

Bridgemate implementation guide

Document revision 1.16
May 2nd, 2008

© 2002-2008 Bridge Systems BV

Contents

Contents	2
Revision history.....	3
General.....	4
Purpose of this document.....	4
Data exchange between Bridgemate and external programs	4
File format.....	4
File extension.....	4
Explanation of used terms	4
Creating the database	6
Database tables	6
“Clients” table	7
“Section” table	7
“Tables” table	7
“RoundData” table	8
“ReceivedData” table	9
“IntermediateData” table	10
“PlayerNumbers” table	10
“Settings” table	11
“BiddingData” table	12
“PlayData” table.....	13
“HandRecord” table	14
Table Layout	15
Filling the tables with session data.....	18
Using command line parameters at startup BM Pro	20
How to deal with missing pairs.....	22
Updating movement information	23
Maximum values	25
Retrieving results	27
Guidelines for data retrieval.....	28
Checklist.....	29
Using the recorder version	30
Example of integration in scoring software.....	32

Revision history

Revision history of this document since document version 1.15:

Document version: 1.15
Release date: November 4th 2007
Bridgemate Pro Control version: 1.7.1

Changes:

- Bridgemate Pro Control 1.7.1 supports both Access97 (DAO 3.51) and Access2000 (DAO 3.60) file formats.

Document version: 1.16
Release date: May 2nd, 2008
Bridgemate Pro Control version: 1.7.15

Changes:

- AutoShutDownBPC field added to the settings table.

General

Purpose of this document

This document provides information about how to implement Bridgemate functionality in external programs (scoring programs, vugraph, etc). It is not a manual for how to use the Bridgemate. By reading this document and examining the sample files, you should be able to integrate the wireless Bridgemate in your own scoring software and make full use of its functionality. We strongly advise to read the English manual of Bridgemate as well in order to have full understanding of the Bridgemate scoring system.

Data exchange between Bridgemate and external programs

All data communication between Bridgemate and external programs is done through a single database file. External programs do not control directly the Bridgemate hardware. The database file is easy to create and read, and therefore reduces the implementation time of Bridgemate in your program to an absolute minimum. Chapter 1 of the Bridgemate manual describes the role of the database file. This document explains the full details and structure of the database.

File format

The Bridgemate database files are in Microsoft Access 97 format (Access version 8.0) or Access 2000 format. The database should not be password protected. Bridgemate Pro Control 1.6.28 and earlier versions only support Access 97 file format. Bridgemate Pro Control 1.7.1 and higher supports both Access 97 and Access 2000 file format.

When accessing the database from within an external program, Microsoft Data Access Objects 3.51 connection method (DAO351) is necessary when file format Access 97 is used; Microsoft Data Access Objects 3.60 (DAO360) is necessary when file format Access 2000 is used.

The Bridgemate database files can be opened using Microsoft Access 2000 or Microsoft Access 97. When using the file format Access 2000, the file cannot be opened by Microsoft Access 97, but only by Access 2000.

Please take note that when a Bridgemate database file is opened in Microsoft Access, you should never use the "convert database" option when you are prompted to do so.

File extension

Extension of the database file is *.bws. However, they can be opened in Access without any problems.

Explanation of used terms

- § BM Pro or Bridgemate Pro Control: Windows software program to communicate with the Bridgemate base station, which communicates with the wireless Bridgemate terminals. Movement data is uploaded by this program to the base station, and results are retrieved from it the other way around.
- § Bws database (wireless session database file): file that is used for storing all information which is needed for communication with the Bridgemate base station and storing results obtained from the base station.
- § (Local) scoring software: Windows software program that is used for scoring bridge sessions, calculating results, printing rankings etc and which is adapted to interface the Bridgemate wireless session database files.

§ External software: can be scoring software, but can also be other software like VuGraph software

Creating the database

Your program should be able to create the database file and fill in the appropriate data (records). Next, the Bridgemate Pro Control program will open up this database, retrieve all the relevant data and store results and other information in the database. Finally, your scoring program should retrieve this information and process it.

Database tables

The database contains ten tables:

Clients

More than one computer can update the database at the same time with data received from the Wireless Bridgemate Server. All computers are registered in this table.

Section

Contains the sections which are being used in the bridge session.

Tables

Contains all the existing tables which are in use in the bridge session.

RoundData

Contains information about the movements that are in use.

ReceivedData

Contains the data that is received from the Wireless Bridgemate server.

IntermediateData

Contains intermediate data (=data after entry of contract/lead card but before result) that is received from the Wireless Bridgemate server. Note that the Intermediate table will never have the final result (e.g. +2, -3, =) stored. The complete board results will be stored in the ReceivedData table.

PlayerNumbers

Contains member numbers of the players which can be entered on the Bridgemate.

BiddingData

Contains bidding data that is received from the Wireless Bridgemate server (recorder version only).

PlayData

Contains play data that is received from the Wireless Bridgemate server (recorder version only).

HandRecord

Contains hand record information that is used to validate when recording the play (recorder version only).

Settings

Contains information about configuration settings for the Bridgemate. Using this table, the scoring program can control the Bridgemate settings.

The tables BiddingData, PlayData and HandRecord are only required when using the "recorder version" of the Bridgemate. The table IntermediateData is required when you want to process the contract and lead card (if in use) directly after entry, but before result is entered. This can be used for showing contracts while play is not finished.

“Clients” table

Purpose of the Clients table is to keep track of the different client computers which can edit the database with data from the Wireless Bridgemate Server.

A new record will be added for each client computer.
The Clients table consists of two fields.

