

AcroT_EX.Net

**The APB
Sub-page Navigation**

D. P. Story

Introduction

The notion of *sub-page navigation* was introduced in the *PDF Reference*, for PDF 1.5, which corresponds to Acrobat/Adobe Reader version 6. With sub-page navigation, you can navigate not only from page to page, but from layer to layer on the same page.

In the previous, and first version of APB, an elaborate system of buttons were created to navigate through the presentation, between layers and between pages. The presenter had to have cursor control at all times to click on buttons to advance the presentation. Remote, wireless devices—popular with Powerpoint presentations—*could not really be used* due to the need for mouse control.

In this latest release of APB, there is an option, `sprnavi`, which causes APB to use sub-page navigation. Sub-page navigation comes into effect only in fullscreen mode. So, in fullscreen, the presentation can be navigated through with the usual controls: left and right arrows, or left and right mouse clicks; consequently, an APB document *now can be controlled by remote wireless presenter devices!* Isn't that swavishly cool!

This document presents some background to sub-page navigation, discusses the APB implementation of it, and demonstrates sub-page navigation.

When in fullscreen, navigation forward and backward as follows:

- Forward Navigation: right-arrow key or left-click on mouse.
- Backward Navigation: left-arrow key or right-click on mouse.

Theory

Sub-page navigation is discussed in section 8.3.3 of the PDF Reference. This section generally discusses presentations and page transition effects.

To use sub-page navigation, the application must create a doubly-linked list of navigation nodes. The root (or primary) node is placed in the page dictionary. Each navigation node is a dictionary (PDF terminology) consisting of five (optional) key-values. These five control are:

1. **NA**: (next action) A sequence of actions executed when the user navigates forward from this node.
2. **PA**: (previous action) A sequence of actions executed when the user navigates backward from this node.
3. **Next**: A pointer to the next node, if any.
4. **Prev**: A pointer to the previous node, if any.
5. **Dur**: The maximum number of seconds before the viewer application should advance forward to the next navigation node. If there is no **Dur** entry, automatic navigation does not occur.

The *PDF Reference* introduces the notion of a *current node*. The following describes the events associated with a request to navigate forward or backward.

- Navigate forward
 1. The sequence of actions as specified by the **NA** key, if present, is executed
 2. The node specified by the **Next** key, if present, becomes the new current node.

- Navigate backward
 1. The sequence of actions as specified by the **PA** key, if present, is executed
 2. The node specified by the **Prev** key, if present, becomes the new current node.

The PDF Reference notes that when navigating between nodes, it is possible to have *transition effects*. This is a very exciting feature that APB supports.

When there is a request to navigate to another page, the following events occur:

1. The root navigation node, if there is one, becomes the current node for the page.
2. If the navigation request was forward or if the navigation request was by a link to another page, the actions of the **NA** key of the root node are executed, and the current node becomes the one pointed to by the **Next** key.
If the navigation request is backward, the actions specified by the **PA** key are executed and the node specified by **Prev** becomes the current node.
3. The viewer application makes a new page, and displays it, using a transition effect, if any is specified.

The important thing to note is that when the viewer application changes page, either forward or backward, *the root node* (at the top of the page) *becomes the current node*.

The next slide illustrates this node structure, as implemented in APB.

APB Implementation

Begin by going into **fullscreen mode**. Forward navigation is accomplished by pressing the right-arrow key, by left-clicking the mouse, or by pressing the Enter key.

The previous slide shows the APB implementation and naming convention. The document author never references the node names, these are internal to APB; nodes are automatically and indirectly created by using the standard “Bld” commands. The next and previous actions are automatically inserted. The forward action normally makes a layer visible and the previous action hides a layer.

In the space below there are a series of build commands, move forward (right-arrow, or left-click on mouse) and backward (left-arrow or right-click on mouse) through them to get a feel for sub-page navigation. Forward click to the next page when you get to the bottom of the list of nodes, come back again using backward navigation. Explore, have fun, there are a few surprises.

This is a continuation of the previous, exploratory page. *Continuez, s'il vous plaît.*

New Options

The sub-page navigation feature, of course, comes with numerous new options, a few for the package, and many for the various standard build commands.

New Options for Package

To use the sub-page navigation feature, you need to use the `snavi` option

```
\usepackage[snavi]{apb}
```

The code will be automatically included. If you plan to present this document as part of a formal “powerpoint-like” talk, there is no need to include various navigation buttons.

When you drop out of fullscreen mode, if the buttons are present, you can use them to navigate from layer to layer, though some functionality is lost, others gained.

When compiling the document *without* the `snavi` option, the document will have to be navigated via the APB buttons, even in fullscreen mode.

For the purpose of development, I created some debugging messages that are written to Acrobat console. You can see these messages by taking the `sptrace` option.

```
\usepackage[snavi,sptrace]{apb}
```

New Options for Standard Build Commands

The standard builds referred to several times prior to this point in the document are general build commands `\Bld`, `\rBld`, `\bBld`, `\fBld`, and the structured commands `\dBld`, `\dPt`, `\ddPt`, `\ddPt`, `\Item`, `\IIItem` and `\IIIItem`. Each has additional optional arguments for controlling the basic components of the current navigation node. These options are listed below.

1. `nextaction`: The JavaScript to be executed when the user requests forward navigation from the current node. The default action is `\setLayerNext`, which shows the associated layer. If custom actions are needed, use the `nextaction` key, but include `\setLayerNext`. For example, earlier in this file, an alert box appeared. That code is

```
\Bld[nextaction={\setLayerNext;app.alert("Welcome to sub-page
navigation AcroTeX style!");},
prevaction={\setLayerPrev;
console.println("Back it up backwards!")},
nextttrans={FlyInDown},
prevttrans={Fade}]
```

2. `prevaction`: The JavaScript to be executed when the user requests backward navigation from the current node. The default action is `\setLayerPrev`, which hides the associated layer. If a custom action is needed, include `\setLayerPrev` as part of the action. See the example above.
3. `nextttrans`: The transition effect to be used with the forward navigation. The value of this key is one of the recognized strings. For example, `nextttrans={FlyInDown}`.

4. `prevtrans`: The transition effect to be used with the backward navigation. For example, `prevtrans={Fade}`, as shown above. See the `apb_man.pdf` for a complete listing.
5. `dur`: The maximum number of seconds before the viewer application automatically should advance forward to the next navigation node. If this key is not present, no automatic advance occurs. The duration key is local to the build command that creates it; there is a global duration command, `\spNaviDuration`. By saying `\spNaviDuration{5}`, navigation automatically advances forward through all nodes, unless the command `\spNaviDurationClear` is executed.
6. `noNode`: This key is not part of the creation of a navigation node, but part of its destruction! Using `noNode` causes the build command not to create a node for itself. This useful when does some special effects such as rearranging the order the layers appear.