ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-pdf



Peter Reutemann

November 21, 2026

©2012-2019



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato



Except where otherwise noted, this work is licensed under
http://creativecommons.org/licenses/by-sa/4.0/

Contents

1 Introduction **7**

2 Flow **9**

3 Tools **13**

4 Troubleshooting **15**

Bibliography **17**

List of Figures

2.1 Flow for creating a PDF file from various sources. 10
2.2 CSV files get added as tables. 11
2.3 Images can get inserted as well. 11
2.4 Plain text files get added as simple text. 12

3.1 Viewer for PDF files. 13

Chapter 1

Introduction

The *pdf* module adds PDF authoring capabilities to ADAMS. This is possible thanks to the iText [2] and jPod Renderer [3] libraries for creating, manipulating and viewing of PDF files.

Chapter 2

Flow

The following source actors are available:

- *PDFNewDocument* – creates an empty PDF document.[1]

The following transformers are available:

- *PDFAppendDocument* – appends a PDF document.[2]
- *PDFCreate* – for creating PDFs, using text files, spreadsheets and images.[3]
- *PDFExtract* – extracts a range of pages.
- *PDFExtractImages* – extracts the images from a PDF.[4]
- *PDFExtractText* – obtaining the plain text of the PDF.[5]
- *PDFGenerate* – similar to *PDFCreate*, but uses the specified generator to create the PDF out of the input objects.
- *PDFInfo* – outputs information about a PDF file.[6]
- *PDFMerge* – merges several PDFs into a single one.[7]
- *PDFPageCount* – determines the number of pages in a PDF.[8]
- *PDFRenderPages* – renders pages of a PDF as images.[9]
- *PDFStamp* – allows to add an overlay to a PDF.[10]

The following sinks are available:

- *PDFCloseDocument* – closes a PDF document, writes out the content to disk.[11]
- *PDFViewer* – for viewing PDFs.[12]

[1] adams-pdf-create_pdf2.flow
[2] adams-pdf-create_pdf2.flow
[3] adams-pdf-create_pdf.flow
[4] adams-pdf-extract_images.flow
[5] adams-pdf-extract_text.flow
[6] adams-pdf-info.flow
[7] adams-pdf-page_count.flow
[8] adams-pdf-page_count.flow
[9] adams-pdf-render_pages.flow
[10] adams-pdf-page_overlay.flow
[11] adams-pdf-create_pdf2.flow
[12] adams-pdf-view_pdf.flow

Creating PDFs

Generating a PDF using the *PDFCreate* transformer is really easy. The transformer takes an array of file names as input, which will all get added to the specified output PDF file. Basic options, like page size and orientation, can be set as well. How and what files get added to the PDF, is determined by the "proclets", i.e., little processor classes, that you specify and configure:

- *SpreadSheet* – for adding spreadsheet files as tables.
- *Headline* – for adding a headline.
- *Image* – for adding images (GIF, JPEG, PNG).
- *PageBreak* – forces a pagebreak.
- *PlainText* – for adding plain text files as paragraphs.
- *Rectangle* – draws a rectangle at specified location.

Figure 2.1 shows a flow[13] that adds all files (i.e., three) found in a directory to a single PDF. Figures 2.2, 2.3 and 2.4 show the resulting pages of the PDF in the viewer.