ID

This is an autonumbering field and contains the ID for each client. Client ID's are automatically assigned each time a new client logs on at the database. The client cannot manually set or edit its ID.

Computer

This is the name of the client computer.

“Section” table

The Section table contains information about the sections which are used in the bridge session. Each section has its own record.

The table consists of 5 fields.

ID

Contains the ID belonging to the section. Mostly it will start from 1, but this is not necessarily the case. ID is directly linked with the letter assigned to the section.

Letter

Contains the letter belonging to the section. Each section has always a letter indicating the uniqueness of the section. A clear link with the ID exists conform the following table:

ID	Letter
1	A
2	B
...	...
26	Z

Tables

Contains the number of tables for each section.

Missing Pair

When there is an odd number of pairs, the number of the missing pair is recorded in this field.

“Tables” table

The table “Tables” contains all the tables of the bridge session. Each table has its own record.

Section

Contains the ID of the section to which the table belongs. This field refers to the ID field in the Section table

Table

The number of the table is stored in this field.

ComputerID

Contains the ID of the client computer which deals with this table. It refers to the field ID in Clients table.

Status

Indicates the status of the table:

- 0 Table not uploaded to server yet
- 1 Table has been uploaded to server

LogOnOff

Indicates whether the table has been logged on or off to the server

- 2 Table has not logged on to server yet
- 1 Table has logged on to server

CurrentBoard (recorder version only)

Internal value used by the recorder version of the Bridgemate to indicate the current board that is being recorded. Writing to this field is only allowed by Bridgemate Pro Control.

CurrentRound (recorder version only)

Internal value used by the recorder version of the Bridgemate to indicate the current round that is being recorded. Writing to this field is only allowed by Bridgemate Pro Control.

UpdateFromRound

Can be set with a value indicating that the RoundData table has been updated from that round and higher. Standard this value is 0.

“RoundData” table

Contains information about the movement which is used in the bridge session. A different record is used for each table/round combination.

Example: a session of 6 rounds and 8 tables uses in total $6 \times 8 = 48$ records.

Section

Contains the ID of the section to which the table belongs. This field refers to the ID field in the Section table

Table

Contains the number of the table.

Round

Contains the number of the round.

NSpair

Indicates the pair which is sitting north-south at this table in this round.

EWpair

Indicates the pair which is sitting east-west at this table in this round.

LowBoard

Contains the lowest number of the set of boards which are being played at this table in this round.

HighBoard

Contains the highest number of the set of boards which are being played at this table in this round.

CustomBoards

This field is used to indicate a series of boards that are not consecutive. Whenever a value is in this field, the fields LowBoard and HighBoard are omitted and the value in CustomBoards is used. The board numbers must be separated using comma's. Example: 1,2,5,6,7,10. Leave this field empty when you want to use LowBoard – HighBoard.

“ReceivedData” table

This table contains the score data which is received by the Wireless Bridgemate Server. Each recorded score is stored in a new record.

ID

Autonumbering field which adds an ID for each score.

Section

Contains the ID of the section to which the score belongs. This field refers to the ID field in the Section table

Table

Contains the number of the table where the score was registered.

Round

Contains the number of round in which the score was registered.

Board

Contains the number of the board at which the score was registered.

PairNS

Contains the number of the north-south pair as it was uploaded to the server according to the movement.

PairEW

Contains the number of the east-west pair as it was uploaded to the server according to the movement.

Declarer

Contains the number of the pair that was the declarer on this board.

NS/EW

Contains the direction in which this score was registered: NS or EW, or either N, S, E, or W.

Contract

Contains the contract of the result. Notations:

t c x (t = 1-7 ; c = C/D/H/S/NT ; x = x/xx)

PASS

Examples :

1 H

2 NT

4 S x

3 D xx

7 C

PASS

Contract field is blank in the following situations :

- arbitral score given by the tournament director
- board not played

Result

Contains the result. Possible notations are:

-1
-2
...
-13
= (contract exactly made)
+1
+2
...
+6

Result field is blank in the following situations :

- arbitral score given by the tournament director
- board not played
- PASS

LeadCard

Text field containing the lead card.

Examples:

S2 (2 of spades)
DA (Ace of diamonds)
C10 (10 of clubs)

Remarks

This field contains special information about:

- Board played in wrong direction: "Wrong direction"
- Arbitral score: "Arbitral score" when undecided, or the arbitral score itself.
- Board not played: "Not played"

See the chapter Retrieving results for more information.

DateLog

Contains the date when the score was registered in the database.

TimeLog

Contains the time when the score was registered in the database.

Processed

Yes/no field which can be used by external programs to indicate whether the record has been processed.

Processed1 ... Processed4

Same field as "Processed" which can be by used by other external programs.

Erased

Yes/no field which determines if the score has been deleted from the server by a remote Bridgемate. Yes value means deleted.

"IntermediateData" table

This table has exactly the same layout as the ReceivedData table.

"PlayerNumbers" table

Before start of the session players can enter their member code in the Bridgemate for automatic registration of the names. These codes are stored in the table PlayerNumbers. It contains the following fields:

Section

Contains the ID of the section to which the table belongs. This field refers to the ID field in the Section table

Table

Contains the number of the table.

Direction

Contains either the letter N, S, E or W indicating the direction of the player.

Number

Contains the number of the player. If the player doesn't enter his number, the field remains blank.

“Settings” table

The Bridgemate has several configuration settings to adjust it each individual needs. By default these settings are configured from the menu Tools → Options in BM Pro. However, it is also possible to configure these settings from the scoring program by adding the table Settings to the database and fill this table with the desired settings.

