

Standard Specification for Paifu Files

For programs in repository

ChemistMikeLam/JapaneseMahjongPaifu

ChemistMikeLam
43129403+ChemistMikeLam@users.noreply.github.com

Version 1.0.0, Dated Unreleased

Table of Contents

1 Introduction	1	4 File structure	8
2 General notations	2	4.1 Pre-match information	9
2.1 Rules and explanations	2	4.1.1 Tournament information	9
2.2 Syntax of rules	2	4.1.2 Match time and place	9
2.3 Recurring elements in rules	2	4.1.3 Recorder information	9
2.3.1 The strLit format	2	4.1.4 Player information	10
2.3.2 The name format	3	4.1.5 Point rules	10
2.3.3 Shorthand tokens	3	4.2 Frame format	11
2.3.4 Frame ID	3	4.2.1 Before frame flow	11
2.3.5 Points representations	5	4.2.2 Frame flow	12
2.3.6 Tile representations	5	4.2.3 After frame flow	14
2.3.7 Hand representations	6	4.3 Post-match information	14
3 File properties	7	5 To-do and possible extensions	14
3.1 File extension	7	A Changelog	15
3.2 File encoding	7	B References	15
3.3 Whitespace	8	C Index of Identifiers	15
3.4 Comments	8		

1 Introduction

Japanese mahjong, also known as riichi mahjong, among other names, is a variant of mahjong originating and popular in Japan. Its popularity outside of Japan is also rising due to its less reliance on luck, as well as promotions by individuals and popular culture works. The establishment of the M league in 2018 pushes it further as a competitive sport aiming for inclusion in the olympics.

As a competitive game of skills, paifus (牌譜 / ぱいふ), or records of play, are very important for this game, just like they are for chess, go, shogi, and other games in this genre. Paifus of professional players can often be found at archives of professional player associations and match organisers, while various online Japanese mahjong platforms also provide for paifu service for its players, with or without charge.

However, (from limited google searches) there seems to lack an open standard for digitally storing paifus in a computer-readable fashion. Many paifus still only exist offline, and for those online, each vendor seem to have their own in-house non-disclosed standards. This is detrimental to the development of the sport, as the rise of popularity of the sport would inevitably require paifus to be publically available and transmittable using a common standard.

While not posing as the perfect solution, this document aims to provide a description for documenting paifus in a computer-readable format in an open standard, upon which paifus can be recorded, transmitted digitally, and viewed through open- (and less preferably closed-) source softwares implemented base on this standard.

2 General notations

2.1 Rules and explanations

In the sections below, paragraphs stating rules and paragraphs explaining rules are intertwined together. Paragraphs stating rules are marked with a \triangle symbol in the margin; each symbol indicates one rule. Although paragraphs explaining rules are not marked, they are not meant to be skipped.

2.2 Syntax of rules

The description of structure of the paifu files employs an EBNF-ish syntax:

Item	Meaning
<i>identifier</i>	An identifier (nonterminal). If it appears left of a $=$, then it expands to the expression on the RHS. If it appears on the right, then it would be explained in another line.
"terminal"	A terminal string, verbatim. \" inside the string refers to \" without the backslash, however. Also, remove the surrounding double quotes before including in the files.
expr expr	Concatenation symbols are omitted for clarity
[...]^(m-n)	Content inside bracket can appear m to n times. m and n are both integers with $0 \leq m \leq n$. If n is omitted then $n = \infty$. If $m = n$ then it have to appear exactly m times.
	XOR. Only one among the list can appear each time
<i>deprecated</i>	Deprecated identifiers that should only appear in changelog.

2.3 Recurring elements in rules

2.3.1 The strLit format

\triangle *strLit* = ["snt[" *strNative* "]"]^(0-1) ["srm[" *strRoman* "]"]^(0-1)

\triangle *strNative* is the relevant name in its native script, wrapped in a pair of double quotes.

\triangle *strRoman* is the relevant name romanized, wrapped in a pair of double quotes.

The *strLit* format is used for (possibly) bilingual string literals in the paifu files.

