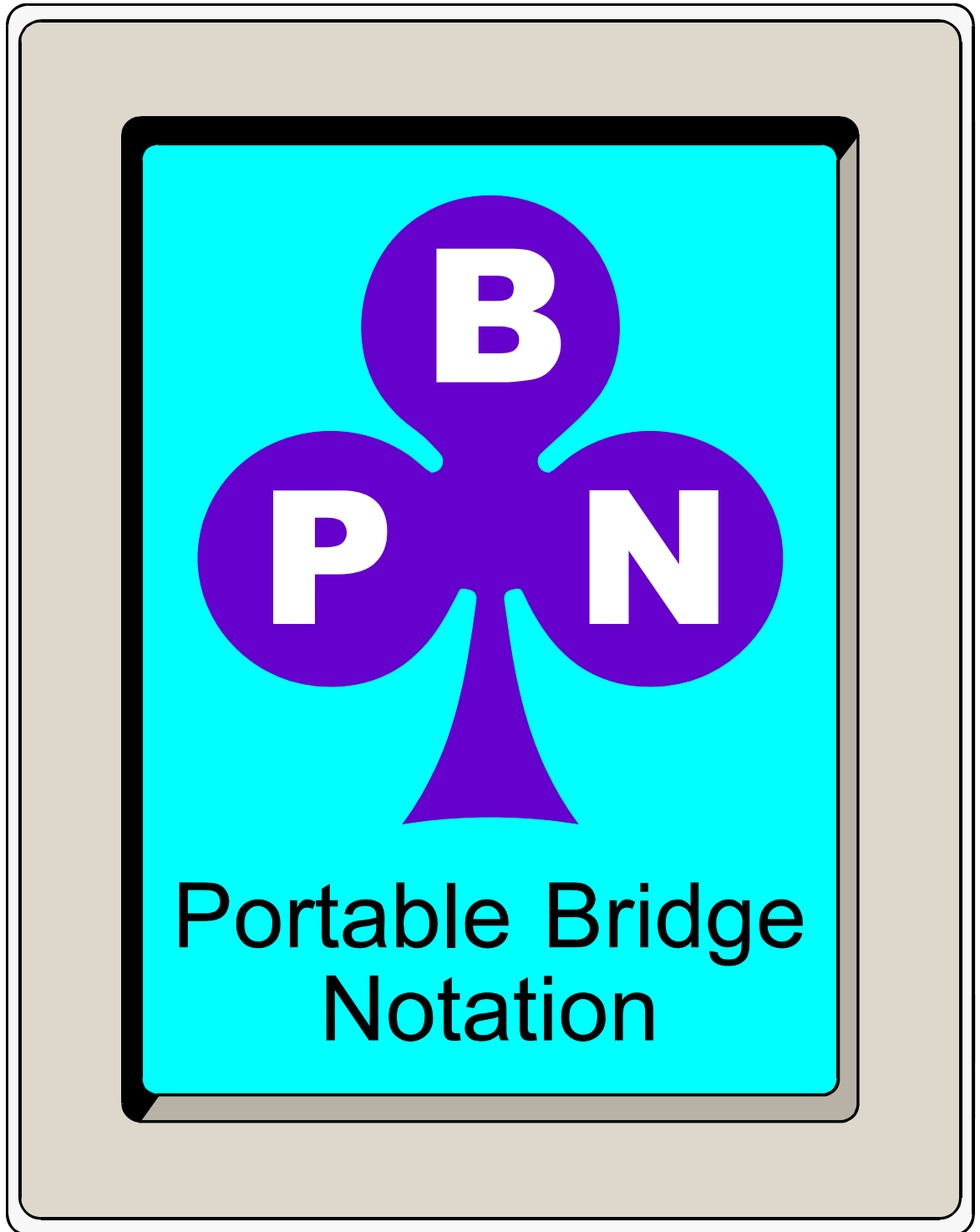


# Version 2.1



Document: Portable Bridge Notation (PBN)  
Version: 2.1.a\_1  
Date: 2007.03.17  
Coordinator: Tis Veugen ( [tis.veugen@gmail.com](mailto:tis.veugen@gmail.com) )  
Co-editor: Kaj G Backas ( [kjb@sackab.fi](mailto:kjb@sackab.fi) )  
Authors: Readers of the Internet newsgroup rec.games.bridge



# Table of Contents

1	Introduction.....	4
1.1	Design criteria.....	4
1.2	Example game.....	5
1.3	Relationship to PGN.....	5
2	File layout.....	5
2.1	Import format and export format.....	6
2.2	Character set.....	6
2.3	Line specification.....	6
2.4	Escape mechanism.....	6
3	Game layout.....	7
3.1	Game sections.....	7
3.2	Tokens.....	8
3.3	Tag pair.....	9
3.4	Identification section.....	9
3.4.1	The Event tag.....	10
3.4.2	The Site tag.....	10
3.4.3	The Date tag.....	10
3.4.4	The Board tag.....	10
3.4.5	The West tag.....	10
3.4.6	The North tag.....	10
3.4.7	The East tag.....	10
3.4.8	The South tag.....	10
3.4.9	The Dealer tag.....	10
3.4.10	The Vulnerable tag.....	11
3.4.11	The Deal tag.....	11
3.4.12	The Scoring tag.....	11
3.4.13	The Declarer tag.....	12
3.4.14	The Contract tag.....	12
3.4.15	The Result tag.....	13
3.5	Auction section.....	13
3.5.1	Auction call.....	13
3.5.2	Auction annotations.....	14
3.5.3	Auction suffix.....	14
3.5.4	Auction note reference.....	14
3.5.5	Auction note explanation.....	14
3.6	Play section.....	15
3.6.1	Play card.....	15
3.6.2	Play annotations.....	15
3.6.3	Play suffix.....	16
3.6.4	Play note reference.....	16
3.6.5	Play note explanation.....	16
3.7	Numeric Annotation Glyphs.....	16
3.8	Commentary.....	17
3.9	Irregularities.....	17
3.10	End positions.....	18
3.11	Case sensitivity.....	18
4	Supplemental tags.....	19
4.1	Game related information.....	19
4.1.1	Tag: Competition.....	19
4.1.2	Tag: DealId.....	19
4.1.3	Tag: Description.....	19
4.1.4	Tag: FrenchMP.....	19
4.1.5	Tag: Generator.....	19
4.1.6	Tag: Hidden.....	19
4.1.7	Tag: Room.....	19
4.1.8	Tag: Termination.....	20
4.2	Score related information.....	20
4.2.1	Tag: Score.....	20
4.2.2	Tag: ScoreIMP.....	20
4.2.3	Tag: ScoreMP.....	20
4.2.4	Tag: ScorePercentage.....	20
4.2.5	Tag: ScoreRubber.....	21



4.2.6	Tag: ScoreRubberHistory .....	21
4.2.7	Tag: OptimumScore.....	21
4.3	Player related information.....	22
4.3.1	Tags: BidSystemEW, BidSystemNS .....	22
4.3.2	Tags: PairEW, PairNS .....	22
4.3.3	Tags: WestNA, NorthNA, EastNA, SouthNA .....	22
4.3.4	Tags: WestType, NorthType, EastType, SouthType .....	22
4.4	Event related information.....	22
4.4.1	Tag: EventDate .....	22
4.4.2	Tag: EventSponsor.....	22
4.4.3	Tag: HomeTeam .....	22
4.4.4	Tag: Round.....	22
4.4.5	Tag: Section .....	22
4.4.6	Tag: Stage .....	22
4.4.7	Tag: Table.....	22
4.4.8	Tag: VisitTeam .....	22
4.5	Time and date related information .....	22
4.5.1	Tag: Time.....	23
4.5.2	Tag: UTCDate.....	23
4.5.3	Tag: UTCTime.....	23
4.6	Time control.....	23
4.6.1	Tag: TimeControl.....	23
4.6.2	Tag: TimeCall .....	23
4.6.3	Tag: TimeCard .....	23
4.7	Miscellaneous .....	24
4.7.1	Tag: Annotator .....	24
4.7.2	Tag: AnnotatorNA .....	24
4.7.3	Tag: Application .....	24
4.7.4	Tag: Mode.....	24
4.8	Tag value inheritance .....	24
4.9	Expandability .....	24
4.10	Compatibility .....	24
5	Supplemental sections.....	26
5.1	General description .....	26
5.2	Description of tables .....	26
5.3	ActionTable.....	27
5.4	AuctionTimeTable .....	27
5.5	InstantScoreTable .....	27
5.6	OptimumPlayTable .....	28
5.7	OptimumResultTable .....	28
5.8	PlayTimeTable.....	28
5.9	ScoreTable .....	28
5.10	TotalScoreTable .....	29
6	File issues.....	29
6.1	File name.....	29
6.2	File embedding.....	29
7	PBN software .....	29



# 1 Introduction

In the past years a lot of bridge computer programs have been developed. There are programs for dealing, bidding, playing, and/or teaching. A widely accepted standard format for bridge games does not exist. Therefore, the output of one program can't be used in another program. This not only holds for programs running on different computer platforms, but even for programs on one and the same computer.