There are in total 17 options of the Bridgemate you can configure through the Settings table. For each option you add a field to the table. Only one record in this table is required and in this record you store the values of the settings. If there is more than one record in this table, record two and higher are ignored by BM Pro. There is no need to add all fields, just add the fields for the settings you want to control by the scoring program.

The fields are as follows. Between parentheses is shown which values can be used:

ShowResults

Show previous results of the board just played (true/false)

ShowOwnResult

Show your own result in this list of previous results (true/false)

RepeatResults

Enable the possibility to repeat the results after all previous results have been shown (true/false)

MaximumResults

Maximize the number of shown previous results (0 till 127. 0 means unlimited)

ShowPercentage

Show the percentage obtained on the board just played (true/false)

GroupSections

Group all sections together. If grouped, results of boards played in other sections will also be shown and the percentage will be calculated across all sections (true/false)

ScorePoints

Show score points from perspective of North-south or from declarer (north-south = 0; declared=1)

EnterResultsMethod

Enter results as number of tricks won/lost compared to the contract (=0) or as total tricks won (=1)

ShowPairNumbers

Show pair numbers in the round information screen (true/false)

IntermediateResults

Send the contract and leadcard directly after entry (true/false)

AutopoweroffTime

Set the autopower-off time (5 through 60)

VerificationTime

Set the time that the verification message is shown on the screen (1 through 7)

ShowContract

Show contract as symbols (=0) or letters (=1)

LeadCard

Enter leadcard (true/false)

MemberNumbers

Enter member numbers (true/false)

MemberNumbersNoBlankEntry

Indicate if blank member number entry is allowed (true/false)

BoardOrderVerification

Verify the board number on correct order of entry (true/false)

HandRecordValidation (recorder use only)

Validate the recording of played cards against the hand records (true/false)

AutoShutDownBPC

Automatically shut down BPC after all tables have logged off (true/false)

Note:

- When adding a field, the value of the field will always overwrite the setting in BM Pro. If your program doesn't offer configuration of a certain setting, then don't include that field in the Settings table.
- You are advised to use the GroupSections field to indicate whether the sections should be combined. When you play different sections and calculate them separately, set this field to false. When you calculate the sections as one, set this field to true.
- AutoShutDownBPC cannot be set through the settings in BPC and should always be set from the scoring program.

“BiddingData” table

This table contains the bidding recording data which is received by the Wireless Bridgemate Server. Each bid is stored in a new record. This table requires the recorder version of the Bridgemate.

ID

Autonumbering field which adds an ID for each bid.

Section

Contains the ID of the section to which the bid belongs. This field refers to the ID field in the Section table

Table

Contains the number of the table where the bid was registered.

Round

Contains the number of round in which the bid was registered.

Board

Contains the number of the board at which the bid was registered.

Counter

Contains a number that indicates the order of the bid in the auction.

Direction

Indicates by which player this bid was made: N, S, E or W

Bid

Contains the bid itself. The bid is noted as:

- 1 to 7 followed by color C, D, H, S, NT. Example: 1C, 2D, 3NT, 7S
- X or XX
- PASS
- SkipBid. This value indicates that one or more players were skipped in making a bid.

DateLog

Contains the date when the bid was registered in the database.

TimeLog

Contains the time when the bid was registered in the database.

Erased

Yes/no field which determines if the bid has been erased by pressing the Cancel button during the recording of the bidding. Yes means the bidding has been erased. The Counter field will mention the correct order number that belongs to the erased bid.

“PlayData” table

This table contains the play recording data which is received by the Wireless Bridgemate Server. Each played card is stored in a new record. This table requires the recorder version of the Bridgemate.

ID

Autonumbering field which adds an ID for each card.

Section

Contains the ID of the section to which the played card belongs. This field refers to the ID field in the Section table

Table

Contains the number of the table where the card was registered.

Round

Contains the number of round in which the card was registered.

Board

Contains the number of the board at which the card was registered.

Counter

Contains a number that indicates the order of the played card in the progress of the game.

Direction

Indicates by which player this card was played: N, S, E or W

Card

Contains the card itself. The card is noted as:

- Color indicated by C,D,H or S, directly followed by 2 till 9, T, J, Q, K or A. Example: D2, H9, ST (ten of spades), CA (ace of clubs).
- Color indicated by C,D,H or S, directly followed by x (small letter x). This means that the operator did not have enough time to register the exact card and indicated it with “small card”.

DateLog

Contains the date when the card was registered in the database.

TimeLog

Contains the time when the card was registered in the database.

Erased

Yes/no field which determines if the played card has been erased by pressing the Cancel button during the recording of the played cards. Yes means the card has been erased. The Counter field will mention the correct order number that belongs to the erased card.

“HandRecord” table

Section

Contains the ID of the section to which the played card belongs. This field refers to the ID field in the Section table

Board

Contains the number of the board at which the card was registered.

NorthSpades

NorthHearts

NorthDiamonds

NorthClubs

EastSpades

EastHearts

EastDiamonds

EastClubs

SouthSpades

SouthHearts

SouthDiamonds

SouthClubs

WestSpades

WestHearts

WestDiamonds

WestClubs

Contains the cards for each color in each direction. Cards are indicated by the letters 2 till 9, T (ten), J (jack), Q (queen), K (king) and A (ace). Example: field *NorthSpades* contains the value AT2, this means that North has the cards ace, ten and two of spades.

The total length of all 16 fields together must be 52.

The total length of all 4 direction fields together must be 13.

