

NiceTables Package

Peter Brookes Chambers

Version 1.0, 2023/12/03

Contents

1	Introduction	3
2	Environments and Macros	3
3	Options	4
4	Examples	5
4.1	Default Styling	5
4.2	Number of Header Rows	6
4.3	Header Cell Style	7
4.4	Every Header Cell Style	9
4.5	Cell Style	9
4.6	Every Cell Style	11
4.7	Header Row Colour	12
4.8	Row Colours	12
4.9	Row Separators	14
4.10	Separator Colour	15

4.11	Array Stretch	15
4.12	Column Separation	16
4.13	Actual Use	17

Introduction

The package is loaded as normal with `\usepackage{nicetables}`.

This package defines a new environment `nicetable` which can be used as a near drop-in replacement for the ‘`tabular`’ environment. It defines a number of new options for styling tables and table content. Under the hood, all tables are a `longtable` environment, so they can span multiple pages. The package relies upon:

- `xcolor` for colouring, with the `table` option
- `longtable` for the actual table rendering
- `environ` for the environment definition
- \TeX 3 syntax, for all options and content processing

Note: This package does not, as of this version, work with `multirow` or `multicolumn` cells.

Environments and Macros

The package defines a single new environment, `nicetable`. This environment takes a single optional argument, which is a key-value list of options, and a required argument which is the table specification. The table specification is the same as for the `longtable` environment, for example “`||c|p{3cm}|.`” Within the environment, the syntax is exactly the same as a `tabular` or `longtable` environment, except that `\hline` is **never** required (and will in fact throw an error). Additionally, the final row should not be followed by a `\\`, as this will also throw an error.

The package defines three new macros:

<code>\setarraystretch</code> <code>{float}</code>	Accepts any positive number for the single argument. This macro sets the array stretch, remembering the previous value. This does not respect nesting – calling <code>\setarraystretch{1.5}</code> then <code>\setarraystretch{2.0}</code> results in the ‘remembered’ value being 1.5, not the original value before the first call.
---	---

<code>\resetarraystretch</code>	Accepts no arguments. This macro resets the array stretch to the value it had before the last call to <code>\setarraystretch</code> . If <code>\setarraystretch</code> has never been called, this macro does nothing.
---------------------------------	--

<code>\nicetablesset</code> <code>{key-value list}</code>	Accepts a key-value list of options described in the following section. This macro resets the default values of the options specified, and applies the new values to all following tables created within the current group.
--	---

Options

The `nicetable` environment takes a number of options, which are described below. These options are passed as a set of comma-separated key-value pairs to the `nicetable` environment. The option is given on the left, below which is the default (the value used if the option is not specified), and the initial value (if any) is given in brackets; this is the value used if the option is specified but no value is given.

Note: all options which use colours use the US spelling “color” rather than the UK spelling “colour” which is used in the text of this documentation. The colours used in examples are as generated by the `qoplots` Python package which can be found here: <https://github.com/pbrookeschambers/qoplots> and the Rosé Pine Dawn colour palette which can be found here: <https://rosepinetheme.com/>.

<code>header rows</code> 1	Accepts any non-negative integer. Specifies the number of rows at the top of the table which are formatted as headers.
<code>header cell style</code> {}	Accepts a comma-separated list of commands to apply to the header rows content, cell by cell. This can be empty, a single value, or multiple values. If the number of styles is less than the number of columns, the styles are repeated, allowing for (for example) alternating styles. The final token in each style may take the content of the cell as an argument, for example <code>\textbf</code> is acceptable.
<code>every header cell style</code> <code>\bfseries</code>	Accepts any commands (not a list of commands), to be applied to every header cell, after any cell-specific styles specified in <code>header cell style</code> . The final token in <code>every header cell style</code> may take the cell contents as an argument, after any cell-specific styles have been applied. In effect, the order is <code>\everyheadercellstyle{\headercellstyle{cell contents}}</code> . This allows for styling specified in <code>header cell style</code> to overwrite the styling specified in <code>every header cell style</code> .
<code>cell style</code> {}	Accepts a comma-separated list of commands to apply to the non-header cell content, cell by cell. This can be empty, a single value, or multiple values. If the number of styles is less than the number of columns, the styles are repeated, allowing for (for example) alternating styles. The final token in each style may take the content of the cell as an argument, for example <code>\textbf</code> is acceptable.
<code>every cell style</code> {}	Accepts any commands (not a list of commands), to be applied to every non-header cell, after any cell-specific styles specified in <code>cell style</code> . The final token in <code>every cell style</code> may take the cell contents as an argument, after any cell-specific styles have been applied. In effect, the order is <code>\everycellstyle{\cellstyle{cell contents}}</code> . This allows for styling specified in <code>cell style</code> to overwrite the styling specified in <code>every cell style</code> .
<code>header row color</code> white	Accepts a single colour, which is applied to the background of the header rows.

