

Javascripting PDF in Scribus

Editor's Note: This document is best displayed in Acrobat 5.x+. Other PDF viewers such as GSView, Xpdf and others do not have support for javascript while viewing PDF documents.

One of the most powerful and interesting capabilities in Scribus is creating javascripted PDF. Yes, javascript and PDF go together nicely, thank you very much. Where Scribus is truly groundbreaking in the Linux/*nix world with its ability to enhance add, edit and embed javascript actions and objects within a PDF. This ability is only found elsewhere in the full Acrobat Editor or hand editing the PDF file – dangerous, unless you are expert with the PDF file format.

The best part ? You do not need to be an expert javascript programmer to add these magical snippets of code. If you understand some of the capabilities of the PDF format, the Scribus tools are easy to use and have some built-in guards against doing the wrong thing.

Concepts

An important concept to remember when designing your PDF document is Acrobat Reader works much like a browser. This includes: mouse driven navigation, internal links, search, browse history and form submission. It is easy to add live web links, as well as validate or calculate fields in forms. You can add nice presentation effects to your documents, but also add link buttons for navigating your docs. Most, but not all of these features are supported on all platforms which have a 4x version of Acrobat Reader. That is almost every PC, Mac or *nix workstation on the planet.

It is helpful to have looked through the sample docs just using the existing form buttons and dialogs. Using bits of javascript extends and enhances these features. Technically, there are six different places where you apply javascript to a PDF. In these examples and most all the time with Scribus' PDF, we are only concerned with two. Global objects are things which affect the whole document like the full screen effect below and adding interactive warnings or info. The other and most common is applying javascript to fields or field objects.

Javascript for Newbies

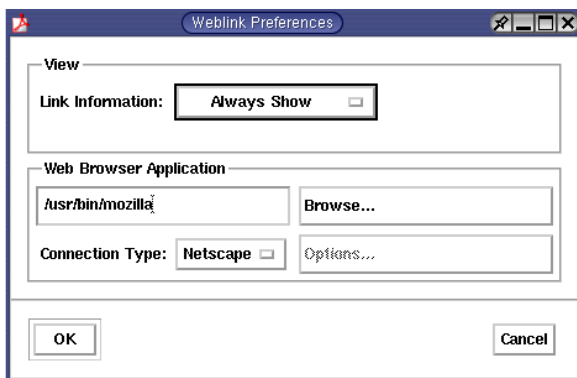
Most of the things which make all this magic work are typically one or two simple “commands” if you will in the form of: *command: variable* or *object: variable*. The syntax is usually quite simple. Better, there is no need for bracketing or other nesting of the code. By contrast, some non-programmers have been known to get dizzy or have their eyes fall out when looking at raw, but well structured C++ code. You can also add comments which are ignored by the javascript interpreter like this:

```
/* This is a comment */  
/* And comments can  
be more than one line */
```

This is, in some cases, a good idea, especially when you want to remember why you added something six months later. Just remember to wrap the comments carefully as above.

Getting Started - Checking Acrobat Reader

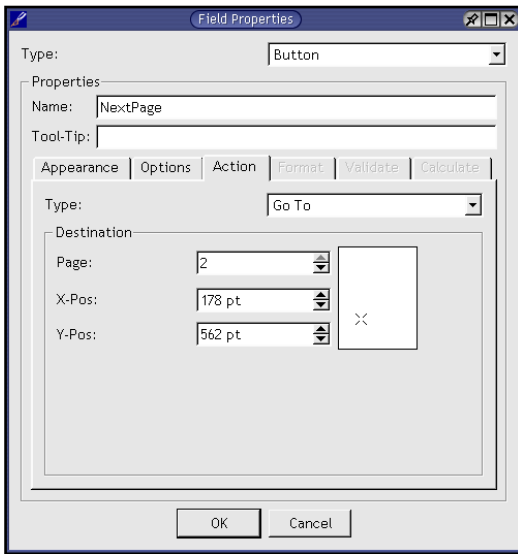
First, open up Adobe Acrobat Reader and go Edit > Preferences > Weblink. As you can see, this sets the preferred browser is used to handle external web links. In this, case, I manually changed the default of Netscape 4.79 to Mozilla. If you change this, close down and restart Acrobat Reader. I have tried both Opera 7.11 and Konqueror 3 and they work just fine too. On Windows, Reader 5.x is usually set to open IE. Later on, we will create examples to launch the browsers to a specific web page. As you browse the objects on these pages, linger over the them momentarily, as they often have tool tips or mouse over effects – another object you can include in to PDF.



