This developer demo shows

    • how to display a multi-page PDF document in a view
    • how to print that view
    • how to let the user scroll a scrollview's contents by dragging

To add this functionality to your application

    (1) add PDFImageView.m and .h to your project
    (2) drag PDFImageView.h to your nib file
    (3) set a custom view's class to PDFImageView
    (4) in your code, to load a PDF document at *somePath*, use
            `[`*`myPDFView`*` loadFromPath: `*`somePath`*`];`
    (5) optionally add DraggableScrollView.m and .h to your project
       (this will allow dragging)

Things to note about the application

• An ordinary NSImageView can display a PDF in an NSImage, but will show only the first page of the PDF.

• PDFImageView is just a subclass of NSImageView, but works with NSPDFImageRep object to intelligently display or print each of its pages.

• PDFImageView automatically sets its size to display all pages of the PDF you load into it, so it works best inside an NSScrollView (and without springs to resize it in either direction).

• PDFImageView overrides **setFrame:** to ensure that the width and height keep the correct proportion.

• DraggableScrollView provides **dragDocumentWithMouseDown:**,

which any view can call (your custom views can, too) to allow dragging the document displayed inside the scroll view.

• DraggableScrollView displays a distinctive cursor when the user could click to drag the content. To update the cursor, it overrides NSScrollView's **tile** method.

Enhancements you might try on a rainy day

- draw lines at page boundaries, and page numbers
- use NSDocument(Controller) and open saved PDF's
- make the PDF resize to fit the scroll-view's width
- buy an umbrella