```
d:\tmp>jruby -S g:/_PROJECTS/Ruby/jAsciiDoctor/src/jAsciiDoctor.jrb d:/tmp/Sonderteile.adoc
CMD => d:/tmp/Sonderteile.adoc
GUI => d:/tmp/Sonderteile.adoc
D:\tmp\Sonderteile.adoc
Mrz 17, 2014 4:42:55 PM org.apache.fop.cli.InputHandler warning
Warnung: Note: namesp. cut : stripped namespace before processing
Mrz 17, 2014 4:42:56 PM org.apache.fop.cli.InputHandler warning
Warnung: Note: namesp. cut : processing stripped document
Mrz 17, 2014 4:42:56 PM org.apache.fop.cli.InputHandler warning
Warnung: Making portrait pages on USletter paper (8.5inx11in)
Mrz 17, 2014 4:42:56 PM org.apache.fop.events.LoggingEventListener proc
Warnung: Font "Symbol,normal,700" not found. Substituting with "Symbol,
Mrz 17, 2014 4:42:56 PM org.apache.fop.events.LoggingEventListener proc
Warnung: Font "ZapfDingbats,normal,700" not found. Substituting with "Z
Mrz 17, 2014 4:42:56 PM org.apache.fop.hyphenation.Hyphenator getHyphen
Schwerwiegend: Couldn't find hyphenation pattern for lang="en".
Mrz 17, 2014 4:42:56 PM org.apache.fop.events.LoggingEventListener proc
Warnung: Line 1 of a paragraph overflows the available area by more tha
Mrz 17, 2014 4:42:56 PM org.apache.fop.events.LoggingEventListener proc
Warnung: Line 1 of a paragraph overflows the available area by more tha
Mrz 17, 2014 4:42:56 PM org.apache.fop.events.LoggingEventListener proc
Warnung: Line 1 of a paragraph overflows the available area by more tha
Mrz 17, 2014 4:42:57 PM org.apache.fop.events.LoggingEventListener proc
Warnung: Line 1 of a paragraph overflows the available area by more tha
Mrz 17, 2014 4:42:57 PM org.apache.fop.events.LoggingEventListener proc
Warnung: Line 1 of a paragraph overflows the available area by more tha
Mrz 17, 2014 4:42:57 PM org.apache.fop.events.LoggingEventListener proc
Warnung: Line 1 of a paragraph overflows the available area by more tha
Mrz 17, 2014 4:42:57 PM org.apache.fop.events.LoggingEventListener proc
Information: Rendered page #1.
Mrz 17, 2014 4:42:57 PM org.apache.fop.events.LoggingEventListener proc
Information: Rendered page #2.
Mrz 17, 2014 4:42:58 PM org.apache.fop.events.LoggingEventListener proc
Information: Rendered page #3.
Mrz 17, 2014 4:42:58 PM org.apache.fop.events.LoggingEventListener proc
Information: Rendered page #4.
Mrz 17, 2014 4:42:58 PM org.apache.fop.events.LoggingEventListener proc
Information: Rendered page #5.
Mrz 17, 2014 4:42:58 PM org.apache.fop.events.LoggingEventListener proc
Information: Rendered page #6.
Mrz 17, 2014 4:42:58 PM org.apache.fop.events.LoggingEventListener proc
Information: Rendered page #7.
Mrz 17, 2014 4:42:59 PM org.apache.fop.events.LoggingEventListener proc
Information: Rendered page #8.
```

jAsciiDoctor 1.0

File **Export** **Help**

Docbook (XML)
PDF

D:\tm...e.adoc

Preview   **Full view**

```
= Sonderteilbearbeitung (COMOS)
Andreas Weber (Wea / C-LI) / andreas.
weber@cac-chem.de
Ausgabe 1 vom 06.03.2014
:encoding: utf-8
:icons:
:toc:
:toc-title: Inhalt
:toc-placement: preamble

image::images/CAC_logo.png[align="center"]


== Sonderteile in COMOS

[NOTE]
Die Sonderteilbearbeitung in COMOS wurde Ende 2013
geändert / überarbeitet.


*COMOS ist das Standard-Tool zum Dokumentieren von
Sonderteilen, d.h. alle zugehörigen Datenblätter*
*werden nur in COMOS erstellt / verwaltet.*

=== Abarbeitung

Sonderteile können sich aus der 2D-Bearbeitung (R&I)
oder aus dem 3D-Design (PDS, SmartPlant, PDMS)
ergeben.
```

