

Surpass and XLWT Manual

Ana Nelson

March 18, 2009

1 Hello World

Let's do a minimal "Hello World" script. We'll need to take care of any imports, initialize a Workbook object, create a Worksheet within the workbook, then write some text. Here's how.

1.1 Surpass

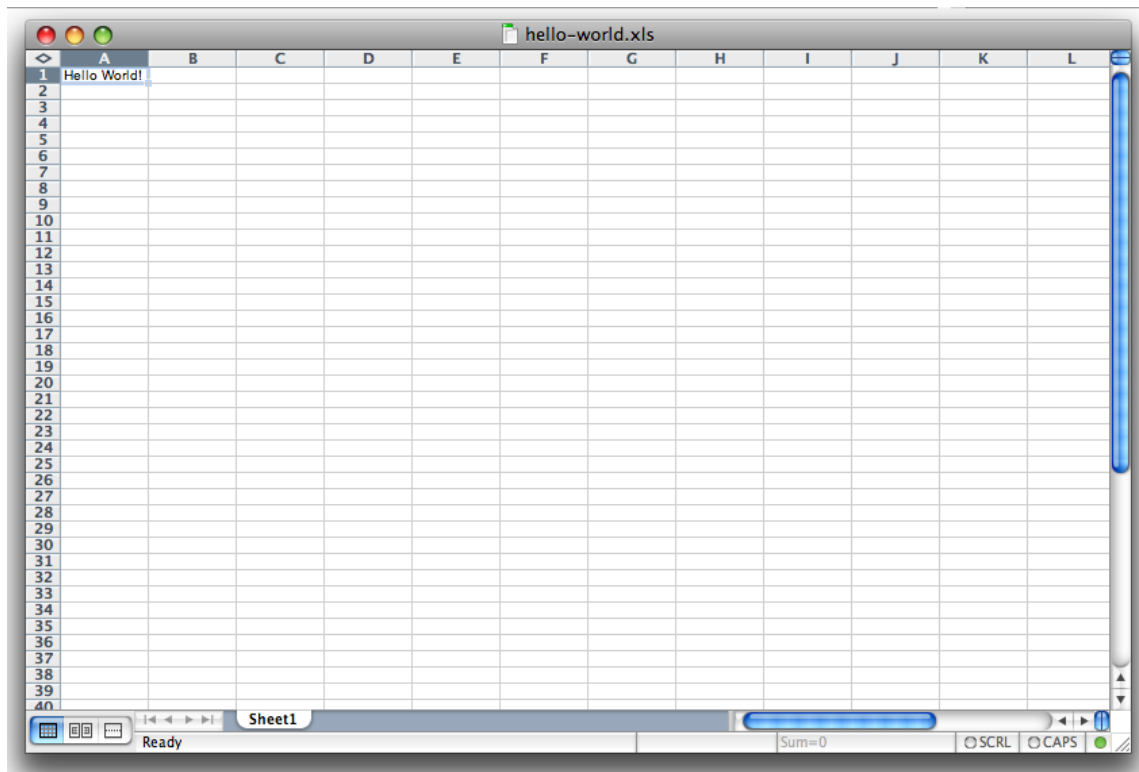
```
1 require 'rubygems'
2 require 'surpass'
3
4 book = Workbook.new
5 sheet = book.add_sheet
6
7 sheet.write(0, 0, "Hello World!")
8
9 book.save("output/examples/hello-world.xls")
```

1.2 XLWT

```
1 from xlwt import *
2
3 book = Workbook()
4 sheet = book.add_sheet("Sheet1")
5
6 sheet.write(0, 0, "Hello World!")
7
8 book.save("output/examples/hello-world-python.xls")
```

1.3 Result

And, here's how it looks.



2 Formatting

The `StyleFormat` class is a wrapper for the various types of formatting you can apply to a cell. `StyleFormat` has attributes:

- `number_format_string`
- `font`
- `alignment`
- `borders`
- `pattern`
- `protection`

Each of these attributes has a corresponding class, and you can look in `lib/formatting.rb` for the source.

There are two basic ways to set formatting options. You can pass a hash with formatting options when you initialize a new `StyleFormat` instance, or you can set individual attributes of the formatting classes. You can combine both approaches. Both of these are demonstrated in the examples in this section.

