Open Foodservice System Consortium

ReceiptLine

Receipt Description Languages

Version 1.0

July 13, 2020 - Release

OFSC Store System Subcommittee
Printer WG

Disclaimer

This document and the information contained herein is provided on an "As Is" basis and OFSC disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

OFSC assume no responsibility for errors or omission sin this publication or other documents which are referenced by, cited by, or linked to this publication.

This publication could include the technical or other inaccuracies or typographical errors.

OFSC reserve the right to make improvements and/or changes to the information herein.

Trademarks

Unicode is a registered trademark of Unicode, Inc. in the United States and other countries.
QR Code is a registered trademark of DENSO WAVE INCORPORATED.
ReceiptLine is a registered trademark of Open Foodservice System Consortium.

Roster

Chairman

Hideo Nakamura	Seiko Epson Corporation

Members

Kazuhiko Yuzawa	CORCOCU CO., LTD
Satoru Higuchi	CITIZEN SYSTEMS JAPAN CO., LTD.
Takashi Imafuku	JustPlanning Inc.
Hiroshi Kakuno	JustPlanning Inc.
Hirofumi Fujiwara	STAR MICRONICS CO., LTD.
Yoshio Ohba	Seiko Instruments Inc.
Takanori Okayasu	Seiko Instruments Inc.
Tadashi Furuhata	Seiko Epson Corporation
Noriyuki Umetsu	FUJITSU ISOTEC LIMITED
Ayumi Miura	FUJITSU ISOTEC LIMITED
Masanori Murai	Putmenu, LLC.

Secretariat

Yasuo Sakami	Open Foodservice System Consortium

Inquiries regarding this document are accepted at the OFSC office.

Table of contents

1. Introduction

1.1 About this document

This document describes the specifications of ReceiptLine, a receipt description language.

ReceiptLine is the receipt description language that expresses the output image of small roll paper. It supports printing paper receipts using a receipt printer and displaying electronic receipts on a POS system or smartphone. It is possible to simply describe images such as receipts, foodservice slips, transaction slips, etc. in text data that does not depend on the paper width.

1.2 Character set

The character set of ReceiptLine can be anything.

This document assumes Unicode. See the annex for properties related to character width.

- The Unicode Consortium
  https://unicode.org/
- Unicode Standard Annex #11 EAST ASIAN WIDTH
  https://www.unicode.org/reports/tr11/

1.3 Character encoding scheme

The character encoding scheme of ReceiptLine can be anything, but when exchanging information between systems, UTF-8 should be used.

When using different character sets and character encodings, be careful of conversion. Pay special attention to the Japanese yen symbol. In Shift_JIS, "\" (0x5C) is used. It is described as "\\" (0x5C, 0x5C) by the ReceiptLine escape sequence. However, in UTF-8, the Unicode yen sign "¥" (U+00A5 YEN SIGN) is used.

1.4 Paper

The medium of paper or screen that outputs the image of ReceiptLine is called paper. The number of characters that can be printed in the horizontal direction of the paper is called characters per line (CPL).

The characters per line is based on a narrow character or halfwidth character. Wide character or fullwidth character is doubled.

The paper is cut at the end of the output image. In addition, manual cutting can be performed

in the middle of the output image.

1.5 Data structure

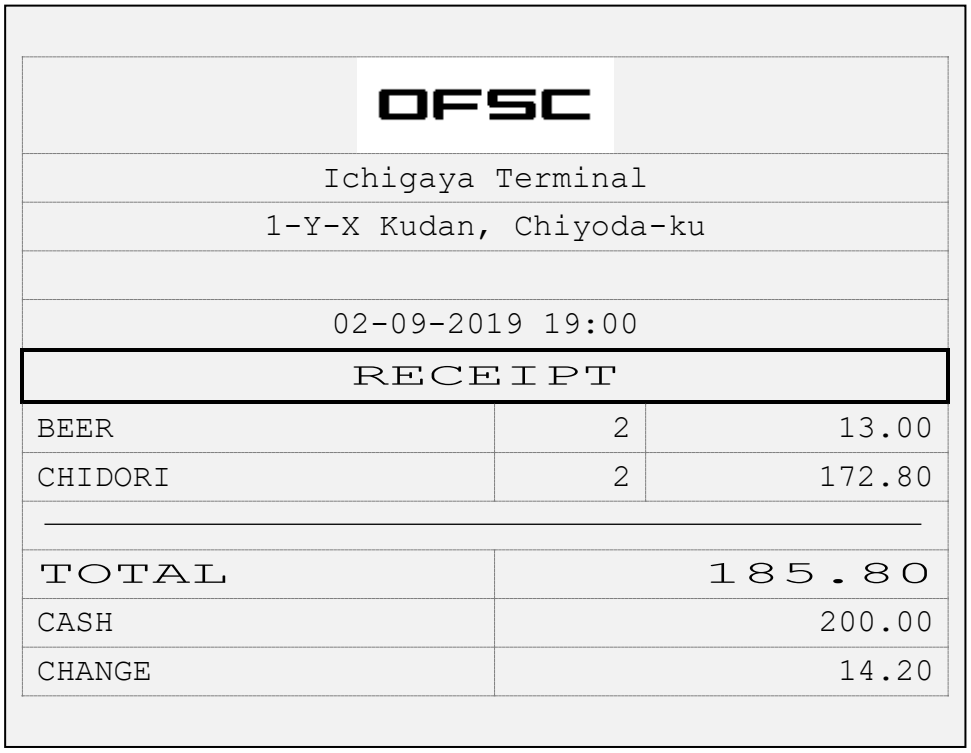
A ReceiptLine document is text data. A document is a set of lines, and the delimiter of the lines is a newline. A line is a set of columns, and the column delimiter is a vertical line "|". A column can have text or properties. A property is a set of key-value pairs, enclosed in curly braces "{" and "}".

To assist in explaining the specification, an example of a ReceiptLine document and an output image are shown. The dotted lines are the boundaries between rows and columns.

