Bridgemate developer's guide

Document revision 37
November 18th, 2013

Contents

Revision history

Revision history of this document since document version 15:

Document version: 37
Release date: November 18th, 2013
Bridgemate Control Software version: 2.8.28
Changes:

- Value for validation of member numbers according to ABF rules
- Value for validation of member numbers according to JCBL rules

Document version: 36
Release date: April 12th, 2013
Bridgemate Control Software version: 2.8.1
Changes:

- ExternalUpdate field added to ReceivedData table.
- Explanation how to update scores from the scoring program

Document version: 35
Release date: May 11th, 2012
Bridgemate Control Software version: 2.7.6
Changes:

- BM2FirstBoardManually added to Settings table.

Document version: 34
Release date: May 5th, 2012
Bridgemate Control Software version: 2.7.1
Changes:

- How to deal with Rover movements

Document version: 33
Release date: April 5th, 2012
Bridgemate Control Software version: 2.7.1
Changes:

- BM2EnterHandRecordWhen added to Settings table.
- Values -256, 256, +256 to UpdateFromRound added for enabling and disabling “waiting for new movement” mode.
- Explanation how to use “waiting for new movement” mode in Swiss events.

Document version: 32
Release date: November 26th, 2011
Bridgemate Control Software version: 2.7.1
Changes:

- strID added to PlayerNames table
- BM2TextBasedNumber added to Settings table

Document version: 31
Release date: June 6th, 2011
Bridgemate Control Software version: 2.6.1
Changes:

- BM2ViewHandrecord added to the Settings table.
- BM2EnterHandrecord added to the Settings table.
- Value 3 for setting BM2NameSource added.

Document version: 30
Release date: February 17th, 2011
Bridgemate Control Software version: 2.5.1
Changes:

- BM2AutoShowScoreRecap setting field added for automatic showing score recap.
- BM2NumberValidation updated with FFB rules.

Document version: 29
Release date: January 17th, 2011
Bridgemate Control Software version: 2.5.1
Changes:

- BM2GameSummary, BM2SummaryPoints setting fields added for summary at end of session.
- Values of BM2Ranking setting changed.
- Length of field “Letter” in table “Section” changed from 2 to 3.
- Strict ordinal relation between section ID and section letter removed.
- Support for double and triple section letters added.

Document version: 28
Release date: November 7th, 2010
Bridgemate Control Software version: 2.3.19
Changes:

- Processed and TimeLog fields added to table PlayerNumbers.

Document version: 27
Release date: October 25th, 2010
Bridgemate Control Software version: 2.3.1
Changes:

- Specifications for missing pair changed.
- Maximum values for Bridgemate II updated.

Document version: 26
Release date: October 10th, 2010
Bridgemate Control Software version: 2.3.1
Changes:

- Startup parameter added for running concurrent Bridgemate Pro system.
- Startup parameter added for specifying which tab to show at startup.
- EWMoveBeforePlay setting moved to Section table (was: BM2EWMoveBeforePlay in Settings table)
- CustomBoards specification amended.

Document version: 25
Release date: September 30th, 2010
Bridgemate Control Software version: 2.2.7
Changes:

- BM2NameSource chapter updated

Document version: 24
Release date: September 5th, 2010
Bridgemate Control Software version: 2.2.5
Changes:

- Bridgemates can be reset by the scoring program.
- Bridgemate settings update can be initiated by the scoring program.
- BM2NameSource setting added.

Document version: 23
Release date: July 5th, 2010
Bridgemate Control Software version: 2.1.24
Changes:

- Bridgemate II setting for validation of member number added.

Document version: 22
Release date: June 5th, 2010
Bridgemate Control Software version: 2.1.10
Changes:

- “Group” field added to table “Tables”.
- “PlayerNames” table added to database.
- Recording settings add to database.
- Information about using names added.

Document version: 21
Release date: December 9th, 2009
Bridgemate Control Software version: 2.0.17
Changes:

- Bridgemate II setting for enabling RESET function key added.

Document version: 20
Release date: October 4th, 2009
Bridgemate Control Software version: 2.0.2
Changes:

- Bridgemate II setting for pair number entry added

Document version: 19
Release date: August 10th, 2009
Bridgemate Control Software version: 1.7.90+ (not released at this date)
Changes:

- Bridgemate II functionality added:
  - Bridgemate II settings (table Settings)
  - Player names (table PlayerNumbers)

Document version: 18
Release date: May 5th, 2009
Bridgemate Control Software version: 1.7.67+ (not released at this date)
Changes:

- Scoring program can set Bridgemate settings separately for each section.

Document version: 17
Release date: November 21st, 2008
Bridgemate Control Software version: 1.7.50
Changes:

- Scoring program can instruct BCS to remove tables from the server.
- Scoring program can instruct BCS to remove rounds only without adding new round data.