<code>row colors</code> <code>{}</code>	Accepts exactly zero, one, or two colours separated by a comma. If zero colours are given, no row colours are applied. If one colour is given, this is applied to every row. If two colours are given, these are applied to alternating rows, always starting on the first non-header row with the first colour given.
<code>row separator</code> <code>true (true)</code>	Accepts either <code>true</code> or <code>false</code> . If <code>true</code> , a horizontal line is drawn between every row. If <code>false</code> , no horizontal lines are drawn within the body of the table. If at least one of <code>row separator</code> and <code>header separator</code> are <code>true</code> , a horizontal line is drawn at the top and bottom of the table.
<code>header separator</code> <code>true (true)</code>	Accepts either <code>true</code> or <code>false</code> . If <code>true</code> , a horizontal line is drawn between the header rows and the body of the table. If both this and <code>row separator</code> are <code>false</code> , no horizontal lines are drawn at all. If at least one of <code>row separator</code> and <code>header separator</code> are <code>true</code> , a horizontal line is drawn at the top and bottom of the table.
<code>separator color</code> <code>black</code>	Accepts a single colour, which is applied to the horizontal lines drawn between rows and any vertical lines between columns.
<code>array stretch</code> <code>1.0</code>	Accepts any positive number. This is the factor by which the default line spacing is multiplied, only applied to the table itself.
<code>column separation</code> <code>1em</code>	Accepts any dimension. This is the distance between columns, and is applied to the entire table.

Examples

Default Styling

With no additional options specified, the table is styled as follows:

```

1 \begin{nicetable}{|l|c|l|}
2   Name           & Age & City           \\
3   John Smith     & 28 & New York       \\
4   Alice Johnson  & 35 & Los Angeles    \\
5   Bob Anderson  & 22 & Chicago        \\
6   Eva Williams  & 30 & San Francisco  \\
7   David Brown   & 25 & Miami          \\
8 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Number of Header Rows

The number of header rows can be specified with the `header rows` option. The default is 1, but this can be changed to any non-negative integer. This allows headers to be styled differently to the rest of the table. For example, the following table has two header rows.

```

1 \begin{nicetable}[
2   header rows = 2
3 ]{|l|c|l|}
4   Name           & Age & City           \\
5   John Smith     & 28 & New York       \\
6   Alice Johnson  & 35 & Los Angeles    \\
7   Bob Anderson  & 22 & Chicago        \\
8   Eva Williams  & 30 & San Francisco  \\
9   David Brown   & 25 & Miami          \\
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

With `header rows = 0`, the entire table will be formatted uniformly:

```

1 \begin{nicetable}[
2   header rows = 0
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Header Cell Style

Styling can be applied to the header row(s) either as a whole or per-column. In the example below, only one style is specified so it is applied to all columns.

```

1 \begin{nicetable}[
2   header cell style = {\scshape\bfseries}
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami
10 \end{nicetable}

```

NAME	AGE	CITY
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

In the example below, an individual style is specified for each column.