2.3.2 The name format

⚠ `name = "(" [lname]^(0-1) "," [fname]^(0-1) ")"`

⚠ `lname = strLit`

⚠ `fname = strLit`

All human names in the file place the last name in front of the first name.

2.3.3 Shorthand tokens

Shorthand tokens are used to represent high-profile matches, teams, etc. They are used in place of certain *strLits* to reduce typing and to save (a little) disc space.

⚠ Programs designed to write paifu files are not required to use these tokens even if a suitable one is available, although its usage is suggested.

⚠ Programs designed to read paifu files of a certain version should be able to parse the shorthand tokens defined in that version.

⚠ Future versions may add new shorthand tokens, but they are not likely to be removed; removing shorthand tokens in use are likely to break backward compatibility.

⚠ Future versions should strive to keep the length of new shorthand tokens within 10 characters; however this is no strict restriction. Use as few characters as possible while staying meaningful and unambiguous.

A list of all shorthand tokens available for use in paifu files is available in Table 1.

2.3.4 Frame ID

In each match of Japanese mahjong, there are many “frames”¹ (局 / きょく) played. In Japanese, they are usually named in sequence from 東一局 (East 1) to 東四局 (East 4), then from 南一局 (South 1) to 南四局 (South 4); however, they sometimes have *n*本場 (honba) suffixed. Therefore, it is important to have a uniform way to refer to the frames in order to ease parsing.

⚠ `frameId = frameSeq "-" frameHonba`

⚠ `frameSeq = frameWind frameNum`

⚠ `frameWind = "E" | "S" | "W" | "N"`

⚠ `frameNum = "1" | "2" | "3" | "4"`

⚠ `frameHonba` refers to the honba number of the frame. It is an integer ≥ 0 .

¹I borrowed this terminology from snooker because I did not see any other translation of it into English.

Identifier	Shorthand	Meaning
<i>tourName</i>	"mlg"	M League / Mリーグ
<i>tourStage</i>	"1rd"	First round (of a general tournament)
	"2rd"	Second round (of a general tournament)
	"qtf"	Quarterfinal (of a general tournament)
	"smf"	Semifinal (of a general tournament)
	"fin"	Final (of a general tournament)
	"mlg-reg"	M League Regular Season Mリーグ レギュラーシーズン
	"mlg-sfs"	M League Semifinal Series Mリーグ セミファイナルシリーズ
	"mlg-fns"	M League Final Series Mリーグ ファイナルシリーズ
<i>matchPlace</i>	"mlg-std"	M League Studio / Mリーグスタジオ 2-1-16 Kaigan, Minato City, Tokyo, Japan 日本東京港区海岸2-1-16
<i>plyerTeam</i>	"mlg-drn"	M League - Akasaka Drivens Mリーグ 赤坂ドリブンズ
	"mlg-exf"	M League - EX Furinkazan Mリーグ EX風林火山
	"mlg-skn"	M League - Kadokawa Sakura Knights Mリーグ KADOKAWAサクラナイツ
	"mlg-mfc"	M League - Konami Mahjong Fight Club Mリーグ KONAMI麻雀格闘倶楽部
	"mlg-abm"	M League - Shibuya Abemas Mリーグ 渋谷ABEMAS
	"mlg-phx"	M League - Sega Sammy Phoenix Mリーグ セガサミーフェニックス
	"mlg-rdn"	M League - Team Raiden Mリーグ TEAM 雷電
	"mlg-prt"	M League - U-Next Pirates Mリーグ U-NEXT Pirates
<i>plyerAffil</i>	"none"	No known affiliation (≠ no data)
	"saikouisen"	最高位戦日本プロ麻雀協会
	"prokyoukai"	日本プロ麻雀協会
	"prorenmei"	日本プロ麻雀連盟
	"rmu"	RMU
	"rengoumu"	麻将連合-μ-
	"101"	101競技連盟
	"kishikai"	日本プロ麻雀棋士会
	"zennihon"	全日本麻雀協会

Table 1: All shorthand tokens used in paifu files.

