

PDF HACKS™

100 Industrial-Strength Tips & Tools



O'REILLY®

Sid Steward

HACK
#15

Bookmark PDF Pages in Reader

Create and maintain a list of PDF pages for rapid access.

Web browsers enable you to bookmark HTML pages, so why doesn't Adobe Reader enable you to bookmark PDF pages? Here is a JavaScript that extends Reader so that it can create bookmarks to specific PDF pages. It works on Windows, Mac, and Linux.



The bookmarks created by this JavaScript aren't PDF bookmarks that get saved with the document. They behave more like web browser bookmarks in that they enable you to quickly return to a specific PDF page.

Bookmark JavaScript for Acrobat and Reader

Visit http://www.pdfhacks.com/bookmark_page/ to download the JavaScript in [Example 2-1](#). Unzip it, and then copy it into your Acrobat or Reader *JavaScripts* directory. “[Customize Acrobat Using JavaScript](#)” [\[Hack #96\]](#) explains where to find this directory on your platform. Restart Acrobat/Reader, and *bookmark_page.js* will add new items to your View menu.

Example 2-1. Adding bookmark functionality to Acrobat and Adobe Reader

```
// bookmark_page.js, ver. 1.0
// visit: http://www.pdfhacks.com/bookmark_page/

// use this delimiter for serializing our array
var bp_delim= '%##%#';

function SaveData( data ) {
    // data is an array of arrays that needs
    // to be serialized and stored into a persistent
    // global string
    var ds= '';
    for( ii= 0; ii< data.length; ++ii ) {
        for( jj= 0; jj< 3; ++jj ) {
            if( ii!= 0 || jj!= 0 )
                ds+= bp_delim;
            ds+= data[ii][jj];
        }
    }
    global.pdf_hacks_js_bookmarks= ds;
    global.setPersistent( "pdf_hacks_js_bookmarks", true );
}

function GetData() {
    // reverse of SaveData; return an array of arrays
    if( global.pdf_hacks_js_bookmarks== null ) {
        return new Array(0);
    }
}
```

Example 2-1. Adding bookmark functionality to Acrobat and Adobe Reader (continued)

```

var flat= global.pdf_hacks_js_bookmarks.split( bp_delim );
var data= new Array();
for( ii= 0; ii< flat.length; ) {
    var record= new Array();
    for( jj= 0; jj< 3 && ii< flat.length; ++ii, ++jj ) {
        record.push( flat[ii] );
    }
    if( record.length== 3 ) {
        data.push( record );
    }
}
return data;
}

function AddBookmark() {
    // query the user for a name, and then combine it with
    // the current PDF page to create a record; store this record
    var label=
        app.response( "Bookmark Name:",
                      "Bookmark Name",
                      "",
                      false );
    if( label!= null ) {
        var record= new Array(3);
        record[0]= label;
        record[1]= this.path;
        record[2]= this.pageNum;

        data= GetData();
        data.push( record );
        SaveData( data );
    }
}

function ShowBookmarks() {
    // show a pop-up menu; this seems to work only when
    // a PDF is already in the viewer;
    var data= GetData();
    var items= '';
    for( ii= 0; ii< data.length; ++ii ) {
        if( ii!= 0 )
            items+= ', ';
        items+= ''+ ii+ ': '+ data[ii][0]+ ''';
    }
    // assemble the command and then execute it with eval()
    var command= 'app.popUpMenu( '+ items+ ' );';
    var selection= eval( command );
    if( selection== null ) {
        return; // exit
    }
}

```

Example 2-1. Adding bookmark functionality to Acrobat and Adobe Reader (continued)

```

// the user made a selection; parse out its index and use it
// to access the bookmark record
var index= 0;
// toString() converts the String object to a string literal
// eval() converts the string literal to a number
index= eval( selection.substring( 0, selection.indexOf(':') ).toString() );
if( index< data.length ) {
  try {
    // the document must be 'disclosed' for us to have any access
    // to its properties, so we use these FirstPage NextPage calls
    //
    app.openDoc( data[index][1] );
    app.execMenuItem( "FirstPage" );
    for( ii= 0; ii< data[index][2]; ++ii ) {
      app.execMenuItem( "NextPage" );
    }
  }
  catch( ee ) {
    var response=
      app.alert("Error trying to open the requested document.\nShould I remove
this bookmark?", 2, 2);
    if( response== 4 && index< data.length ) {
      data.splice( index, 1 );
      SaveData( data );
    }
  }
}
}

function DropBookmark() {
  // modeled after ShowBookmarks()
  var data= GetData();
  var items= '';
  for( ii= 0; ii< data.length; ++ii ) {
    if( iil= 0 )
      items+= ', ';
    items+= ''+ ii+ ': '+ data[ii][0]+ ''';
  }
  var command= 'app.popUpMenu( '+ items+ ' );';
  var selection= eval( command );
  if( selection== null ) {
    return; // exit
  }

  var index= 0;
  index= eval( selection.substring( 0, selection.indexOf(':') ).toString() );
  if( index< data.length ) {
    data.splice( index, 1 );
    SaveData( data );
  }
}
}

```

Example 2-1. Adding bookmark functionality to Acrobat and Adobe Reader (continued)

```
function ClearBookmarks() {
    if( app.alert("Are you sure you want to erase all bookmarks?", 2, 2 )== 4 ) {
        SaveData( new Array(0) );
    }
}

app.addItem( {
    cName: "-", // menu divider
    cParent: "View", // append to the View menu
    cExec: "void(0);" } );

app.addItem( {
    cName: "Bookmark This Page &5",
    cParent: "View",
    cExec: "AddBookmark();",
    cEnable: "event.rc= (event.target != null);" } );

app.addItem( {
    cName: "Go To Bookmark &6",
    cParent: "View",
    cExec: "ShowBookmarks();",
    cEnable: "event.rc= (event.target != null);" } );

app.addItem( {
    cName: "Remove a Bookmark",
    cParent: "View",
    cExec: "DropBookmark();",
    cEnable: "event.rc= (event.target != null);" } );

app.addItem( {
    cName: "Clear Bookmarks",
    cParent: "View",
    cExec: "ClearBookmarks();",
    cEnable: "event.rc= true;" } );
```

Running the Hack

When you find a PDF page you want to bookmark, select View → Bookmark This Page, or use the Windows key combination Alt-V, 5. A dialog will ask you for the bookmark's name. Enter a title and click OK.



The bookmarks you create persist across Acrobat/Reader sessions because the data is stored in a persistent, global JavaScript variable. Acrobat/Reader stores persistent data in a text file named *glob.js* in the *JavaScripts* directory.

To activate a bookmark, some PDF document (*any* PDF document) must already be open in the viewer. This is due to a quirk in Acrobat's JavaScript pop-up menu. Select View → Go To Bookmark, or use the Windows key

Bookmark PDF Pages in Reader

combination Alt-V, 6. A pop-up menu appears, showing your bookmark titles. Select a title, and that PDF will open in the viewer and then advance to the bookmarked page.

Remove a bookmark by selecting View → Remove a Bookmark. A pop-up menu will show all of your current bookmarks. Select a bookmark, and it will be removed permanently.



The pop-up menu of bookmarks appears as a floating menu next to where the View menu was. That floating menu does work. Also, you'll need to have a PDF file open for the bookmark choices on the View menu to be active.