- Example of a ReceiptLine document

```
{image:iVBORw0KGgoAAAANSUhEUgAAAIAAAAAwAQMAAADjO
uD9AAAABlBMVEUAAAD///+l2Z/dAAAAZklEQVQoz2P4jwYYR
rrABwYGOwYG5gMMDBUMDPxAgQcMDDJAgQYGhgJcAv//yMj//
9/8//+HerAZRAsAzUASAJoGMhRF4AC6ANCIAhQz8AkAXQoUO
IDidBQBkG8hAj8gAqPJAa8AAGjulhOsX97yAAAAAElFTkSuQ
mCC}
         Ichigaya Terminal
      1-Y-X Kudan, Chiyoda-ku
---------------------------------------
02-09-2019 19:00
{border:line}
^RECEIPT
{border:space}
{width:*,2,10}
BEER                 | 2|     13.00
CHIDORI               | 2|    172.80
---------------------------------------
{width:*,20}
^TOTAL            |         ^185.80
CASH              |          200.00
CHANGE            |           14.20
```

- Example of an output image



OFSC

Ichigaya Terminal

1-Y-X Kudan, Chiyoda-ku

02-09-2019 19:00

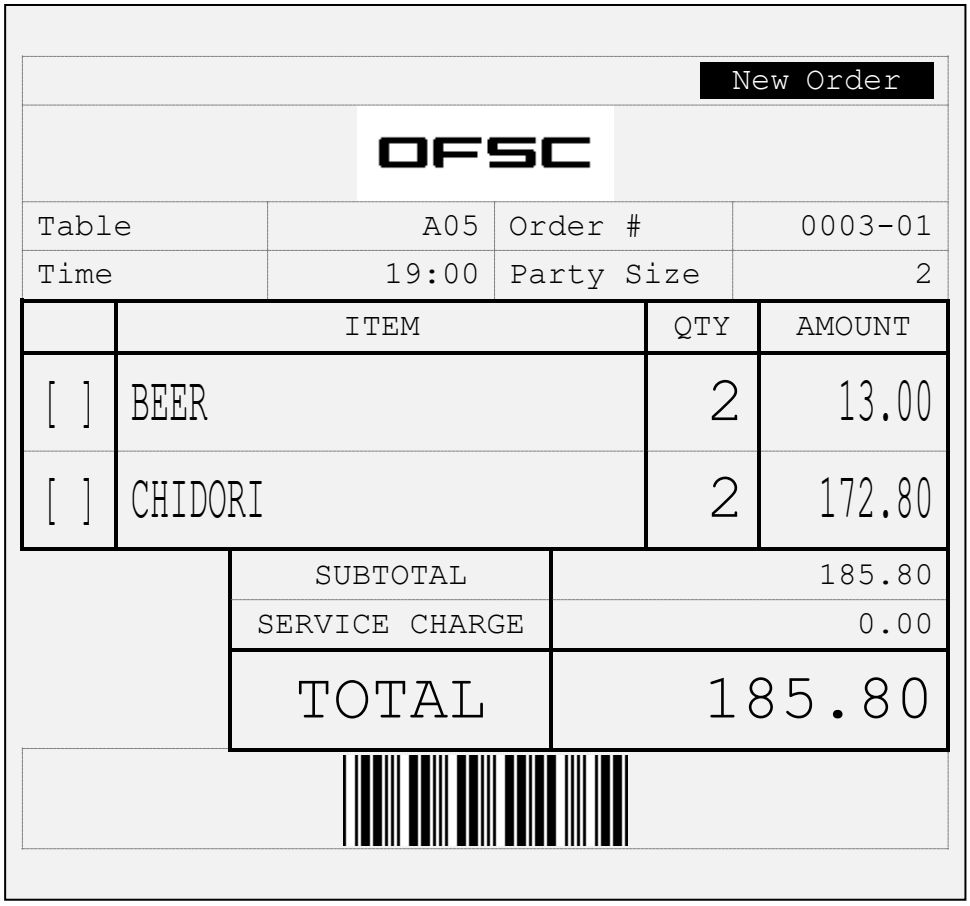
RECEIPT

BEER	2	13.00
CHIDORI	2	172.80
TOTAL		185.80
CASH		200.00
CHANGE		14.20

- Example of a ReceiptLine document

```
|                    `~~New Order~~|
{image:iVBORw0KGgoAAAANSUhEUgAAAIAAAABAAQMAAADoG
O08AAAABlBMVEUAAAD///+l2Z/dAAAAaUlEQVQ4y2P4jwYYR
gUGh8AHBgY7BgbmAwwMFQwM/ECBBwwMMkCBBgaGAlwC///Iy
P//3/z//4d6sBlECwDNQBIAmgYyFEXgALoA0IgCFDPwCQBdC
hQ4gOJ0FAGQbyECPyACo8lhSAgAACgAlSKnht6nAAAAAElFT
kSuQmCC}
|Table |    A05|Order #    |  0003-01|
|Time  |  19:00|Party Size |       2|
{width:3,*,4,9; border:line}
---------------------------------------
|     |    ITEM     | QTY |  AMOUNT  |
---------------------------------------
|^^[ ]|^^BEER       | ^^^2|   ^^13.00|
|^^[ ]|^^CHIDORI    | ^^^2|  ^^172.80|
---------------------------------------
{width:14,18; align:right}
      |    SUBTOTAL    |     185.80|
      | SERVICE CHARGE |       0.00|
      -------------------------------
      | ^^^TOTAL |        ^^^185.80|
      -------------------------------
{width:*; border:none; align:center}
{code:0003; option:code39,48}
```

- Example of an output image



2. Syntax

2.1 Notation

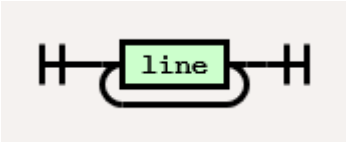
This document uses the simplified Backus-Naur form and syntax diagram style adopted in "Introducing JSON".