2.3.5 Points representations

⚠ *pt* is a 1-decimal-place number, whose value is $\frac{1}{1000}$ of a point inside a match.

Because Japanese mahjong is highly dependent on the points used in the match, we have to use some kind of convention to represent this information. This convention makes the points in the paifu to be in the same size as most tournaments. If it is not clear enough, in our notation, the value of a riichi stick (立直棒) is represented as 1.0, while each honba stick (本場棒) typically adds 0.3 to the winner.

⚠ *ptTour* = "(" [*ptTourPrsnl*]⁽⁰⁻¹⁾ "," [*ptTourTeam*]⁽⁰⁻¹⁾ ")"

⚠ *ptTourPrsnl* = *pt*

⚠ *ptTourTeam* = *pt*

ptTour represents the personal and team tournament points.

2.3.6 Tile representations

⚠ *realTiles* = *numTiles* | *wordTiles* | *unknownTiles*

⚠ *numTiles* = *circleTiles* | *bambooTiles* | *charTiles*

⚠ *circleTiles* =
"1p" | "2p" | "3p" | "4p" | "5p" | "6p" | "7p" | "8p" | "9p" | "0p"

⚠ *bambooTiles* =
"1s" | "2s" | "3s" | "4s" | "5s" | "6s" | "7s" | "8s" | "9s" | "0s"

⚠ *charTiles* =
"1m" | "2m" | "3m" | "4m" | "5m" | "6m" | "7m" | "8m" | "9m" | "0m"

⚠ *wordTiles* = *windTiles* | *dragonTiles*

⚠ *windTiles* = "ew" | "sw" | "ww" | "nw"

⚠ *dragonTiles* = "wd" | "gd" | "rd"

⚠ *unknownTiles* = "uk"

As mahjong paifus, there must be some representation for the mahjong tiles. The number tiles are self-evident; 0 represents the red five tiles. The suffix refers to *pinzu* (筒子 / ピンズ, the circle tiles), *souzu* (索子 / ソウズ, the bamboo tiles) and *manzu* (萬子 / マンズ, the character tiles) respectively. The wind tiles have their first letter specifying direction, while the dragon tiles have their first letter specifying color (white, green and red). An unknown tile refers to a tile which identity is unknown to the paifu recorders.

2.3.7 Hand representations

⚠ *hand* =
"hnd[" *tileInHand*^(1-13) "," *tileTsumo*^(0-1) ","
 tileFuro^(0-4) "]"

⚠ *tileInHand* = *realTiles*

⚠ There are no rules regarding the order of tiles in the *tileInHand* field.

⚠ *tileTsumo* = *realTiles*

tileTsumo refers to “the 14th tile” which the player gets in one’s own round. When a player wins by ron, although the tile technically does not count as tsumo, this field is still used to document that tile.

In some cases this 14th tile might not be distinguished from the other tiles in hand. One common situation is at the beginning of a frame, when the dealer (親) may get all 14 tiles at once. In such cases, any one tile may go into this field, and the rest goes into *tileInHand*.

⚠ *tileFuro* = *furoChi* | *furoPon* | *furoKan*

tileFuro documents fūro (副露) in the hand.

⚠ When there are more than one fūro, the newest one should be documented first, with them getting older down the list.

⚠ *furoChi* = "chi[" *tileChiIn* "," *tileChiHand*^(2-2) "]"

⚠ *tileChiIn* = *realTiles*

⚠ *tileChiHand* = *realTiles*

furoChi refers to a fūro made by chi-ing a discard tile (*tileChiIn*) from the player on the left (上家) with 2 tiles in you hand (*tileChiHand*), forming a shuntsu (順子).

⚠ *furoPon* = "pon[" *tilePonIn* "," *tilePonHand*^(2-2) "," *tileFrom* "]"

⚠ *tilePonIn* = *realTiles*

⚠ *tilePonHand* = *realTiles*

⚠ *tileFrom* = "k" | "t" | "s"

furoPon refers to a fūro made by pon-ing a discard tile (*tilePonIn*) from any other player when one have two other of the same tile in hand (*tilePonHand*). The field *tileFrom* identifies the source of the tile: "k" for kamicha (上家), the player on the left; "t" for toimen (対面), the player on the oppositeside of the table; and "s" for shimocha (下家), the player on the right.