Document version: 16
Release date: May 2nd, 2008
Bridgemate Control Software version: 1.7.15
Changes:

- AutoShutDownBPC field added to the settings table.

Document version: 15
Release date: November 4th 2007
Bridgemate Control Software version: 1.7.1
Changes:

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

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 Control Software 1.6.28 and earlier versions only support Access 97 file format. Bridgemate Control Software 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

- BCS or Bridgemate Control Software: 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 Control Software 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.

PlayerNames
Contains names of the players. Names can be preloaded by the scoring program or after the member numbers are entered on the Bridgemates.

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

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

HandRecord
Contains hand record information that is used to validate when recording the play (recorder version and Bridgemate II only) or to show or enter the hand record (Bridgemate II 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 Pro or the recording features of the Bridgemate II. 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 four fields.

ID
Contains the ID belonging to the section. Mostly it will start from 1, but this is not nescessarily the case. ID must be a value between 1 and 78 and unique for each section.

Letter
Contains the letter assigned to the section. Each section has always a letter indicating the uniqueness of the section. A unique section letter and a unique section ID must be supplied because they are referred to in different tables. Section letters can be single, double or triple. The following letters can be used, in total 78 different combinations:
A, B, C, …. to Z
AA, BB, CC, …. to ZZ
AAA, BBB, CCC, …. to ZZZ
Note that in double or triple section letters, the letters must be identical. For example BB and CCC, is allowed, AB, or AAB is not allowed.

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. See section "How to deal with missing pairs" for more information.

EWMoveBeforePlay
Setting to indicate if and how many tables the EW pairs move up or down from their home table before playing the first round.
0 = EW pair does not move
-1 …. = EW pair move tables down equivalent to the absolute value in this field
1 … = EW pair move tables up equivalent to the value 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 Control Software.

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 Control Software.

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.

Group
Can be set with a value between 0 and 63 and allows for dividing tables in different groups. Each group is separated from the other groups and calculated individually by the Bridgemate server.

"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 x 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. Numbers must be in ascending order. 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
Containts 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/updated in the database.

TimeLog
Contains the time when the score was registered/updated in the database.

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

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

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 Bridgemate. 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.

Name
Contains the name of the player.

Updated
Boolean field used by the scoring program to signal BCS that the name has been updated/added.

TimeLog
Contains the time when the player number was registered/updated in the database.

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

Bridgemate II fields:
Name
Contains the name of the player. This information must come from the scoring program.

Updated
When the scoring program sets the name or updates the name in field “Name”, it must set this field to true in order to let BCS process the (updated) name.

“PlayerNames” table

(Bridgemate II only)
This table can be used for preloading the player names by the scoring program. It contains the following fields:

ID
Contains the number belonging to a player name in number format (long).

Name
Contains the name of the player.

strID
Contains the number belonging to a player name in text format. Number may contain leading zeros.

“Settings” table

The Bridgemate has several configuration settings to adjust it each individual needs. By default these settings are configured in BCS from the menu Tools **à** tab Options for Bridgemate Pro and Bridgemate II settings, and tab Bridgemate II for specific settings of Bridgemate II only. 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.

For each option you add a field to the table. There is no need to add all fields, just add the fields for the settings you want to control by the scoring program.

Settings can be set for each section separately by adding a record for each section. In case there is only one record, the settings will apply to all sections.

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

Settings for both Bridgemate Pro and II:
Section
Contains the ID of the section to which the settings in this record belong (1 to 78).
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), or as American style (=2).
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 BCS after all tables have logged off (true/false)

Bridgemate II settings:

BM2PINcode
Pin code to access TD-menu (string, always containing 4 digits)
BM2ConfirmNP
Confirm entry of No Play by TD pin code (true/false)
BM2RemainingBoards
Show remaining number of boards to go after each entry (true/false)
BM2NextSeatings
Show seatings for next round at end of round (true/false)
BM2ScoreRecap
Allow players to retrieve the round scores they have entered (true/false)
BM2AutoShowScoreRecap
Show score recap automatically at end of each round and end of session (true/false)
BM2ScoreCorrection
Allow players to make corrections to their round rounds (true/false)
BM2AutoBoardNumber
Automatic input of board number in the entry screen (true/false)
BM2FirstBoardManually
Disable the automatic input of board number when no boards are entered yet in current round (true/false; true = override automatic input, false = automatic input). This setting has no use when BM2AutoBoardNumber is set to false.
BM2ResultsOverview
Type of results overview:
0 = frequency list, 6 lines, 1 score column
1 = frequency list, 6 lines, 2 score columns
2 = frequency list, 4 lines, 1 score column
3 = traveler, 6 lines, 1 score column
4 = traveler, 6 lines, 2 score columns
5 = traveler, 4 lines, 1 score column
BM2ShowPlayerNames
Show the names of the players:
0 = don't show player names
1 = show player names at each round
2 = show player names only at first round
BM2Ranking
Show current ranking for the two pairs
0 = don't show ranking
1 = show ranking after each round
2 = show ranking at end of session
BM2GameSummary
Enable/disable the summary after end of session. (Enabled requires BM2Ranking=1 or 2)
BM2SummaryPoints
Specifies what type of points should be shown in the summary:
0 = matchpoints obtained for each board
1 = percentage achieved for each board
BM2PairNumberEntry
Setting for entering pair number as part of the declarer
0 = don't enter pair number
1 = entry of pair number is optional
2 = entry of pair number is compulsory
BM2ResetFunctionKey
Enable/disable the RESET function key.
BM2RecordBidding
Enable the recording of the bidding.
BM2RecordPlay
Enable the recording of the cards.
BM2ValidateRecording
Activate validation of the recording against the hand records.
BM2ShowHands
Allow the hand record to be shown during the recording process.
BM2NumberValidation
Use validation for member numbers.
0 = no validation
1 = validation according to ACBL number rules