This document specifies a universal notation for bridge games, called "Portable Bridge Notation" (PBN). PBN is based on "Portable Game Notation" (PGN), a standard for the representation of chess games.

The current PBN 2.0 standard has been released on 1999.09.01 and it will be frozen until 2000.09.01. At that time an evaluation will be made for a possible update.

The PBN 1.0 standard has been released on 1998.04.01 and it has been frozen until 1999.04.01. The PBN 2.0 standard is an update to PBN 1.0. This does not mean that the PBN 1.0 standard is invalid. PBN files obeying the PBN 1.0 standard are fully compliant to the PBN 2.0 standard.

With PBN 2.1 the possibility to add optimum score information was added.

Versio 21a\_1 has only changed Email addresses.

## 1.1 Design criteria

PBN has been designed with several criteria in mind. Portability is of course the main criterion. Other criteria are:

- (1) PBN is an open standard. It is publicly available for users and software developers. PBN is not subject to any copyrights.
- (2) The PBN notation must be straightforward and comprehensible. A human must be able to read and write the PBN notation easily.
- (3) PBN must be suitable for a variety of computer programs. It can be used by dealing programs, bidding programs, playing programs, etc. The PBN files range from single deals to large databases. Examples of database contents are:
  - (a) tournament database
  - (b) famous player database
  - (c) database with 7NT deals
- (4) PBN must be able to serve as an interface between competing computer programs.
- (5) The PBN standard must be expandable.
- (6) The PBN notation must be worldwide applicable.
- (7) The syntax of the PBN notation must be easy to parse by a computer program.

The PBN standard is based on textual files, instead of binary files. A lot of PBN data, including comment, are text strings. Therefore, low priority is given to minimization of file sizes.



## 1.2 Example game

An example game is described below. The tags are specified in detail in chapter 3.

```
[Event "International Amsterdam Airport Schiphol Bridgetournament"]
[Site "Amsterdam, The Netherlands NLD"]
[Date "1995.06.10"]
[Board "1"]
[West "Podgor"]
[North "Westra"]
[East "Kalish"]
[South "Leufkens"]
[Dealer "N"]
[Vulnerable "None"]
[Deal "N:.63.AKQ987.A9732 A8654.KQ5.T.QJT6 J973.J98742.3.K4 KQT2.AT.J6542.85"]
[Scoring "IMP"]
[Declarer "S"]
[Contract "5HX"]
[Result "9"]
{
    S
    H 6 3
    D A K Q 9 8 7
    C A 9 7 3 2

    S K Q 10 2
    H A 10
    D J 6 5 4 2
    C 8 5

    S A 8 6 5 4
    H K Q 5
    D 10
    C Q J 10 6

    S J 9 7 3
    H J 9 8 7 4 2
    D 3
    C K 4
}
[Auction "N"]
1D 1S 3H =1= 4S
4NT =2= X Pass Pass
5C X 5H X
Pass Pass Pass
[Note "1:non-forcing 6-9 points, 6-card"]
[Note "2:two colors: clubs and diamonds"]
[Play "W"]
SK =1= H3 S4 S3
C5 C2 C6 CK
S2 H6 S5 S7
C8 CA CT C4
D2 DA DT D3
D4 DK H5 H7
- - - H2
* [Note "1:highest of series"]
```

## 1.3 Relationship to PGN

PBN is derived from "Portable Game Notation" (PGN). Although the name PGN suggests that it is a standard for arbitrary games, it is dedicated for representing chess games. PGN has been very successful. There are already many chess programs that understand PGN, and PGN databases with thousands of games are publicly available.

The specification of PGN can be downloaded from <http://www.iae.nl/users/veugent/pgn.zip>.

One standard, namely the "Portable Draughts Notation" (PDN), has already been derived from PGN. Board games like chess and draughts have much similarity, so the derivation of PDN from PGN was straightforward.

The derivation of PBN from PGN is less straightforward for the following reasons:

- the number of players is 4 instead of 2
- the "initial position", i.e. the cards, must be given
- the auction must be described, and possibly some calls must be explained
- the "moves", i.e. the calls and the played cards, must be described as two separate sections. The moving player must be indicated.

## 2 File layout

A PBN file contains one or more text lines. A text line consists of zero or more visible characters. An empty text line has zero visible characters. A visible character is defined as a character that a user can edit. A text line also



contains non-visible characters, needed to represent a text line. The representation of a text line differs among platforms. The non-visible characters can be linefeeds, carriage returns, zero bytes, padding bytes or even bytes telling the length of a line.

## 2.1 Import format and export format

The PBN standard discerns two formats: the "import" format and the "export" format.

The import format is rather flexible and is used to describe data that may have been prepared by hand. The order of tags is unimportant; the use of (superfluous) tabs and space characters does not matter, etc. A PBN file in import format can have text lines according to the platform's native line representation. A PBN parser should be able to handle a PBN file with import format, that has been created on a computer, the parser is running on.

The export format is rather strict and is used to describe data that is usually prepared under program control. A file with export format can be read by a PBN parser on any computer! So, a PBN parser with the capability of writing a PBN file in export format guarantees the portability of PBN files.

## 2.2 Character set

PBN data is represented using a subset of the eight bit ISO 8859/1 (Latin 1) character set. ("ISO" is an acronym for the International Standards Organization.) This set is also known as ECMA-94 and is similar to other ISO Latin character sets. ISO 8859/1 includes the standard seven bit ASCII character set for the 32 control character code values from zero to 31. The 95 printing character code values from 32 to 126 are also equivalent to seven bit ASCII usage. (Code value 127, the ASCII DEL control character, is a graphic character in ISO 8859/1; it is not used for PBN data representation.)

The 32 ISO 8859/1 code values from 128 to 159 are non-printing control characters. They are not used for PBN data representation. The 32 code values from 160 to 191 are mostly non-alphabetic printing characters and their use for PBN data is discouraged as their graphic representation varies considerably among other ISO Latin sets. Finally, the 64 code values from 192 to 255 are mostly alphabetic printing characters with various diacritical marks; their use is encouraged for those languages that require such characters. The graphic representations of this last set of 64 characters is fairly constant for the ISO Latin family.

Printing character codes outside of the seven bit ASCII range may only appear in string data and in commentary. They are not permitted for use in symbol construction.

The horizontal tab and vertical tab are permitted. No other ASCII control characters are permitted, except for the (non-visible) control characters for the line representation.

Because some PBN users' environments may not support presentation of non-ASCII characters, PBN game authors should refrain from using such characters in critical commentary or string values in game data that may be referenced in such environments. PBN software authors should have their programs handle such environments by displaying a question mark ("?") for non-ASCII character codes. This is an important point because there are many computing systems that can display eight bit character data, but the display graphics may differ among machines and operating systems from different manufacturers.

## 2.3 Line specification

A text line contains any characters from the allowed character set. However, tab characters may not appear inside of string data.

The export format obeys some additional rules. In export format a text line contains printing characters concluded by an end-of-line indicator. This end-of-line indicator equals the CRLF character-pair consisting of the ASCII control character CR (Carriage Return, decimal value 13, hexadecimal value 0x0d) and the ASCII control character LF (LineFeed, decimal value 10, hexadecimal value 0x0a).

Tab characters, both horizontal and vertical, are not permitted in the export format. This is because the treatment of tab characters is highly dependent upon the particular software in use on the host computing system.

The length of a PBN text line is maximally 255 characters, including the non-visible characters, especially the end-of-line indicator. Lines with 80 or more printing characters are strongly discouraged because of the difficulties experienced by common text editors with long lines.

## 2.4 Escape mechanism

The escape character "%" in the first column means that the rest of the line can be ignored. A percent sign appearing in any other place other than the first position in a line does not trigger the escape mechanism.