⚠ `furoKan = furoDmk | furoKkn | furoAnk`

⚠ `furoDmk = "dmk[" tileDmkIn "," [tileDmkHand]^(3-3) "," tileFrom "]"`

⚠ `tileDmkIn = realTiles`

⚠ `tileDmkHand = realTiles`

`furoDmk` refers to a *fūro* made by *daiminkan* (大明槓), an action that can only be done when another player discards a tile (`tileDmkIn`) which one has all the other three of the same tile in hand (`tileDmkHand`).

⚠ `furoKkn =
"kkn[" tileKknIn "," tilePonIn "," [tilePonHand]^(2-2) ","
tileFrom "]"`

⚠ `tileKknIn = realTiles`

`furoKkn` refers to a *fūro* made by *kakan* (加槓), an action that can only be performed when one has pon-ed a tile, and then draws the forth of that tile into hand (`tileKknIn`).

⚠ `furoAnk = "ank[" [realTiles]^(4-4) "]"`

`furoAnk` refers to an *ankan* (暗槓), an action that can only performed when one has all four of a tile in hand. This technically does not count as a *fūro*, but it has the same effect of removing the tiles from the hand, so it is documented in the same way as a *fūro*.

3 File properties

3.1 File extension

⚠ The paifu files should bear the extension “.jmdp”.

The extension refers to Japanese *mahjong paifu*. A four-letter extension is chosen for reduced likelihood of clashing with other file extensions.

3.2 File encoding

⚠ The paifu files should be plain text files encoded in UTF-8.

While many parts of the files would only require ASCII characters, the file can contain comments as well as other strings that might not be encodable using pure ASCII (eg names of players, which will most likely be in Japanese). UTF-8 is selected for its support of non-ASCII contents, as well as relative space efficiency over other unicode encodings over a predominantly ASCII plain text file.

⚠ The use of byte order mark (BOM) is not forbidden but is also not recommended.

⚠ Programs designed to read paifu files should ignore BOM at start of file if it exists.

- ⚠ Programs designed to write paifu files should not write BOM at start of file.

The use of BOM in UTF-8 is not suggested. However, since some other programs (especially legacy ones) might insert BOM at the start of the file on creation, reader programs are expected to be able to handle its presence. However, the presence or absence of the mark should not have any effect on the reading of an otherwise syntactically correct paifu file. As for writer programs, they are expected not to insert the mark at anytime. If they are rewriting an existing file, the modified file should not contain the mark, regardless of whether it originally contained the mark or not.

3.3 Whitespace

- ⚠ Whitespaces are ignored except in string literals wrapped in a pair of double quotes.

3.4 Comments

- ⚠ `//` marks the beginning of a comment. Such comments extends to the end of the same line. However, if the symbol is enclosed within a pair of double quotes, it will not be parsed as such.
- ⚠ Any such comments are to be ignored by parsers.

This provides for C-style comments that are ignored by parsers, which might be useful for hand-written files. Do not confuse this with the `frameCmt` field explained below in Section 4.2.3; this field contains comment that the parser would read verbatim as strings.

4 File structure

- ⚠ `file = "jmjp[" version "]" [match]^(1-)`

- ⚠ It is strongly recommended *against* including multiple matches in one single paifu file.

While syntactically valid, it is not recommended to store more than one match in a single paifu file, for readability and archive management concerns. However, if your use case strongly requires the storage of multiple matches within one single paifu file, you may do so. This also allow for simple merging and splitting of paifu files on demand.

However, there must be at least one match in each paifu file. This restriction is imposed considering that a paifu file without a match is meaningless.