2 = validation according to FFB number rules
3 = validation according to ABF rules
4 = validation according to JCBL rules
BM2NameSource
Setting for where to source for names:
0 = table “PlayerNames” in .bws file
1 = “BMPlayerDB.mdb” database lookup file
2 = no name source, names are preset/updated by the scoring program
3 = first look in .bws file, then look in BMPlayerDB.mdb database
BM2TextBasedNumber
Specifies whether name lookups should be performed based on number comparison (false) or text comparison (true).
BM2ViewHandRecord
Allow the players to view the hand record of the board they just played.
BM2EnterHandRecord
Enable manual entry of the hand record of the boards played in a round. Entered hand records are stored in the HandRecord table.
BM2EnterHandRecordWhen
Specifies when the hand records are entered:
0 = at end of round
1 = at end of board
(requires BM2EnterHandRecord = true)

Note:

- When adding a field, the value of the field will always overwrite the setting in BCS. 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. Bridgemate II system allows more specific grouping instructions through the Group field in Tables table. If you use that field to specify the groupings, the value of GroupSections field in Settings is ignored.
- AutoShutDownBPC, GroupSections and MaximumResults are global settings and not per section. Only the values in the first record will be used, and any other values in subsequent records will be ignored.
- Note that BM2ShowHands is used during the recording process, and BM2ViewHandRecord is used by the players during normal score entry.

“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 12 tables which were described earlier in this document. Their names are:

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

(The tables BiddingData, PlayData and HandRecord are for the recorder version of the Bridgemate or Bridgemate II with recording function only. PlayerNames is only used by Bridgemate II)

These tables have the following fields definitions:

Note:

- No indexes are needed on any field. BCS will add them automatically when needed.
- 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	3	no	
Tables	integer			
MissingPair	integer			0
EWMoveBeforePlay	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
Group	integer			0

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
ExternalUpdate	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	
Name	text	18	yes	
Updated	true/false			false
TimeLog	date/time			
Processed	true/false			false

Table PlayerNames

Field name	Type	Field size	AllowZeroLength	DefaultValue
ID	long integer			
Name	text	18	yes	
strID	text	18	yes	

Table Settings

Field name	Type	Field size	AllowZeroLength	DefaultValue
Section	integer			
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
BM2PINcode	text	4		“0000”
BM2ConfirmNP	true/false			false
BM2RemainingBoards	true/false			true
BM2NextSeatings	true/false			true
BM2ScoreRecap	true/false			false
BM2AutoShowScoreRecap	true/false			false
BM2ScoreCorrection	true/false			false
BM2AutoBoardNumber	true/false			true
BM2FirstBoardManually	true/false			false
BM2ResultsOverview	integer			0
BM2ShowPlayerNames	integer			0
BM2Ranking	integer			0
BM2GameSummary	true/false			false
BM2SummaryPoints	integer			0
BM2PairNumberEntry	integer			0
BM2ResetFunctionKey	true/false			true
BM2RecordBidding	true/false			false
BM2RecordPlay	true/false			false
BM2ValidateRecording	true/false			false
BM2ShowHands	true/false			false
BM2NumberValidation	integer			0
BM2NameSource	integer			0
BM2ViewHandRecord	true/false			false
BM2EnterHandRecord	true/false			false
BM2EnterHandRecordWhen	integer			0
BM2TextBasedNumber	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 unique 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 single, double or triple character, which can be A through Z, AA through ZZ, AAA through ZZZ (3 x 26 combinations). The character code and the ID are compulsory, you cannot omit these.
The ID must be a value between 1 and 78. There is no strict relation between the ID and character code, you can assign any of the possible 78 IDs to a character code. Usually you will assign ID = 1 to section code = A, but it is also allowed to assign for example ID = 55 to section code = A.
The Tables field contains the number of tables you have in each section.
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). See section “How to deal with missing pairs” for more information.
EWMoveBeforePlay indicates how many tables EW moves up or down from their home table before they play the first round. This field is used when EW first takes place at their home table (for example: 1 EW sits at table 1), enters their member numbers at that table, then moves on to a higher/lower table and starts playing round 1. This can happen when board duplication is done by the players at the table, or at team events where both team pairs take place at their home table to enter their line-up, and then EW moves to another table. Default value is 0. Only write a value to this field when member numbers are used and EW enters their member number at another table than where they play in the first round. Otherwise, this field can be left at 0.