The total length of all 4 color fields together must be 13.

When there are no cards in a specific direction and color, leave the field empty.

It is not allowed to use “10” to indicate the then card. “T” must be used instead.

See the next chapter for more information on the table and field layout.

Table Layout

Database type, table and field definitions

The database should be generated as a Microsoft Access 97 database, which is DAO 3.51, or as a Microsoft Access 2000 database, which is DAO 3.60. Do not make the database password protected.

The database contains 11 tables which were described earlier in this document. Their names are:

- \$ Clients
- \$ Section
- \$ Tables
- \$ RoundData
- \$ ReceivedData / IntermediateData
- \$ PlayerNumbers
- \$ Settings
- \$ BiddingData
- \$ PlayData
- \$ HandRecord

(The tables BiddingData, PlayData and HandRecord are for the recorder version of the Bridgemate only)

These tables have the following fields definitions:

Note:

- \$ No indexes are needed on any field
- \$ The property "required" is set to false for any field

Table Clients

Field name	Type	Field size	AllowZeroLength	DefaultValue
ID	long integer, autonumber			
Computer	text	255	no	

Table Section

Field name	Type	Field size	AllowZeroLength	DefaultValue
ID	integer			
Letter	text	2	no	
Tables	integer			
MissingPair	integer			0

Table Tables

Field name	Type	Field size	AllowZeroLength	DefaultValue
Section	integer			
Table	integer			
ComputerID	integer			0
Status	integer			0
LogOnOff	integer			2
UpdateFromRound	integer			0

CurrentRound integer 0
 CurrentBoard integer 0
 (the last two fields are for the recorder version of the Bridgemate only)

Table RoundData

Field name	Type	Field size	AllowZeroLength	DefaultValue
Section	integer			
Table	integer			
Round	integer			
NSPair	integer			
EWPair	integer			
LowBoard	integer			
HighBoard	integer			
CustomBoards	text	255	yes	

Table ReceivedData / IntermediateData

Field name	Type	Field size	AllowZeroLength	DefaultValue
ID	long integer, autonumber			
Section	integer			
Table	integer			
Round	integer			
Board	integer			
PairNS	integer			
PairEW	integer			
Declarer	integer			
NS/EW	text	2	yes	
Contract	text	10	yes	
Result	text	10	yes	
LeadCard	text	10	yes	
Remarks	text	255	yes	
DateLog	date/time			
TimeLog	date/time			
Processed	true/false			false
Processed1	true/false			false
Processed2	true/false			false
Processed3	true/false			false
Processed4	true/false			false
Erased	true/false			false

Table PlayerNumbers

Field name	Type	Field size	AllowZeroLength	DefaultValue
Section	integer			
Table	integer			
Direction	text	2		
Number	text	16	yes	

Table Settings

Field name	Type	Field size	AllowZeroLength	DefaultValue
ShowResults	true/false			true
ShowOwnResult	true/false			true
RepeatResults	true/false			false
MaximumResults	integer			0

ShowPercentage	true/false		true
GroupSections	true/false		false
ScorePoints	integer		1
EnterResultsMethod	integer		0
ShowPairNumbers	true/false		true
IntermediateResults	true/false		false
AutopoweroffTime	integer		20
VerificationTime	integer		2
ShowContract	integer		1
LeadCard	true/false		false
MemberNumbers	true/false		false
MemberNumbersNoBlankEntry		true/false	false
BoardOrderVerification	true/false		true
HandRecordValidation	true/false		true
AutoShutDownBPC	true/false		false

Table BiddingData

Field name	Type	Field size	AllowZeroLength	DefaultValue
ID	long integer, autonumber			
Section	integer			
Table	integer			
Round	integer			
Board	integer			
Counter	integer			
Direction	text	2	yes	
Bid	text	10	yes	
DateLog	date/time			
TimeLog	date/time			
Erased	true/false			false

Table PlayData

Field name	Type	Field size	AllowZeroLength	DefaultValue
ID	long integer, autonumber			
Section	integer			
Table	integer			
Round	integer			
Board	integer			
Counter	integer			
Direction	text	2	yes	
Card	text	10	yes	
DateLog	date/time			
TimeLog	date/time			
Erased	true/false			false

Table HandRecord

Field name	Type	Field size	AllowZeroLength	DefaultValue
Section	integer			
Board	integer			
NorthSpades	text	13	yes	
NorthHearts	text	13	yes	
NorthDiamonds	text	13	yes	
NorthClubs	text	13	yes	
EastSpades	text	13	yes	
EastHearts	text	13	yes	

EastDiamonds	text	13	yes
EastClubs	text	13	yes
SouthSpades	text	13	yes
SouthHearts	text	13	yes
SouthDiamonds	text	13	yes
SouthClubs	text	13	yes
WestSpades	text	13	yes
WestHearts	text	13	yes
WestDiamonds	text	13	yes
WestClubs	text	13	yes

Filling the tables with session data

In this document we assume you are using only one computer and one Bridgemate server in your session. The tables Clients, Section, Tables, RoundData, HandRecord are to be loaded with data from your scoring software. The ReceivedData, PlayerNumbers, BiddingData, PlayData table should remain empty, but will receive data from the Bridgemates during play.

Table Clients

The Clients table contains the names of the computers which are used in the Bridgemate network. Each record contains one computer name. The name of the computer is stored in the field "Computer" and must be identical to the name produced by the Windows API call. The field is case sensitive.

After adding the record and storing the computer name, retrieve the value of the ID field (which should be 1 for the first record added).