The escape convention can be applied to embed non-PBN commands and data in PBN streams. In PBN, there are two proposed uses:



- (1) It indicates the version of the PBN standard used by a computer program. The syntax is:  
% PBN <major version #>.<minor version #>
- (2) It indicates that a PBN file has export format. The syntax is:  
% EXPORT

### 3 Game layout

A PBN file contains zero or more PBN games. Each game, except for the first game, starts with a semi-empty line. A semi-empty line is either an empty text line (no characters), or a line having only spaces and/or tab characters. In export format, each game (except for the first) must start with an empty text line.

So, a parser (and a human) knows that after a semi-empty line a new game begins. There is one exception to this situation, namely a comment (chapter 3.8) may include semi-empty lines that do not end the game.

Every game contains all its required data. Several games in a database file may contain identical information. This costs more bytes, but the syntax can be kept simple to cope with a large diversity of databases. A PBN file does not have a header with general file information.

#### 3.1 Game sections

A PBN game consists of several sections:

- (1) the identification section,
- (2) the auction section,
- (3) the play section, and
- (4) supplemental sections.

The identification section provides information that identifies the game by defining the values associated with a set of standard parameters.

The auction section gives the optionally annotated bids. It starts with tag name 'Auction' and ends at the next tag or the end of the game. The auction section is optional. The outcome of the auction can be given by the tags 'Declarer' and 'Contract'. If the auction section is present, then the tag 'Dealer' must be filled before the auction section.

The play section shows the played cards. It starts with tag name 'Play' and ends at the next tag or the end of the game. The play section is optional. If the play section is present, then the tags 'Deal', 'Declarer' and 'Contract' must be filled before the play section.

The supplemental sections are optional sections containing tables or user defined sections.

In import format, no tags are mandatory. The tags can be given in any order. As indicated above, in case of an auction section or a play section the needed tags and their order are prescribed.

In export format, the sections must be given in the above order. The supplemental sections are sorted alphabetically by their tag name. The identification section contains 15 mandatory tags.



## 3.2 Tokens

PBN character data is organized as tokens. A token represents a basic semantic unit. A token consists of a contiguous sequence of printing characters belonging to the allowed character set. Tokens may be separated from adjacent tokens by white space characters. (White space characters include space, newline, and tab characters.) A white space character can never be part of a token. As a consequence, a token does not cross a text line boundary, and its length is limited to a maximum of 255 characters.

Some tokens are self delimiting and do not require white space characters to separate them from the next token.

The left and right bracket characters ("[" and "]") are tokens. They are used to delimit tag pairs (see below). Both tokens are self terminating.

A name token consists of one or more name characters. These name characters are letter characters ("A-Za-z"), digit characters ("0-9"), and the underscore ("\_"). All characters in a name are significant. A name token is terminated just prior to the first non-name character.

A string token is a sequence of zero or more printing characters delimited by a pair of quote characters (ASCII decimal value 34, hexadecimal value 0x22). An empty string is represented by two adjacent quotes. (Note: an apostrophe is not a quote.) A quote inside a string is represented by the backslash immediately followed by a quote. A backslash inside a string is represented by two adjacent backslashes. Strings are commonly used as tag pair values (see below). A string token is terminated by its closing quote. Note that, non-printing characters like newline and tab are not permitted inside of strings, because a string is a special case of a token.

A call token is a sequence of one or more letter characters ("A-Za-z"), digit characters ("0-9"), and the hyphen character ("-"). A call token is terminated just prior to the first non-call character. The call tokens are defined in chapter 3.5.1.

A card token is a sequence of one or more letter characters ("A-Za-z"), digit characters ("0-9"), and the hyphen character ("-"). A card token is terminated just prior to the first non-card character. The card tokens are defined in chapter 3.6.1.

A suffix token consists of one or two exclamation marks "!" and/or question marks "?". A suffix token is terminated after the second suffix character, or just prior to the first non-suffix character.

A note token is a sequence of one or more digit characters ("0-9") preceded as well as succeeded by the equal sign ("="). This token terminates just after the succeeding equal sign.

A Numeric Annotation Glyph ("NAG", see below) is a token; it is composed of a dollar sign character ("\$\$") immediately followed by one or more digit characters. It is terminated just prior to the first non-digit character following the digit sequence.

An asterisk character ("\*") is a token by itself. It is used as a termination marker of an incomplete auction section or an incomplete play section. The token is self terminating.

A plus character ("+") is a token by itself. It is used to denote the next call in an auction that must be continued, or to denote the next card of a play that must be continued. The token is self terminating.

An irregularity token starts with the caret character ("^") and is followed by one of the characters "I", "S", "R", or "L". This token terminates just after the second character.

A section token is a sequence of one or more ASCII characters from 33 to 126 with the exclusion of '[', ']', '{', '}', ';', '"', '%'. This limited set of characters is needed to prevent problems in parsing a supplemental section. The excluded characters are used for indicating tag pairs, comments, string tokens, and the escape mechanism. The backslash is allowed, but has no special meaning.

A column name token is a sequence of zero or more printing characters with the exception of ';', because it serves as separator between 2 column name tokens. The first character '+' or '-' has a special meaning: it indicates the order in the column. Also character '\ in the column name token has a special meaning: it must be followed by an integer that represents the minimum width of the data in the column. Optionally, this integer can be followed by 'L' or 'R' to indicate left or right alignment of column data.





### 3.3 Tag pair

The tag pair (or shortly 'tag') is the basic information unit. A tag pair is composed of four consecutive tokens: a left bracket token, a name token, a string token, and a right bracket token. The name token is the tag name and the string token is the tag value associated with the tag name. (There is a standard set of tag names and semantics described below.)

Tag names are case sensitive. All tag names begin with an upper case letter.

For PBN import format, there may be zero or more white space characters between any adjacent pair of tokens in a tag pair.

For PBN export format, there are no white space characters between the left bracket and the tag name, there are no white space characters between the tag value and the right bracket, and there is a single space character between the tag name and the tag value.

PBN import format may have multiple tag pairs on the same line and may even have a tag pair spanning more than a single line. Export format requires each tag pair to appear left justified on a line by itself.

Some tag values may be composed of a sequence of items. For example, a game may have more than one annotator. When this occurs, the single character ";" (semicolon) appears between adjacent items.

### 3.4 Identification section

The identification section contains all the tag pairs, that are not involved with the auction section and the play section.

There is a set of 15 tags defined for storage of PBN data, called Mandatory Tag Set (MTS). These tags are mandatory for a PBN file in export format. The MTS is the common ground that all programs should follow for public data interchange. When a tag value of a mandatory tag is unknown (because it was not given in the import format), then the tag is given with tag value "?". When a tag value of a mandatory tag is inappropriate (e.g. a dealing program only gives the tags 'Dealer', 'Vulnerable', and 'Deal'), then the tag is given as an empty string.

For import format, the order of tag pairs is not important. For export format, the order of tag pairs is:

- (1) the mandatory tags in a fixed order
- (2) the supplemental tags, sorted alphabetically by their tag name

The 15 tag names of the MTS are (in order):

- (1) Event (the name of the tournament or match)
- (2) Site (the location of the event)
- (3) Date (the starting date of the game)
- (4) Board (the board number)
- (5) West (the west player)
- (6) North (the north player)
- (7) East (the east player)
- (8) South (the south player)
- (9) Dealer (the dealer)
- (10) Vulnerable (the situation of vulnerability)
- (11) Deal (the dealt cards)
- (12) Scoring (the scoring method)
- (13) Declarer (the declarer of the contract)
- (14) Contract (the contract)
- (15) Result (the result of the game)

In export format, tag pairs may only occur once. In import format, a tag pair that already occurred, is ignored.



### 3.4.1 The Event tag

The Event tag value should be reasonably descriptive. Abbreviations are to be avoided unless absolutely necessary. A consistent event naming should be used to help facilitate database scanning.

### 3.4.2 The Site tag

The Site tag value should include city and region names along with a standard name for the country. The use of the ISO 3166 (or, EN 23166) three letter names (instead of the country name) is suggested for those countries where such codes are available. A comma may be used to separate a city from a region. No comma is needed to separate a city or region from the ISO 3166 country code.

The ISO 3166 standard is continuously updated. A copy can be found on the PBN homepage.

### 3.4.3 The Date tag

The Date tag value gives the starting date for the game. (Note: this is not necessarily the same as the starting date for the event.) The date is given with respect to the local time of the site given in the Event tag. The Date tag value field always uses a standard ten character format: "YYYY.MM.DD". The first four characters are digits that give the year, the next character is a period, the next two characters are digits that give the month, the next character is a period, and the final two characters are digits that give the day of the month. If any of the digit fields are not known, then question marks are used in place of the digits.

### 3.4.4 The Board tag

The Board tag value is a positive integer, indicating the board number of the deal.