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”.
The Group field can be used to assign the table to a specific group. Groups are used by the Bridgemate server to distinguish tables from each other. Scores which are displayed on the Bridgemate, and overview of previous board results will only include scores that belong to the same group. This feature allows multiple groups of tables that belong together, for example section A and B play an event, and section C and D play another event. Also this can be used in team matches to create groups of two tables which make up a match. The Group value ranges from 1 to 63. If it is left to 0, it is ignored by BCS and the overall “GroupSections” setting is used.

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 BCS 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. Numbers must always be in ascending order. If failed to do so, BCS will return an error when uploading the rounds. 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. Note that when CustomBoards is used, LowBoard and HighBoard must still contain a value higher than zero. Best practice would be to include the lowest board number of CustomBoards in LowBoard, and the highest number of CustomBoards in HighBoard.
In case no play is happening at this table in a round (because the movement leaves this table empty for reasons), you add 0 for the fields NSPair, EWPair, LowBoard and HighBoard.
In case no play is happening at this table in a round because of a sit-out (one of the two pairs is missing), write 0 in the field NSPair or EWPair (whichever pair is missing), but do write the number of the pair that is attending. Bridgemate will still skip this round, but it does know that the pair of which the number has been provided, is playing in the session. This information can be useful when member number entry is required; the pair that is playing but has a sit-out will be asked at the first round to enter their member numbers. (Bridgemate II only)

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 en empty string. When player register their number in the Bridgemate, this field will be used to store the member number.
The Processed field can be used by the scoring program as a flag to indicate that this field has been processed. Whenever a new number is added or updated, its Processed value is set to false.
The TimeLog field contains the system time of the moment when the number was added or updated.

Functionality for Bridgemate II only:
Bridgemate II has the capability of showing the names of the players. The names are stored in the field Name and can be maximum 18 characters long. When a name is stored or updated in the record, the field Updated must be set to true.
There are two possible procedures for storing names:

1. Sessions with pre-registration; all the names are already known before the start of the session. At such events, the names can be stored in the database by the scoring program at the moment of creation of the database. You must set the Updated field to true in order to have the names processed by BCS.
2. Sessions without pre-registration; players enter their number at the start of the session. The database is created without player names. Players enter their number and these numbers are stored by BCS in the table PlayerNumbers. The scoring program processes these numbers and returns the names to the database and sets the field Updated to true.

Table PlayerNames (Bridgemate II only)

The table PlayerNames can be used to store the player database in the .bws file for easy and quick retrieval by BCS without the need for the scoring program to update the names continuously. If the table PlayerNames exists, BCS will search this table for a matching player number and retrieve the name from this table and store it in the table PlayerNumbers. It provides a quick method to update the PlayerNumbers table with the names if no pre-registration is used, and subsequently automatic upload of the names to the Bridgemate (Bridgemate II only).

Each name must be accompanied by a player number. This number can be stored in number (long) format in the field ID, or as a text number in field strID. Text numbers may contain leading zeros. Text number “1” and text number “01” are considered as different player numbers.

Table HandRecord (only recorder version and Bridgemate II)

This table contains the hand records for the boards played in each section. Hand records are used in the Bridgemate Pro recorder version, and the Bridgemate II.
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 x 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.

When players use the manual entry of hand records in Bridgemate II, the entered hand records are stored in this table. If the table does not exist, BCS will create the table.

Hand records of pre-dealt boards can be stored by the scoring program in this table. When executing BCS, use the parameter /h to load all the hand records in the server (see section Command Line Parameters at Startup). Once the hand records are loaded into the server, they can be viewed by the TD using the TD-menu option 7, or by players if option BM2ViewHandRecord is enabled.

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 BCS

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

/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 BCS 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.

/pi
Only valid when using Bridgemate II system together with a Bridgemate Pro system.
This parameter will instruct BCS to run a second instance of the program operating a Bridgemate Pro server. The default instance will start Bridgemate II USB server. The second instance will start the Bridgemate Pro system which is concurrently connected to the same computer.

/t1 … /t5
Specifies which tab should be visible when starting BCS:
/t1 **à** Results
/t2 **à** Round monitor
/t3 **à** Board monitor
/t4 **à** Result matrix
/t5 **à** Player names

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 round.