- Introducing JSON
  https://www.json.org/
- BNF, Backs-Naur Form
  https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form
- Syntax diagram, Railroad diagram
  https://en.wikipedia.org/wiki/Syntax_diagram

2.2 Document

A document is a chunk of information that represents a single output image.
A document consists of a set of lines.

document




```
document
    lines

lines
    line
    line lines
```

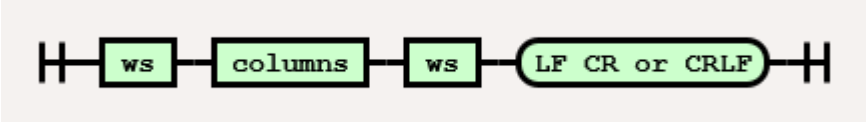
2.3 Line

A line is a row of information that consists a document.
A line has a set of columns and is terminated with a newline. Whitespaces (ws) can be

inserted before and after a set of columns to improve readability.

The characters of newline are LF (U+000A LINE FEED), CR (U+000D CARIDGE RETURN), or CRLF (U+000D CARIDGE RETURN, U+000A LINE FEED).

line




```
line
    ws columns ws newline

newline
    '000a'
    '000d'
    '000d' '000a'
```

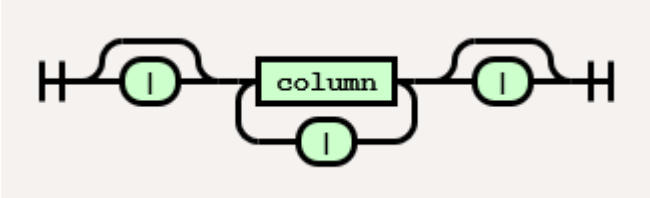
2.4 Column

A column is information that constitutes a set of columns.

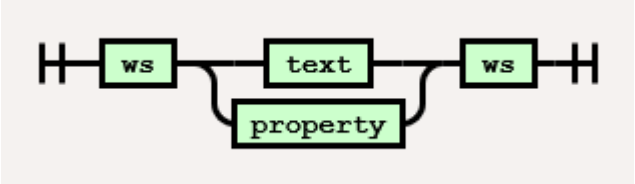
The column delimiter is a vertical line "|" (U+007C VERTICAL LINE). The vertical lines at the beginning and end of the set of columns can be omitted.

A column can have text or property. Whitespaces (ws) can be inserted before and after text and properties to specify alignment.

columns



column




```
columns
    content
    content '|'
    '|' content
    '|' content '|'

content
    column
    column '|' content

column
    ws text ws
    ws property ws
```

2.5 Text

A text is one of the information that column can have.
A text is a set of characters and is a character string including an empty string.

text

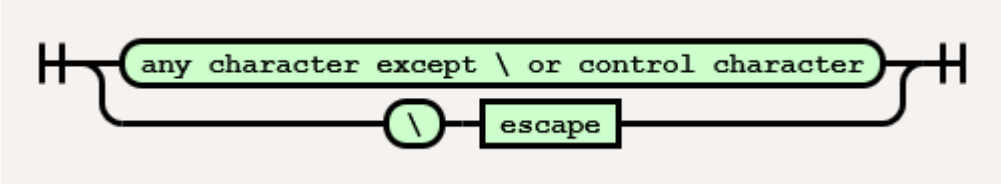
```
text
    ""
    chars

chars
    char
    char chars
```

2.6 Character

A character (char) is information that constitutes a text.

A character can be any single character except control characters, or an escape sequence consisting of a backslash "\" (U+005C REVERSE SOLIDUS) and an escape character.

char



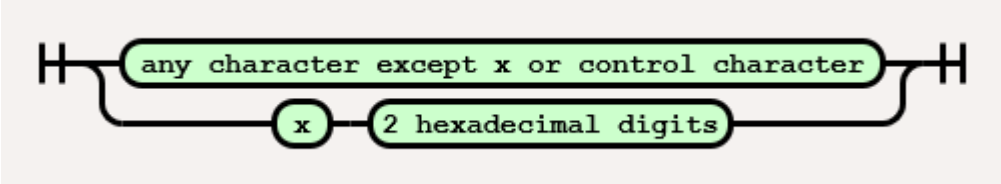

```
char
    '0020' . '10ffff' - '\'
    '\' escape
```

2.7 Escape character

An escape character is a character that constitutes an escape sequence.

Escape characters output characters with special meanings. If the escape character starts with "x", specify two hexadecimal digits consecutively.

escape



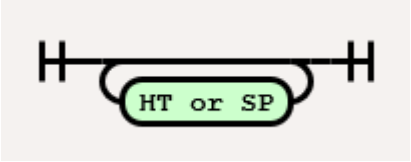

```
escape
    '0020' . '10ffff' - 'x'
    'x' hex hex

hex
    '0' . '9'
    'A' . 'F'
    'a' . 'f'
```

2.8 Whitespace

A whitespace (ws) is a non-printing character that is inserted to increase readability and specify column alignment.

A whitespace is a set of one or more horizontal tabs (U+0009 HORIZONTAL TABULATION) or spaces (U+0020 SPACE), including an empty string.

ws




```
ws
    ""
    '0009' ws
    '0020' ws
```

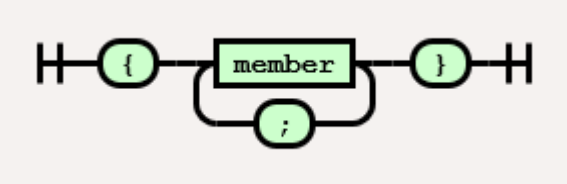
2.9 Property

A property is one of the information that a column can have.

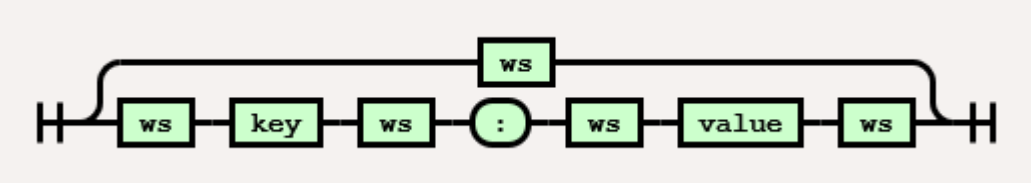
A property begins with a left brace "{" (U+007B LEFT CURLY BRACKET), has a member, and ends with a right brace "}" (U+007D RIGHT CURLY BRACKET). If the property has multiple members, the members are separated by a semicolon ";" (U+003B SEMICOLON).