Sonderteilbearbeitung (COMOS)

Andreas Weber (Wea / C-LI) / andreas.weber@cac-chem.de
Ausgabe 1 vom 06.03.2014

Inhalt

- Sonderteile in COMOS
  - Abarbeitung
  - COMOS
- Sonderteilarten
  - Sonderteil (inkl. Datenblatt) (A)
  - Grafisches Sonderteil (R&I) (B)
- Arbeitsteilung / Bearbeitungsumfang
- R&I-Bearbeitung
  - Sonderteil (Grafik) anlegen (B)
  - Sonderteil anlegen / verlinken (Ax)
- Sonderteil-Bearbeitung

Sonderteile in COMOS

Note   Die Sonderteilbearbeitung in COMOS wurde

Sonderteilbearbeitung (COMOS)

Andreas Weber (Wea / C-LI) / andreas.weber@cac-chem.de

Table of Contents

Figure 1.

C·A·C

Sonderteile in COMOS

Note

Die Sonderteilbearbeitung in COMOS wurde Ende 2013 geändert / überarbeitet.

COMOS ist das Standard-Tool zum Dokumentieren von Sonderteilen, d.h. alle zugehörigen Datenblätter werden nur in COMOS erstellt / verwaltet.

Abarbeitung

Sonderteile können sich aus der 2D-Bearbeitung (R&I) oder aus dem 3D-Design (PDS, SmartPlant, PDMS) ergeben.