NS and EW have equal pair numbers

à Leave the MissingPair field in Section table to 0, and instead, set the value in the field NSPair or EWPair in the RoundData table to 0 for the pair that is *missing*. Bridgemate will recognize that this pair is missing and will skip those rounds. However, in case member number entry is used, the pair that has a sit-out in the first round will be asked to enter their member numbers. (Bridgemate II only.) Then it will skip the first round and move on the next round to be played.

Note:
In previous versions of BCS it was custom to set both NSpair = 0 and EWPair = 0 to indicate that the round was not played. From BCS 2.3.1 and higher it is recommended only to set either NSPair or EWPair to zero, whichever pair is missing.

How to deal with Rover movements

A rover movement is a movement with a sit-out in a slightly different format. In a normal sit-out, the number of tables in the movement is equal to the number of pairs rounded up to the next even number divided by two. For example, a session with 25 pairs uses a movement for 13 tables.

In a rover movement, instead, the number of tables is determined by rounding the total number of pairs downwards to the first even number. In the example mentioned above, a rover movement contains only 12 tables. Every round, the pair sitting out is excluded from the movement. This sit-out pair is called the “rover”. The difference with a normal sit-out movement is that a rover pair can be added without adding extra boards.

To facilitate a rover in the Bridgemate scoring system, an extra table must be added to the database. In the example above, instead of 12 tables, 13 tables are needed. Tables 1 to 12 are taken by the non-rover pairs in each round. The rover pair is placed at table 13, usually at the NS position.

This has the following consequences for the data written by the scoring program to the database (numbers refer to the example mentioned above):

- Table Section. Field Tables is 1 higher. (13 instead of 12)
- Table Tables. Add one extra record for this table number. It is advised to number this table 1 higher than the highest table in the movement.
- Table Rounddata. Round records for this extra table are added. Write the rover pair number in NS, and leave EW to 0. LowBoard and HighBoard are left to 0 as well.
- Table PlayerNumbers. If you pre-load the records for the member numbers, also include 4 records for the extra rover table.

At the start of the session, the first rover pair can enter their member numbers in the Bridgemate at the rover table. These numbers will be used to show the names of the players correctly throughout the game (Bridgemate II only). After that, the Bridgemate will show end of session since there are no boards to be played at that table.

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. BCS 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.

Removing movement data from existing tables without adding new movement data

If you want to remove rounds from an existing table (for example in a Swiss match you want to erase the last added round) and you don't want to upload new movement data, do as follows:

1. Remove the records of the rounds for this table from the table RoundData.
2. Change the field UpdateFromRound in table Tables to the number of the lowest round you have removed, but set the number negative instead of positive.

Example:
There are five rounds in the server and you have just added the sixth round. Now, a mistake in round six line-up has been discovered, and you want to remove round six first. To do so, you first remove the record of round six from the RoundData table. Next, you set the UpdateFromRound field to -6, which instructs BCS to remove all rounds starting from round 6. No new data will be uploaded, only the old movement data will be removed.

Note:
This method only deals with removing old movement data without uploading new (updated) movement data. In case you want to replace old movement data with new data, there is no need to remove

movement data using a negative number, but instead you follow the procedure described after “Update existing movement data”.

Adding new tables

When due to late arriving pairs the number of tables is increased, 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.

Removing unused tables

When the number of tables is decreased due to pairs not showing up, the scoring program can instruct BCS to remove those tables from the server. Update the movement information as follows:

1. First remove the records for this table from the table RoundData.
2. Change the field UpdateFromRound to value 999.

The value 999 is recognized by BCS as an instruction to remove the table. The table will be removed from the server, and the table record will be removed as well.

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. BCS 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.

Special operations

Besides updating movements and removing tables, the UpdateFromRound field can also be used for the following operations.

Reset Bridgemate

(Bridgemate II only)

To reset a Bridgemate from the scoring program, write value 998 to UpdateFromRound field. At the next communication with the server, the Bridgemate will receive instruction to reset itself. Resetting means that the Bridgemate will return to the main screen, but no data will be erased from the server.

Update settings Bridgemate

To update Bridgemates with new settings, update first the settings in the Settings table, then write value 997 to UpdateFromRound field. At the next communication with the server, the Bridgemate will copy the new settings (Bridgemate II only). Bridgemate Pro will require a manual reset using the TD-key in order to apply the updated settings.

Adding rounds in Swiss events

The following feature is only available in the Bridgemate II system.

In a Swiss event, the pairs or teams are pre-assigned in round 1 and assignments for subsequent rounds are based on their results in the previous rounds. The consequence is that round 2 can only be added to the server after all results of round 1 are collected and the ranking has been calculated. Based on the ranking, the assignments for the next round are added to the server.

The Bridgemates always follow the movement information of the server. When the last available round has been fully scored by a Bridgemate, the Bridgemate will show end of session. However, in a Swiss event, this may be not the case as more rounds may be added to the server and it is desired that while the event is in progress, the Bridgemate will show end of round and will wait for new movement information instead of showing end of session and returning to the main screen.