Table Section

This table contains basic information about the sections of your session. For every section, you add one record to this table. Usually the sections are numbered as A, B, C, etc. Every section contains its own ID, for which we recommend to start with 1 and increment with 1 for each section. However, if you have any special reasons (e.g. to equal section numbering with your own scoring software), you are free to choose other numbers. The ID is used in other tables of this database to refer to the section.

Besides an ID, every section is marked with a character, which can be A through Z. The character code and the ID are compulsory, you cannot omit these.

The Description is optional however. At the moment it is not used by BM Pro, so you can leave this field blank.

The Tables field contains the number of tables you have in each section.

Finally, the MissingPair field contains the number of the missing pair. If there is no missing pair in this section, leave this as 0 (=standard value).

Table Tables

This table contains information about the tables of all sections. For every table you add one record to the table.

The Section field contains the ID of the section to which the table belongs. This number must be equal to the number in the ID field of the Section table.

The Table field contains the number of the table. If you have tables 1 till 7, you add seven records with table numbers 1 through 7. If your tables are numbered 101 till 107, you add seven records but with table numbers running from 101 to 107. The maximum number you can assign to a table is 511.

Each table is assigned to a specific computer in the Bridgemate network. The ComputerID field refers to the ID field of the Client table and indicates which computer (=client) will handle this table. The number in this field must equal the ID number of the client for being assigned to that client.

The fields Status, LogOnOff, CurrentRound, CurrentBoard and UpdateFromRound are automatically filled up with their default values and don't require any change. For more information on UpdateFromRound, see chapter "Updating round information".

Table RoundData

The RoundData table contains information about the movements which are in use. Each record holds the pair and board numbers for a specific table and round.

The Section and Table field refer again to an existing table in this session. The combination of this two should exist in the Tables table. For every table present in the Tables table, there should be information in the RoundData table.

The Round field contains the number of the round for which information is stored in this record. Always include all rounds, even when a round will not be played at a table for reasons of a missing pair or empty table.

The NSPair and EWPair fields contain the numbers of the pairs who are expected to sit at this table. The LowBoard and HighBoard fields contain the numbers of the lowest board and highest board which are to be played. Version 1.5.x and lower of BM Pro can only handle series of boards, e.g. 1-4, 7-8, 11-15, etc. It cannot handle non-consecutive board series in one round like board 1 and 4 till 6. Version 1.6.x and higher can handle series of board numbers that are non-consecutive. Use the field CustomBoards to enter the individual board numbers, using comma's to separate them. Example: 1,2,5,6,7,10. Whenever the field CustomBoards contains a value, the fields LowBoard and HighBoard are neglected and the CustomBoards value is used. When you leave this field empty, LowBoard and HighBoard are used.

In case of an "empty table" (this means according to the movement there won't be any play at this table during a specific round), you add 0 for the fields NSPair, EWPair, LowBoard and HighBoard.

Table PlayerNumbers

This table contains information about the numbers of the players which are to be registered in the Bridgemate.

The Section field contains the ID of the section to which the table belongs. This number must be equal to the number in the ID field of the Section table.

The Table field contains the number of the table and refers to the Tables table.

For every table you add four records to the table. The Section and table field contain the same value for a table, the Direction field makes the distinguish between the four directions. Populate this field with the values N, S, E and W in each record.

Leave the member field blank or set it to an empty string. When player register their number in the Bridgemate, this field will be used to store the member number.

Table HandRecord (only recorder version)

This table contains the hand records for the boards played in each section. This hand record is uploaded to the recorder server when the Hand Record Validation function is used.

Recorder Bridgemates will receive this hand record information when they start a new board and will validate the recorded played cards against the hand record.

For each board per section you add a record to the database. In case different sections play the same duplicated board, it is still required to add multiple records for the same board.

Example: there are three sections each playing 24 boards. The boards are duplicated across all three sections. You add $3 \times 24 = 72$ records.

The Section field contains the ID of the section to which the table belongs. This number must be equal to the number in the ID field of the Section table. The Board field contains the number of the board.

The 52 cards of a board are divided among the 16 fields ranging from NorthClubs to WestSpades. Each set of four fields in the same direction must have in total 13 cards, corresponding to the 13 cards of the player. Each set of four fields in the same card color must also have in total 13 cards, corresponding to the 13 cards in a color.

Each card is represented by one character only, namely: 2 to 9, T (ten), J (jack), Q (queen), K (king), A (ace). The ten card must be represented by the character T, not the digits "10".

There is no prescribed order in which the cards should be registered for a direction and color. However, it is advised to start with the highest card and end with the lowest card. Example: AJT52. When there are no cards in a direction and color, you leave that field empty.

Modifying information in the database

After generating the database, and maybe even after uploading information to the Bridgemate base station, there is always the possibility of a change in the number of pairs, tables, sections, or in the movement being used. In such a case, your scoring software should be able to modify the database information.

To modify the data, one can choose to alter the records which require a change. This can be quite a time-consuming code programming task. We suggest to go for the easy way of deleting all records of the Section, Tables and RoundData table and add these records again with the modified data. Information about the table Status and LogOnOff values will be deleted, but they are easily restored by using the Synchronize function.

Using command line parameters at startup BM Pro

The following command line parameters can be used at startup of BM Pro:

`/f:[path + database file name]`

Opens the database specified between [].

`/s`

(only together with `/f`)

Resets server and uploads tables to server.

`/r`

(only together with `/f`)

Start retrieving data from server and don't show the report after uploading tables and/or hand records.

`/m`

Minimize BM Pro after startup.