A member can have a key-value pair. The key and value are separated by a colon ":" (U+003A COLON). Whitespaces (ws) can be inserted to improve readability.

property



member




```
property
    '{' members '}'

members
    member
    member ';' members

member
    ws
    ws key ws ':' ws value ws
```

2.10 Key-value

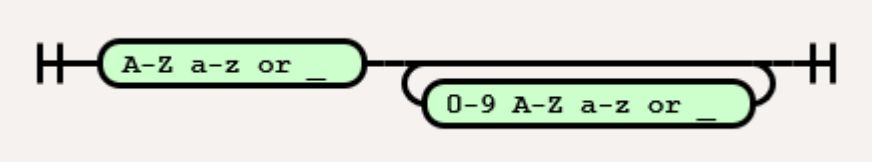
A key is information that identifies a member of a property.

Characters that can be used as keys are numbers, alphabetic characters, and the underscore "_" (U+005F LOW LINE). However, the first letter must be an alphanumeric character or an underscore.

If the key of a property is duplicated, the value described later has priority. Ignore unknown properties.

A value is a non-empty string.

key



value


```
key
    word
    word alphanumeric

alphanumeric
    '0' . '9'
    word

word
    'A' . 'Z'
    'a' . 'z'
    '_'

value
   chars
```

3. Special and escape characters

3.1 Special characters

A special character is a symbol that has a special meaning in a document. Special characters are assigned to characters that are used infrequently on receipts.

Common	Description
\	Character escape
|	Column delimiter
{	Property delimiter (start)
}	Property delimiter (end)

Specific (Text)	Description
- (1 or more, no other characters)	Horizontal rule • Valid for lines with a single column
= (1 or more, no other characters)	Paper cut • Valid for lines with a single column
~	Space • To output space at the beginning or end of text
_	Underline
"	Emphasis
`	Inversion
^	Enlargement

Specific (Property value)	Description
:	Key-value separator
;	Key-value delimiter

3.2 Escape characters

When embedding special characters or control characters in a document, use an escape sequence that starts with the character escape "\".

Ignore invalid escape sequences where characters do not follow the "\" correctly. For example, non-hexadecimal "\xnn", and "\", "\x", and "\xn" just before a newline are invalid.

Common	Description
`\\`	"\" (U+005C REVERSE SOLIDUS)
`|`	"|"
`\{`	"{"
`\}`	"}"
`\xnn`	Hexadecimal character code • "nn" is case-insensitive

Specific (Text)	Description
`\-`	"−" (Cancel horizontal rule)
`\=`	"=" (Cancel paper cut)
`\~`	"~"
`\_`	"_"
`\"`	""
`\``	"`"
`\^`	"^"
`\n`	Wrap text manually
`\`*char* (Other characters)	Ignore

Specific (Property value)	Description
`\;`	";"
`\n`	LF (U+000A LINE FEED)
`\`*char* (Other characters)	Ignore

4. Line

4.1 Line and column

A line can have one or more columns using the column delimiter "|".

Line	Description
column |*column*| |*column* *column*|	Single column • Column content is text or property
column|*column* |*column*|*column*| |*column*|*column* *column*|*column*|	Double column • Column content is text • Ignore lines that contain properties
column| ... |*column* |*column*| ... |*column*| |*column*| ... |*column* *column*| ... |*column*|	Multiple columns • Column content is text • Ignore lines that contain properties

- Example of a ReceiptLine document

```
|CREDIT CARD RECEIPT
{width:10 *}
AMOUNT |^123.45
ACCOUNT# |XXXXXXXXXXXX7890
DATE |Feb. 22, 2019
APPROVAL |0123456
{width:auto}
CUSTOMER COPY|
```

- Example of an output image

CREDIT CARD RECEIPT	
AMOUNT	１２３．４５
ACCOUNT#	XXXXXXXXXXXX7890
DATE	Feb. 22, 2019
APPROVAL	0123456
	CUSTOMER COPY

- Example of a ReceiptLine document

```
{width:21,10}
|DATE                |SLIP #      |
|Feb. 22, 2019        |       12345|
{width:10,10,10}
|MERCHANT  | TRAN TYPE |PAY COND     |
|   0123456| PARCHASE  |         123|
|^^SEQ #   |           | ^^[Lump-sum]|
|     7890|AMOUNT     |       12.34|
|APP CODE  |TAX/OTHERS |          0|
| ^^0001111|^^TOTAL    |     ^^12.34|
```

- Example of an output image

DATE		SLIP #
Feb. 22, 2019		12345
MERCHANT	TRAN TYPE	PAY COND
0123456	PARCHASE	123
SEQ #		[Lump-sum]
7890	AMOUNT	12.34
APP CODE	TAX/OTHERS	0.00
0001111	TOTAL	12.34

4.2 Line width

Line width is the sum of the width of each column set by the width property and the spacing of the column border set by the border property. When this total exceeds the characters per line (CPL) or includes the columns with automatic setting width, adjust the width of each column to match the characters per line (CPL).

Line width affects images, barcodes / 2D codes, horizontal lines, and commands. For text, applies column width.

- Example of a ReceiptLine document

```
{width: 12 12; border: none}
{image:iVBORw0KGgoAAAANSUhEUgAAAIAAAAAwAQMAAADjO
uD9AAAABlBMVEUAAAD///+l2Z/dAAAAZklEQVQoz2P4jwYYR
rrABwYGOwYG5gMMDBUMDPxAgQcMDDJAgQYGhgJcAv//yMj//
9/8//+HerAZRAsAzUASAJoGMhRF4AC6ANCIAhQz8AkAXQoUO
IDidBQBkG8hAj8gAqPJAa8AAGjulhOsX97yAAAAAElFTkSuQ
mCC} |
|TOTAL | 12.34|
-
| {code: 0003; option: 48}
```

- Example of an output image

4.3 Line alignment

Line alignment can be set with the align property.

