XmlGraphicsCommonComponents

XML Graphics common components

The problem

The FOP codebase contains two Batik transcoders for PDF and PostScript which can be used independently of FOP and are distributed with Batik. The Batik team should be empowered to help maintain these transcoders, for example, after changes in Batik itself. Since Batik and FOP are both implementing an XML standard that creates graphical output there is some overlap between the two efforts. Last but not least, a Batik release didn't involve a FOP release until now which is something that must change.

The idea

The idea now is to create a common area where the parts of common interest between the two subprojects are jointly developed and maintained. XML Graphics Commons will deliver a set of well-defined libraries of which some might provide some supporting tools (such as the TTFReader and the PFMReader, but which will eventually vanish when they are not needed anymore). Other potentially reusable software like utilities for build support, ant tasks, plugins for IDEs, test frameworks and data files might also qualify for finding a home in XML Graphics Commons.

Benefits

- Clear dependencies
- More structured codebase
- Easier for newbies to contribute to certain parts (They are afraid now of the big blob of code)
- New (and now easily visible) use cases for certain components (examples: JPS services to allow Java applications to print to PDF and PS through Printables, PDF lib can be used independetly of FOP, etc.)
- Faciliating collaboration with "competing" projects like FOray and Folio, i.e. work together on undisputed parts of an XSL-FO processor.

The plan

Common repository

For the joint operations we need a common repository. Since we now have migrated both subprojects to Subversion we are in a good starting position for this.

Parts affected in FOP

directly:

- PDF transcoder (org.apache.fop.svg)
- PostScript transcoder (org.apache.fop.render.ps)

Dependencies:

- PDF library (org.apache.fop.pdf, by PDF transcoder)
- PS generation code (org.apache.fop.ps, by PS transcoder)
- font support (org.apache.fop.fonts, by PDF lib, PDF and PS transcoders)
- IO helpers (org.apache.fop.util, by PDF and PS transcoders) [1]
- Image adapters (org.apache.fop.image, by PDF and PS transcoders)
- Basic FOP exceptions (org.apache.fop.apps)
- RTF library (org.apache.fop.render.rtf.rtflib, based on user feedback, generally usable)

[1] Some of these classes could be candidates to go into Jakarta Commons IO.

External dependencies:

- Avalon Framework (for the configuration subpackage only) [2]
- Jakarta Commons Logging [3]
- Jakarta Commons IO

[2] There's enough opposition here that warrants a reevaluation. We'll try to do without this.

[3] During the move it should be tried to remove this dependency, i.e. no logging at all in the basic shared components.

Parts affected in Batik

The transcoders depend on several packages inside Batik. Obviously, we can't create a clean hierarchy without dependencies on Batik, at least for the PDF and PS transcoders. But we should do that for the parts where this is possible (PDF lib, fonts, images etc.).

Possible components coming from Batik:

- ParsedURL (org.apache.batik.util, could be used by FOP, too. Benefits: RFC2397 data: URLs, automatic gzip decompression (SVGZ))
- image codecs (org.apache.batik.ext.awt.image.codec)
- Base64 encoder and decoder (org.apache.batik.util)
- SVGGraphics2D
- ...

Organization and naming

Naming the individual parts

Apache XML Graphics Commons

- xmlgraphics-commons-pdf (PDF library)
- xmlgraphics-commons-ps (PostScript generation and manipulation/post-processing code)
- xmlgraphics-commons-fonts (Font library)
- xmlgraphics-commons-codecs (Image codecs)
- xmlgraphics-commons-image-adapter (Image library adapters and analyzers)
- xmlgraphics-commons-utils (IO classes, Exception classes, ParsedURL)
- xmlgraphics-commons-java2d (Graphics2D implementation for PDF and PS, base for the Transcoders)
- xmlgraphics-commons-rtf (RTF library)

Layout in SVN

```
http://svn.apache.org/repos/asf/xmlgraphics
  +-- commons
        +-- branches
        +-- tags
        +-- trunk
             +-- src
                  +-- java (all shared components under this directory)
                       +-- org.apache.xmlgraphics
                            +-- codecs
                            +-- fonts
                            +-- image-adapters
                            +-- java2d (Graphics2D implementations, gradient/(pattern) extensions)
                                 +-- pdf (PDF implementation)
                                 +-- ps (PS implementation)
                                 +-- svg (SVG implementation)
                            +-- pdf
                            +-- ps
                            +-- rtf
                            +-- utils
                  +-- java-1.4 (all JDK 1.4 dependant code under this directory)
  +-- batik
        +-- branches
        +-- tags
        +-- trunk
  +-- fop
        +-- branches
        +-- tags
        +-- trunk
  +-- site
```

Notes:

- In this scenario, the PDF and PS transcoders are transferred to the Batik subproject under their repository.
- Basic Graphics2D implementations and supporting code (gradient/(pattern) classes) go into Commons (Batik only gets the Transcoders)
- this layout has the effect that we have three major products (batik, fop and xmlgraphics-commons), where axgc may have additional (sub-) products for each individual components (if necessary).
- axgc builds some components individually (xmlgraphics-commons-pdf.jar, xmlgraphics-commons-utils.jar etc.), as well as a collective (xmlgraphics-commons.jar). A release is only done as a collective for the whole of XML Graphics Commons.
- All releases are always coordinated on the project level (i.e. on level Apache XML Graphics).

Notes on additional use cases for the separated components

- PDFDocumentGraphics2D and PSDocumentGraphics2D can be used to create streamed print services for JPS which would allow arbitrary Java applications to create PDF and PostScript by simply printing to JPS.
- The PDF library could be used to create PDF documents from scratch (iText is probably better suited for that).
- The PDF library could be extended to modify existing documents (same comment as above)
- The PS library code could be extended to support PostScript post-processing.
- The RTF library is reported to be used separately. There is desire for a separate JAR with only the RTF generation code.

work items

closed

- Move Batik and FOP to Subversion.
- Complete and discuss this plan.

open

- Vote on the plan.
- Clean up dependencies.
- Create basic exception classes so the dependency on org.apache.fop.apps disappears.
- Create common repository.
- Move affected parts (see above) over to their new location.
- Create build system for all components. (TBD: one for all or one for each?)
- Documentation for the new parts