### 3.4.5 The West tag

The West tag value is the name of the player or players at the West direction. The names are given as they would appear in a telephone directory. The family or last name appears first. If a first name or first initial is available, it is separated from the family name by a comma and a space. Finally, one or more middle initials may appear. (Wherever a comma appears, the very next character should be a space. Wherever an initial appears, the very next character should be a period.)

The intent is to allow meaningful ASCII sorting of the tag value that is independent of regional name formation customs. If more than one person is playing West, the names are listed in alphabetical order and are separated by the semicolon character between adjacent entries. A player who is also a computer program should have appropriate version information listed after the name of the program.

### 3.4.6 The North tag

The North tag value is the name of the player or players at the North direction. The names are given here as they are for the West tag value.

### 3.4.7 The East tag

The East tag value is the name of the player or players at the East direction. The names are given here as they are for the West tag value.

### 3.4.8 The South tag

The South tag value is the name of the player or players at the South direction. The names are given here as they are for the West tag value.

### 3.4.9 The Dealer tag

The Dealer tag value is the direction of the game's dealer. The tag value is "W" (West), "N" (North), "E" (East), or "S" (South).



### 3.4.10 The Vulnerable tag

The Vulnerable tag value defines the situation of vulnerability. The following tag values are possible:

"None" , "Love" or "-" no vulnerability  
"NS" North-South vulnerable  
"EW" East-West vulnerable  
"All" or "Both" both sides vulnerable

In export format the tag values "None" and "All" are applied.

### 3.4.11 The Deal tag

The Deal tag value gives the cards of each hand. The tag value is defined as "<first>:<1st\_hand> <2nd\_hand> <3rd\_hand> <4th\_hand>". The 4 hands are given in clockwise rotation. A space character exists between two consecutive hands. The direction of the <1st\_hand> is indicated by <first>, being W (West), N (North), E (East), or S (South). The cards of each hand are given in the order: spades, hearts, diamonds, clubs. A dot character "." exists between two consecutive suits of a hand. The cards of a suit are given by their ranks. The ranks are defined as (in descending order):

A , K , Q , J , T , 9 , 8 , 7 , 6 , 5 , 4 , 3 , 2 .

Note that the 'ten' is defined as the single character "T". If a hand contains a void in a certain suit, then no ranks are entered at the place of that suit.

Not all 4 hands need to be given. A hand whose cards are not given, is indicated by "-". For example, only the east/west hands are given:

```
[Deal "W:KQT2.AT.J6542.85 - A8654.KQ5.T.QJT6 -" ]
```

In import format, the ranks of a suit can be given in any order; the value of <first> is free. In export format, the ranks must be given in descending order; <first> is equal to the dealer.

### 3.4.12 The Scoring tag

This tag gives the used scoring method. It is an essential part of the game since the tactics of the players depend on the scoring method.

There are a lot of scoring systems with all kind of variations, refer to Bridge Encyclopedia. New scoring systems evolve for coping with all kind of irregularities, see e.g.:

<http://www.gallery.uunet.be/hermandw/bridge/hermtd.html>.

The wealth of scoring systems makes standardisation difficult. Therefore, the specification of the tag value is open ended: only example values are given. The tag value consists of fields separated by semicolons. A field indicates either a basic scoring system or a modifier. Examples of basic scoring systems are:

MP MatchPoint scoring  
MatchPoints identical to 'MP'  
IMP IMP scoring (since 1962)  
Cavendish Cavendish scoring  
Chicago Chicago scoring  
Rubber Rubber scoring  
BAM Board-A-Match  
Instant apply InstantScoreTable



Examples of modifiers are:

Butler the trick point score is IMPed against the average value of all scores

Butler-2 as 'Butler', but the 2 extreme scores are not used in computing the average value

Experts the trick point score is IMPed against a datum score determined by experts

Cross the trick point score is IMPed against every other trick point score, and summed

Cross1 value of 'Cross' , divided by number of scores

Cross2 value of 'Cross' , divided by number of comparisons

Mean the datum score is based on a (normal) average value

Median the datum score is based on the median value

MP1 MatchPoints are computed as: the sum of points, constructed by earning 2 points for each lower score, 1 point for each equal score, and 0 points for each higher score.

MP2 MatchPoints are computed as: the sum of points, constructed by earning 1 point for each lower score, 0.5 points for each equal score, and 0 points for each higher score.

OldMP NO bonus of 100 (Doubled) or 200 (Redoubled) for the fourth and each subsequent undertrick, when not vulnerable

Mitchell2 see <http://www.gallery.uunet.be/hermandw/bridge/hermtd.html>

Mitchell3 idem

Mitchell4 idem

Ascherman idem

Bastille idem

EMP European MatchPoints

IMP\_1948 IMP scoring used between 1948 and 1960

IMP\_1961 IMP scoring revised in 1961

### 3.4.13 The Declarer tag

The Declarer tag value is the direction of the declarer of the contract. The tag value is "W" (West), "N" (North), "E" (East), or "S" (South).

The Declarer tag can also cope with the irregularity that the declarer and the dummy are swapped. This may happen when e.g. South is declarer, but by accident East plays the first card and South puts his cards on the table. The tag value becomes a caret (^) followed by the direction of the irregular declarer: "^W", "^N", "^E", resp. "^S".

When all 4 players pass, then the tag value is an empty string.

### 3.4.14 The Contract tag

The Contract tag value can be "Pass" when all players pass, or a 'real' contract defined as: "<k><denomination><risk>"

with

<k> the number of odd tricks, <k> = 1 ... 7

<denomination> the denomination of the contract, being S (spades), H (Hearts), D (Diamonds), C (Clubs), or NT (NoTrump)

<risk> the risk of the contract, being void (undoubled), X (doubled), or XX (redoubled)



### 3.4.15 The Result tag

The Result tag value gives the result of the game in number of tricks.

The possible tag values are:

"<result>"	number of tricks won by declarer
"EW <result>"	number of tricks won by EW
"NS <result>"	number of tricks won by NS
"EW <result> NS <result>"	number of tricks won by EW resp. by NS
"NS <result> EW <result>"	number of tricks won by NS resp. by EW with

<result> = 0 ... 13.

The <result> must match the actual number of won tricks. However, the players could accidentally agree on a wrong number of tricks. A caret character ("^") preceding one of the above tag values indicates that the <result> differs from the actual number of won tricks.

When all 4 players pass, then the tag value is an empty string.

In export format the tag value contains the number of tricks won by declarer.

The Result tag normally gives the final result after the play has ended. This is the case when all 52 cards have been played, or when the Play section ends with '\*'. The Result tag can also be used to give a partial result. When the play has not ended, then the Result tag indicates the number of won tricks for the completed, played tricks in the play section.

Usage of '+' in the play section would make it explicitly clear that the Result tag is based on a partial result.

## 3.5 Auction section

The auction section gives the calls, together with the optional annotation. The start of the auction section is indicated by the tag pair with tag name 'Auction'. The Auction tag is followed by multiple auction lines containing the calls. The direction of the player making the first call is given as tag value in the Auction tag pair: "W" (West), "N" (North), "E" (East), or "S" (South).

In export format this player is always the dealer. Each auction line contains 4 calls. Only due to much annotation, a line may contain less than 4 calls. The call of the dealer always starts left justified at a new line.

In import format the player in the Auction tag value need not be the dealer. For example, the calls are given in a table of 4 columns with West in the first column. In that case, each player before the dealer has a hyphen ("-") in the first auction line.

The auction section ends with all passes, or with "\*". The "\*" character indicates that no further calls will or can be given. In this case, the result of the auction can be given in the tags 'Declarer' and 'Contract'.

It is also possible that the auction has not been completed yet, with the intention to continue another time. For example, a game might be saved in a PBN file so that another computer program might make the next call. The '+' character is used to indicate explicitly this intention. The '+' replaces the next call to be made. Hence, it is placed after the last call (and its annotations). At the continuation, the '+' will be replaced by the made call, and possibly a new '+' can be inserted. In this situation, the '\*' character is forbidden to end the auction section. The usage of '+' is not obligatory, only recommended.

### 3.5.1 Auction call

A call is represented by a call token. The possibilities of a call token are:

AP	all players pass
Pass	the player passes
X	the player doubles
XX	the player redoubles
<k><denomination>	the player bids a contract, where <k> and <denomination> are defined as in the Contract tag
-	it is not yet player's turn to make a call



### 3.5.2 Auction annotations

In import format, the following annotations can be added after each call, namely:

- (1) at most one suffix annotation
- (2) at most one note reference
- (3) zero or more NAGs

These annotations are described below. In import format, the annotations can be given in any order. In export format, the order is: the optional note reference, and then the NAGs in increasing order of the NAG values. The optional suffix has been translated into the corresponding NAG (chapter 3.7). For example, in an import format PBN file the construction '1S !! =1= \$25' becomes in export format '1S =1= \$3 \$25'.

