1 Introduction

This is a short introduction to yet another table mechanism built in ConTeXt. It is a variant of the so called natural tables but it has a different configuration. Also, the implementation is completely different. The reason for writing it is that in one of our projects we had to write styles for documents that had tables spanning 30 or more pages and apart from memory constraints this is quite a challenge for the other mechanisms, if only because splitting them into successive floats is not possible due to limitations of \TeXX. The extreme table mechanism can handle pretty large tables and split them too. As each cell is basically a \texttt{\textbackslash framed} and as we need to do two passes over the table, this mechanism is not the fastest but it is some two times faster than the natural tables mechanism, and in most cases can be used instead.

2 The structure

The structure of the tables is summarized here. There can be the usual head, body and foot specifications and we also support the optional header in following pages.

\texttt{\textbackslash definextable [tag] | [tag][parent]}
\texttt{\textbackslash setupxtable [settings] | [tag][settings]}
\texttt{\textbackslash startxtable[tag|settings]}
\texttt{\textbackslash startxtablehead|next|body|foot[tag|settings]}
\texttt{\textbackslash startxrowgroup[tag|settings]}
\texttt{\textbackslash startxrow[settings]}
\texttt{\textbackslash startxcellgroup[tag|settings]}
\texttt{\textbackslash startxcell[settings] ... \textbackslash stopxcell}
\texttt{\textbackslash stopxcellgroup}
Contrary to what you might expect, the definition command defines a new set of command. You don’t need to use this in order to set up a new settings environment. Settings and definitions can inherit so you can build a chain of parent-child settings. The grouping is nothing more than a switch to a new set of settings.

3 Direct control

A simple table with just frames is defined as follows:

\startxtable
\startxrow
  \startxcell one \stopxcell
  \startxcell two \stopxcell
\stopxrow
\startxrow
  \startxcell alpha \stopxcell
  \startxcell beta \stopxcell
\stopxrow
\stopxtable

You can pass parameters for tuning the table:

\startxtable[offset=1cm]
\startxrow
  \startxcell one \stopxcell
  \startxcell two \stopxcell
\stopxrow
\startxrow
  \startxcell alpha \stopxcell
  \startxcell beta \stopxcell
\stopxrow
\stopxtable
You can (for as much as they make sense) use the same settings as the `\texttt{framed}` command, as long as you keep in mind that messing with the frame related offsets can have side effects.

### 4 Sets of settings

Instead of directly passing settings you can use a predefined set. Of course you can also combine these methods.

\begin{verbatim}
definextable [myxtable]
definextable [myxtable:important]
definextable [myxtable]
setupxtable [myxtable]
setupxtable [myxtable:important]
setupxtable [myxtable:important]
\end{verbatim}

We can use these settings in table. Although it is not really needed to define a set beforehand (i.e. you can just use the setup command) it is cleaner and more efficient too.

\begin{verbatim}
\startxtable[myxtable]
  \startxrow[foregroundcolor=green]
  \startxcell one \stopxcell
  \startxcell two \stopxcell
  \startxcellgroup[foregroundcolor=blue]
    \startxcell tree \stopxcell
    \startxcell four \stopxcell
  \stopxcellgroup
  \stopxrow
\startxrow
  \startxcell alpha \stopxcell
  \startxcell beta \stopxcell
  \startxcellgroup[myxtable:important]
    \startxcell gamma \stopxcell
    \startxcell delta \stopxcell
  \stopxcellgroup
  \stopxrow
\stopxtable
\end{verbatim}

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>one</td>
<td>two</td>
<td>tree</td>
<td>four</td>
</tr>
<tr>
<td>alpha</td>
<td>beta</td>
<td>gamma</td>
<td>delta</td>
</tr>
</tbody>
</table>
The overhead of (chained) settings is quite acceptable and it helps to keep the source of the table uncluttered from specific settings.