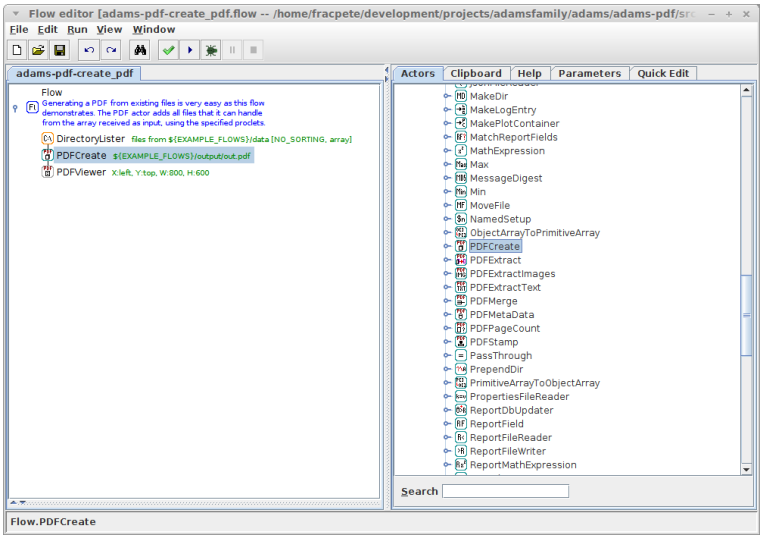


Figure 2.1: Flow for creating a PDF file from various sources.

[13] adams-pdf-create_pdf.flow

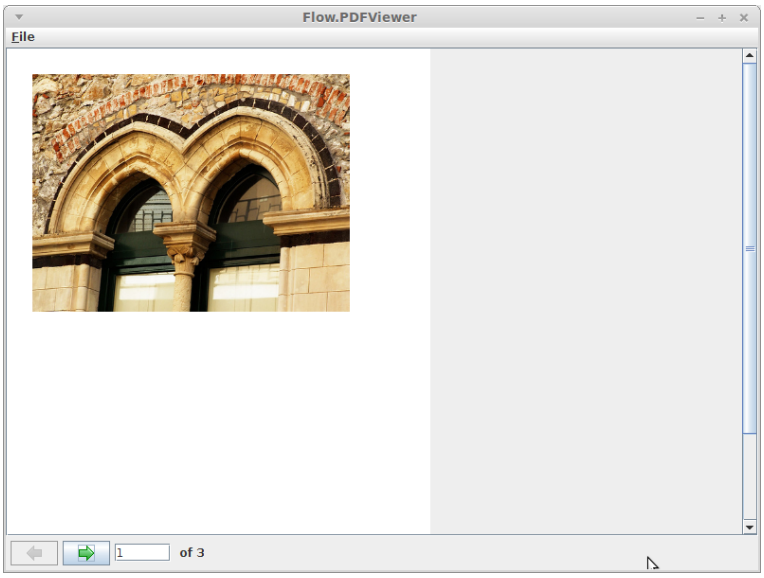


Figure 2.2: CSV files get added as tables.

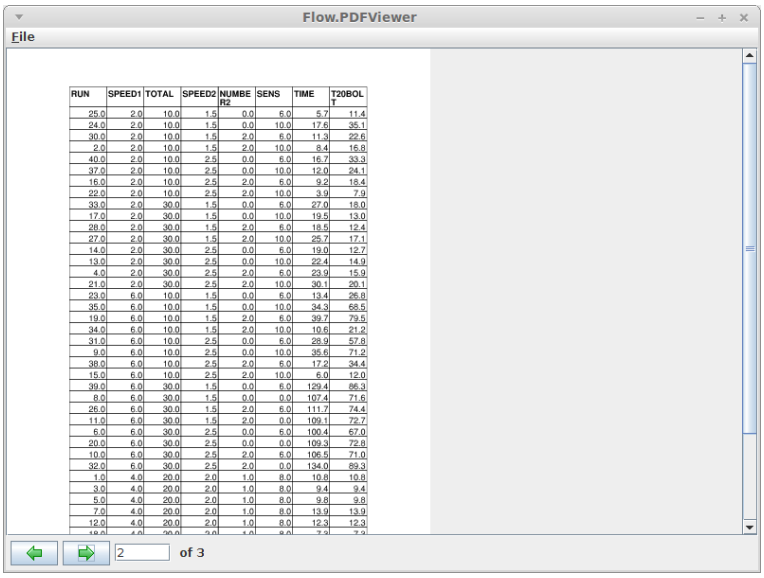


Figure 2.3: Images can get inserted as well.