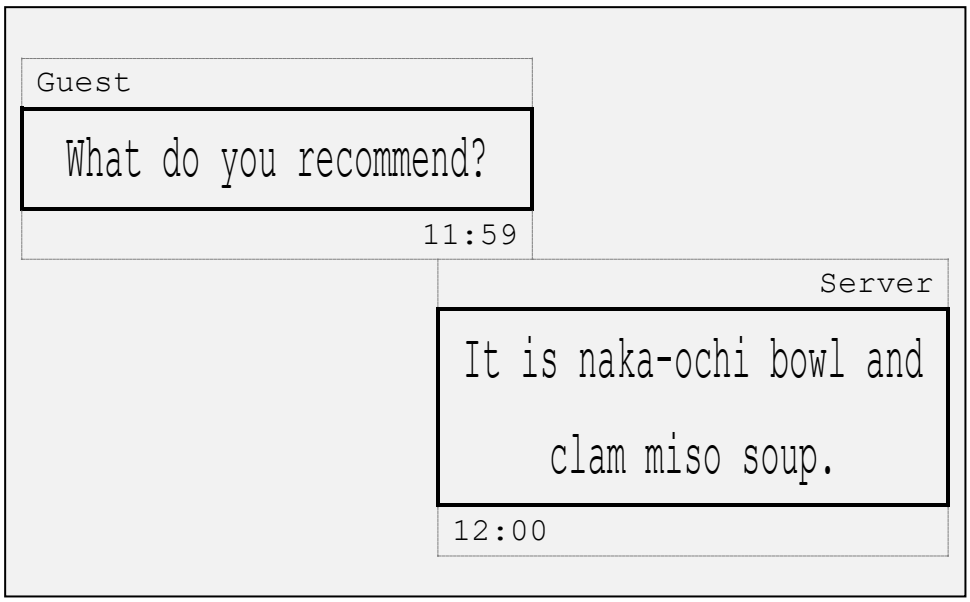
If the line width is less than the characters per line (CPL), it specifies the position to place the line.

Line alignment	Description
`left`	Left
`center` (Default)	Center
`right`	Right

- Example of a ReceiptLine document

```
{width: 25}
{align: left}
|Guest
{border: line}
^^What do you recommend?
{border: space}
11:59|
{align: right}
Server|
{border: line}
^^It is naka-ochi bowl and clam miso soup.
{border: space}
|12:00
```

- Example of an output image



5. Column

5.1 Column width

Column width can be set with the width property.

Column width affects text. For images, barcodes / 2D codes, horizontal lines, and commands, apply line width.

Column width	Description
width, *width*, ...	Set the output column and its width
`auto` (Default)	Set all column widths to "`*`"

When a column width is set manually, the widths are separated by commas or one or more whitespaces. The types of column width values are shown below. The output column is determined from the number of widths and their values.

Column width value	Description
`*`	Set the column width automatically
An integer greater than or equal to `1`	Set the column width and output
`0` or omitted	Do not output the column

When the column width is set automatically, the printable width is equally distributed to each column.

The printable width is the characters per line (CPL) minus the column border spacing set by the border property.

- Example of a ReceiptLine document

```
{width:auto}
Boiled salted peanuts | 1| 5.00
Tanashi cabbage with salted flavor | 1| 5.00
Shinshu lettuce salad | 1| 5.00
Ishigaki strawberry rolled cake | 1| 5.00
Peach juice from Date | 1| 5.00
```

- Example of an output image

Boiled salted peanuts	1	5.00
Tanashi cabbage with salted flavor	1	5.00
Shinshu lettuce salad	1	5.00
Ishigaki strawberry rolled cake	1	5.00
Peach juice from Date	1	5.00

If columns with automatic width and columns with manual width are mixed, the width of the automatic setting column is equally distributed by subtracting the width of the manual setting column from the printable width.

- Example of a ReceiptLine document

```
{width:*,2,*}
Boiled salted peanuts | 1| 5.00
Tanashi cabbage with salted flavor | 1| 5.00
Shinshu lettuce salad | 1| 5.00
Ishigaki strawberry rolled cake | 1| 5.00
Peach juice from Date | 1| 5.00
```

- Example of an output image

Boiled salted peanuts	1	5.00
Tanashi cabbage with salted flavor	1	5.00
Shinshu lettuce salad	1	5.00
Ishigaki strawberry rolled cake	1	5.00
Peach juice from Date	1	5.00

If the total width of the manual setting columns exceeds the printable width, reduce the width to match the printable width. The minimum width after reduction is one character.

- Example of a ReceiptLine document

```
{width:100,2,10}
Boiled salted peanuts | 1| 5.00
Tanashi cabbage with salted flavor | 1| 5.00
Shinshu lettuce salad | 1| 5.00
Ishigaki strawberry rolled cake | 1| 5.00
Peach juice from Date | 1| 5.00
```

- Example of an output image

Boiled salted peanuts	1	5.00
Tanashi cabbage with salted flavor	1	5.00
Shinshu lettuce salad	1	5.00
Ishigaki strawberry rolled cake	1	5.00
Peach juice from Date	1	5.00

Do not output zero-width columns. If all columns have zero width, output an empty line.

- Example of a ReceiptLine document

```
{width:*,0,10}
Boiled salted peanuts | 1| 5.00
Tanashi cabbage with salted flavor | 1| 5.00
Shinshu lettuce salad | 1| 5.00
Ishigaki strawberry rolled cake | 1| 5.00
Peach juice from Date | 1| 5.00
```

- Example of an output image

Boiled salted peanuts	5.00
Tanashi cabbage with salted flavor	5.00
Shinshu lettuce salad	5.00
Ishigaki strawberry rolled cake	5.00
Peach juice from Date	5.00

If the column with is set but the column does not exist, fill the empty column.