Using the UpdateFromRound value, the scoring program is able to control the behaviour of the Bridgemate when the last available round has been completed. Based on this value, the Bridgemate will either consider the last available round as the last round of the session and show the end of session screen, or it will remain in end of round screen and wait for new movement information.
Each time the scoring program adds a new round to the database file, it includes also instructions for the Bridgemate how to behave after the last available round has been completed. This is explained below.

The normal procedure for updating a round is:

1. Write the new movement information to Rounddata table.
2. Set UpdateFromRound to the round number just added.

For example, if round 3 has to be added to the server, movement data of round 3 is stored in Rounddata, and UpdateFromRound is set to 3.

When a new round is added to the database file, and Bridgemates should wait for new rounds after this round, the above procedure becomes:

1. Write the new movement information to Rounddata table.
2. Set UpdateFromRound to (round number just added + 256).

Example:
In case of adding round 3, the UpdateFromRound value is not 3, but 259 (= 3 + 256). This will add the movement of round 3 to the server and simultaneously instruct the Bridgemate to wait for new movement information after the last available round (which currently is 3) is completed.

Adding round 1 to the server + instructions to wait for round 2 movement information

Round 1 is added to the server together with the instructions to start BCS using the command line parameters /s /r. As UpdateFromRound is usually 0 when BCS is started, this will not instruct the Bridgemate to wait for movement information after round 1 is completed. To provide these instructions to the Bridgemate, the scoring program should set UpdateFromRound to 257 (256 + 1). The Bridgemates will begin the game as usual, but will be instructed to wait for new movement information after they completed the last available round (in this case, round 1).

Adding last round to the server + instructions to end the game after this round

When the last round of the session is added to the server and Bridgemates should finish the session once the last available round has been completed, the scoring program applies the standard procedure for updating the database file:

1. Write the last round to the Rounddata table
2. Set UpdateFromRound to the number of the last round. Do not add 256 to this number.

BCS will instruct the Bridgemates they should no longer wait for new movement information after the last round is completed, but instead finish the session.

Remarks

- The value +256 is an overall true/false instruction for the Bridgemate to wait for new movement information after the last available round has been completed, and is not related to the value of the updated round number. For example, if rounds 3, 4 and 5 are added in a single update procedure and UpdateFromRound is set to 259 (256 +3), Bridgemates will move to round 3, after that to round 4 and round 5, and only after round 5 has been completed, it will wait for new movement information.
- The true/false instruction to wait for new movement information can be altered without adding new movement to the database file. This is useful when a previous instruction has to be corrected without modifying the movement data.
    - Enable waiting for new movement information: set UpdateFromRound to 256.
    - Disable waiting for new movement information: set UpdateFromRound to -256.

Showing player names on the Bridgemate

This feature is only available in the Bridgemate II system.

Bridgemate II provides functionality to show the player names at the start of a new round. To be able to do so, the player names must be stored in the .bws file and forwarded to the Bridgemate II server. There are four methods to handle names and show them on the Bridgemates. The field "BM2NameSource" in the Settings table determines which name source is used.

1. Pre-loading names in the table PlayerNames (BM2NameSource value = 0)
The scoring program loads all the names of its player database in the table PlayerNames at the start of the session, and players enter their number when they start to play. When players enter their number in the Bridgemate, BCS will search for the corresponding name, and store this name in the correct record in PlayerNumbers and upload the name to the server. This is an easy method as the scoring program's only task is to create a PlayerNames table with all possible names. The disadvantage of this method is that it may take a long time and create large-size .bws files when the player database contains many players. This method is recommended only for player databases with 10,000 players or less.

2. Pre-loading names in the database BMPlayerDB.mdb (BM2NameSource value = 1)
When using large national databases, a separate database lookupfile can be used in order to keep the size of the .bws file limited. This file is BMplayerdb.mdb, and exists in the same folder where bmpro.exe is installed. Instead of loading the names in each separate .bws file, the scoring program can load only once (or update periodically) the names in the lookup file. BCS will search through this lookup file to retrieve the names.

3. Pre-registering names for each table (BM2NameSource value = 2)
This method is useful when the players pre-register for the event and the table position of the players is known at the start of the event. The scoring program stores the names in the PlayerNumbers table in the correct records matching their actual seating in the first round. The table PlayerNames is not used.
For each name, the scoring program stores the following information: section number, table number where the player takes position in the first round, direction, name, number (optional), and the value true in field Updated. When BCS is started with this information, it will upload the names to the Bridgemate II server and will make them available to the Bridgemates.
When using this method, the setting "Enter player numbers" should be disabled (the names have been made available before).

