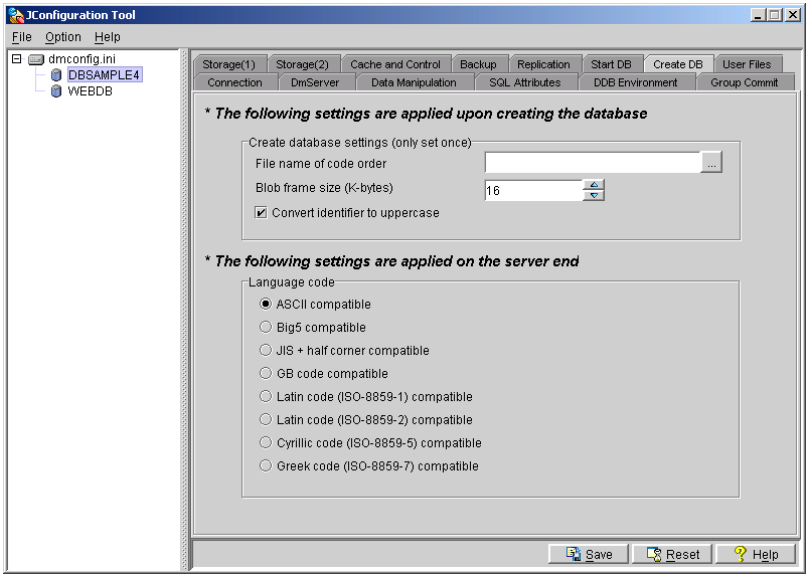


Figure 3-13 The Create DB (database) Page

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 as well as 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. For the above example, 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, the creator should be careful to 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 whole database to another machine, remember to 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 the new path into the text field, or by clicking on *The Browse Button* () next to the text field. This setting corresponds to the **DB_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: 16 (KB) for UNIX, Windows 95 or later

8 (KB) for Windows 3.1

valid range: 8-256 (**KB**) for UNIX, Windows 95 or later

8 (**KB**) for Windows 3.1

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, triggers, etc. User passwords and User-defined Functions are always case sensitive, and the database name is always case insensitive, regardless of this setting. Disabling the check box indicates the database all 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*.

Language Code

The language code for data tables can only be chosen at the time of the database creation by selecting one of the eight options in the **Language Code** field. The language code will affect 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)), **GB code compatible** (Simplified Chinese), **Latin code (ISO-8859-1)** compatible, **Latin code (ISO-8859-2)** compatible, **Cyrillic code (ISO-8859-5)** compatible, or **Greek code (ISO-8859-7)** compatible. For more in-depth information, 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, which displays the **User Files** dialog box. The options on this page are only applied to the server.