- Example of a ReceiptLine document

```
{width:*,,2,10}
Boiled salted peanuts | 5.00| 2| 10.00|#
Tanashi cabbage with salted flavor | 5.00| 1| 5.00
Shinshu lettuce salad | 5.00| 2
Ishigaki strawberry rolled cake | 5.00
Peach juice from Date |
```

- Example of an output image

Boiled salted peanuts	2	10.00
Tanashi cabbage with salted flavor	1	5.00
Shinshu lettuce salad	2	
Ishigaki strawberry rolled cake		
Peach juice from Date		

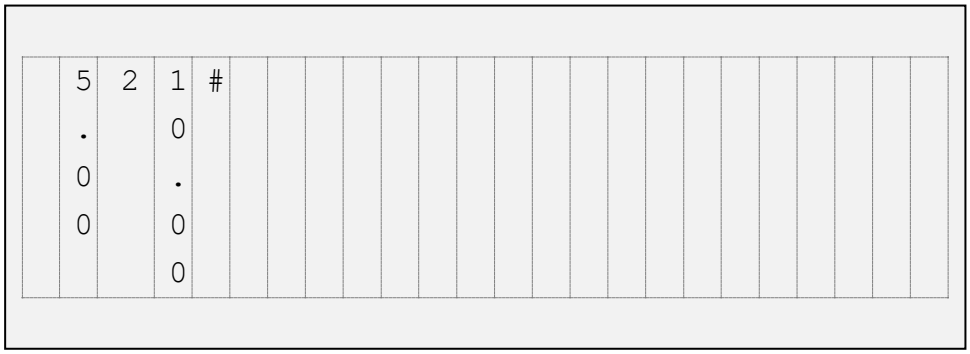
If the number of columns exceeds the printable width, the width of the excess columns is set to 0.

Do not output characters with a width that exceeds the column width, such as enlarged characters and Kanji characters.

- Example of a ReceiptLine document

```
{width:*,1,2,3,4,5,6,7,8,9,*,1,2,3,4,5,6,7,8,9,*
,1,2,3,4,5,6,7,8,9}
^Boiled salted peanuts | 5.00| 2| 10.00|#
```

- Example of an output image



5.2 Column border

Column border can be set with the border property.

Column border affects text. Does not apply to images, barcodes / 2D codes, commands, and horizontal lines.

Column border	Description

`0` `none`	No border • Columns are adjacent
`1` `space` (Default)	Space • Column to column spacing is one character wide
`2`	Wide space • Column to column spacing is 2 characters wide
`line`	Ruled line • Draw a vertical line between columns • Column to column spacing is one character wide • Convert a horizontal rule to a horizontal ruled line • Ignore horizontal rules without text • Draw an outline around the effective area • Leave one character wide for outline drawing • Draw outlines before and after the paper cut and the end of the document

- Example of a ReceiptLine document

```
{border: space; width: * 4 8}
-
|ITEM|QTY|AMOUNT|
-
Boiled salted peanuts | 1| 5.00
Tanashi cabbage with salted flavor | 1| 5.00
Shinshu lettuce salad | 1| 5.00
Ishigaki strawberry rolled cake | 1| 5.00
Peach juice from Date | 1| 5.00
-
```

- Example of an output image

ITEM	QTY	AMOUNT
Boiled salted peanuts	1	5.00
Tanashi cabbage with salted flavor	1	5.00
Shinshu lettuce salad	1	5.00
Ishigaki strawberry rolled cake	1	5.00
Peach juice from Date	1	5.00

- Example of a ReceiptLine document

```
{border: line; width: * 4 8}
-
|ITEM|QTY|AMOUNT|
-
Boiled salted peanuts | 1| 5.00
Tanashi cabbage with salted flavor | 1| 5.00
Shinshu lettuce salad | 1| 5.00
Ishigaki strawberry rolled cake | 1| 5.00
Peach juice from Date | 1| 5.00
-
```

- Example of an output image

ITEM	QTY	AMOUNT
Boiled salted peanuts	1	5.00
Tanashi cabbage with salted flavor	1	5.00
Shinshu lettuce salad	1	5.00
Ishigaki strawberry rolled cake	1	5.00
Peach juice from Date	1	5.00

5.3 Column alignment

Column alignment is determined by the presence or absence of one or more whitespaces "␣" adjacent to the column delimiter "|". The contents of the column are attracted by the vertical line and repelled by whitespaces.

Column alignment affects text, images, barcodes / 2D codes, and commands. Does not apply to horizontal lines.

Column alignment	Description
column |*column*| |␣*column*␣|	Center
|*column* |*column*␣| *column*␣|	Left
column| |␣*column*| |␣*column*	Right

6. Text

6.1 Text wrapping

Text wrapping can be set with the text property.

Text wrapping	Description
`wrap` (Default)	Wrap text
`nowrap`	Do not wrap text

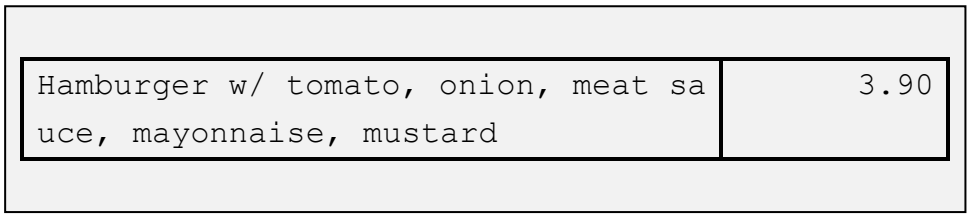
When wrapping a character string, the character string that exceeds the column width or the character string after "`\n`" is output by creating a row inside the column.

As the rows increase in the column, the height of the column increases, and the height of the line increases. The rows of each column in the same line are aligned top. The characters of each column in the same line are aligned with the baseline.