- ⚠ `version` refers to the version number string `big.small`.
- ⚠ Patch updates of the standard should not create any incompatibility between versions.
- ⚠ `match = "(" preMatchInfo [frame]^(0-) postMatchInfo ")"`

`frame` is not a necessary component of `match`. While it might be counter-intuitive to allow matches without frame data in the paifu files, this arrangement allows for paifu files to document matches where the actual playing data are not available, but other match-related information exists (eg tournament point changes). This can further allow for homogeneity in the layer of files in an archive of paifus.

4.1 Pre-match information

⚠ `preMatchInfo =`
 `[tourInfo]^(0-1) [matchTP]^(1-1) [recerInfo]^(0-1) [plyerInfo]^(4-4)`
 `[ptRule]^(0-1)`

4.1.1 Tournament information

⚠ `tourInfo =`
 `"tnm[" [tourName]^(0-1) "," [tourYear]^(0-1) "," [tourStage]^(0-1)`
 `"," [matchNumInStage]^(0-1) "," [matchNumInDay]^(0-1) "]"`

⚠ `tourName = "mlg" | strLit`

⚠ `tourYear` is the year in which the tournament began, in Gregorian calendar and in format YYYY.

⚠ `tourStage =`
 `"1rd" | "2rd" | "qtf" | "smf" | "fin" | "mlg-reg" | "mlg-sfs" |`
 `"mlg-fns" | strLit`

⚠ `matchNumInStage` is the match number in the current stage, in arabic numerals.

⚠ `matchNumInDay` is the match number in the same day, in arabic numerals.

4.1.2 Match time and place

⚠ `matchTP =`
 `"mtp[" [matchDate]^(0-1) "," [matchDayOfWk]^(0-1) ","`
 `[matchTime]^(0-1) "," [matchPlace]^(0-1) "]"`

⚠ `matchDate` is the date of the day on which the match started, in the Gregorian calendar and in the format YYYYMMDD.

⚠ `matchDayOfWk = "sun" | "mon" | "tue" | "wed" | "thu" | "fri" | "sat"`

⚠ `matchTime` is the time at which the match started, in local time and in 24-hour format as HHMM.

⚠ `matchPlace = "mlg-std" | strLit`

Note that `matchDate` and `tourYear` does not necessarily overlap. For example, the matches for M league 2018 extended well into 2019.

4.1.3 Recorder information

⚠ `recerInfo = "rec[" [recerName]^(0-1) "," [framesRecd]^(0-1) "]"`

recerInfo records the information of the person who records the paifu. It is common for a whole match to be recored by the same person. However, some matches might have multiple people recording the match together, and some matches might have different recorder for different frames. Therefore, this piece of information is allowed to appear multiple times in *preMatchInfo*, while also possessing a field for noting the frames this person is responsible for recording. This may also be used as a form of crediting the paifu writer/recorder.

⚠ *recerName* = name

⚠ *framesRecd* = "all" | [*frameId*]⁽⁰⁻⁾

4.1.4 Player information

⚠ *plyerInfo* =
"ply[" *plyerId* "," [*plyerName*]⁽⁰⁻¹⁾ "," [*plyerTeam*]⁽⁰⁻¹⁾ ","
[*plyerAffil*]⁽⁰⁻¹⁾ "," [*ptTourPrev*]⁽⁰⁻¹⁾ "]"

⚠ *plyerId* = "0" | "1" | "2" | "3"

plyerId refers to the seating in frame E1-0. The person seating at the East seat (東家) is "0"; South seat (南家), "1"; West seat (西家), "2"; and North seat (北家), "3". With this information, the seating arrangement in each frame can be determined easily.

⚠ There are no fixed order for the four *plyerInfo*; however it is strongly recommended to list in ascending order of *plyerId*.

⚠ *plyerName* = name

⚠ *plyerTeam* =
"mlg-drn" | "mlg-exf" | "mlg-skn" | "mlg-mfc" | "mlg-abm" |
"mlg-phx" | "mlg-rdn" | "mlg-prt" | *strLit*

⚠ *plyerAffil* =
"none" | "saikouisen" | "prokyoukai" | "prorenmei" | "rmu" |
"rengoumu" | "101" | "kishikai" | "zennihon" | *strLit*

The shorthand tokens of *plyerAffil* listed above provide for the professional associations as listed in Wikipedia^[1].