### 3.5.3 Auction suffix

Import format PBN allows for the use of suffix annotations for calls. A suffix token exists for each suffix annotation. There are exactly six such annotations available:

- ! good call
- ? poor call
- !! very good call
- ?? very poor call
- !? speculative call
- ?! questionable call

These suffixes correspond with the NAG values \$1 - \$6. A one-to-one correspondence is needed, because the suffixes are translated to NAG values for the export format.

### 3.5.4 Auction note reference

A call can be followed by at most one note reference. A note is used for indicating alerts (e.g. transfer bids). PBN does not distinguish between partner-alerts and self-alerts.

The note reference is given by means of a note token, defined as an integer surrounded by equal signs: =<note\_index>= The value of <note\_index> lies between 1 and 32. Several calls may use the same note reference.

In export format, a single space character is entered after the note reference, if it is followed by another token in the text line.

### 3.5.5 Auction note explanation

An auction note reference with index <note\_index> is explained in a note tag. This note tag has tag name 'Note', and the tag value of the note tag equals: "<note\_index>:<note\_text>"

Note tags may occur more than once (unlike other tags). They must differ in the value of the <note\_index>.

The note tags, dedicated to the auction, are placed in the auction section at the end of the auction calls. They may not be placed in the identification section, nor the play section.



### 3.6 Play section

The play section gives the played cards, together with the optional annotation. The start of the play section is indicated by the tag pair with tag name 'Play'. The Play tag is followed by multiple play lines containing the played cards. The direction of the player playing the first card is given as tag value in the Play tag pair: "W" (West), "N" (North), "E" (East), or "S" (South).

In export format this player is always the opening leader, being declarer's left hand opponent. Each play line contains 4 cards. Only due to much annotation, a line may contain less than 4 cards. The card of the opening leader always starts left justified at a new line. A dash '-' is used for each card that is unknown or that has not been played (yet).

In import format the player in the Play tag value need not be the opening leader. For example, the cards are given in a table of 4 columns with West in the first column. In that case, each player before the opening leader has a hyphen ("-") in the first play line.

The play section ends after 13 tricks, or with "\*\*\*". The "\*\*\*" character indicates that no further cards will or can be given. In this case, the result of the play can be given in the tags 'Result'.

It is also possible that the play has not been completed yet, with the intention to continue another time. For example, a game might be saved in a PBN file so that another computer program might play the next card. The '+' character is used to indicate explicitly this intention. The '+' replaces the next card to be played. At the continuation, the '+' will be replaced by the played card, and possibly a new '+' can be inserted. Note that the '+' need not be the last character in the Play section, e.g. in the fragment:

```
[Play "W]
H2 H3 H4 HA
+ - - CQ
```

South has played the last card, and the game must be continued by West playing the next card.

The usage of '+' is not obligatory, only recommended. When '+' is not used, then instead a '-' must be used to represent the card to be played.

#### 3.6.1 Play card

A card is represented by a card token. The possibilities of a card token are:

- the played card does not matter

<suit><rank> the played card

with

<suit> = S , H , D , C

<rank> = A , K , Q , J , T , 9 , 8 , 7 , 6 , 5 , 4 , 3 , 2 .

#### 3.6.2 Play annotations

The play annotations are analogous to the auction annotations.

In import format, the following annotations can be added after each card, namely:

- (1) at most one suffix annotation
- (2) at most one note reference
- (3) zero or more NAGs

These annotations are described below. In import format, the annotations can be given in any order. In export format, the order is: the optional note reference, and then the NAGs in increasing order of the NAG values. The optional suffix has been translated into the corresponding NAG (chapter 3.7). For example, in an import format PBN file the construction 'SK !! =1= \$200' becomes in export format 'SK =1= \$9 \$200'.



### 3.6.3 Play suffix

The play suffixes are analogous to the auction suffixes.

Import format PBN allows for the use of suffix annotations for played cards. A suffix token exists for each suffix annotation. There are exactly six such annotations available:

!	good card
?	poor card
!!	very good card
??	very poor card
!?	speculative card
?!	questionable card

These suffixes correspond with the NAG values \$7 - \$12. A one-to-one correspondence is needed, because the suffixes are translated to NAG values for the export format.

### 3.6.4 Play note reference

The play note references are analogous to the auction note references.

A card can be followed by at most one note reference. A note is used for indicating lead conventions, signals, etc. The note reference is given by means of a note token, defined as an integer surrounded by equal signs: =<note\_index>= The value of <note\_index> lies between 1 and 32. Several cards may use the same note reference.

In export format, a single space character is entered after the note reference, if it is followed by another token in the text line.

### 3.6.5 Play note explanation

The play note explanations are analogous to the auction note explanations.

A play note reference with index <note\_index> is explained in a note tag. This note tag has tag name 'Note', and the tag value of the note tag equals: "<note\_index>:<note\_text>"

Note tags may occur more than once (unlike other tags). They must differ in the value of the <note\_index>.

The note tags, dedicated to the played cards, are placed in the play section at the end of the played cards. They may not be placed in the identification section, nor the auction section.

## 3.7 Numeric Annotation Glyphs

An NAG (Numeric Annotation Glyph) is an element that is used to indicate a simple annotation in a language independent manner. The NAG token consists of the dollar sign ("\$\$") followed by a non-negative decimal integer. The value of this integer must be between zero and 255.

In export format, a single space character is entered after the NAG if it is followed by another token in the text line. Note: the number assignments listed below should be considered preliminary in nature; they are likely to be changed as a result of reviewer feedback.

NAG	Interpretation
0	'no annotation'
1	good call (same as "!")
2	poor call (same as "?")
3	very good call (same as "!!")
4	very poor call (same as "??")
5	speculative call (same as "!?")
6	questionable call (same as "?!")
7	good card (same as "!")
8	poor card (same as "?")
9	very good card (same as "!!")
10	very poor card (same as "??")
11	speculative card (same as "!?")
12	questionable card (same as "?!")
13	call has been corrected manually
14	card has been corrected manually
15	\
..	used for commenting calls and played cards
255	/

The NAG values \$1 ... \$6 and \$13 may only be used for calls in the auction section. The values \$7 ... \$12 and \$14 may only be used for played cards in the play section.





The NAG interpretations are available in a file called "pbn\_eng.nag". This file will be publicly available. It is possible to construct similar files in other languages. In this way, the expandability of NAG values is guaranteed.

The format of the file is as follows. The file consists of lines, ending by the CRLF character. Each line has the syntax:

```
<NAG_value> <NAG_text>
```

The line begins with a left justified non-negative integer. A single space character separates the NAG value from the NAG text.

### 3.8 Commentary

Comment text may appear in PBN data. There are two kinds of comments. The first kind is the 'rest of line' comment; this comment type starts with a semicolon character ";" and continues to the end of the line. The second kind starts with a left brace character "{" and continues to the next right brace character "}". Comments cannot appear inside any token, nor inside a tag pair.

Brace comments do not nest; a left brace character appearing in a brace comment loses its special meaning and is ignored. A semicolon appearing inside of a brace comment loses its special meaning and is ignored. Braces appearing inside of a semicolon comments lose their special meaning and are ignored. Furthermore, inside a brace comment, "%" in the first column is considered as a normal character, not as an escape character.

Bridge comments often refer to a suit. It would be nice to present a suit by a graphical symbol. The notation for a suit is defined as a backslash followed by the first character of the suit name. Hence:

```
\S = Spades, \H = Hearts, \D = Diamonds, \C = Clubs.
```

The position of comments is important. A comment refers to the preceding event tag, auction call, played card, suffix, note reference, or nag. Also a comment can be positioned at the begin of the game (before any tag). The position of the comment must be preserved when a PBN file is converted from import format to export format. This holds especially when the order of the tags is changed to obey the prescribed tag order in export format. In export format comments succeeding an event tag are printed at the begin of a new line. When a comment does not start at a new line, then a single space precedes the comment begin character ";" or "{".

There are no rules for converting comments from import format to export format. A conversion program has the freedom to merge comments, etc. After a conversion, the import file won't be used anymore; the export file is used, and the reader observes the comments as they are.