`/h:[1,2,3...]`

(only together with `/s`)

Upload the hand records to the server for the sections specified between [] and separated by commas. Sections must be specified by using their section ID number.

How to deal with missing pairs

If you have a missing pair in your session, you can instruct Bridgemate to skip the round. Depending on the pair numbering in your session, do as follows:

NS and EW have different pair numbers

à Set the MissingPair field in Section table to the number of the missing pair. Bridgemate will detect this number in the RoundData table and automatically skip that movement.

NS and EW have equal pair numbers

à Leave the MissingPair field in Section table to 0, and instead, set the values in the fields PairNS and PairEW in the RoundData table to 0 for the specific round you want to skip.

Updating movement information

The scoring program should be able to make changes to the movement after it has been uploaded to the server. Some reasons for doing this are:

- Wrong movement applied to session and uploaded to server
- Adding rounds in a Swiss tournament
- Late arriving pairs or pairs not showing up results in the use of different movement

The general procedure for updating the movement information is as follows:

1. Update records of or add records to the RoundData table containing the correct movement information.
2. Set the field UpdateFromRound in table Tables to the value from which round you want to perform the update.
3. BPC will automatically detect any value unequal to zero in the UpdateFromRound field and will start updating the movement information for that table starting at that round. After having successfully updated the server, the value in this field will be reset to zero.

This is now explained for the following three situations:

Updating existing movement data

If the movement that has been uploaded to the server, is incorrect, it can be updated as follows:

1. Edit the records in the RoundData table for the tables that have a wrong movement and make sure they contain the correct information.
2. Change the field UpdateFromRound in table Tables to the number of the round you want to start updating from. If you want to update the complete movement and erase the results of round 1, you enter the value 1. If you want to leave the results of round 1 unchanged, enter 2. The movement will now be updated from round 2 upwards.

Adding movement data to existing tables

If you want to add rounds to movement data of an existing table (for example when you play Swiss), do as follows:

1. Add the records in the RoundData table for the rounds that you want to add. Make sure there is not gap between the rounds. You can add multiple rounds at once for a single table.
2. Change the field UpdateFromRound in table Tables to the number of the round you have just added. All subsequent rounds will be added to the server.

Adding new tables

When due to late arriving pairs the number of tables is enlarged, you update the movement information as follows:

1. Add the records in the RoundData table containing the movement for the newly added tables.
2. Add the records in the Tables table for the new tables.
3. Change the field UpdateFromRound to 1. This table will now be added to the server.

Note:

- Always make changes to the RoundData table first, then to the table Tables.

- You can combine these update situations in one single update procedure. BPC will automatically detect what kind of movement update should be made.
- All results are erased from the start update round. All results from rounds that are not erased, remain in the server.

Maximum values

The Bridgemate has limits for the numbers of pairs, boards, rounds and tables which you should take into account when implementing Bridgemate in your software.

Sections

Sections range from A to Z, so the maximum number of sections you can have is 26.

Tables

The maximum number of tables you can add to one single server is 128. Table numbers can range from 1 to 511.

Pairs

Pair numbers can range from 1 to 999.

Boards

The highest board number you can use is 63. The maximum boards per round is 32.

Rounds

Rounds can range from 1 to 63. You must always add rounds starting from round 1, e.g. when you add round 2, round 1 must be added as well.

The following additional limitations apply to the data uploaded to the server:

Limitations on number of tables and sections

The number of sections and number of tables per section that are uploaded to a single server are limited as follows:

Number of sections	Maximum number of tables per section
Between 1 and 2	128
Between 3 and 4	64
Between 5 and 8	32
Between 9 and 16	16
Between 17 and 26	8

The maximum number of tables per section holds for all sections that are uploaded to the server. The maximum number of tables per section is defined by the number of tables of the largest section.

Note that your .bws file is allowed to contain numbers that exceed these limitations. In that case you should use multiple servers to run the session completely.

Limitations on number of rounds and total number of boards (per table)

$$(number\ of\ rounds \times 3) + (total\ number\ of\ boards \times 4) \leq 479$$

“Number of rounds” is the total number of rounds that will be played at the table. Please note the rounds limitation above saying that when uploading round number 2, you also have to add round number one. Empty rounds are counted in this sum.

“Total number of boards” is the sum of all boards that will be played at the table. Empty rounds do not add to this sum since there are no boards to be played in such a round.

In case you want a session to start from round 11, you can add empty rounds for round 1 to 10. These rounds are omitted and the Bridgemate will start directly from round 11.

This limitation is per table. You can have different combinations of number of rounds and total number of boards for the tables uploaded to the server.

Retrieving results

Results which are entered in the Bridgemate are stored in the table ReceivedData. When the option "Intermediate results" is activated, the contract and lead card will also be stored in the table IntermediateData directly after they are entered in the Bridgemate. This is particularly of interest to developers of vugraph and rama programs that require immediate presentation of contracts.

The format of the ReceivedData table is simple and straightforward. Contracts are stored in the 'contract' field and results in the 'result' field. Results are noted as number of tricks won or lost compared to the contract. When the contract was just made, an equal (=) sign is noted. When there is a pass-out, the contract is PASS and the result field remains empty.

In the following special cases, the contract and result field stay empty and the remarks field contains information:

Board not played

Remarks field contains "Not played".

Notification of arbitral score

The TD has entered a notification of an arbitral decision, but the decision itself was not entered. The remarks field contains "Arbitral score".

Arbitral score entered as percentage

When a percentage was awarded to the pairs, the remarks field can contain one of the following:

40%-40%

50%-40%

60%-40%

40%-50%

50%-50%

60%-50%

40%-60%

50%-60%

60%-60%

Note that the percentage sign is included in the text.