4. Player names update during session (BM2NameSource value = 2)
The scoring program continuously queries the table PlayerNumbers for new numbers (without names yet), and when a new number is found, it updates the Name field and sets Updated to true. The PlayerNames table is not used, nor are the names pre-registered before. This method is useful when the scoring program has a large player database and pre-loading all the names in PlayerNames would create too large .bws files.
When using this method, the entry of player numbers in the Bridgemate is enabled, and the scoring program continuously polls the PlayerNumbers table for new number. When it finds a number without a name, it retrieves the number, searches in its databases for the matching name, and stores the name in the same record in the PlayerNumbers table. Finally it sets the Updated field to true, which instructs BCS to upload the name to the server. Use this method only when you have large player databases that you don't want to load in the .bws file.

EW pairs move up or down from their home table before playing first round

It is not uncommon in team matches that the team's NS and EW pairs enter their member numbers in the same Bridgemate at their home table, and afterwards EW pairs move up or down a certain number of tables before round 1 starts. Write to the field "EWMoveBeforePlay" in the Section table the number of tables that EW moves up or down in that section. Use a negative number to indicate moving down. For example, 3 means moving up 3 tables, -2 means moving down 2 tables. 0 (=default) means EW pairs don't move.

Maximum values

The Bridgemate has limits for the numbers of pairs, boards, rounds an tables which you should take into account when implementing Bridgemate in your software. Bridgemate Pro and Bridgemate II have different maximum values.

Sections
Bridgemate Pro: Sections range from A to Z, the maximum number of sections you can have is 26.
Bridgemate II: Sections range from A to Z, AA to ZZ, AAA to ZZZ, the maximum number of sections you can have is 78.

Tables
Bridgemate Pro: The maximum number of tables you can add to one single server is 128. Table numbers can range from 1 to 511.
Bridgemate II: The maximum number of tables you can add to one single server is 256. Table numbers can range from 1 to 4095.

Pairs
Bridgemate Pro: Pair numbers can range from 1 to 999.
Bridgemate II: Pair numbers can range from 1 to 4095.

Boards
Bridgemate Pro: The highest board number you can use is 63. The maximum number of boards per round is 32.
Bridgemate II: The highest board number you can use is 127. The maximum number of boards per round is 54.

Rounds
Bridgemate Pro: Rounds can range from 1 to 63.
Bridgemate II: Rounds can range from 1 to 255.
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:

Bridgemate Pro: 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.

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

(number of rounds X 3) + (total number of boards X 4) <= 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 no 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.

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

For Bridgemate II, this equation is as follows:

(number of rounds) + (total number of boards in all rounds) <= 256

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 BCS 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 BCS.

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 as 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 a Processed field which you can use to keep track of which records you have processed and which not. Whenever BCS updates or adds a record, it sets the Processed field to False. Set it to True in your scoring program to indicate that you have processed this record. The TimeLog field records the time of record update/addition.

Updating scores through the scoring program

Scores entered on the Bridgemate can be corrected in various ways:

1. On the Bridgemate itself. The score must be deleted first (through TD-menu or the score recap screen) and then the correct score can be entered.
2. In BCS by modifying the score from the Results tab or Matrix tab.
3. By the scoring program to modify the score record in the ReceivedData table.

Method 1 and 2 will automatically update the result in the server and the updated score will be shown on all Bridgemates. A score updated by the scoring program requires the following procedure in order to update the server:

1. Update the record in ReceivedData table which contains the score with the new score details. Make sure the data is consistent (e.g. 7 NT +1 is not consistent). BCS will not validate any correction made by the scoring program in the ReceivedData table.
2. Set the field ExternalUpdate to -1. This will signal to BCS that the record has been modified.

It is best practice to perform the above two steps in one single update query.

BCS will scan every 15 seconds for score updates and update the server with the modified scores. It is required that ExternalUpdate is set to -1 of modified records. After the update is performed, the Bridgemates will show the updated score.

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 BCS with the session file from your scoring program
- Retrieval of results
- Retrieval of player numbers
- Storing player names in the .bws file

And also:

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

Using the recorder version

The recorder version of the Bridgemate is available as a separate type of Bridgemate Pro firmware, but is also available as additional feature in the Bridgemate II. The main differences with normal scoring 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 Control Software 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 BCS 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 (Bridgemate Pro) or BM2ValidateRecording (Bridgemate II) 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 Control Software 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 BCS 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. BCS will take care of this automatically when you use the parameters /s and /h.