5 Defining

If needed you can define your own encapsulating commands. The following example demonstrates this:

```latex
\definextable[mytable]

We now can use the mytable wrapper:

```latex
\startmytable[height=4cm,width=8cm,align={middle,lohi}]
   \startxrow
      \startxcell test \stopxcell
   \stopxrow
\stopmytable
```

One drawback of using buffers is that they don’t play well in macro definitions. In that case you need to use the following wrapper:

```latex
\starttexdefinition MyTable #1#2#3#4
   \startembeddedxtable
      \startxrow
         \startxcell #1 \stopxcell
         \startxcell #2 \stopxcell
      \stopxrow
      \startxrow
         \startxcell #3 \stopxcell
         \startxcell #4 \stopxcell
      \stopxrow
   \stopembeddedxtable
\stoptexdefinition
```

This macro is used as any other macro with arguments:

```latex
\MyTable{one}{two}{three}{four}
```

<table>
<thead>
<tr>
<th>one</th>
<th>two</th>
</tr>
</thead>
<tbody>
<tr>
<td>three</td>
<td>four</td>
</tr>
</tbody>
</table>
6 Stretching

If you don't give the width of a cell, the widest natural size will be taken. Otherwise the given width applies to the whole column.

\startxtable
\startxrow
\startxcell[width=1cm] one \stopxcell
\startxcell[width=2cm] two \stopxcell
\startxcell[width=3cm] tree \stopxcell
\startxcell[width=4cm] four \stopxcell
\stopxrow
\startxrow
\startxcell alpha \stopxcell
\startxcell beta \stopxcell
\startxcell gamma \stopxcell
\startxcell delta \stopxcell
\stopxrow
\stopxtable

<table>
<thead>
<tr>
<th>one</th>
<th>two</th>
<th>tree</th>
<th>four</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>beta</td>
<td>gamma</td>
<td>delta</td>
</tr>
</tbody>
</table>

You can let the cells stretch so that the whole width of the text area is taken.

\startxtable[option=stretch]
\startxrow
\startxcell[width=1cm] one \stopxcell
\startxcell[width=2cm] two \stopxcell
\startxcell[width=3cm] tree \stopxcell
\startxcell[width=4cm] four \stopxcell
\stopxrow
\startxrow
\startxcell alpha \stopxcell
\startxcell beta \stopxcell
\startxcell gamma \stopxcell
\startxcell delta \stopxcell
\stopxrow
\stopxtable

The available left over space is equally distributed among the cells.

<table>
<thead>
<tr>
<th>one</th>
<th>two</th>
<th>tree</th>
<th>four</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>beta</td>
<td>gamma</td>
<td>delta</td>
</tr>
</tbody>
</table>

An alternative is to distribute the space proportionally:

\startxtable[option={stretch,width}]
\startxrow
\stopxtable
\begin{table}
\begin{tabular}{|l|l|l|l|}
\hline
one & two & tree & four \\
\hline
alpha & beta & gamma & delta \\
\hline
\end{tabular}
\end{table}

Just to stress the difference we show both alongside now:

\begin{table}
\begin{tabular}{|l|l|l|l|}
\hline
one & two & tree & four \\
\hline
alpha & beta & gamma & delta \\
\hline
\end{tabular}
\end{table}

You can specify the width of a cell with each cell but need to keep into mind that that value is then used for the whole column:

\begin{table}
\begin{tabular}{|l|l|l|l|}
\hline
one & two & tree & four \\
\hline
alpha & beta & gamma & delta \\
\hline
\end{tabular}
\end{table}

You can enforce that larger columns win via the option parameter:

\begin{verbatim}
\startxtable[option=\max]
\end{verbatim}
7 Spacing

It is possible to separate the cells by horizontal and/or vertical space. As an example we create a setup.

\setupxtablenew[myztable]
\[option=stretch, 
  foregroundcolor=blue, 
  columndistance=10pt, 
  leftmargindistance=20pt, 
  rightmargindistance=30pt\]

You can use the textwidth parameter to set a specific maximum width. We now apply the previous settings to an extreme table:

\startxtablenew[myztable]
\startxrow
\startxcell[width=1cm] one \stopxcell
\startxcell[width=2cm,distance=5pt] two \stopxcell
\startxcell[width=3cm] tree \stopxcell
\startxcell[width=4cm] four \stopxcell
\stopxrow
\startxrow
\startxcell[width=1cm] alpha \stopxcell
\startxcell[width=2cm] beta \stopxcell
\startxcell[width=3cm] gamma \stopxcell
\startxcell[width=4cm] delta \stopxcell
\stopxrow
\stopxtablenew

As you can see here, we can still locally overload the settings but keep in mind that these apply to the whole column then, not to the specific cell.
Vertical spacing is (currently) setup differently, i.e. as an argument to the \texttt{\textbackslash blank} command.