One of the default plug-ins in Acrobat Reader 4.x+ is the weblink plug-in. This handles external links to other sections of a document or even to an HTML page on a web site. In the examples below, we are going to make some simple javascript calls to enhance our PDF. We will also show the Acrobat version of Hello World!

Example 1 - A Navigational Text Link

The easiest way to start is with a simple text link. This might be for a different part of a document, perhaps a glossary. To create web-like hyperlinks, the easiest way is to make a simple text field remove the borders and color the text blue, as it is common in browsers. Where you want to add a text link, click the PDF Text button, draw a button and format your text. Next, right click Field Properties > Action and then select the drop down box for Goto as below. Then choose the page and the destination on that page. Then click OK to save.

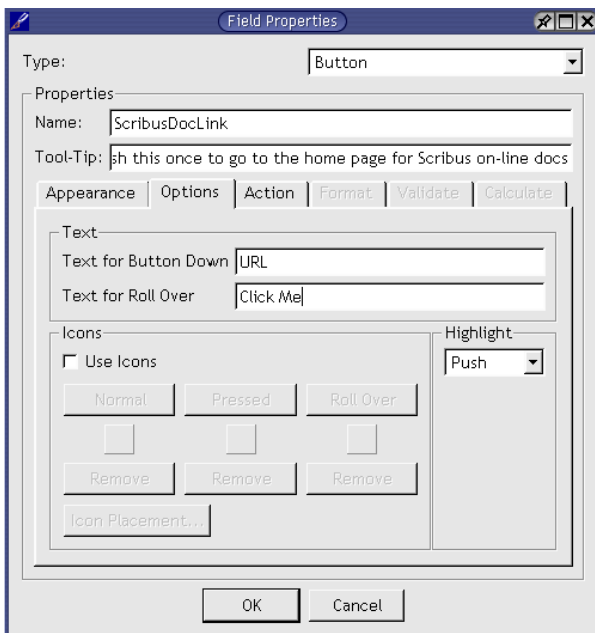


Drag the cross-hair and select the page and area which is the target of your link. When you export and then open the PDF, the cursor in the reader will turn into a small hand pointer as you pass over the link. To speed up document creation, once you have a button you like, send it to the scrapbook for reuse on each page. Then edit just the target of the link.

When you are creating navigation links in a PDF, X/Y coordinates are always expressed as points. When selecting your target destination, it might help to have the target page open in another window to more precisely define your target. Yes, you can open a Scribus doc in more than one window. Just remember to close the second window before saving.

Example 2 A web button link.

Click the OK button on the tool bar and draw a button to the size you want. Then, release the mouse button and double click the button to edit the field properties. Here you have two style choices: You can draw an artistic button or simply spell out the actual text of the link. The author is challenged by crayons and paint by numbers. so text it is...



In the Field Properties dialog, you set all the options for your button. In this example, I have added rollover text, a tool tip and a highlight with separate text when the button is pushed. With more advanced javascript or a group of differently color buttons, you could add highlighting to buttons, much like CSS menus in HTML.

Roll your mouse slowly over the link below and slowly click the link to see the effects. If you weblink plug-in is set properly, it will launch your browser and take you to the destination.

Now, how did I get the browser to launch and add the URL? Remember the part where I said it took just small snippets of code ? It was this simple. In the Action tab, click the Edit button, which launches a simple text editor. Then, I added the following text:

```
getURL("http://atlantictechsolutions.com/scribusdocs/index.html")
```

That is all. You can also create the same type of html link with a text field. This in some respects is easier if you are creating a list of links. Why ? One challenge with formatting buttons vs. text links is buttons automatically align the text to centered. So, this makes making a long list of links like the topics index at the beginning of the documentation more of a challenge to align correctly. What is lost by making a text link instead of a button? The main difference is the visual cues for the reader. Text links which are "click-able" do not change the cursor to the familiar hand pointer, while a button does. Subtle, but something to consider for readers who might not be familiar with Acrobat Reader. Workaround: Use tool tips to catch the reader's eye.

