AcroTEX.Net

Seeding a Signature Field with AeB Pro

D. P. Story

AeB Blog March 28, 2009

http://www.acrotex.net
Published: March 28, 2009

To: Honorable Barrister Maxwell Frimpong
From: D. P. Story
Subject: On Business Proposal

March 23, 2009

Dear Mr. Frimpong;

Thank you for thinking of me concerning an "important business proposal" in your recent and brief email to me on March 23, 2009. Recovering $12,000,000 (twelve million Us dollars) in claims sounds intriging and exciting to me. Such a large amount of money would certainly come in handy in these tough times. Yet, regrettably, I must decline you kind offer; though I am in retirement, I am, none-the-less, quite busy lately sorting my button collection, and don't really have the time to pick up all this easy money.

Thank you again, Barrister Frimpong, for your offer. Please keep me in mind should future opportunities arise.

Best regards,

Note: Acrobat required so sign

Dr. D. P. Story
Department of Mathematics,
The Talking University
Talkville, FL 12345
dpstory@uakron.edu

In this example, we seed the signature field but do not sign it. Signing, perhaps, comes later through the user interface.

As in the previous blog articles on signatures, we use `\sigField` to create the field.

```
Best regards,\\
\sigField{sigOfDPS}{2.5in}{4\baselineskip}\\[3pt]
Dr.\ D. P. Story\\
Department of Mathematics,\\
The Talking University\\
Talkville, FL 12345\\
\texttt{dpstory@uakron.edu}
```

The field is created then seeded using the `\signatureSetSeedValue` command of AeB Pro. This command expands to execute the security restricted JavaScript method *`Field`*`.signatureSetSeedValue`. The field is seeded as part of the `docassembly` event.

```
\begin{docassembly}
var sv={
    mdp: "defaultAndComments",
    reasons: ["This is a reason", "This is a better reason"],
    flags:0x8
};
var oSigField=this.getField("sigOfDPS");
\signatureSetSeedValue(sv);
\end{docassembly}
```

We create a JavaScript object `sv`, and populate it with key-value pairs that are to be passed to *`Field`*`.signatureSetSeedValue`. The first property (with key `mdp`) makes this into a certifying signature field, the `reasons` key restricts the reason the `signer` can use when he/she sign the field. The `flags` property makes it mandatory that one of these listed reasons be used. Next, we get the field object `oSigField` for the signature field; the field object *must be named* `oSigField`, this is the JavaScript variable that `\signatureSetSeedValue` expects the field object to be named. We finish up by passing this object `\signatureSetSeedValue(sv)` and we're done.

Information on signatures can be found at the Acrobat Developer Center; or go to the Acrobat Security page; look for the document titled *Digital Signature User Guide for Acrobat 9.0 and Adobe Reader 9.0*.

The *JavaScript for Acrobat API Reference*[1] for details on these methods and their parameters.

Now, back to my retirement.

[1] Available through the Acrobat Developer Center.