



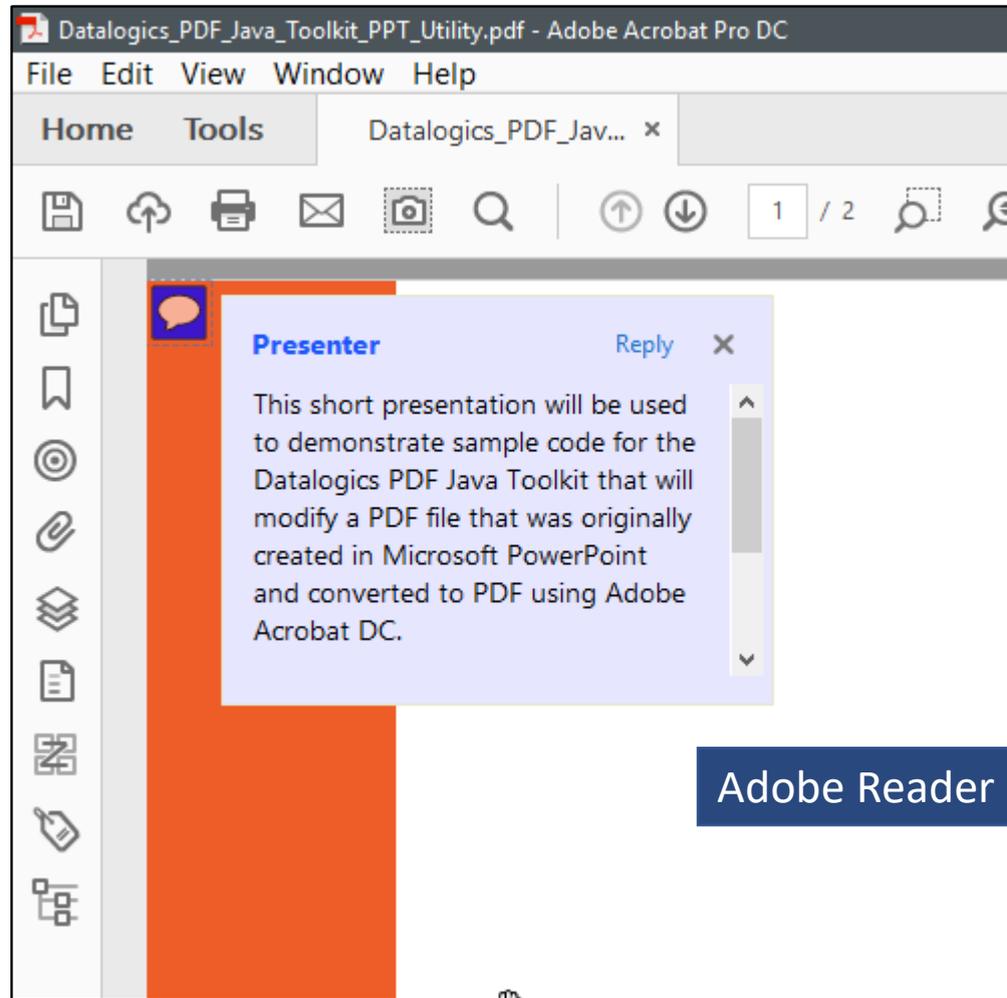
Datalogics PDF Java Toolkit

PowerPoint Notes to Content

Datalogics

Slide Notes:

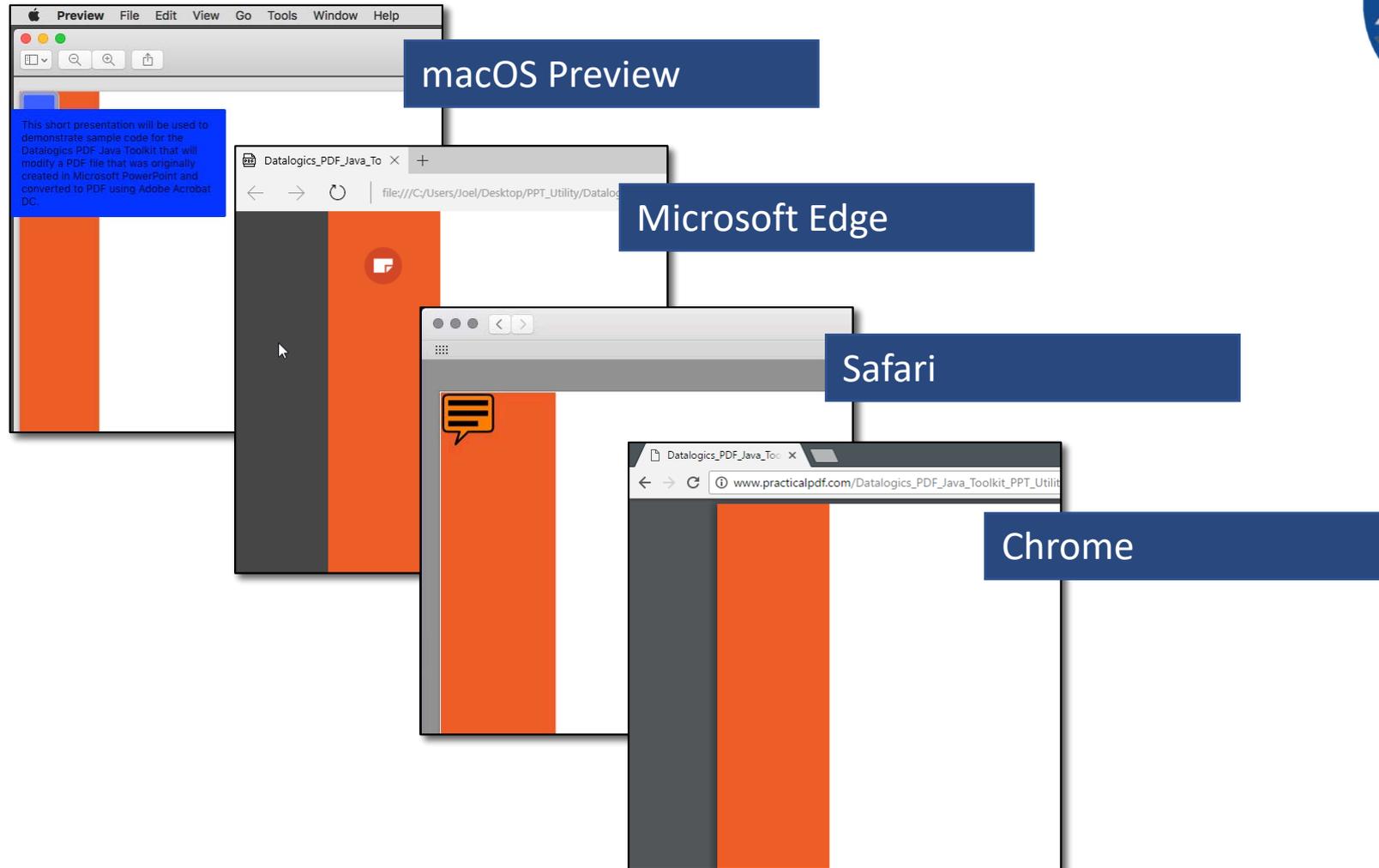
This short presentation will be used to demonstrate sample code for the Datalogics PDF Java Toolkit that can modify a PDF file that was originally created in Microsoft PowerPoint and converted to PDF using Adobe Acrobat DC. The problem is that desktop PDF viewing applications, hosted PDF viewing applications, and web browsers with native PDF support all treat comments in a PDF file differently as you'll see in the following slides. The goal of the sample is to use the output from Acrobat as the input to the sample and produce a PDF file that behaves consistently across various PDF viewers and web browsers.



Slide Notes:

The image on this slide is what the comment will look like in Adobe Acrobat or Reader. As the document author, this is what I expect the document recipients to see.

Datalogics



Slide Notes:

Unfortunately, the same PDF file looks different in different viewers. In macOS Preview, I can see the content of the note annotation but it shows up as black text on a dark blue background, it's too hard to read. The annotation shows up in the Microsoft Edge browser but it doesn't do anything. The same is true of the Safari browser and the annotation doesn't even appear in the Chrome browser. If I want my audience to understand the graphics in the presentation, they need to be able to read the notes but the default output from Acrobat is really only useful in an Adobe viewer.



Print

Info
New
Open
Save
Save As
Save as Adobe PDF
History
Print
Share
Export
Close
Account
Options
Feedback

Print

Copies: 1

Printer: Adobe PDF Ready

Settings

Print All Slides: Print entire presentation

Slides:

Notes Pages: Print slides with notes

Collated: 1,2,3 1,2,3 1,2,3

Portrait Orientation

Color



This short presentation will be used to demonstrate the Datalogics PDF Java Toolkit that can modify a PDF file that was originally created and converted to PDF using Adobe Acrobat DC. The toolkit takes a PDF file from Acrobat as the input to the sample and produces a new PDF file that works across various PDF viewers and web browsers.

Datalogics

Slide Notes:

The only way to guarantee that the PDF will display with the notes so all PDF viewers can display them is to make the notes part of the normal page content. You can easily do this by "printing" the "Notes Pages" to PDF. Unfortunately, the slides themselves get rendered as bitmaps so often times the type can get jagged and ugly looking even when using the Adobe PDF printer. After trying a few different ways of solving the problem manually and realizing the horror of doing that for every presentation I create, I decided to write an application that could modify the standard PDF file output by Acrobat from PowerPoint and reformat it into the layout I wanted but with the slide notes as part of the page. Because I have access to the Datalogics PDF Java Toolkit, this task was pretty easy.



A thumbnail of a presentation slide. The slide has a white background with a blue vertical bar on the right and an orange vertical bar on the left. In the top right corner, there is a small version of the toolbox icon. The main text is centered: "Datalogics PDF Java Toolkit" in a large black font, and "PowerPoint Notes to Content" in a smaller black font below it. In the bottom left corner, the word "Datalogics" is written in a bold blue font.

Slide Notes:

This short presentation will be used to demonstrate sample code for the Datalogics PDF Java Toolkit that can modify a PDF file that was originally created in Microsoft PowerPoint and converted to PDF using Adobe Acrobat DC. The goal of the sample is to use the output from Acrobat as the input to the sample and produce a PDF file that behaves consistently across various PDF viewers and web browsers.

Slide Notes:

Using the Datalogics PDF Java Toolkit, I was able to resize the page making it wider to accommodate a column of text on the right. Then I added a few filled boxes to the newly added page area to make it more graphically appealing, then I simply grabbed the text out of the annotation on each page, deleted the annotation and used the content to set the value of a text field that I also added to the page. Finally, I flattened all the new text fields to turn them into page content. The result looks like the image on this slide... and this PDF.

Datalogics