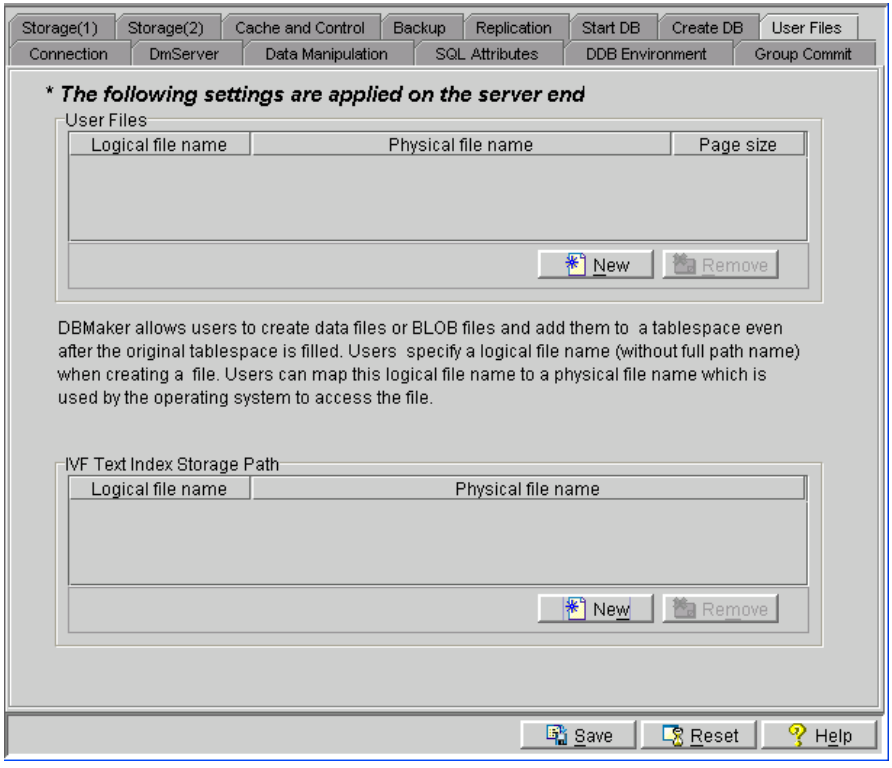


Figure 3-14 The User Files Page

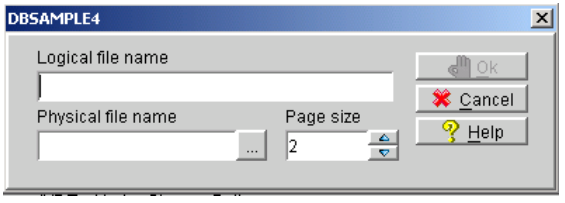
Creating a User-Defined File

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 that is being used by the operating system in order to access the file. This setting corresponds to the **User-defined filename** keyword in the **dmconfig.ini** file.

➲ **To create a user-defined file:**

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

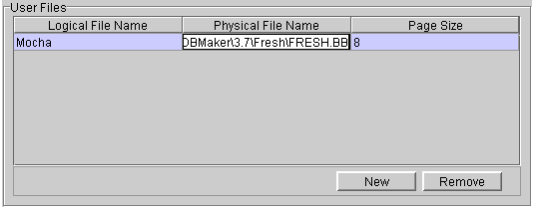


2. Enter the logical file name in the **Logical file name** field (this is the path referenced by SQL commands and by DBMaker to reference the data).

3. Enter a Physical File Name by entering a new path into the **Physical file name** field, or by clicking on *The Browse Button* (...). This is the operating system path. The maximum length for the physical file name path is 256 characters.

4. Enter a number of data pages into the **Page size** box. This indicates the number of data pages allocated for the user-defined file. The range of values for the number of data pages is between 2 and 524,287 bytes.

5. Click **OK.** A new user-defined file will appear 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 on the text allows the user to edit it. This is useful if data files are moved within the operating system directory; DBMaker will not recognize a change in the physical file name unless it is specified here.



6. To delete a user-defined file click **Remove**. DBMaker will no longer recognize 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** on 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 by clicking on *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 will appear 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 on the text allows the user to edit it. This is useful if an IVF file is moved within the operating system directory; DBMaker will 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 will no longer recognize the logical file name.

Glossary

Asynchronous Table Replication
A set of columns selected to form a projection that is replicated to the destination tables of another database. Asynchronous table replication occurs according to a schedule. Changes made to the source table are saved to a log file that is used to periodically update the destination tables.

BLOB
Acronym for Binary Large Objects. A table that consists of one column of LONGVARCHAR or LONGVARBINARY type data.

BLOB Frames
Measurement unit used to allocate disk space for BLOB type data.

Concurrency Control
A system of locks placed on objects to prevent multiple users from simultaneously manipulating the same data sets.

Coordinator Database
In a distributed database environment, the database that the client connects to. If the client accesses data from another database then that database is a participant database.

Daemon
A routine that automatically executes at a set time interval.

Data Pages
Data measurement unit for allocation of disk space.