⚠ Shorthand token "none" of field *plyerAffil* is used for “no known affiliation”; for no data leave the whole field out.

⚠ *ptTourPrev* = *ptTour*

ptTourPrev represent the personal and team tournament points right before the match begins.

4.1.5 Point rules

⚠ *ptRule* = "ptr[" *ptStart* "," *ptRetn* "," *ptPostn* "," *ptHonba* "]"

⚠ *ptStart* = *pt*

ptStart refers to the points each player begins with (配給原点). Typical matches use 25.0.

⚠ *ptRetn* = *pt*

ptRetn refers to the points each player needs to “return” after the match (原点). Typical matches use 30.0.

The excess ($4 \times ptRetn - 4 \times ptStart$) is called オカ (oka) in Japanese, and is often awarded to the first place player. In rare cases that oka is negative, the corresponding amount is deducted from the fourth place player.

⚠ *ptPostn* = "(" *ptPostn14* , *ptPostn23* ")"

⚠ *ptPostn14* = *pt*

⚠ *ptPostn23* = *pt*

ptPostn refers to the points that player exchange based on their rankings in the match (ウマ / 順位点). *ptPostn14* is the amount that the fourth place player gives to the first place player, while *ptPostn23* is the amount that the third place player gives to the second place player. It is usually the case that *ptPostn14* > *ptPostn23*, such that the rankings do not change because of the exchange.

⚠ *ptHonba* = *pt*

This field refers to the additional point (積み符) for the winning hand per honba stick (積み棒). Typical matches use 0.3, but a limited number of local rules might use 1.5.

4.2 Frame format

⚠ *frame* =
"frm[" *frameId* "," [*kyoutak*]⁽⁰⁻¹⁾ "," [*dice*]⁽⁰⁻¹⁾ ","
[*ptFrmStart*]⁽⁰⁻¹⁾ "," [*dora*]⁽⁰⁻¹⁾ "," [*frameFlow*]⁽⁰⁻⁾ ","
[*ptFrmEnd*]⁽⁰⁻¹⁾ "," [*frameCmt*]⁽⁰⁻¹⁾ "]"

4.2.1 Before frame flow

⚠ *kyoutak* = *pt*

kyoutak refers to the number of riichi stick left on the table (供託). Since each stick has a value of 1.0, we use the point value to represent it. Note that the honba sticks does not count; this information is already encoded in *frameId*.

⚠ *dice* = *diceRes* "-" *diceRes*

⚠ *diceRes* = "1" | "2" | "3" | "4" | "5" | "6"

dice represents the results of the two dices separately.

⚠ `ptFrmStart = "pfs[" pt [" , " pt]^(3-3) "]"`

`ptFrmStart` indicates the points each player has at the beginning of the frame, in the order of East seat, South seat, West seat and finally North seat.

⚠ `dora = [realTiles]^(10-10)`

`dora` documents the actual doras, not the indication tiles. The ten tiles should be arranged in the following order: normal dora, ura dora, kan dora 1, kan ura 1, kan dora 2, ..., kan ura 4.

4.2.2 Frame flow

⚠ `frameFlow = ["(" seat " , " startHand ")"]^(4-4) [frameAct]^(0-) ["(" seat " , " endHand ")"]^(4-4)`

⚠ `seat = "e" | "s" | "w" | "n"`

⚠ `startHand = hand`

⚠ `frameAct = "(" seat " , " drawTile " , " [discTile]^(0-1) ")"`

`frameAct` represents a complete act from drawing tile to discarding tile. Except for special cases, there should always be a discard tile.

⚠ `drawTile = realTiles | drawActs`

⚠ `drawActs = actChi | actPon | actDmk | actRsn | actRon | actOya`

⚠ `actChi = "ch[" [realTiles]^(2-2) "]"`

`actChi` refers to the case where a player chi-ed the discard tile from the previous player. Therefore the chi-ing player will have to take out the 2 documented tiles from hand to finish the act.