- Example of a ReceiptLine document

```
{width: *,10; border: line}
Hamburger w/ tomato, onion, meat sauce, mayonnaise,
mustard | 3.90
```

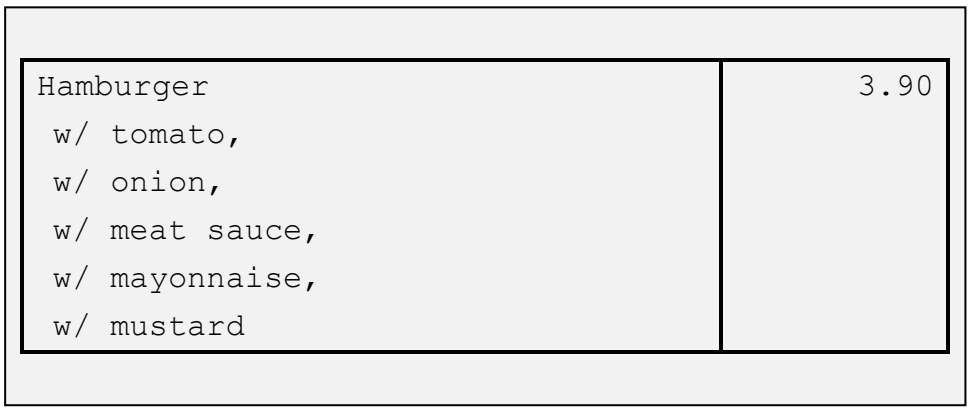
- Example of an output image



- Example of a ReceiptLine document

```
{width: *,10; border: line}
Hamburger\n w/ tomato,\n w/ onion,\n w/ meat
sauce,\n w/ mayonnaise,\n w/ mustard | 3.90
```

- Example of an output image

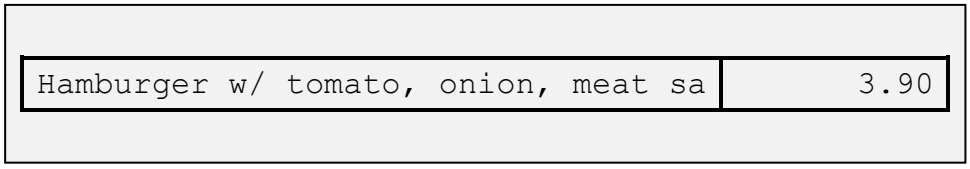


If the character string is not wrapped, the character string exceeding the column width or the character string after "\n" is not output.

- Example of a ReceiptLine document

```
{text: nowrap}
{width: *,10; border: line}
Hamburger w/ tomato, onion, meat sauce, mayonnaise,
mustard | 3.90
```

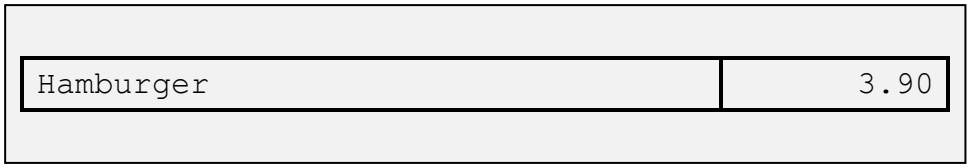
- Example of an output image



- Example of a ReceiptLine document

```
{text: nowrap}
{width: *,10; border: line}
Hamburger\n w/ tomato,\n w/ onion,\n w/ meat
sauce,\n w/ mayonnaise,\n w/ mustard | 3.90
```

- Example of an output image



6.2 Text decoration

By using special characters, some character strings in the text can be decorated.

The types of character decoration are underline, emphasis, inversion, and enlargement. Each character decoration can be combined by specifying a start position and an end position.

The scope of text decoration is a column. Text decoration in the previous column does not affect the next column. Text decorations that specify only the start position are released at the end of the column.

When characters with different heights are mixed on the same line, align them based on the baseline.

If the output device does not support double width or double height, it can be replaced with another character decoration expression that keeps the character width.

The following table shows the pattern of character decoration. The characters to be decorated are indicated by *chars*.

Text decoration	Description
_*chars*	Underline (start)
*chars*	Underline (start, end)
__	Underline (empty)
"*chars*	Emphasis (start)
"*chars*"	Emphasis (start, end)
""	Emphasis (empty)
`*chars*	Inversion (start)
`*chars*`	Inversion (start, end)
``	Inversion (empty)
^*chars*	Double width (start)
^*chars*^	Double width (start, end)
^^*chars*	Double height (start)
^^*chars*^^	Double height (start, end)
^^^*chars*	Double width and double height (start)
^^^*chars*^^^	Double width and double height (start, end)
^^^^*chars*	Enlargement (3 times, start)
^^^^*chars*^^^^	Enlargement (3 times, start, end)
^^^^^*chars*	Enlargement (4 times, start)
^^^^^*chars*^^^^^	Enlargement (4 times, start, end)
^^^^^^*chars*	Enlargement (5 times, start)
^^^^^^*chars*^^^^^^	Enlargement (5 times, start, end)
^^^^^^^*chars*	Enlargement (6 times, start)

(7 or more)	
^^^^^^^*chars*^^^^^^^^ (7 or more)	Enlargement (6 times, start, end)

Underline, emphasis, inversion, and enlargement can be used in combination.

- Example of a ReceiptLine document

```
{width: *,10; border: line}
^^^"Hamburger"^^^ _w/_ ^^tomato^^, `^^onion`^^,
"^^meat sauce"^^, mayonnaise, "mustard | 3.90
```