```
\begin{xtabular}{|c|c|c|c|}
\hline
one & two & tree & four \\
\hline
alpha & beta & gamma & delta \\
\hline
\end{xtabular}
```

Specifying spacing this way improves consistency with the rest of the document spacing.

```
\begin{xtabular}{|c|c|}
\hline
one & two + three \\
\hline
\end{xtabular}
```

\section{8 Spanning}

Of course we can span cells horizontally as well as vertically. Future versions might provide more advanced options but the basics work okay.

```
\begin{xtabular}{|c|c|c|c|c|}
\hline
one & two + three & four & five \\
\hline
alpha + beta + gamma & delta + epsilon \\
\hline
\end{xtabular}
```
The next example gives a span in two directions:

\begin{table}
\begin{tabular}{cccc}
alpha 1 & beta 1 & gamma 1 & delta 1 \\
alpha 2 & whatever & & delta 2 \\
alpha 3 & & & delta 3 \\
alpha 4 & beta 4 & gamma 4 & delta 4 \\
\end{tabular}
\end{table}

Of course, spanning is always a compromise but the best fit found by this mechanism takes natural width, given width and available space into account.

\section{Partitioning}

You can partition a table as follows:

\begin{table}
\begin{tabular}{cccc}
\hline
head one & head two & head tree & head four \\
\hline
\end{tabular}
\end{table}
There can be multiple such partitions and they are collected in head, next, body and foot groups. Normally the header ends up at the beginning and the footer at the end. When a table is split, the first page gets the header and the following pages the next one.

You can let headers and footers be repeated by setting the \texttt{header} and/or \texttt{footer} parameters to \texttt{repeat}.

\setupxtable
\[split=yes,\]
\[header=repeat,\]
\[footer=repeat\]

The table can be flushed in the running text but also in successive floats. Given that the table is in a buffer:

\texttt{\placetable[here,split]{A big table.}\{\getbuffer}}

When you specify \texttt{split} as \texttt{yes} the caption is taken into account when calculating the available space.

There are actually three different split methods. The \texttt{yes} option works in text mode as well as in floats, but in text mode no headers and footers get repeated. If you want that feature in a text flush you have to set \texttt{split} to \texttt{repeat} as well.

You can keep rows together by passing a \texttt{samepage} directive. This parameter can get the values \texttt{before}, \texttt{after} and \texttt{both}.

\texttt{\startxtable[split=yes]}
\texttt{\startxrow \startxcell \tttf .01. \stopxcell \stopxrow}
\texttt{\startxrow \startxcell \tttf .... \stopxcell \stopxrow}
On the average a table will come out okay but you need to keep in mind that when (complex) spans are used the results can be less than optimal. However, as normally one pays attention to creating tables, the amount of control provided often makes it possible to get what you want.

In the following situations, the first cell width is determined by the span. It is possible to make a more clever analyzer but we need to keep in mind that in the same column there can be entries that span a different amount of columns. Not only would that be inefficient but it would also be rather unpredictable unless you know exactly what happens deep down. The following two examples demonstrate default behaviour.

\begin{verbatim}
\begin{table}
\begin{tabular}{|c|c|c|}
\hline
1/2/3 \\
\hline
\end{tabular}
\end{table}
\end{verbatim}

\begin{verbatim}
\begin{table}
\begin{tabular}{|c|c|c|}
\hline
1 & 2 & 3 \\
\hline
\end{tabular}
\end{table}
\end{verbatim}
In practice you will set the width of the columns, as in:

```
\startxtable
  \startxrow
    \startxcell[nx=3]
      1/2/3
    \stopxcell
  \stopxrow
  \startxrow
    \startxcell[width=\dimexpr\textwidth/3] 1 \stopxcell
    \startxcell[width=\dimexpr\textwidth/3] 2 \stopxcell
    \startxcell[width=\dimexpr\textwidth/3] 3 \stopxcell
  \stopxrow
\stopxtable
```

<table>
<thead>
<tr>
<th>1/2/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

But, if you want you can control the enforced width by setting an option:

```
\startxtable
  \startxrow
    \startxcell[nx=3,option=tight]
      1/2/3
    \stopxcell
  \stopxrow
  \startxrow
    \startxcell 1 \stopxcell
    \startxcell 2 \stopxcell
    \startxcell 3 \stopxcell
  \stopxrow
\stopxtable
```

<table>
<thead>
<tr>
<th>1/2/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>
11 Nesting

Extreme tables can be nested but you need to keep an eye on inheritance here as the inner table uses the settings from the encapsulating cell. The widths and heights of the inner table default to fit. We could cook up a more complex nesting model but this one is easy to follow.
Here we just manipulate the offset a bit.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>one</td>
<td>two</td>
<td>two</td>
</tr>
<tr>
<td>alpha</td>
<td>beta</td>
<td></td>
</tr>
<tr>
<td>alpha</td>
<td></td>
<td>one two</td>
</tr>
<tr>
<td></td>
<td>alpha beta</td>
<td></td>
</tr>
</tbody>
</table>

**12 Buffers**

When you don’t want to clutter your document source too much buffers can be if help:

\begin{verbatim}
\startbuffer[test]
\startxtable
\startxrow
\startxcell[width=1cm] one \stopxcell
\startxcell[width=2cm] two \stopxcell
\stopxrow
\startxrow
\startxcell alpha \stopxcell
\startxcell beta \stopxcell
\stopxrow
\stopxtable
\stopbuffer
\end{verbatim}

One way of getting this table typeset is to say:

\begin{verbatim}
\getbuffer[test]
\end{verbatim}

Normally this is quite okay. However, internally extreme tables become also buffers. If you don’t like the overhead of this double buffering you can use the following command:

\begin{verbatim}
\processxtablebuffer[test]
\end{verbatim}
This can save you some memory and runtime, but don’t expect miracles. Also, this way of processing does not support nested tables (unless {} is used).

13 XML

The following example demonstrates that we can use this mechanism in XML too. The example was provided by Thomas Schmitz. First we show how a table looks like in XML:

```xml
<table>
  <tablerow>
    <tablecell>
      One
    </tablecell>
    <tablecell>
      Two
    </tablecell>
  </tablerow>
  <tablerow>
    <tablecell>
      <b>Three</b>
    </tablecell>
    <tablecell>
      Four
    </tablecell>
  </tablerow>
</table>
```

We need to map these elements to setups:

```latex
\startxmlsetups xml:testsetups
  \xmlsetsetup{main}{b|table|tablerow|tablecell}{xml:*}
\stopxmlsetups

\xmlregistersetup{xml:testsetups}

The setups themselves are rather simple as we don’t capture any attributes.

```latex
\startxmlsetups xml:b
  \bold{\xmlflush{#1}}
\stopxmlsetups

\startxmlsetups xml:table
  \startembeddedxtable
    \xmlflush{#1}
  \stopembeddedxtable
\stopxmlsetups

\startxmlsetups xml:tablerow
  \xrow
    \xmlflush{#1}
  \stopxrow
```
We now process the example. Of course it will also work for files.

\xmlprocessbuffer{main}{test}{}

The result is:

<table>
<thead>
<tr>
<th>One</th>
<th>Two</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three</td>
<td>Four</td>
</tr>
</tbody>
</table>

14 Natural tables

For the impatient a small additional module is provided that remaps the natural table commands onto extreme tables:

\usemodule[ntb-to-xtb]

After that:

\bTABLE
  \bTR
    \bTD[background=color,backgroundcolor=red] one \eTD
    \bTD[width=2cm] two \eTD
  \eTR
  \bTR
    \bTD[width=5cm] alpha \eTD
    \bTD[background=color,backgroundcolor=yellow] beta \eTD
  \eTR
\eTABLE

Will come out as:

<table>
<thead>
<tr>
<th>one</th>
<th>two</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>beta</td>
</tr>
</tbody>
</table>

You can restore and remap the commands with the following helpers:

\restoreTABLEfromxtable
\mapTABLEtoxtable

Of course not all functionality of the natural tables maps onto similar functionality of extreme tables, but on the average the result will look rather similar.
15 Alignment

There is limited support for aligning numbers in columns. When you flush right and have a fixed number of digits it’s not a problem, but otherwise it might come in handy. In any case, you might want to use the following as part of your style:

\addfeature{tabularnumbers}

The basic alignment mechanism can be demonstrated with the following examples:

\startcharacteralign[], % or \[character={,}]
   \checkcharacteralign{123.456,78} \par
   \checkcharacteralign  {456} \par
   \checkcharacteralign  {23.456} \par
   \checkcharacteralign  {78,9} \par
   \checkcharacteralign  {78} \par
\stopcharacteralign

\begin{verbatim}
  123.456,78
  456
  23.456
  78,9
  78
\end{verbatim}

\startcharacteralign[leftsample=123.456,rightsample=00,character={,}]
   \checkcharacteralign{123.456,78} \par
   \checkcharacteralign  {456} \par
   \checkcharacteralign  {23.456} \par
   \checkcharacteralign  {78,9} \par
   \checkcharacteralign  {78} \par
\stopcharacteralign

\begin{verbatim}
  123.456,78
  456
  23.456
  78,9
  78
\end{verbatim}

\startcharacteralign[leftwidth=100pt,rightwidth=30pt]
   \checkcharacteralign{123.456,78} \par
   \checkcharacteralign  {456} \par
   \checkcharacteralign  {23.456} \par
   \checkcharacteralign  {78,9} \par
   \checkcharacteralign  {78} \par
\stopcharacteralign

\begin{verbatim}
  123.456,78
  456
  23.456
  78,9
  78
\end{verbatim}
Samples win over width settings. When no sample or width is given a two pass analyzer kicks in. This is not the case in the following examples.

\begin{table}[h]
\centering
\begin{tabular}{|c|}
\hline
1
\hline
1.1
\hline
11.11
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|}
\hline
123.456,78 x
\hline
456 x
\hline
23.456 x
\hline
78,9 x
\hline
78 x
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|}
\hline
123.456,78
\hline
456
\hline
23.456
\hline
78,9
\hline
78
\hline
\end{tabular}
\end{table}
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>123.456</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>78,9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
16 Depth

Take the following example. Here we have some text that, when typeset narrow, will have a last line with no (or, depending on the font, hardly) depth.

\startxtable[frame=on]
\startxrow
\startxcell[width=3cm]
here filled with some words to force a rather long line
\stopxcell
\stopxrow
\startxrow
\startxcell
here filled with some words to force a rather long line
\stopxcell
\stopxrow
\startxrow
\startxcell
here filled with some words to force a rather long line
\stopxcell
\stopxrow
\stopxtable

The problem is that TeX doesn’t add depth that that last line and adding it automatically can be done in grid mode but one doesn’t always want that. Some mechanisms add so called struts automatically. As the table cells are framed boxes indeed that happens here, unless you add an empty line (as in the last row). When you embed
Don’t count on all mechanisms used inside a cell to provide this but with itemize you’re lucky.
17 Swapping

The next two examples demonstrate a feature added for Taco, who wanted to input vertical instead of horizontal. It’s a bit of a hack but it works okay for simple cases. We assume these settings:

\setupxtable[one] [foregroundcolor=darkred]
\setupxtable[two] [foregroundcolor=darkgreen]
\setupxtable[three][foregroundcolor=darkblue]
\setupxtable[four] [foregroundcolor=darkorange]

The first example has no spans:

\startxtable[frame=on]
  \startxcolumn[one]
  \startxcell A1 \stopxcell
  \startxcell B1 \stopxcell
  \startxcell C1 \stopxcell
  \startxcell D1 \stopxcell
  \startxcell E1 \stopxcell
  \stopxcolumn
  \startxcolumn[two]
  \startxcell A2 \stopxcell
  \startxcell B2 \stopxcell
  \startxcell C2 \stopxcell
  \startxcell D2 \stopxcell
  \startxcell E2 \stopxcell
  \stopxcolumn
  \startxcolumn[three]
  \startxcell A3 \stopxcell
  \startxcell B3 \stopxcell
  \stopxcolumn
while the second one has:

\startxtable[frame=on]
\startxcolumn[one]
\startxcell A \stopxcell
\startxcell B \stopxcell
\startxcell C \stopxcell
\startxcell D \stopxcell
\startxcell E \stopxcell
\stopxcolumn
\startxcolumn[two]
\startxcell A \stopxcell
\startxcell[ny=2] BC \stopxcell
\startxcell[ny=2] DE \stopxcell
\stopxcolumn
\startxcolumn[three]
\startxcell A \stopxcell
\startxcell[ny=4] BCDE \stopxcell
\stopxcolumn
\startxcolumn[four]
\startxcell A \stopxcell
\startxcell B \stopxcell
\startxcell[ny=2] CD \stopxcell
\startxcell E \stopxcell
\stopxcolumn
\stopxtable

The results are shown in figure 1.