2.1 Specifying Colours

Here is a list of available built-in colours:

- `aqua`
- `black`
- `blue`

- blue_grey
- bright_green
- brown
- coral
- cornflower_blue
- cyan
- dark_blue
- dark_green
- dark_red
- dark_teal
- dark_yellow
- fuchsia
- gold
- gray
- green
- grey
- grey_25_percent
- grey_40_percent
- grey_50_percent
- grey_80_percent
- indigo
- lavender
- lemon_chiffon
- light_blue
- light_cornflower_blue
- light_green
- light_orange
- light_turquoise
- light_yellow
- lime
- magenta
- maroon
- navy
- olive_green

- orange
- orchid
- pale_blue
- pink
- plum
- purple
- red
- rose
- royal_blue
- sea_green
- silver
- sky_blue
- tan
- teal
- turquoise
- violet
- white
- yellow

You can refer to any of these colours by name.
At this time, there is no support for custom colours.

2.2 Border Formats

Here is a list of available border line types:

- none
- thin
- medium
- dashed
- dotted
- thick
- double
- hair
- medium-dashed
- thin-dash-dotted
- medium-dash-dotted
- thin-dash-dot-dotted
- medium-dash-dot-dotted
- slanted-medium-dash-dotted

2.3 Surpass

```
1 require 'rubygems'
2 require 'surpass'
3
4 book = Workbook.new('output/examples/formatting.xls') # You can pass a filename here too.
5 sheet = book.add_sheet("Demo Worksheet") # You can name your worksheets.
6
7 # Let's set up some formatting.
8
9 # Remember to use Excel-style formatting directives, not sprintf.
10 date_format = StyleFormat.new(:number_format_string => "DDDD DD MMM YYYY")
11
12 fancy_format = StyleFormat.new(
13   :font_name => 'Times New Roman',
14   :font_colour => 'green',
15   :font_italic => true
16 )
17
18 sheet.write(0, 0, "Hello World!", fancy_format)
19 sheet.write(0, 1, Date.today, date_format)
20
21 # You can also set up formatting by passing attributes directly to the constituents of StyleFormat
22
23 # Font colours.
24 Formatting::COLOURS.keys.each_with_index do |c, i|
25   format = StyleFormat.new
26   format.font.name = 'Verdana'
27   format.font.color = c
28   format.font.size = i + 5
29   sheet.write(i, 5, c, format)
30 end
31
32 # Font underlining.
33
34 [:none, :single, :single_accounting, :double, :double_accounting, nil, true, false].each_with_index
35   format = StyleFormat.new
36   format.font.underline = u
37   sheet.write(i, 7, u.to_s, format)
38 end
39
40 # Font bold, italic, strikethrough, outline are simple booleans.
41 [:bold, :italic, :struck_out, :outline].each_with_index do |s, i|
42   attribute = ("font_" + s.to_s).to_sym
43   sheet.write(i, 8, s.to_s, StyleFormat.new(attribute => true))
44 end
45
46 # Cell alignment.
47 sheet.write(15, 2, "top left", :text_align => 'top left',
48   :border_top => 'pink',
49   :border_left => 'pink'
50 )
51 sheet.write(15, 3, "top center", :text_align => 'top center')
52 sheet.write(15, 4, "top right", :text_align => 'top right')
53 sheet.write(16, 2, "bottom left", :text_align => 'bottom left')
54 sheet.write(16, 3, "bottom centre", :text_align => 'bottom centre')
```

```

55 sheet.write(16, 4, "bottom right", :text_align => 'bottom right',
56   :border_bottom => 'pink',
57   :border_right => 'pink'
58 )
59
60
61 # Borders
62 sheet.write(3, 1, "borders",
63   :border_right => 'medium blue',
64   :border_left => 'yellow', # thin by default
65   :border_top => 'dotted purple',
66   :border_bottom => 'dashed' # black by default
67 )
68
69 # Or the hash-free option.
70 crazy_border_format = StyleFormat.new
71 crazy_border_format.borders.all = 'slanted-medium-dash-dotted grey'
72
73 sheet.write(5, 1, "borders", crazy_border_format)
74
75 sheet.write(7, 1, "fill", :fill_color => 'yellow')
76
77 book.save

```

And, here's how it looks.

