



DOCUMENTATION (Source)

=====

Project: SystemPath

Folder: g:/\_PROJECTS/Ruby/SystemPath/src

Status: 04.12.2013

-----

SystemPath.ini	04.12.2013 14:12	---
SystemPath.rb	04.12.2013 15:15	# SystemPath

```
1: ---
2: paths:
3: - c:\AVEVA\Review12.1.1
4: - c:\GS\Foxit Reader
5: - c:\GS\gs9.07\bin
6: - c:\Program Files (x86)\Comos\91\Current\BIN
7: - c:\Program Files (x86)\FreePDF_XP
8: - c:\Program Files (x86)\Java\jre7\bin
9: - c:\Program Files (x86)\Mozilla Firefox
10: - c:\Program Files (x86)\SmartPlant\SPLM\Bin
11: - c:\Program Files\AVEVA\Plant\PDMS12.0.SP6
12: - c:\Program Files\AutoCAD2010
13: - c:\Windows\System32
14: - c:\Windows\System32\nls\english
15: - c:\win32app\ingr\PDSHELL\share
16: - c:\win32app\ingr\RIS06.01\bin
17: - d:\Oracle\product\11.2.0\client_1
18: - d:\Programmierung\Graphviz\bin
19: - d:\Programmierung\Gtk+\bin
20: - d:\Programmierung\Ruby200-x64\bin
21: - d:\Programmierung\Ruby_Pik
22: - d:\cygwin\bin
23: settings:
24:   reg32: c:/Windows/System32/reg.exe
25:   reg64: c:/Windows/SysWOW64/reg.exe
26:   64bit: true
27:   system: true
```

```
1: # SystemPath
2: #!/usr/bin/env ruby
3: #
4: require 'yaml'
5:
6: # Exit while compiling
7: exit if Object.const_defined?(:Odra)
8:
9: @cfile = 'SystemPath.ini'
10: @setx = 'c:/Windows/System32/setx'
11:
12: def append(folder)
13:   config()
14:   puts "[APPEND]"
15:   path = "#{@path};#{folder};"
16:   regwrite(path)
17: end
18:
19: def backup()
20:   config()
21:   xpath = ENV["Path"].split(";").sort
22:   if xpath.size > 0
23:     puts "[BACKUP] #{xpath.size.to_s}"
24:     File.rename(@cfile, "#{@cfile}.old") if File.exists?(@cfile)
25:     data = {"paths" => xpath, "settings" => @set}
26:     File.open(@cfile, "w") { |f| f.write(data.to_yaml) }
27:     puts "Backup file: '" + @cfile + "'"
28:   end
29: end
30:
31: def config()
32:   if File.exists?(@cfile)
33:     @cfg = YAML.load_file(@cfile)
34:     @set = @cfg['settings']
35:     @sys = @cfg['system']
36:   else
37:     puts 'Error loading file ' + @cfile
38:     exit
39:   end
40: end
41:
42: def regwrite(path)
43:   if @set['64bit']
44:     regexe = @set['reg64']
45:   else
46:     regexe = @set['reg32']
```

```
47:         end
48:         if @set['system']
49:             regp = '"HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment"'
50:         else
51:             regp = '"HKCU\Environment"'
52:         end
53:         cmd = "#{regexe} add #{regp} /v Path /t REG_EXPAND_SZ /d " + '"' + path + '" /f'
54:         system(cmd)
55:         puts "PATH update finished."
56:     end
57:
58:     def restore()
59:         config()
60:         paths = @cfg['paths']
61:         if paths.size > 0
62:             path = ""
63:             paths.sort.each do |p|
64:                 path = "#{path}#{p};"
65:             end
66:             puts "[RESTORE] #{paths.size.to_s}"
67:             regwrite(path)
68:         end
69:     end
70:
71:     #####
72:
73:     @path = ENV["Path"]
74:     puts "SystemPath (2013) ruby.gruena.net"
75:     puts "*" * 40
76:
77:     if ARGV.length < 1
78:
79:         puts "Command is missing (only listing)."
```

```
80:         puts "Please run:\t'SystemPath.exe <command>'"
81:         puts "Commands:\t(b)ackup / (r)estore\n\t\t(a)ppend <folder>"
82:         puts "*" * 40
83:         paths = @path.split(";")
84:         if paths.size > 0
85:             puts "[LIST] #{paths.size.to_s}"
86:             paths.sort.each { |p| puts p }
87:             system("pause")
88:         end
89:     else
90:         comm = ARGV[0]
91:         if comm[0..1] == "b"
92:             backup()
```

```
93:      elsif comm[0..1] == "r"
94:          restore()
95:      elsif comm[0..1] == "a"
96:          if ARGV[1]
97:              append(ARGV[1].strip)
98:          end
99:      end
100: end
```