## Shared String Table

A [workbook](workbook.docx) may contain thousands of cells containing string (non-numeric) data. Furthermore this data is very likely to be repeated across many rows or columns. The goal of implementing a single string [table](table.docx) that is shared across the [workbook](workbook.docx) is to improve performance in opening and saving the file by only reading and writing the repetitive information once.

Consider for example a [workbook](workbook.docx) summarizing information for cities within various countries. There may be a column for the name of the country, a column for the name of each city in that country, and a column containing the data for each city. In this case the country name is repetitive, being duplicated in many cells. In many cases the repetition is extensive, and a tremendous savings is realized by making use of a shared string [table](table.docx) when saving the workbook. When displaying text in the spreadsheet, the [cell](cell.docx) [table](table.docx) will just contain an index into the string [table](table.docx) as the value of a [cell](cell.docx), instead of the full string.

The shared string [table](table.docx) contains all the necessary information for displaying the string: the text, formatting properties, and phonetic properties (for East Asian languages).

Most strings in a [workbook](workbook.docx) have formatting applied at the [cell](cell.docx) level, that is, the entire string in the [cell](cell.docx) has the same formatting applied. In these cases, the formatting for the [cell](cell.docx) is stored in the styles part, and the string for the [cell](cell.docx) can be stored in the shared strings table. In this case, the strings stored in the shared strings [table](table.docx) are very simple text elements, and the following xml illustrates the example.

[Example:

<[sst](sst.docx) xmlns=http://schemas.openxmlformats.org/spreadsheetml/2006/5/[main](main.docx)
 count="8" uniqueCount="4">
 <[si](si.docx)>
 <[t](t.docx)>United States</[t](t.docx)>
 </[si](si.docx)>

 <[si](si.docx)>
 <[t](t.docx)>Seattle</[t](t.docx)>
 </[si](si.docx)>

 <[si](si.docx)>
 <[t](t.docx)>Denver</[t](t.docx)>
 </[si](si.docx)>

 <[si](si.docx)>
 <[t](t.docx)>New York</[t](t.docx)>
 </[si](si.docx)>
</[sst](sst.docx)>

end example]

In the above example we can see that the string [table](table.docx) is just a collection of string [items](items.docx) that consist of simple text elements. Note that any numeric data in the [workbook](workbook.docx) is not shown in the shared string table.

Some strings in the [workbook](workbook.docx) may have formatting applied at a level that is more granular than the [cell](cell.docx) level. For instance, specific characters within the string may be bolded, have coloring, italicizing, etc. In these cases, the formatting is stored along with the text in the string [table](table.docx), and is treated as a unique entry in the table. The following xml illustrates this.

[Example:

<[sst](sst.docx) xmlns=http://schemas.openxmlformats.org/spreadsheetml/2006/5/[main](main.docx)
 count="8" uniqueCount="4">
 <[si](si.docx)>
 <[r](r.docx)>
 <[t](t.docx) xml:space="preserve">United </[t](t.docx)>
 </[r](r.docx)>

 <[r](r.docx)>
 <[rPr](rPr.docx)>
 <[sz](sz.docx) [val](val.docx)="11"/>
 <[color](color.docx) rgb="FFFF0000"/>
 <[rFont](rFont.docx) [val](val.docx)="Calibri"/>
 <[family](family.docx) [val](val.docx)="2"/>
 <[scheme](scheme.docx) [val](val.docx)="minor"/>
 </[rPr](rPr.docx)>

 <[t](t.docx)>States</[t](t.docx)>
 </[r](r.docx)>
 </[si](si.docx)>

 <[si](si.docx)>
 <[t](t.docx)>Seattle</[t](t.docx)>
 </[si](si.docx)>

 <[si](si.docx)>
 <[t](t.docx)>Denver</[t](t.docx)>
 </[si](si.docx)>

 <[si](si.docx)>
 <[t](t.docx)>New York</[t](t.docx)>
 </[si](si.docx)>
</[sst](sst.docx)>

In the above example you can see that this time, the text "United States" has specific, colored, formatting applied to the text, "States." end example]