18 Examples

On the following pages we show some examples of (experimental) features. For this we will use the usual quotes from Ward, Tufte and Davis etc. that you can find in the distribution.
<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th></th>
<th>A</th>
<th>A</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td></td>
<td>B</td>
<td>BC</td>
<td>BCDE</td>
</tr>
<tr>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>C4</td>
<td></td>
<td>C</td>
<td></td>
<td>CD</td>
</tr>
<tr>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td></td>
<td>D</td>
<td>DE</td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>E2</td>
<td>E3</td>
<td>E4</td>
<td></td>
<td>E</td>
<td></td>
<td>E</td>
</tr>
</tbody>
</table>

**Figure 1**  Entering columns instead of rows.
<table>
<thead>
<tr>
<th>The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>We thrive in information-thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.</td>
</tr>
<tr>
<td>This nation, turning 100 years old, had no Odyssey, no St. George slaying the dragon, no Prometheus. The emerging American genius for making a lot of money was a poor substitute for King Arthur and his knights (although the Horatio Alger myth of rags to riches was good for a lot of mileage). Without a mythology and set of ancient heroes to call its own, America had to manufacture its heroes. So the mythmaking machinery of nineteenth-century American media created a suitable heroic archetype in the cowboys of the Wild West. The image was of the undaunted cattle drivers living a life of reckless individualism, braving the elements, staving off brutal Indian attacks. Or of heroic lawmen dueling with six-guns in the streets at high noon. This artificial Wild West became America’s Iliad.</td>
</tr>
</tbody>
</table>
We thrive in information-thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

This nation, turning 100 years old, had no Odyssey, no St. George slaying the dragon, no Prometheus. The emerging American genius for making a lot of money was a poor substitute for King Arthur and his knights (although the Horatio Alger myth of rags to riches was good for a lot of mileage). Without a mythology and set of ancient heroes to call its own, America had to manufacture its heroes. So the mythmaking machinery of nineteenth-century American media created a suitable heroic archetype in the cowboys of the Wild West. The image was of the undaunted cattle drivers living a life of reckless individualism, braving the elements, staving off brutal Indian attacks. Or of heroic lawmen dueling with six-guns in the streets at high noon. This artificial Wild West became America’s Iliad.
\startxtable[option=stretch]
  \startxrow
  \startxcell bla \stopxcell
  \startxcell bla bla \stopxcell
  \startxcell bla bla bla \stopxcell
  \stopxrow
\stopxtable

<table>
<thead>
<tr>
<th>bla</th>
<th>bla bla</th>
<th>bla bla bla</th>
</tr>
</thead>
</table>

\startxtable[option={stretch, width}]
  \startxrow
  \startxcell bla \stopxcell
  \startxcell bla bla \stopxcell
  \startxcell bla bla bla \stopxcell
  \stopxrow
\stopxtable

<table>
<thead>
<tr>
<th>bla</th>
<th>bla bla</th>
<th>bla bla bla</th>
</tr>
</thead>
</table>
\setupxtable[suffix][align=middle,foregroundcolor=red]
\setupxtable[blabla][foregroundstyle=slanted]
\setupxtable[crap] [foregroundcolor=blue]
\setupxtable[bold] [crap][foregroundstyle=bold]
\startxtable % [frame=off]
  \startxtablehead
    \startxrow[bold]
      \startxcell[suffix] head a \stopxcell
      \startxcell[blabla] head b \stopxcell
      \startxcell head c \stopxcell
    \stopxrow
  \stopxtablehead
  \startxtablebody
    \startxrow
      \startxcell[suffix][ny=2] cell a 1 \stopxcell
      \startxcell cell b 1 \stopxcell
      \startxcell cell c 1 \stopxcell
    \stopxrow
    \startxrow
      \startxcell cell b 2 \stopxcell
      \startxcell cell c 2 \stopxcell
    \stopxrow
    \startxrow
      \startxcell[suffix] cell a 3 \stopxcell
      \startxcell cell b 3 \stopxcell
      \startxcell cell c 3 \stopxcell
    \stopxrow
    \startxrow
      \startxcell[suffix] cell a 4 \stopxcell
      \startxcell cell b 4 \stopxcell
      \startxcell cell c 4 \stopxcell
    \stopxrow
    \startxrow
      \startxcell[suffix] cell a 5 \stopxcell
      \startxcell cell b 5 \stopxcell
      \startxcell cell c 5 \stopxcell
  \stopxtablebody
\stopxtable