Example 3 Starting a Presentation PDF

This example is how to use javascript to begin a presentation PDF.

Create a button on the first page of your document and create a text label Full Screen. Then right click the button and select Field Properties. Then, Action > Javascript . I recommend you keep the default Mouse Up. This is to avoid accidental clicks. Then Open > Edit which brings up the Javascript editor and add this:

```
app.fs.defaultTransition = "WipeDown";  
app.fs.isFullScreen = true;
```

Save and close the editor. While you can enable this from the dialog boxes in the export to PDF, this explicitly allows you to start from a title page. You could also hide this by blending the text into a background color with a similar shade and color.

Example 4 Hello World in Acrobat

This example shows how to add an informational pop-up or warning. In the real world, you would probably use this to offer info to the end user when validating data for a field. This could be done with mouse enter to give hints for a field or to ask a user to double check a value entered in the field. For our purposes, we will demonstrate this with a button. Create a button as above. Then, in the Field Properties, select Action > Javascript and click edit to open the editor. Insert the following on one line:

```
app.alert({ cMsg: "Hello World!", cTitle: "Hello World Example" });
```

Now, let's extend this a bit further and add a useful reader response which can then be used for input for a form field or validate a value.

```
app.alert({ cMsg: "Have you learned anything ?", cTitle:
"Question!", nIcon: 2, nType: 3});
```

In this example, you could use the input of the above to evaluate true or false in a boolean. Cancel creates no input.

Other PDF Features in Scribus

Articles are another method of navigating a document. Where it is especially useful is when creating multi-column or threaded text. For example, it is quite common to thread a newspaper or magazine story with one or two paragraphs to a section a few pages towards the back of the paper. Within Acrobat Reader, select Window > Article and a palette with the linked articles are shown. Double clicking on the article, brings you to the beginning of the article with a closer zoom level. This does not require javascript, but is a check box selection in the PDF Export dialog.

Bookmarks

Included in Acrobat Reader is a Bookmark panel, much like bookmarks in a web browser. However, bookmarks are specific to a document and have to be created within the PDF originating application or added in Acrobat. In Scribus, creating bookmarks are straightforward. Simply, right click any text frame and select is PDF Bookmark. Bookmarks ease navigating a document, especially longer documents and allow you to direct a reader to the beginning of section titles. In this example PDF, all the section heads are bookmarks. To view, simply click on the bookmark tab on the far left of the Reader screen. Thumbnails of each page can also be used to navigate from page to page.

PDF Q and A

How can create a link to a certain page of a document ?

<http://www.atlantictechsolutions.com/scribusdocs/pdfs/javascript101.pdf#page=5>
would take you to the page you are reading if it were not open already.

How do I know which versions of Acrobat Reader support which options and PDF functions?

The PDF javascript reference clearly shows which options are supported by version number and application. There are some javascript functions which are only available with the full version of Acrobat. Do not think you are missing out. Most of the full Acrobat functions are for automating file and field creation in dynamically created PDF's.

Will we see a version 6.0 of Adobe Reader for Linux or other Unixes ?

Adobe is very guarded about announcing application releases in advance. Historically, they have released a version for Linux and other Unixes after additional languages are released for Windows and Macs.

Which version of javascript are Adobe Acrobat Reader's PDF features based on ?

Acrobat Reader features are based on the core of JavaScript version 1.5 of ISO-16262, also known as ECMAScript.

Conclusions

Scribus eases the creation of interactive PDF forms, presentations and other PDF features with an approachable, user friendly design. You are encouraged to read through the Javascript reference for Acrobat Reader. There are numerous examples, bits of sample code and the features are well documented. Hopefully this tutorial will encourage further exploring.

References and Further Learning

1.8 Mb This is the main Javascript reference for PDF.

10.0 Mb This is the Third Version of the PDF file format reference. 978 pages.

Home Page for everything Javascript at Netscape's developer site.

This document was created 100% with Scribus 1.0RC2
First Release 01:07:2003 for Scribus 1.0

Copyright (c) 2003 Peter Linnell and Franz Schmid. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later with elected option number 1 (the latest version is presently available at <http://www.opencontent.org/openpub/>).