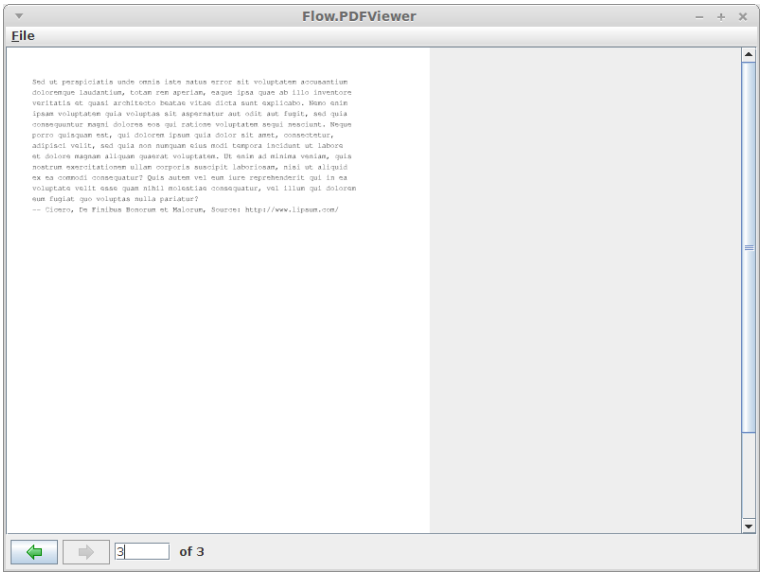


Figure 2.4: Plain text files get added as simple text.

Chapter 3

Tools

The *PDF viewer* can be used to load and browse PDF files. Figure 3.1 shows a screenshot of the viewer with a PDF file loaded.

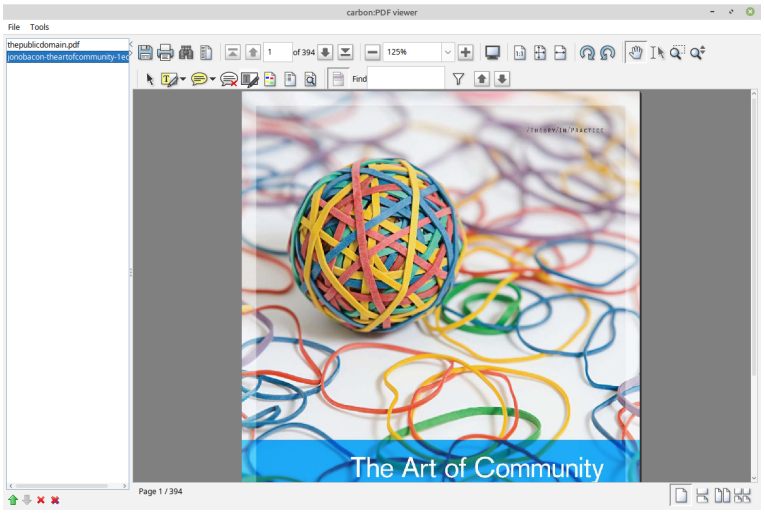


Figure 3.1: Viewer for PDF files.

Chapter 4

Troubleshooting

- **Problem:** 64bit Linux shows error message when viewing PDF that 'libfreetype.so' library was not found ("UnsatisfiedLinkError").
  Solution: Make sure that /usr/lib/libfreetype.so exists. If not, add a symbolic link to the 64bit library, using a similar command as follows (for Kubuntu 11.10 or LinuxMint 11, 13):

  ```
  sudo ln -s /usr/lib/x86_64-linux-gnu/libfreetype.so.6 /usr/lib/libfreetype.so
  ```

  And restart the application.

Bibliography

[1] *ADAMS* – Advanced Data mining and Machine learning System
`https://adams.cms.waikato.ac.nz/`

[2] *iText* – A library that allows you to create and manipulate PDF documents
`http://itextpdf.com/`

[3] *jPod Renderer* – A PDF rendering implementation for AWT and SWT.
`http://opensource.intarsys.de/home/en/index.php?n=JPodRenderer.HomePage`