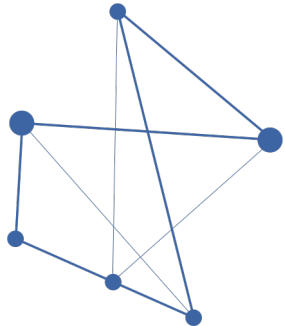




A GROOVY way to enhance your SAS life



Your 1st choice
of global FSP partner

Author:
Karl Kennedy

11 November 2019



GCE

an IQVIA business

Agenda

- Introduction
- What is PROC GROOVY?
- Program Components
- Advantages & Disadvantages of using PROC GROOVY
- Key Components of the TFL PDF Document
 - 1) Appending PDF files
 - 2) Creating Bookmarks
 - 3) Creating a Table of Contents
- Conclusion
- Questions

Would PROC GROOVY have enabled me to:

- Fully automate the TFL PDF process
- Include standard document navigation features e.g. toc, bookmarks
- Provide an all-round cleaner program and solution



PROC GROOVY is relatively new (SAS 9.3+) and little-known SAS procedure that enables SAS code to execute Groovy/Java code on the Java Virtual Machine.

```
proc groovy;  
  add classpath = "C:\groovy\pdfbox-app-2.0.15.jar";  
  submit;  
    // Place Groovy code within submit statements.  
  endsubmit;  
quit;
```





GCE

an IQVIA business

Program Components



Advantages	Disadvantages
Handles images/graphs	Initial set-up & configuration
Access to extensive Java APIs	Working knowledge of Groovy/Java
Extensive online Java support	

Code excerpt for appending all TFLs into one PDF document.

```
File tflFile = new File(file.getCanonicalPath());
PDDocument tempDoc = PDDocument.load(tflFile);

// Stores the first page of TFL which is used as a bookmark destination.
PDPage page = (PDPage) tempDoc.getDocumentCatalog().getPages().get(0);

int numOfPages = tempDoc.getNumberOfPages();
for(int i=0; i < numOfPages; i++)
{
    PDPage tflPage = (PDPage) tempDoc.getDocumentCatalog().getPages().get(i);
    doc.addPage(tflPage);
    // Code for creating bookmarks & TOC links is discussed in the next sections.
}
```



Code excerpt for setting the first page of a TFL to be the bookmark destination.

```
PDPageDestination dest = new PDPageFitWidthDestination();  
dest.setPage(page);  
PDOutlineItem bookmark = new PDOutlineItem();  
bookmark.setDestination(dest);  
String title = file.getName().substring(0,file.getName().indexOf("."));  
bookmark.setTitle(title);  
pagesOutline.addLast(bookmark);
```

Bookmarks

▼ All TFLs

L16020103

L16020301

T140301

G109457



```
// Sets up a specific PDF 'go-to' action within TOC.
action = new PDActionGoTo();
action.setDestination(tocMain);
pdest = new PDPageFitWidthDestination();
pdest.setPage(thePage);
// Create link for an individual TFL title in TOC.
link = new PDAnnotationLink();
link.setAction(action);
// Use x,y coordinates to place the PDRectangle over the TFL title.
rect = new PDRectangle();
rect.setLowerLeftX(llx);
rect.setLowerLeftY(lly);
rect.setUpperRightX(ux);

rect.setUpperRightY(ury);
link.setRectangle(rect);
// Add link to TOC.
toc.getAnnotations().add(link);
```

Listing 16.2.1.3 Case-Level Modes

Listing 16.2.3.1 Protocol Deviations



With the benefit of this exploratory program I feel PROC GROOVY would have provided a better solution than my original SAS program.

PROC GROOVY isn't a procedure that would be used for a SAS programmers' day to day tasks but:

- A viable option for possible automated tasks e.g. TFL document generation.
- It could possibly be used for Excel or database tasks using the appropriate Java APIs and drivers.