Use of the field 'Remarks'

Although the field Remarks is used by BM Pro for abovementioned special cases, you can use it for your own specific purpose as well. There are no further limitations on the use of this field by BM Pro.

Guidelines for data retrieval

Reading results

The table "ReceivedData" is the most important table for data retrieval. Here is a description to easily retrieve data from the table:

1. Make a connection to the database using DAO351. Be sure the connection does not lock the database for other users.
2. Make a query on the table "ReceivedData" which includes all fields. Use the "WHERE 'Processed'=false" clause to retrieve only records which have not been processed yet by the external program. If multiple external programs are retrieving data from the database at the same time, the "Processed1" till "Processed4" fields can be used. Every external program uses its own dedicated "Processed" field. Which field to use is decided by the external program.
3. Retrieve the data from all records and process it in the external program.
4. Set the 'Processed' field to true.
5. Close the query recordset.

Repeat this periodically, say for example every five seconds. By setting the Processed field to true, the query will not include the records the next time.

Creating a designated "Processed" field

In case your external program wants to use a special process field, an extra 'Processed' field may be necessary. Your external program can add without problems an extra true/false column to the table definition. Please note the following requirements:

1. fields must start with "custom_" (without the quotes, all in lowercase). For example: "custom_VueGraph"
2. custom fields MUST be placed at the end of the existing columns. It is not allowed to assign a field index which interferes with existing columns.
3. the fixed field "Processed" must remain intact.

Reading table PlayerNumbers

If the players enter their member number in the Bridgemate, the table PlayerNumbers contains these numbers. As the players register themselves at the start of the first round, we advise to read this table completely at once just after the start of the first round. There is no "Processed" field to keep track of processed records. However, if required, you may add your own custom_ field to this table to keep track of processed records.

Checklist

For full support of Bridgemate, make sure you have the following functionality in your program:

- Creation of .bws database
- Adding session data to database by adding the appropriate records
- Automatic start-up of BM-Pro with the session file from your scoring program
- Retrieval of results
- Retrieval of player numbers

And also:

- Updating movement in database when movement in session changes
- Restart BM Pro with current database file without restarting Bridgemate system (convenient when user accidentally closes BM Pro during a session)

Using the recorder version

The recorder version of the Bridgemate is a special type of Bridgemate. Although the hardware is identical to the normal version of the Bridgemate, the firmware in both server and Bridgemate terminal is significantly different from the normal version. The main differences are:

- Normal Bridgemate is used by the players, recorder Bridgemate is used by a fifth person at the table.
- Normal Bridgemate allows for entry of contract, result and lead card only. Recorder Bridgemate registers complete bidding and play.
- Member number function, adjusted score, results overview functions are not available in the recorder version.

Database

Bridgemate Pro Control uses the same .bws file format for both recorder Bridgemates and normal Bridgemates. The following fields and tables are used only by the recorder and must be present in the database when using this version:

Tables:

BiddingData

PlayData

HandRecord (optional, only when you validate recording of the cards)

Fields:

CurrentBoard (in table Tables)

CurrentRound (in table Tables)

HandRecordValidation (in table Settings, optional)

The tables BiddingData and PlayData are filled by BPC during the session. You only read data from this table, you don't edit it.

The table HandRecord should be filled by the scoring program with the appropriate hand records. You only do so when you use the HandRecordValidation function. When this function is switched off, the recorded cards are not validated against the hand record, which means that every card entry is allowed (also entering the same card multiple times).

When the hand record validation function is used, all data that is stored in the table PlayData is validated and complies with the hand record. The mark X for small card will not be used, all small cards are converted to their respective small cards. X however is used and stored in the database when no validation is used and the operator pressed X on the Bridgemate keyboard.

Hand records must be stored in the HandRecord table for each section separately. Also when you use duplicated boards for multiple sections, you still need to store the hand records for all those sections.

Starting session

You start the session in the same way as for the normal Bridgemate. In case you want to use the hand record validation, Bridgemate Pro Control must be instructed to upload the hand record before uploading the tables. You do so by using the parameter /h:[], where the section numbers are noted between the [] separated by commas. (for example /h:[1,2,3].

When you instruct BPC to upload hand records after you have uploaded the table entries, you may experience an error saying that upload is not allowed anymore. Due to the internal memory allocation within the server, hand records must always be uploaded before any of the Bridgemates has initiated connection with the server. Therefore, you are safe when you

upload hand records before uploading tables. BPC will take care of this automatically when you use the parameters /s and /h.