<table>
<thead>
<tr>
<th>head a</th>
<th>head b</th>
<th>head c</th>
</tr>
</thead>
<tbody>
<tr>
<td>cell a 1</td>
<td>cell b 1</td>
<td>cell c 1</td>
</tr>
<tr>
<td></td>
<td>cell b 2</td>
<td>cell c 2</td>
</tr>
<tr>
<td>cell a 3</td>
<td>cell b 3</td>
<td>cell c 3</td>
</tr>
<tr>
<td>cell a 4</td>
<td>cell b 4</td>
<td>cell c 4</td>
</tr>
<tr>
<td>cell a 5</td>
<td>cell b 5</td>
<td>cell c 5</td>
</tr>
</tbody>
</table>
\begin{table}[option=stretch]
\begin{tabular}{|l|l|l|}
\hline
first cell & 101 & 102 & 103 \\
\hline
2\textsuperscript{nd} cell & a & b & c \\
\hline
\end{tabular}
\end{table}
\startxtable
\startxrow
    \startxcell[demo][nx=4] 1 / 2 / 3 / 4 \stopxcell
\stopxrow
\startxrow
    \startxcell[demo][nx=3] 1 / 2 / 3 \stopxcell
    \startxcell 4 \stopxcell
\stopxrow
\startxrow
    \startxcell 1 \stopxcell
    \startxcell[demo][nx=3] 2 / 3 / 4 \stopxcell
\stopxrow
\startxrow
    \startxcell[demo][nx=2] 1 / 2 \stopxcell
    \startxcell 3 \stopxcell
    \startxcell 4 \stopxcell
\stopxrow
\startxrow
    \startxcell 1 \stopxcell
    \startxcell[demo][nx=2] 2 / 3 \stopxcell
    \startxcell 4 \stopxcell
\stopxrow
\startxrow
    \startxcell 1 \stopxcell
    \startxcell 2 \stopxcell
    \startxcell[demo][nx=2] 3 / 4 \stopxcell
\stopxrow
\startxrow
    \startxcell[demo][nx=2] 1 / 2 \stopxcell
    \startxcell[demo][nx=2] 3 / 4 \stopxcell
\stopxrow
\startxrow
    \startxcell 1 \stopxcell
    \startxcell 2 \stopxcell
    \startxcell 3 \stopxcell
    \startxcell 4 \stopxcell
\stopxrow
\stopxtable
\setupxtable[\textwidth] [option=tight]

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
1 & 2 & 3/4 \\
\hline
1 & 2 & 3 \\
\hline
1 & 2/3 & 4 \\
\hline
1 & 2 & 3/4 \\
\hline
1 & 2/3 & 4 \\
\hline
1 & 2 & 3/4 \\
\hline
1 & 2 & 3 & 4 \\
\hline
\end{tabular}
\end{table}

\setupxtable[\textwidth] [option=]

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
1/2/3/4 \\
\hline
1/2/3 & 4 \\
\hline
1 & 2/3/4 \\
\hline
1/2 & 3 & 4 \\
\hline
1 & 2/3 & 4 \\
\hline
1 & 2 & 3/4 \\
\hline
1/2 & 3/4 \\
\hline
1 & 2 & 3 & 4 \\
\hline
\end{tabular}
\end{table}
\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
(1,3) & (1,1) \\
(1,1) & (1,1) \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
(1,2) & (1,1) \\
(1,1) & (1,1) \\
(1,1) & (2,1) \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
(1,2) & (1,1) \\
(1,1) & (1,1) \\
(2,1) & \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
(1,2) & (1,1) \\
(1,1) & (1,1) \\
(1,1) & (3,1) \\
\hline
\end{tabular}
\end{table}
\begin{center}
\begin{tabular}{l}
\hline
(1,2) & (1,1) & (1,1) & (1,1) \\
\hline
 & (3,1) \\
 & (4,1) \\
\hline
\end{tabular}
\end{center}
19 Colofon

author Hans Hagen, PRAGMA ADE, Hasselt NL
version October 3, 2018
website www pragma ade nl - www contextgarden net
copyright cc by sa nc