⚠ `actPon = "pn[" [realTiles]^(2-2) "]"`

`actPon` refers to the case where a player pon-ed the discard tile in the previous `frameAct`. The pon-ing player would have to take out the 2 documented tiles from hand to finish the act.

⚠ `actDmk = "dk[" [realTiles]^(3-3) "]"`

`actDmk` refers to the case where a player has declared a daiminkan on the previous discard tile. The kan-ing player would have to take out the 3 documented tiles from hand to finish the act. There should be no discard tile in this `frameAct`, and the next `frameAct` should also be by this player drawing an `actRsn`.

⚠ `actRsn = "rs[" realTiles "]"`

actRsn refers to the case where a player did a kan in the previous *frameAct*, and then drew a rinshan tile (嶺上牌) as documented.

⚠ *actRon* = "ro"

actRon refers to the case where a player rons on the previous discard tile (or kakan tile for stealing kan (搶槓)). If a player has declared ron, it should be documented, regardless of whether atama hane (頭跳ね) rules are in place. There should be no discard tile in this *frameAct*.

⚠ *actOya* = "oy"

actOya refers to the case of the first round for the dealer. If the dealer starting hand already contains 14 tiles, then he/she would not draw an additional tile in the first round, in which case the draw tile would be substituted with *actOya*. However, if the dealer start hand only have 13 tiles, then he/she would still draw a tile as usual.

⚠ *discTile* = *realTiles* | *discActs*

⚠ *discActs* = *actTmg* | *actKrg* | *actKkn* | *actAnk* | *actTsm* | *actRch*

⚠ *actTmg* = "tg"

actTmg refers to the case where a player has carried out tsumogiri (ツモ切り), or discarding the draw tile directly. It is specially documented because the act can imply meanings for other players on the table.

⚠ *actKrg* = "kg"

actKrg refers to the case where a player has carried out karagiri (空切り), or discarding a tile in hand that is identical to the draw tile. This act is often done to conceal tsumogiri, and thus it also carries special meaning, leading to it being coumented specially.

⚠ *actKkn* = "kk[" *realTiles* "]"

actKkn refers to the case where a player declares a kakan with the documented tile. Provided that the kakan stands (that is, no one steals it), the next *frameAct* should be by the same player, drawing an *actRsn*.

⚠ *actAnk* = "ak[" [*realTiles*]^(4-4) "]"

actAnk refers to the case where a player declares an ankan with the 4 documented tiles. Provided that the ankan stands (that is, no one steals it), the next *frameAct* should be by the same player, drawing am *actRsn*. It should be noted that ankan usually stands, since the limited variant of rules that alllows stealing ankan only allows kokushi (国士無双) to do so.

⚠ *actTsm* = "tm"

actTsm refers to the case where a player has won by tsumo on the draw tile of this *frameAct*.

⚠ `actRch = "rc[" realTiles "]"`

`actRch` refers to the case where a player declares riichi on discarding the documented tile.

⚠ `endHand = hand`

4.2.3 After frame flow

⚠ `ptFrmEnd = "pfe[" pt [", " pt]^(3-3) "]"`

`ptFrmEnd` is analogous to `ptFrmStart` except that it represents points at end of frame.

⚠ `frameCmt = strLit`

`frameCmt` is used for comments on the frame. This field contains comment that would be read verbatim by parsers as strings (`strLit`), which might be displayed or outputted depending on program settings. On the contrary, the file comments described in Section 3.4 would be outright ignored by parsers.

4.3 Post-match information

⚠ `postMatchInfo = [ptMatchEnd]^(0-1) [ptTourNew]^(0-1)`

⚠ `ptMatchEnd = "pme[" pt [", " pt]^(3-3) "]"`

`ptMatchEnd` refers to the points each player has after all leftover kyoutak, oka and position points exchanges are dealt with. The points are arranged in ascending order of corresponding `plyerId`.

⚠ `ptTourNew = "ptn[" ptTour [", " ptTour]^(3-3) "]"`

`ptMatchEnd` refers to the new personal and team tour points after the match, arranged in ascending order of corresponding `plyerId`.