Destination Database
The database that receives data for table replication. A database containing destination tables for a synchronous or asynchronous replication.
Destination Tables
Tables that receive replicated data from a source table. The table on the destination database that the data is replicated to.
Distributed Database Environment
A system of networked remote databases that allow for any table in the system to be accessed by a client on any of the participant (remote) databases.
Exclusive Lock (X Lock)
An access block placed on a database object that prevents other users from accessing the object.
Foreign Key
A column or set of columns whose rows contain the same values as the set of columns in the primary key or unique index of another table.
Fragment
Also called a horizontal partition, a fragment is the replication of a given range of data tuples.
Journal Blocks
Internal data measurement unit (514 bytes) that DBMaker uses to manage journal data.
Journal Buffer
Upper memory where current journal blocks are stored before writing them to disk.
Journal Pages
Measurement unit for allocation of disk space for journal files.
Lock
Locks allow only one user update and delete permission on the locked object at the time the object is locked.
Page Buffer
Upper memory allocated for data pages accessed by a user.

Participant Database
In a distributed database environment, a database that is accessed by the client through a coordinator database.
Primary Database
The source database for database replication.
Primary key
A column or sets of columns in a table that contain values that uniquely identify the rows in the table.
Projection
The selected columns from a base table chosen for replication.
Publication
A data set on the source table available for the replication.
Remote Database
A database located on another server from that being accessed by the client.
Remote Tables
Tables on a database located on a server that is different from the one the client is connected to.
Replication Domain
The replication fragment (horizontal partition) and projection (vertical partition) together are called a replication domain. It is the range of a table's data that is replicated.
Select Lock (S Lock)
An access block placed on a database object that allows other users to browse the object but not to update or delete any part of it.
Slave Database
A read-only database that receives data from a database replication.
Source Database
The database that contains source tables used to replicate.
Source Table
The table on the source database that the replicated data is from.
Subscription
The data set on the destination table to receive a publication.

Synchronous Table Replication

A set of columns selected to form a projection that is replicated to the destination tables of another database. Synchronous table replication occurs simultaneously – changes made to the source table are simultaneously made to the destination table(s).

Target Database

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

Index

A

Address
  Server, 3-6
Asynchronous Table Replication
  Enabling, 3-40
  Log File Directory, 3-40
  Port Number, 3-40
  Reset Asynchronous Table Replication System, 3-47
  Settings, 3-39
Auto Commit, 3-3

B

Backup
  Directory, 3-33
  Enabling the Backup Server, 3-33
  Full Backup Interval, 3-34
  Incremental Backup Settings, 3-34
  Main Console, 3-31
  Setting the start time for Full Backup, 3-33
  Storage location of previous Full Backup files, 3-34
BLOB File
  User File Size, 3-22
BLOB Files
  Extending User File, 3-22
  Frame Size, 3-51–3-52
  System File Location, 3-20
  User File Location, 3-21
Browse
  The browse button, 2-8
Browse Mode, 3-10

C

Cache and Control
  Main Console, 3-27
Case Sensitivity, 3-52
Catalog Cache Turbo Mode, 3-29
CHAR Output
  Remove space padding, 3-9
Connection
  Allowing Multiple Users, 3-46–3-47
  Detection interval, 3-3
  Differentiating Multiple Connections, 3-3
  Maximum Number, 3-29
  The Connection Page, 3-2

Time-out, 3-4
To remote databases, time-out value, 3-15
Create Database
BLOB Frame Size, 3-51–3-52
Case Sensitivity, 3-52
Main Console, 3-49
Order Definition File Location, 3-49–3-51
Cursor Behavior, 3-10

D

Data Buffer, 3-28
Data Files
Extending User File, 3-22
System File Location, 3-20
User File Location, 3-21
User File Size, 3-21
Data Manipulation
Console, 3-8
Database Directory, 3-20
Database Replication, 3-40–3-41
Clearing Journal Files, 3-43
Receive pirt Number, 3-42
Setting the number of times to reconnect to Remote Database, 3-42
Source Database IP Number, 3-41
Start Database Mode
Source Database, 3-46
Target Database, 3-46
Start Time Settings, 3-42
Target Database IP and Port Number, 3-41–3-42
Detection Interval
Connection, 3-3
Differentiating Multiple Connections, 3-3
Display Mode
SQL Statements, 3-30
Distributed Databases
Enabling, 3-15
Lock Time-out value, 3-15
Time-out value for establishing a connection, 3-15
dmServer
Address, 3-6
Console, 3-5
Network Encryption, 3-6
Port Number, 3-6
Saving Log Files, 3-7