Processing data

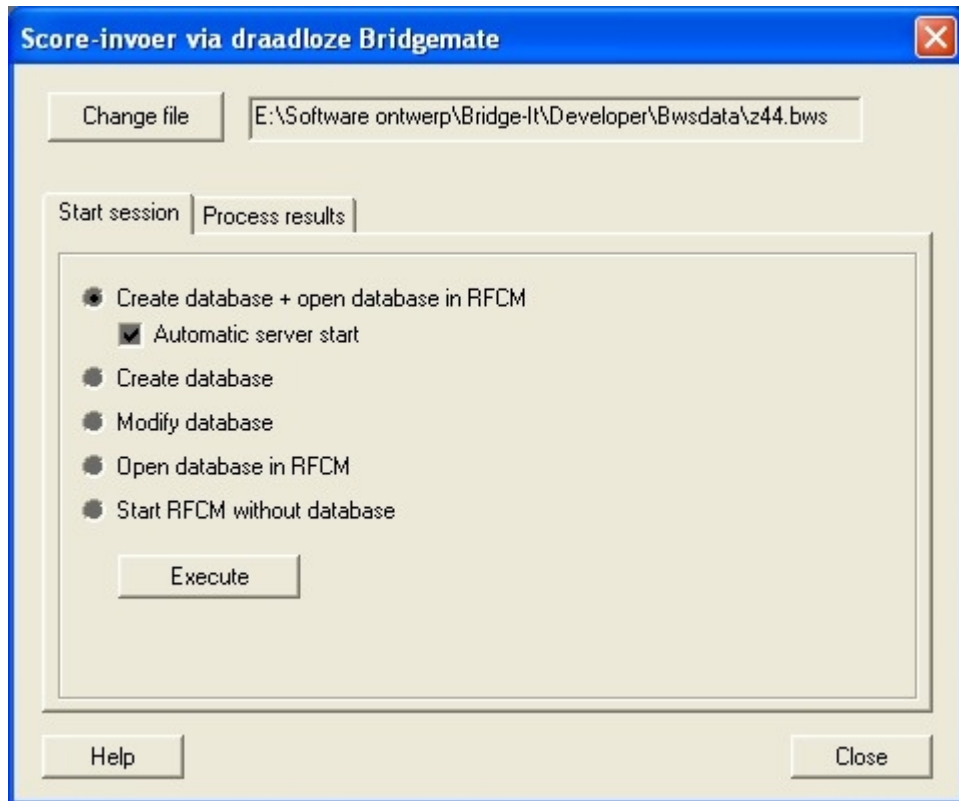
The recorded data is stored in the tables BiddingData and PlayedData. Every bid and card is represented by its own record. There is not "processed" field that you can use. You are not recommended to add such a field yourself. The table is meant to be read-only for other programs than Bridgemate Pro Control.

Every record has its own autonumber ID. You use this number to keep track of the records you have processed. At every subsequent query you select those records which have an ID higher than the ID of the records you previously queried.

Example of integration in scoring software

The way how to integrate the database create/read functionality differs for each scoring program. In this chapter we provide an example of how to add this functionality. This is only to give you an idea of a possible solution.

Add a form (window) to your program which looks like this:

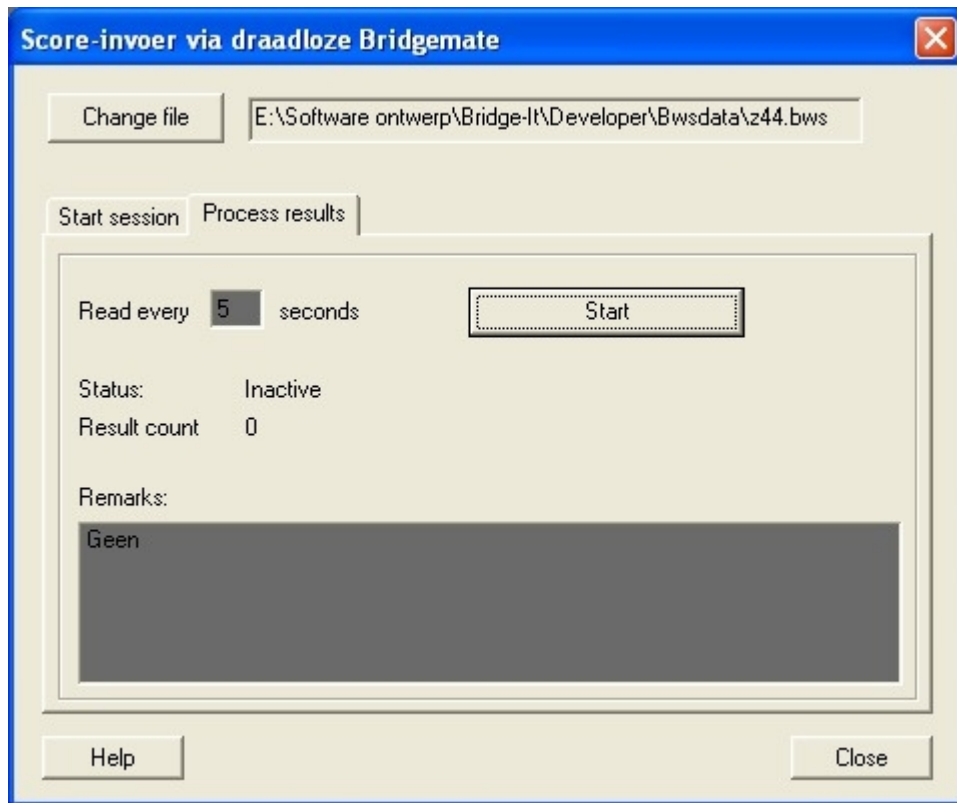


(RFCM is the same as Bridgemate Pro Control)

The .bws database file can be a standard file or automatically assigned by the scoring software. A good solution is to have the same name as the scoring session file, but with extension .bws. The user can change to another database file by clicking the “Change file” button”.

The screen has two tabs. The tab “Start session” contains several options to create the database file, modify it, and open the database file in BM Pro. All options are related to the database file mentioned at the top of the screen. The meaning of the options should speak for themselves. “Automatic server start” means that at startup of BM Pro, the tables are immediately uploaded to the server.

The second tab shows this:



To start the results retrieval process, the "Start" button is pressed. Pressing again will stop the process. Data retrieval happens at a set interval, which is currently 5 seconds. Any remarks like board not played, wrong direction entered etc is mentioned in the text box below.