5 To-do and possible extensions

- This document may have many typos, which needs correction
- A lot of explanations might be inadequate and needs to be expanded
- The shorthand tokens currently only covers a subset of Japanese stuffs, but considering that Japanese mahjong is expanding and becoming international, more tokens might be beneficial

A Changelog

1.0.0 ChemistMikeLam (Unreleased)

Added

- Added requirement for software to be able to parse all sorthand tokens in a supported version.
- Added functionality of file comments that would be ignored by parsers.
- (For future use) Added a new class and corresponding style of identifiers, for when some keywords becomes deprecated but still needs formatting in changelog.

Changed

- *version* now refers to the big . small version string instead of big version only.
- Representations of *furoPon*, *furoDmk*, *furoKkn*, *furoAnk*; *actPon*, *actDmk*, *actAnk* have been changed to accomodate the possibility of red five tiles in them.
- *name* now uses parenthesis instead of square brackets to surround *lname* and *fname*.
- The changelog, references and index sections are now put as appendix.

0.0.0 ChemistMikeLam (Tue 23 Jun 2020)

- Initial version

B References

- [1] <https://ja.wikipedia.org/wiki/%E7%AB%B6%E6%8A%80%E9%BA%BB%E9%9B%80>.
Accessed Sun 21 Jun 2020.

C Index of Identifiers

actAnk, 13	discTile, 12, 13	furoChi, 6
actChi, 12	dora, 11, 12	furoDmk, 7
actDmk, 12	dragonTiles, 5	furoKan, 6, 7
actKkn, 13	drawActs, 12	furoKkn, 7
actKrg, 13	drawTile, 12	furoPon, 6
actOya, 12, 13		
actPon, 12	endHand, 12, 14	hand, 6, 12, 14
actRch, 13, 14		kyoutak, 11
actRon, 12, 13	file, 8	
actRsn, 12, 13	fname, 3	lname, 3
actTmg, 13	frame, 8, 11	
actTsm, 13	frameAct, 12, 13	match, 8
	frameCmt, 8, 11, 14	matchDate, 9
bambooTiles, 5	frameFlow, 11, 12	matchDayOfWk, 9
	frameHonba, 3	matchNumInDay, 9
charTiles, 5	frameId, 3, 10, 11	matchNumInStage, 9
circleTiles, 5	frameNum, 3	matchPlace, 4, 9
	frameSeq, 3	matchTime, 9
dice, 11	framesRecd, 9, 10	matchTP, 9
diceRes, 11	frameWind, 3	
discActs, 13	furoAnk, 7	name, 3, 10

numTiles, 5	ptStart, 10, 11	tileDmkIn, 7
playerAffil, 4, 10	ptTour, 5, 10, 14	tileFrom, 6, 7
playerId, 10, 14	ptTourNew, 14	tileFuro, 6
playerInfo, 9, 10	ptTourPrev, 10	tileInHand, 6
playerName, 10	ptTourPrsnl, 5	tileKknIn, 7
playerTeam, 4, 10	ptTourTeam, 5	tilePonHand, 6, 7
postMatchInfo, 8, 14	realTiles, 5-7, 12-14	tilePonIn, 6, 7
preMatchInfo, 8-10	recerInfo, 9, 10	tileTsumo, 6
pt, 5, 11, 12, 14	recerName, 9, 10	tourInfo, 9
ptFrmEnd, 11, 14	seat, 12	tourName, 4, 9
ptFrmStart, 11, 12, 14	startHand, 12	tourStage, 4, 9
ptHonba, 10, 11	strLit, 2, 3, 9, 10, 14	tourYear, 9
ptMatchEnd, 14	strNative, 2	unknownTiles, 5
ptPostn, 10, 11	strRoman, 2	version, 8
ptPostn14, 11	tileChiHand, 6	windTiles, 5
ptPostn23, 11	tileChiIn, 6	wordTiles, 5
ptRetn, 10, 11	tileDmkHand, 7	
ptRule, 9, 10		