E

Error Message files, 3-12
Error Report
Email address, 3-46
Escalation
Lock Escalation, 3-30

F

File
Select, 2-8
File Objects
Enabling User File Objects, 3-24
Files per FO subdirectory, 3-25
Storing System File Objects, 3-25
File Type Data
Mapping, 3-10
Files per FO subdirectory, 3-25
Forced Start Database
Enabling, 3-47

Frame Size, 3-51–3-52
Full Backup
  Default Directory, 3-33
  Enabling the Backup Server, 3-33
  Start Time, 3-33
  Storage location of previous Full Backup files, 3-34
  Time Interval, 3-34

G

Global Transaction Recovery Daemon
  .Time interval, 3-16

I

I/O Server
  Enabling, 3-47
Incremental Backup
  Backup Interval, 3-36
  Default Directory, 3-33
  Enabling Compact Backup, 3-37
  Enabling the Backup Server, 3-33
  File Format, 3-36
  Journal File Full Threshold Value, 3-37
  Start Time, 3-36

J

Journal Buffer, 3-28
Journal Files, 3-24
  Start Database Mode, 3-45

L

Lock
  Time-out value for remote databases, 3-15
Lock Escalation
  Page to Table, 3-29
  Row To Page, 3-30
Locks
  Escalation from Page to Table, 3-29
  Escalation from Row to Page, 3-30
  Mode on "Select ... for Update" statements, 3-30
  System Control Area, 3-29
Log Files
  dmServer, 3-7

M

Maximum Number of Connections, 3-29
Menu, 2-5

N

Network Encryption, 3-6
Notification
  Error report, 3-46

O

Object Names
  Case Sensitivity, 3-52
Order Definition File
  Location, 3-49–3-51

P

Password
  Default, 3-2
Path

Select, 2-8
Port Number, 3-6

R

Replication
Asynchronous Table Replication, 3-39
Database Replication, 3-40–3-41
Main Console, 3-38
Setting the number of times to reconnect to Remote Databases, 3-42
Restore Database
Start Database Mode, 3-45

S

Save, 2-8
SConcurrency Control
System Control Area, 3-29
SELECT
Lock Mode, 3-30
SELECT Statement
Browse mode, 3-10
Server
Address, 3-6
Network Encryption, 3-6
Port Number, 3-6
Space Padding
Remove for CHAR output, 3-9
Remove for String Concatenation, 3-9
SQL Attributes
Format, Date and Time, 3-12
SQL Statements
Display Mode, 3-30
Start Database
Allowing Multiple Users, 3-46–3-47
Enabling Forced Start, 3-47
Main Console, 3-44
Mode, 3-45
Reset Asynchronous Table Replication System, 3-47
Start Database Mode, 3-45
Database Replication Source, 3-46
New Journal, 3-45
Normal Start, 3-45
Read Only, 3-46
Restore Database, 3-45
Target Database, 3-46
Storage
Database Directory, 3-20
Main Console, 3-19
System File Objects, 3-25
Temporary Files, 3-26
User Defined Functions, 3-25
Stored Procedures
Directory, 3-12
Error Message files, 3-12
String Concatenation
Space Padding Removal, 3-9
String Size
Maximum, 3-9
System Control Area, 3-29
Catalog Cache Turbo Mode, 3-29
System File Objects
Storage, 3-25
System Files
BLOB File Location, 3-20
Data File Location, 3-20

T

Technical Support:, 1-4
Temporary Files
  Storage Location, 3-26
Time-out
  Connection, 3-4
  When establishing a connection to a remote database, 3-15
  When requesting a lock on a remote database, 3-15
Tree, 2-6

U

Update Statistics, 3-47–3-48
User Defined Function
  Maximum String Size, 3-9
User Defined Functions
  Storage, 3-25
User File
  Creating, 3-54–3-55
User File Objects
  Enabling, 3-24
User Files, 3-54–3-55
  BLOB File Location, 3-21
  BLOB File Size, 3-22
  Data File Location, 3-21
  Data File Size, 3-21
  Extending, 3-22
User ID
  Default, 3-2