### 3.9 Irregularities

Several irregularities may occur during the auction and the card playing. An irregularity is considered a problem if it violates the "normal" bridge rules. Five problems have been identified:

- (A) An insufficient bid, accepted by the opponents.
  - (B) A call out of rotation, accepted by the opponents.
  - (C) A revoke happened, without being noticed in time.
  - (D) A lead out of turn, accepted by the opponents.
  - (E) The declarer and the dummy are swapped, refer to chapter 3.4.13.
- NB. Irregular (re)double is impossible, and MUST be taken back. When an irregularity (an insufficient bid that has been rejected) is solved, then there is no problem for PBN. The solved irregularity and its consequences (e.g. partner must keep passing) can be described as comment. The following notation handles the irregularity (A) - (D):
- ^I preceding a bid indicates that the bid is insufficient (A)
  - ^S indicates that a player was not able to call, because another player has done a call out of rotation (B); the ^S replaces the skipped player's call
  - ^R preceding a card indicates a revoke (C)
  - ^L preceding a card indicates a lead out of turn (D). The remaining 3 cards are "normally" played in clockwise order after the erroneous lead.



In the following examples West is dealer (situations (A) and (B)), or West must play the lead card (situations (C) and (D)) :

(A)	1D	^I	1C	West opens 1 Diamonds, North bids 1 Clubs
(B)	^S	1D		West is dealer, but North opens 1 Diamonds
	1D	^S	2D	West opens 1 Diamonds, East bids 2 Diamonds
	1D	^S	^I 1C	West opens 1 Diamonds, East bids 1 Clubs
	1D	^S	^S 2D	West opens 1 Diamonds, South bids 2 Diamonds
	1D	^S	^S ^I 1C	West opens 1 Diamonds, South bids 1 Clubs
(C)	CA	^R	S6 C3 C4	West plays Clubs Ace, North discards Spades 6
(D)	CA	C2	^L C3 C4	East instead of West plays the first card in a trick

### 3.10 End positions

PBN can be used to describe end positions. A game is considered an end position when the maximum number of cards of each hand in the Deal tag is less than 13.

The trump is given by the Contract tag, but now the tag value only contains the denomination (without number of odd tricks and without risk) : S, H, D, C, NT.

The tag value of the play section indicates the lead player at the first trick.

### 3.11 Case sensitivity

PBN is case sensitive. Suits, sides, ranks, etc are given by uppercase characters. In import format, lowercase characters are allowed, to ease human input. In export format, only uppercase characters are allowed. The following table shows the tags where lowercase characters can be used:

Upper	Lower	Description	Tags
W	w	West	Dealer, Deal, Declarer, Auction, Play
N	n	North	Dealer, Deal, Declarer, Auction, Play
E	e	East	Dealer, Deal, Declarer, Auction, Play
S	s	South	Dealer, Deal, Declarer, Auction, Play
S	s	Spades	Deal, Contract, Auction, Play
H	h	Hearts	Deal, Contract, Auction, Play
D	d	Diamonds	Deal, Contract, Auction, Play
C	c	Clubs	Deal, Contract, Auction, Play
NT	nt	NoTrump	Contract, Auction
X	x	double	Contract, Auction
XX	xx	redouble	Contract, Auction
Pass	pass	Pass	Contract, Auction
AP	ap	AllPass	Auction
A	a	Ace	Deal, Play
K	k	King	Deal, Play
Q	q	Queen	Deal, Play
J	j	Jack	Deal, Play
T	t	Ten	Deal, Play
None	none	no vuln.	Vulnerable
Love	love	no vuln.	Vulnerable
NS	ns	NS vuln.	Vulnerable
EW	ew	EW vuln.	Vulnerable
All	all	All vuln.	Vulnerable
Both	both	All vuln.	Vulnerable



## 4 Supplemental tags

The following tags can be used optionally. They are not contained in the mandatory set MTS.

The title of each chapter contains the tag names. The tag value of each tag is a string token without any restrictions unless stated otherwise.

### 4.1 Game related information

#### 4.1.1 Tag: Competition

This tag describes the type of competition. Examples tag values are:

"Cavendish", "Chicago", "Individuals", "Pairs", "Rubber", "Teams".

#### 4.1.2 Tag: DealId

This tag identifies a deal by a unique value; no two PBN deals should have the same tag value for DealId. The syntax of the tag value is left unspecified; typically, it will contain values from tags such as Round, Board, Event, Date, Generator, etc.

Adding this tag to games allows programs to easily find all games belonging to the same deal, for instance to do comparisons for duplicate scoring. Usage is strongly recommended, when there is a chance that the PBN games are copied in other PBN files.

#### 4.1.3 Tag: Description

The purpose of this tag is to give arbitrary game description. It can be used when no other tag is suited.

#### 4.1.4 Tag: FrenchMP

This tag indicates if French scoring is used, default is 'No'. The only two possible tag values are: "Yes" and "No".

#### 4.1.5 Tag: Generator

This tag indicates how the cards have been generated. It is intended for bridge computer programs and especially hand generators. The tag value may include the name of the program and possibly a seed value.

#### 4.1.6 Tag: Hidden

This tag indicates the directions of the hands that should be hidden by a viewer after the PBN game has been loaded. The idea is to reveal the hands on user request, after the user has solved some task using only the visible cards. The tag value is a sequence of characters representing the directions to be hidden. The possible characters are: W (West), N (North), E (East), and S (South). For example: [Hidden "WE"] means, that initially the cards of E/W must be hidden.

#### 4.1.7 Tag: Room

This tag can be used in a teams bridge tournament. The only two possible tag values are: "Open" and "Closed".



### 4.1.8 Tag: Termination

This takes a string that describes the reason for the conclusion of the game. While the Result tag gives the result of the game, it does not provide any extra information and so the Termination tag is defined for this purpose.

Examples are:

"abandoned"	abandoned game.
"adjudication"	result due to third party adjudication process
"death"	losing player called to greater things, one hopes
"emergency"	game concluded due to unforeseen circumstances
"normal"	game terminated in a normal fashion
"rules infraction"	administrative forfeit due to losing player's failure to observe either the Laws of Bridge or the event regulations
"time forfeit"	loss due to losing player's failure to meet time control requirements
"unterminated"	game not terminated

## 4.2 Score related information

### 4.2.1 Tag: Score

This tag gives the number of points of the game based on the trick score. The score can be given in 5 possible formats:

"<score>"	score of declarer
"EW <score>"	score of EW
"NS <score>"	score of NS
"EW <score> NS <score>"	score of EW resp. NS
"NS <score> EW <score>"	score of NS resp. EW

where <score> is the integer number of points.

The score can deviate from the normal value due to irregularities. This can be explained in commentary.

### 4.2.2 Tag: ScoreIMP

This tag gives the score in International MatchPoints based on the difference between score points (given in tag Score). The score can be given in the 5 possible formats, defined in the Score tag. Now, <score> is the number of IMPs. The value of <score> is normally an integer number. It can also be a broken number due to averaging of IMP values. In that case, it is represented as a decimal number having a decimal dot.

The modifiers in the Scoring tag specify how the IMPs are computed.

### 4.2.3 Tag: ScoreMP

This tag gives the MatchPoints based on the ranking of score points (given in tag Score). The score can be given in the 5 possible formats, defined in the Score tag. Now, <score> is the number of MatchPoints. The value of <score> is normally an integer number. It can also be a broken number due to averaging of MatchPoints values. In that case, it is represented as a decimal number having a decimal dot.

The computation of the MatchPoints is computed as: the sum of points, constructed by earning 2 points for each lower score, 1 point for each equal score, and 0 points for each higher score.

This definition is indicated in the Scoring modifiers as 'MP1'.

### 4.2.4 Tag: ScorePercentage

This tag gives the percentage score. The score can be given in the 5 possible formats, defined in the Score tag. Now, <score> is the percentage, a decimal value between 0 and 100.



#### 4.2.5 Tag: ScoreRubber

This tag gives the game's score for Rubber bridge or Chicago bridge. The score can be given in the 5 possible formats, defined in the Score tag. Now, <score> is defined as:

<score> = <non-negative integer>/<non-negative integer>

where the integer before "/" is the score above-the-line (bonus points or premium points), and the integer after "/" is the score below-the-line (trick points).