Ggf. müssen Sonderteile, die im 3D entstehen, auch im R&I nachgetragen werden.

```ruby
 1  # jAsciiDoctor
 2  #!/usr/bin/env jruby
 3  #encoding: utf-8
 4  #
 5  include Java
 6  import javax.swing.JOptionPane
 7  require 'jrubyfx'
 8  require 'asciidoctor'
 9  require 'erb'
10
11  fxml_root File.dirname(__FILE__)
12  PROG = Dir.pwd
13  AUTHOR = "Andreas Weber"
14  MAIL = "ruby@gruena.net"
15  VERSION = "1.0"
16  TITLE = "jAsciiDoctor" + " #{VERSION}"
17
18  class GUI < JRubyFX::Application
19    #~ property_accessor :file
20
21    #~ def initialize
22    #~   fASD = parameters.raw[0].to_s
23    #~   puts "GUI => #{fASD}"
24    #~   @file = fASD
25    #~ end
26
27    def start(stage)
28      stage.title = TITLE
29      stage.fxml GUIController
30      stage.show()
31      file = parameters.raw[0].to_s
32      puts "GUI => #{file}"
33    end
34  end
35
36  class GUIController
37    include JRubyFX::Controller
38    fxml "jAsciiDoctor.fxml"
39
40    #~ puts "Contr. => #{@@cFILE}"
41    #~ file = GUI.file
42    #~ if file != ""
43    #~   puts "X => #{file}"
44    #~   GUIController.openFile(file.to_s)
45    #~ else
46    #~   GUIController.openFileDlg()
47    #~ end
48
49    def about
50      msg = "#{TITLE}\nby #{AUTHOR}\n<#{MAIL}>"
51      JOptionPane.showMessageDialog(nil, msg, "About jAsciiDoctor", JOptionPane::
52      INFORMATION_MESSAGE)
53    end
54
55    def help
56      html = 'help/help.html'
57      system("start #{html}")
58    end
59
60    def newFile
61      @editor.text = "= Title\nauthor <email>\nrevision, date"
62      @file = saveFileAs()
        @txtFile.text = @file.to_s
```

```ruby
63    end
64
65    def openFile(f)
66      @file = f
67      puts f
68      @txtFile.text = @file.to_s
69      File.open(@file.to_s, "r:utf-8") { |f| @text = f.read }
70      @editor.text = @text
71    end
72
73    def openFileDlg
74      chooser = FileChooser.new()
75      chooser.setTitle("Select Asciidoc file")
76      chooser.add_extension_filter("ADOC", "*.adoc")
77      file = chooser.showOpenDialog(nil)
78      openFile(file)
79    end
80
81    def saveFile
82      File.open(@file.to_s, "w") { |f| f.puts @editor.text}
83    end
84
85    def saveFileAs
86      chooser = FileChooser.new()
87      chooser.setTitle("Save file as (*.adoc)")
88      chooser.add_extension_filter("ADOC", "*.adoc")
89      file = chooser.showSaveDialog(nil)
90      file = file.to_s + ".adoc" if File.extname(file.to_s) == ""
91      File.open(file.to_s, "w+") { |f| f.puts @editor.text}
92      return file
93    end
94
95    def closeFile
96      ret = JOptionPane.showConfirmDialog(nil, "Do you want to save ?", "jAsciiDoctor",
97      JOptionPane::YES_NO_OPTION)
98      saveFile() if ret == 0
99      @txtFile.text = "Open Asciidoc file ..."
100     @editor.text = ""
101     @html = ""
102     @preview.getEngine.loadContent(@html)
103   end
104
105   def preview
106     saveFile()
107     asd = @file.to_s.gsub(/.adoc/, '.html')
108     html = asd.gsub(/.adoc/, '.html')
109     Asciidoctor.render_file(asd,
110       :in_place => true,
111       :icons => true,
112       :backend => "html5"
113     )
114     File.open(html, "r:utf-8") { |f| @html = f.read }
115     @preview.getEngine.loadContent(@html)
116   end
117
118   def fullview
119     asd = @file.to_s.gsub(/\\/, '/')
120     html = asd.gsub(/.adoc/, '.html')
121     system("start #{html}")
122   end
123
124   def exportXML
        saveFile()
```

```ruby
125     asd = @file.to_s.gsub(/\\/, '/')
126     xml = asd.gsub(/.adoc/, '.xml')
127     Asciidoctor.render_file(asd,
128                 :in_place => true,
129                 :icons => true,
130                 :backend => 'docbook5'
131         )
132     JOptionPane.showMessageDialog(nil, "Docbook export finished => #{xml}",
133             "jAsciiDoctor", JOptionPane::INFORMATION_MESSAGE)
134
135   end
136   def exportPDF
137     exportXML()
138     asd = @file.to_s.gsub(/\\/, '/')
139     xml = asd.gsub(/.adoc/, '.xml')
140     xsl = "g:/Tools/OpenWriting/docbook_xsl/fo/docbook.xsl"
141     pdf = asd.gsub(/.adoc/, '.pdf')
142     cmd = "fop -xml #{xml} -xsl #{xsl} -pdf #{pdf}"
143     #~ puts cmd
144     system(cmd)
145     JOptionPane.showMessageDialog(nil, "PDF export finished => #{pdf}",
146             "jAsciiDoctor", JOptionPane::INFORMATION_MESSAGE)
147     system("start #{pdf}")
148
149   end
150
151 if ARGV.size > 0
152     aFILE = ARGV[0]
153 else
154     aFILE = "d:/tmp/Sonderteilex.adoc"
155     #~ aFILE = ""
156 end
157 #~ xfile = aFILE.gsub(/\\/, '\\')
158
159 puts "CMD => #{aFILE}"
160 GUI.launch(aFILE)
161 end
```