Processing data

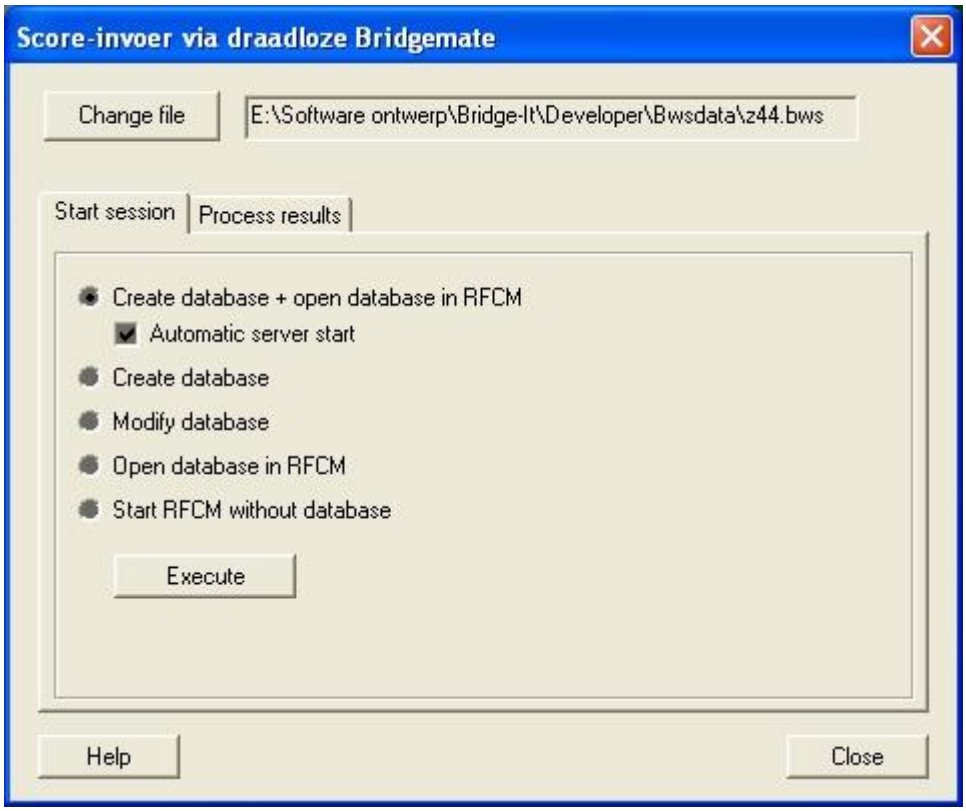
The recorded data is stored in the tables BiddingData and PlayedData. Every bid and card is represented by its own record. There is not "processed" field that you can use. You are not

recommended to add such a field yourself. The table is meant to be read-only for other programs than Bridgemate Control Software.
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:

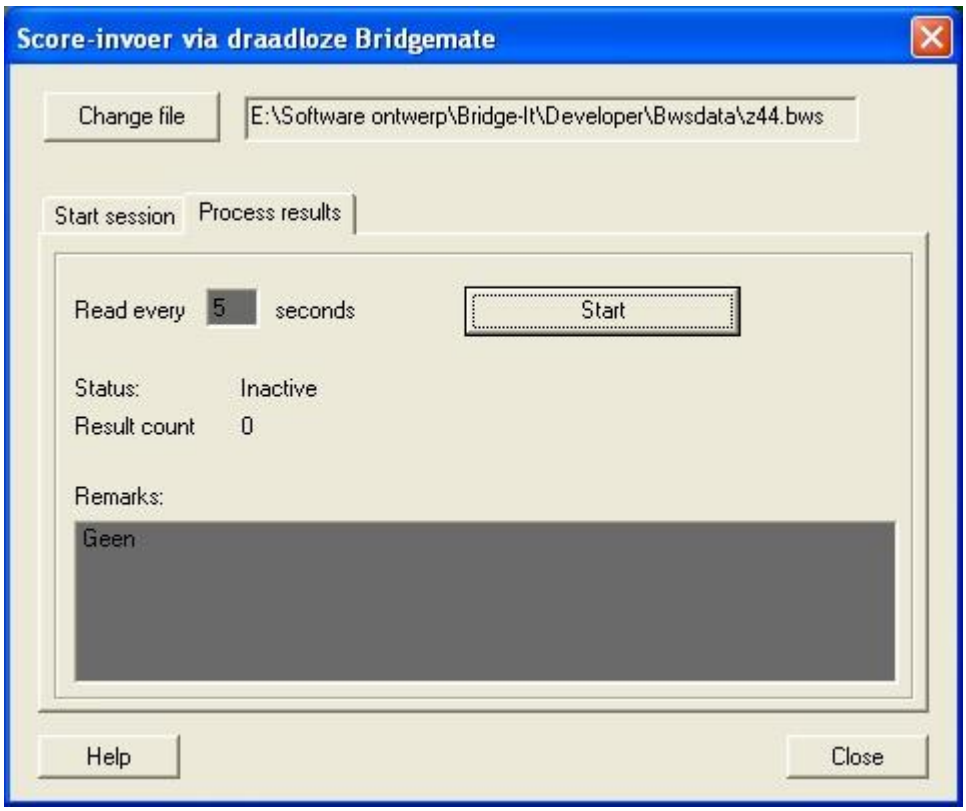


(RFCM is the same as Bridgemate Control Software)

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 BCS. 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 BCS, 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.