- Example of an output image

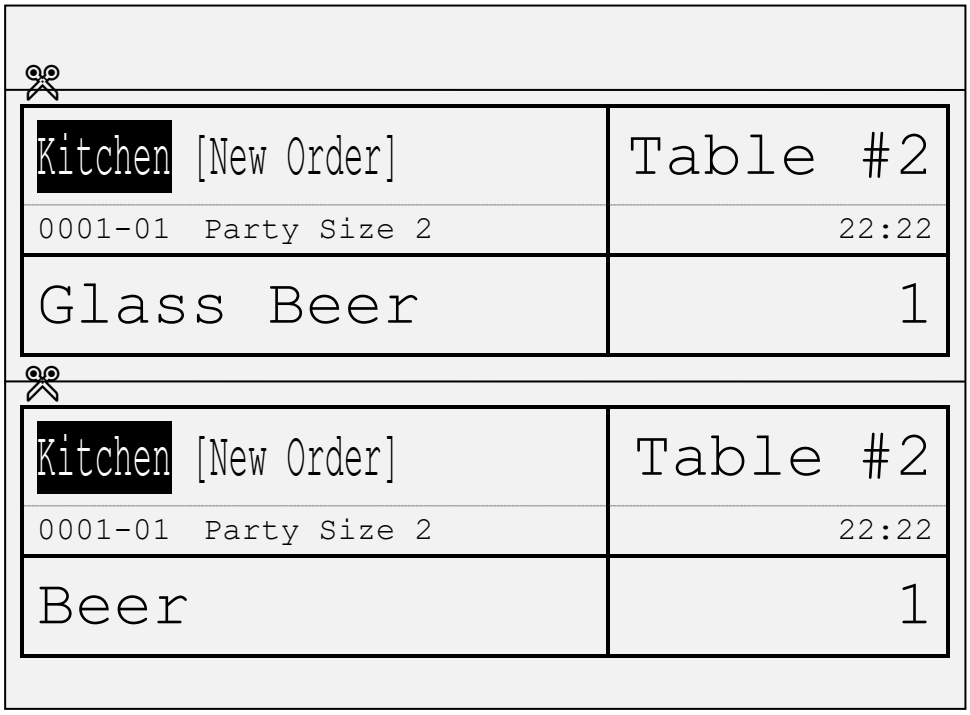
6.3 Document break

Document breaks can be added using special characters.
The types of document breaks are horizontal line and paper cut.

- Example of a ReceiptLine document

```
{width: *,10; border: line}
====================================
^^`Kitchen` [New Order] | ^^Table #2
0001-01  Party Size 2   |     22:22
------------------------------------
^^^Glass Beer           |      ^^^1
====================================
^^`Kitchen` [New Order] | ^^Table #2
0001-01  Party Size 2   |     22:22
------------------------------------
^^^Beer                 |      ^^^1
```

- Example of an output image



7. Property

7.1 Property types

The property is valid for lines with a single column. If the property key is duplicated, the value described later is valid. Ignore unknown properties.

Property	Description
`{ image:` *base64data* `}` `{ i:` *base64data* `}`	Image • The value is base64 data in PNG format • Monochrome and no ancillary chunks are recommended • Actual pixels • The priority of the same line is `image` > `code` > `command`
`{ code:` *textdata* `}` `{ c:` *textdata* `}`	Barcode / 2D code • The value is a character string of the barcode data • Patterns may be different even for the same data • No quiet zone • The priority of the same line is `image` > `code` > `command`
`{ option:` *textdata* `}` `{ o:` *textdata* `}`	Barcode / 2D code options • The values are in no particular order and can be omitted • Options are separated by commas or one or more whitespaces • Case-insensitive • Values of the same type are valid as those described earlier • Ignore invalid values • The setting is also valid after the next line `-----` Barcode specific settings • The values are type, module width, module height, and HRI (human readable interpretation) • The types are shown in the following table • The module width is an integer between "`2`" and

	"4" • The module height is an integer between "24" and "240" • The HRI values are "hri" and "nohri" • "hri" is a barcode with human readable interpretation • Default values are "2", "72", and "nohri" ----- 2D code specific settings • The values are type, cell size, and error correction level • The types are shown in the following table • The cell size is an integer between "3" and "8" • The error correction level of QR Code is "l", "m", "q", "h" • Default values of QR code are "3" and "l"
{ align: *textdata* } { a: *textdata* }	Line alignment settings • The values are "left", "center" and "right" • Case-insensitive • Invalid value is regarded as "center" • The default value is "center" • The setting is also valid after the next line
{ width: *textdata* } { w: *textdata* }	Column width settings • The value is the width of each column specified by the number of characters, or "auto" • Widths are separated by commas or one or more whitespaces • Case-insensitive • Each column width is an integer greater than or equal to "0" or "*" • "*" sets the column width automatically • Invalid values are regarded as "0" • When "auto" is included, set all column widths to "*" • The default value is "auto" • The setting is also valid after the next line
{ border: *textdata* }	Column border settings

{ b: *textdata* }	• The values are "line", "space", and "none" • "nowrap" does not wrap character string • To specify the column spacing by the number of characters, use an integer between "0" and "2" • "space" is the same as "1" • "none" is the same as "0" • Case-insensitive • Invalid values are regarded as "space" • The default value is "space" • The setting is also valid after the next line
{ text: *textdata* } { t: *textdata* }	Text wrap settings • The values are "wrap" and "nowrap" • "nowrap" does not wrap character string • Case-insensitive • Invalid values are regarded as "wrap" • The default value is "wrap" • The setting is also valid after the next line
{ command: *textdata* } { x: *textdata* }	Command • Not recommended due to loss of compatibility • The value is a character string of the device-specific command • The priority of the same line is image > code > command
{ comment: *textdata* } { _: *textdata* }	Comment • The value is any character string

7.2 Barcode / 2D code types

The types that can be specified by the barcode / 2D code setting property and the valid data in the barcode / 2D code property are as follows.

Barcode / 2D code	Description
upc	UPC • UPC-A is an 11-12 digits • UPC-E is a 7-8 digits starting with "0" • Check digit can be omitted • Check digit error is corrected
ean	EAN (JAN)

`jan`	• EAN-13 (JAN-13) is a 12-13 digits • EAN-8 (JAN-8) is a 7-8 digits • Check digit can be omitted • Check digit error is corrected
`code39`	CODE39 • Numbers, alphabets (uppercase), symbols (`-.$+%`), spaces
`itf`	Interleaved 2 of 5 • an even digit
`codabar` `nw7`	Codabar (NW-7) • Start code (`ABCD`) • Numbers, symbols (`-.$:/+`) • Stop code (`ABCD`)
`code93`	CODE93 • Numbers, alphabets, symbols, spaces, control characters (`~\x7f`)
`code128` (Default)	CODE128 • Numbers, alphabets, symbols, spaces, control characters (`~\x7f`)
`qrcode`	QR Code • Numeric, alphanumeric, 8-bit byte, Kanji