```

1 \begin{nicetable}[
2   header cell style = {\bfseries, \texttt, \color{Red_2}}
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami         \\
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

If there are more columns than styles, the styles are repeated. For example, the following table has three columns but only two styles specified, so the third column cycles back to the first style.

```

1 \begin{nicetable}[
2   header cell style = {\color{Red}, \color{Blue}},
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami         \\
10 \end{nicetable}

```


Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Every Header Cell Style

Functionally, this is the same as passing a single style to `header cell style`. However, both `every header cell style` and `header cell style` can be specified in the same table, which can be useful for applying some style to the entire table, as well as specific styles to individual columns. For example:

```

1 \begin{nicetable}[
2   every header cell style = \color{Blue},
3   header cell style = {\textbf, \textit, \bfseries\color{Red}}
4 ]{|l|c|l|}
5   Name          & Age & City          \\
6   John Smith    & 28 & New York      \\
7   Alice Johnson & 35 & Los Angeles   \\
8   Bob Anderson & 22 & Chicago       \\
9   Eva Williams  & 30 & San Francisco \\
10  David Brown   & 25 & Miami         \\
11 \end{nicetable}

```

Name	<i>Age</i>	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Cell Style

Styling can be applied to the non-header cells either as a whole or per-column. In the example below, only one style is specified so it is applied to all columns.

```

1 \begin{nicetable}[
2   cell style = {\texttt}
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami         \\
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

In the example below, an individual style is specified for each column.

```

1 \begin{nicetable}[
2   cell style = {\bfseries, \ensuremath, \ttfamily}
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami         \\
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

If there are more columns than styles, the styles are repeated. For example, the following table has three columns but only two styles specified, so the third column cycles back to the first style. In this example specifically, the first style specified is empty, so no styling is applied to the first and third columns.

```

1 \begin{nicetable}[
2   cell style = {, \color{Red}},
3 ]{lllc|ll}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Every Cell Style

Functionally, this is the same as passing a single style to `cell style`. However, both `every cell style` and `cell style` can be specified in the same table, which can be useful for applying some style to the entire table, as well as specific styles to individual columns. For example:

```

1 \begin{nicetable}[
2   every cell style = \color{Blue},
3   cell style = {\texttt, \ensuremath, \color{Red}}
4 ]{lllc|ll}
5   Name          & Age & City          \\
6   John Smith    & 28 & New York      \\
7   Alice Johnson & 35 & Los Angeles   \\
8   Bob Anderson & 22 & Chicago       \\
9   Eva Williams & 30 & San Francisco \\
10  David Brown  & 25 & Miami
11 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Header Row Colour

This option specifies the background colour of the header row(s). The default is white, but any colour can be specified, including `xcolor`-style colour combinations such as `red!50!blue`. For example:

```

1 \begin{nicetable}[
2   header row color = BackgroundColour
3 ]{|l|c|l|}
4   Name           & Age & City           \\
5   John Smith     & 28 & New York       \\
6   Alice Johnson  & 35 & Los Angeles    \\
7   Bob Anderson   & 22 & Chicago        \\
8   Eva Williams   & 30 & San Francisco  \\
9   David Brown    & 25 & Miami          \\
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Row Colours

The option `row colors` specifies the background colours of the non-header rows. With one colour specified, this is applied to every row. For example:

```

1 \begin{nicetable}[
2   row colors = {BackgroundColour}
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

With two colours specified, these are applied to alternating rows, always starting on the first non-header row with the first colour given. For example:

```

1 \begin{nicetable}[
2   row colors = {Blue_4, Red_4}
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

In the example above, the first row coloured will always be blue, regardless of the number of header rows. I.e, the colours are applied to odd and even rows as though any header rows did not exist.

Row Separators

The options `row separator` and `header separator` control the inclusion of horizontal lines within the table. If `row separator` is `true` (which is the default) then a horizontal line will be included between each row (including headers), and at the top and bottom of the table. If it is `false`, then lines will not be drawn between rows.

However, if `header separator` is `true`, then a horizontal line will still be drawn between the last header row and the first non-header row, and at the top and bottom of the table. If it is `false`, then no horizontal lines will be drawn at all. If `row separator` is `true`, then `header separator` is ignored.

For example:

```
1 \begin{nicetable}[
2   row separator      = false,
3   header separator   = true
4 ]{|l|c|l|}
5   Name              & Age & City          \\
6   John Smith        & 28 & New York      \\
7   Alice Johnson     & 35 & Los Angeles  \\
8   Bob Anderson      & 22 & Chicago      \\
9   Eva Williams      & 30 & San Francisco \\
10  David Brown       & 25 & Miami
11 \end{nicetable}
```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

With `header separator` also turned off:

```
1 \begin{nicetable}[
2   row separator      = false,
3   header separator   = false
4 ]{|l|c|l|}
5   Name              & Age & City          \\
6   John Smith        & 28 & New York      \\
7   Alice Johnson     & 35 & Los Angeles  \\
8   Bob Anderson      & 22 & Chicago      \\
9   Eva Williams      & 30 & San Francisco \\
10  David Brown       & 25 & Miami
11 \end{nicetable}
```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Separator Colour

The option `separator color` specifies the colour of the horizontal lines drawn between rows and any vertical lines between columns. For example:

```

1 \begin{nicetable}[
2   separator color = Red
3 ]{|l|c|l|}
4   Name           & Age & City           \\
5   John Smith     & 28 & New York      \\
6   Alice Johnson  & 35 & Los Angeles   \\
7   Bob Anderson   & 22 & Chicago       \\
8   Eva Williams   & 30 & San Francisco \\
9   David Brown    & 25 & Miami
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Array Stretch

Specifying the option `array stretch = {value}` is equivalent to calling `\renewcommand{\arraystretch}{value}` within the table environment. The value before the `nicetable` environment is remembered and restored after the environment ends. For example:

```

1 \begin{nicetable}[
2   array stretch = 2
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami
10 \end{nicetable}

```

Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Column Separation

Specifying the option `column separation = {value}` is equivalent to calling `\setlength{\tabcolsep}{value}` within the table environment. The value before the `nicetable` environment is remembered and restored after the environment ends. For example:

```

1 \begin{nicetable}[
2   column separation = 2em
3 ]{|l|c|l|}
4   Name          & Age & City          \\
5   John Smith    & 28 & New York      \\
6   Alice Johnson & 35 & Los Angeles   \\
7   Bob Anderson & 22 & Chicago       \\
8   Eva Williams & 30 & San Francisco \\
9   David Brown  & 25 & Miami
10 \end{nicetable}

```


Name	Age	City
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

Actual Use

The following uses the `\nicetablesset` macro to set the default options for all tables in the document, specifying a set of reasonable defaults.

```

1 \nicetablesset{
2   array stretch      = 1.5,
3   header cell style   = \color{BackgroundColour}\scshape\bfseries,
4   header row color    = Accent4,
5   row colors          = {Surface5, BackgroundColour},
6   every cell style    = \color{ForegroundColour},
7   separator color     = ForegroundColour,
8   row separator       = false
9 }
10
11 \begin{nicetable}{{lcl}}
12   Name           & Age & City           \\
13   John Smith     & 28 & New York       \\
14   Alice Johnson  & 35 & Los Angeles    \\
15   Bob Anderson   & 22 & Chicago        \\
16   Eva Williams   & 30 & San Francisco  \\
17   David Brown    & 25 & Miami          \\
18 \end{nicetable}

```

NAME	AGE	CITY
John Smith	28	New York
Alice Johnson	35	Los Angeles
Bob Anderson	22	Chicago
Eva Williams	30	San Francisco
David Brown	25	Miami

The cell styling can be quite powerful; for example, it can be combined with the `siunitx` package to render numbers in scientific notation¹:

```

1 \begin{nicetable}[
2   cell style = {, \num, \ensuremath},
3 ]{|llc|}
4   Planet & Mass (\unit{\kilo\gram}) & Number of Moons & \\
5   Mercury & 3.30e23 & 0 & \\
6   Venus & 4.87e24 & 0 & \\
7   Earth & 5.97e24 & 1 & \\
8   Mars & 6.42e23 & 2 & \\
9   Jupiter & 1.90e27 & 92 & \\
10  Saturn & 5.68e26 & 83 & \\
11  Uranus & 8.68e25 & 27 & \\
12  Neptune & 1.02e26 & 14 & \\
13  Pluto & 1.31e22 & 5 & \\
14 \end{nicetable}

```

PLANET	MASS (kg)	NUMBER OF MOONS
Mercury	3.30×10^{23}	0
Venus	4.87×10^{24}	0
Earth	5.97×10^{24}	1
Mars	6.42×10^{23}	2
Jupiter	1.90×10^{27}	92
Saturn	5.68×10^{26}	83
Uranus	8.68×10^{25}	27
Neptune	1.02×10^{26}	14
Pluto	1.31×10^{22}	5

¹Data taken from <https://nssdc.gsfc.nasa.gov/planetary/factsheet/>