#### 4.2.6 Tag: ScoreRubberHistory

This tag gives the scores above-the-line (bonus points or premium points) and the scores below-the-line (trick points) for both sides at the beginning of a deal in a rubber match. The syntax of the tag is:

[ScoreRubberHistory "NS <HistoryScore> EW <HistoryScore>"]

Each <HistoryScore> consists of 3, 4 or 5 score units (being non-negative integers), depending on the number of finished rubber games in the current rubber. The syntax of <HistoryScore> is:

<above\_previous> <above\_current>/<below-1> <below-2> <below-3>

where

<above_previous>	all above-the-line points in previous rubbers (if any)
<above_current>	all above-the-line points in the current rubber
<below-n>	the below-the-line points, either at begin of the current deal in game 'n' or at the end of finished game 'n'

During the first game, the values for <below-2> and <below-3> are omitted.

During the second game, the values for <below-3> are omitted. The slash "/" separates the scores above-the-line from the scores below-the-line just as in tag ScoreRubber.

Note that the number of games won defines the vulnerability. This must be consistent with the Vulnerable tag (if present).

#### 4.2.7 Tag: OptimumScore

This tag gives the optimum number of points of a deal based on the trick score of the optimum result, with open cards (double dummy). This result is computed by comparing all possible (doubled) contracts, including all passes. The number of tricks for a certain contract - and hence the corresponding score - may be derived from the OptimumResultTable.

The optimum score can be given in 4 possible formats:

"EW <score>"	score of EW
"NS <score>"	score of NS
"EW <score> NS <score>"	score of EW resp. NS
"NS <score> EW <score>"	score of NS resp. EW

where <score> is the integer number of points. Note that the NS <score> is opposite to the EW <score>. For example, if NS can make 3NT with 10 tricks and 5D with 11 tricks with scores of 630 respectively 600 and EW can play 5C doubled for a score of -500 then NS's optimum score is 600.



### 4.3 *Player related information*

#### 4.3.1 Tags: BidSystemEW, BidSystemNS

These tags describe the bidding system for each side.

#### 4.3.2 Tags: PairEW, PairNS

These tags describe the partnerships East/West and North/South. Suitable tag values are the names of the players, when it is unknown who is sitting North, East, South, West. Other suitable tag values are the values of the PairId\_NS and PairId\_EW columns from the ScoreTables, or the values of the PairId column from the TotalScoreTable.

#### 4.3.3 Tags: WestNA, NorthNA, EastNA, SouthNA

These tags are the e-mail or network addresses of the players.

#### 4.3.4 Tags: WestType, NorthType, EastType, SouthType

These tags describe the player types. Two typical example tag values are:

"human"	a human player
"program"	an algorithmic (computer) player

### 4.4 *Event related information*

The following tags are used for providing additional information about the event.

#### 4.4.1 Tag: EventDate

This tag describes the starting date of the event. The used format is the same as for the Date tag.

#### 4.4.2 Tag: EventSponsor

This tag gives the name of the sponsor of the event.

#### 4.4.3 Tag: HomeTeam

This tag gives the name of the home team.

#### 4.4.4 Tag: Round

The Round tag value gives the playing round for the game. The round indicator consists of letter characters, digit characters and the underscore.

Some organizers employ unusual round designations and have multipart playing rounds and sometimes even have conditional rounds. In these cases, a multipart round identifier can be made from a sequence of round indicators separated by periods. The leftmost indicator represents the most significant playing round, and succeeding indicators represent playing rounds in descending hierarchical order.

#### 4.4.5 Tag: Section

This tag is used for the playing section of a tournament. Examples are "Open", "Ladies" or "Reserve".

#### 4.4.6 Tag: Stage

This tag is used for the stage of a multistage event. Examples are "Preliminary" or "Semifinal".

#### 4.4.7 Tag: Table

This tag identifies the table in a tournament. The tag value will normally be a positive integer.

#### 4.4.8 Tag: VisitTeam

This tag gives the name of the visiting team.

### 4.5 *Time and date related information*

The following tags assist with further refinement of the time and data information associated with a game.



#### 4.5.1 Tag: Time

This uses a time-of-day value in the form "HH:MM:SS"; similar to the Date tag except that it denotes the local clock time (hours, minutes, and seconds) of the start of the game. Note that colons, not periods, are used for field separators for the Time tag value. The value is taken from the local time corresponding to the location given in the Site tag pair.

#### 4.5.2 Tag: UTCDate

This tag is similar to the Date tag except that the date is given according to the Universal Coordinated Time standard.

#### 4.5.3 Tag: UTCTime

This tag is similar to the Time tag except that the time is given according to the Universal Coordinated Time standard.

### 4.6 Time control

The following tags are used to help describe the time control used with the game.

#### 4.6.1 Tag: TimeControl

This tag indicates how many games must be played within a certain time limit. The tag value has the syntax: "<NrGames>/<NrMinutes>" This means that the number of games, given by <NrGames>, must be finished before a time in minutes, given by <NrMinutes>.

For example: [TimeControl "4/30"] means, that 4 games must be finished in half an hour.

When there is no time limit at all, then use "" as tag value.

#### 4.6.2 Tag: TimeCall

This tag is used to limit the time for making a single call. The tag value is defined in number of seconds.

This tag can typically be applied for computer programs.

#### 4.6.3 Tag: TimeCard

This tag is used to limit the time for playing a single card. The tag value is defined in number of seconds.

This tag can typically be applied for computer programs.



## 4.7 Miscellaneous

These are tags that can be briefly described and that don't fit well in other chapters.

### 4.7.1 Tag: Annotator

This tag identifies the annotator or annotators of the game.

### 4.7.2 Tag: AnnotatorNA

This tag is the e-mail or network addresses of the annotator.

### 4.7.3 Tag: Application

The intention of this tag is to fill in a particular application (especially a computer program). It can be used as an anchor for Application specific data, that are added immediately after this tag in comments.

### 4.7.4 Tag: Mode

This tag gives the playing mode of the game. Examples are:

"EM" electronic mail  
"IBS" Internet Bridge Server  
"OKB" OK Bridge  
"TABLE" normal table  
"TC" general telecommunication

## 4.8 Tag value inheritance

A tag with the special tag value "#" inherits the tag value of the corresponding tag in the nearest previous game. This process repeats until a tag has been found with a 'real' tag value different from "#". If no previous tag can be found, then the tag value is set to "".

There are some exceptions to this feature. The tag value "#" is forbidden for the tags 'Dealer', 'Vulnerable', 'Deal', 'Declarer', 'Contract', 'Auction', 'Play', and the note tags in the auction section and the play section.

A program may show "#" on the user interface instead of the 'real' inherited tag value. The user should be aware of this. Thanks to the exceptions, the PBN game can still be played.

This optional feature not only minimizes the file size. It also stresses that a PBN game is related to a previous PBN game.

The concept of the tag value "#" is defined one step further. When a tag has a tag value beginning with ##, then the (optional) text after ## can be interpreted as: use this text in all subsequent games. So, all the subsequent games need not repeat the tag with a "#". This feature minimizes the file size a lot more.

The tags that are forbidden to use "#" (see above), are also forbidden to use ##.

The tags with ## at the top of the file can be seen as a header. Such tags can still redefine their tag value in the middle of the file. The redefined tag value need not begin with ##.

Example: at the top of the file, the date tag is given using ##. At the middle of the file the date tag is redefined because the next 24 boards are played at the next day.

Note that in export format a mandatory tag can be omitted when there is a corresponding tag using ## in a previous game.

## 4.9 Expandability

PBN is intended to be expandable. Future versions can have new tag pairs, that are a valuable addition to the current version. When a PBN file includes new tag pairs, then these tags might not be recognized by PBN parsers. A PBN parser is robust when it just skips such tags, and perhaps shows a warning.

The syntax of a tag is designed so that it must be able to parse any tag without interpreting its contents. The same remark holds for supplemental sections (chapter 5). The syntax of a section element allows a parser to skip all the section data until a tag has been found or until the end of the game.

## 4.10 Compatibility

The PBN standard is subject to change. Newer versions clarify ambiguities observed in applying the standard. Also, usage of the PBN standard indicates shortcomings in the possibilities of the PBN notation.





The differences between the PBN versions lead to compatibility problems. The intention is to restrict these problems to a minimum.

Backward compatibility is maintained by the following rule: a PBN file complying to a certain PBN version must also be valid for any higher PBN version.

Forward compatibility can be guaranteed by honouring the expandability feature. This implies that the syntax of tags and sections must obey the following rules. The general syntax of a tag (as defined in chapter 3.3) may not change. And, the general syntax of a section (as defined in chapter 5) may not change.



## 5 Supplemental sections

### 5.1 General description

A supplemental section is a unit of information consisting of a tag pair and section data following this tag pair. In general, the section data are so large that they don't fit in a tag value.

The section data consist of so-called section elements. A section element has 2 representations. Either it is a string token, or a section token. A string token can be used for section data consisting of characters, that are not allowed for section tokens. Thanks to the double quotes at the begin and end of the string token, it is clear where such a section element begins and ends.

The white space characters serve as separator between section elements. Section tokens do not include white space characters. The string token can have spaces, and is used for e.g. names.

Only the tag pair of the section may be followed by commentary. Commentary after a section element is forbidden. To avoid confusion with commentary, the characters ';', '{', and '}' are forbidden in section tokens.

In export format, all the section elements in one text line are separated by a single space character. There is an exception to this rule, namely in case of tables that specify a minimum column width.

### 5.2 Description of tables

The supplemental sections below can be viewed as tables. The section data of tables have some additional characteristics. For clearness' sake, the section elements of tables are called table elements.

The section data of a table are structured in rows and columns. A table element belongs to one row and one column. The tag value of a table describes the columns of the table's section data. A table has at least 1 column. The tag value consists of 1 or more column name tokens separated by the semicolon ';':

Each table below has a set of predefined column names. It is not needed to use all these predefined columns, any subset is possible. Furthermore, it is allowed to add columns with user defined column names. Such a name may not have ';' since this character serves as separator.

The rows of a table do not need to have a specific order. In most cases there will be a numerical order, e.g. based on a score. When the table elements in a column have a descending or ascending order, then this can be indicated explicitly by entering a '+' or '-' just before the column name. The character '+' means descending (i.e. highest value at first row), and the character '-' means ascending (i.e. lowest value at first row).

The minimum width of the table elements in a column can be specified. This is needed to format a table nicely in export format. The width is indicated by entering at the end of the column name a backslash '\' followed by an integer being the minimum width. Extra spaces must be added before/after the table element to reach the minimum width. Optionally, the alignment can be indicated. After the minimum width integer, character 'L' indicates left alignment (only extra spaces after the table element), and 'R' indicates right alignment (only extra spaces before the table element).

An unknown table element is indicated by '?', and an irrelevant table element is indicated by '!'. A table element representing an artificial score value is preceded by the caret character (^).

In import format, there is no restriction on the number of table elements in a text line.

In export format, all the table elements of a row are contained in one text line, and they are separated by a single space character. There can be more than one space character, but only when a column has defined a minimum column width.

The following chapters describe the defined supplemental sections. The title of each chapter contains the tag name of the section.



### 5.3 ActionTable

This table gives a scale of percentages for certain actions in a game. Some comment in the game can indicate the user's task. After the user has made his choice, the table can be shown or even processed with the user's input. The possible actions are:

- (a) Making a call. All previous calls are given ending with '+'.  
 (b) Playing a card (especially the first lead card). All previous played cards are given ending with '+'.  
 (c) Bidding a contract. Comment can be added to tell opponent bidding. So, the final contract might be declared by the opponents, possibly doubled.

The predefined column names are:

Rank	rank order of action
Call	call as defined in auction section
Card	card as defined in play section
Declarer	declarer as defined in Declarer tag
Contract	contract as defined in Contract tag
Percentage	percentage for action

### 5.4 AuctionTimeTable

This table gives the thinking times of the calls during the auction. An Auction section must be given before this table section. Each row gives the thinking time of one call. The order of the thinking times is exactly the same as the order of the call tokens in the Auction section.

The predefined column names are:

Time	thinking time for call
TotalTime	total time since start of game

The syntax of these 2 fields equals (compare tag Time in chapter 4.5.1) :

HH:MM:SS	hours:minutes:seconds , or
MM:SS	minutes:seconds , or
SS	seconds

### 5.5 InstantScoreTable

This table shows ranges of scores, so no contracts. After playing a game, the realised score can be looked up immediately in the table and translated to a percentage.

The score ranges do not overlap, and all the score ranges together cover the whole score space.

The predefined column names are:

Rank	rank order of score
ScoreRange_NS	<score_range> of NS
ScoreRange_EW	<score_range> of EW
IMP_NS	IMP score of NS
IMP_EW	IMP score of EW
Percentage_NS	percentage score of NS
Percentage_EW	percentage score of EW

where <score\_range> can be:

<score>	score equal to <score> (as in Score tag)
<score1>,*	scores higher than <score1>
<score1>,<score2>	scores higher than <score1> and lower than <score2>
*,<score2>	scores lower than <score2>



## 5.6 *OptimumPlayTable*

This table shows all possible cards for optimum play based on the Contract tag, with open cards (double dummy). The first row gives the optimum cards of the opening leader at the first trick. Each next row gives the optimum cards for the current player based on the actual, previously played cards according to the Play section. The order of the rows corresponds with the order of the cards being played. The optimum cards are given by their ranks (A, K, Q, J, T, 9, 8, 7, 6, 5, 4, 3, 2) for each suit. Character '-' means that a suit has no optimum cards. So, the table has 4 predefined columns with names:

S ranks of optimum cards of suit Spades  
H ranks of optimum cards of suit Hearts  
D ranks of optimum cards of suit Diamonds  
C ranks of optimum cards of suit Clubs

## 5.7 *OptimumResultTable*

This table shows the optimum result of a deal in number of tricks, with open cards (double dummy). For each direction, being the declarer, and for each denomination the maximum number of tricks is given based on optimum declarer play and optimum defence. A full table consists of 20 rows. The table has 3 predefined columns with names:

Declarer direction of declarer: W, N, E, S  
Denomination S, H, D, C, NT  
Result number of tricks won by declarer: 0...13

## 5.8 *PlayTimeTable*

This table gives the thinking times of the cards during the play. A Play section must be given before this table section. Each row gives the thinking time of one played card. The order of the thinking times is exactly the same as the order of the card tokens in the Play section.

The predefined column names are:

Time thinking time for card  
TotalTime total time since start of game  
The syntax of these 2 fields equals (compare tag Time in chapter 4.5.1) :

HH:MM:SS hours:minutes:seconds , or  
MM:SS minutes:seconds , or  
SS seconds

## 5.9 *ScoreTable*

This table presents the duplicate game results of a tournament. After playing a game, the result can be compared with the games that have actually been played in a tournament.

The advantage is that a deal can be described in a single game without auction and play section. The input for such a game may come from a tournament administration program. When a game has an auction and play section, then the game still occurs as entry in the ScoreTable.

The predefined column names are:

Rank rank order of score; or best rank in case of ties  
PairId\_NS pair identification of NS players  
PairId\_EW pair identification of EW players  
Names\_NS <names> of NS players  
Names\_EW <names> of EW players  
Contract as in Contract tag: <k><denomination><risk>  
Declarer direction of declarer: W, N, E, S  
Result number of tricks won by declarer  
Score\_NS trick score of NS  
Score\_EW trick score of EW  
IMP\_NS IMP score of NS  
IMP\_EW IMP score of EW  
MP\_NS MatchPoints score of NS  
MP\_EW MatchPoints score of EW  
Percentage\_NS percentage score of NS  
Percentage\_EW percentage score of EW  
Multiplicity the multiplicity of the score



where <names> is defined as: a string token containing the names of the 2 players separated by ';' , or the name of the partnership.

Also artificial scores due to irregularities can be included. In such a case, possibly enter '-' for the fields Contract, Declarer, and Result.

### 5.10 TotalScoreTable

This table shows the total scores of (a part of) a tournament that have actually been realised.

The predefined column names are:

Rank	rank order of score; or best rank in case of ties
PairId	pair identification of players
Names<names>	of players (as defined in ScoreTable)
TotalScore	cumulative value of scores
TotalIMP	cumulative value of IMPs
TotalMP	cumulative value of MatchPoints
TotalPercentage	cumulative value of percentage
MeanScore	mean value of scores
MeanIMP	mean value of IMPs
MeanMP	mean value of MatchPoints
MeanPercentage	mean value of percentage
NrBoards	the number of played games
Multiplicity	the multiplicity of the score

In general, a mean value is averaged over the number of played games.

## 6 File issues

### 6.1 File name

A file with PBN games has extension .pbn.

### 6.2 File embedding

A PBN file or a part of it might be embedded in another document, e.g. an HTML file, or a word-processor document. In order to recognize the PBN text in that document, a set of delimiters can be applied. The delimiter <PBN> signals the begin of the PBN text, and the delimiter </PBN> signals the end of the PBN text. (Note, that the characters '<' and '>' are part of the delimiters.) The delimiters do not occur in a PBN file itself.

## 7 PBN software

Refer to the PBN homepages: <http://www.tistis.nl/pbn> and <http://www.backas.fi>.

