

AcroTeX.Net

Certification Signing using AeB Pro

D. P. Story

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 intriguing 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.

Regards,

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

1. Creating and Signing a Certification Signature Field

The eforms package can create a signature field with the `\sigField` command, and using the `aeb_pro` package with its `docassembly` environment, can also sign the field from a \TeX source.

The \TeX code for creating the signature field of this document is

```
Best regards,\n\n\sigField{sigOfDPS}{2.5in}{4\baselineskip}\n\n[3pt]\n\nDr.\ D. P. Story\n\nDepartment of Mathematics,\n\nTalking University\n\nTalkville, FL 12345\n\n\texttt{dpstory@uakron.edu}
```

The `\sigField` command appears in the second line, and uses the usual syntax for form fields, as defined in the eforms package. The optional parameter, shown used in this example, is used to change the appearance of the (unsigned) field, and to associate JavaScript actions.

Once the field is created, it can be signed using the Acrobat user interface, or programmatically, from the \TeX source file.

What is certification signing? To quote *Digital Signature User Guide for Acrobat 9.0 and Adobe Reader 9.0* found at the Acrobat Security

Certifying a document enables the first signer attest to its contents and specify the types of changes permitted for the document to remain certified. Certification helps document authors and recipients determine that documents are legitimate and tamper-proof, thereby enabling trustworthy online transactions and more secure communications.

To obtain certification signing, the field method `signatureSetSeedValue` is used to seed the field (before signing) with the `mpd` key which is used to force a certification signature. The values of `mpd` can be `allowNone`, `default`, `defaultAndComments`, and `allowAll`. Again from *Digital Signature User Guide for Acrobat 9.0 and Adobe Reader 9.0*

- `allowAll`: Do not use `allowAll` unless you want to force an approval signature since this value results in MDP not being used for the signature and therefore doesn't force a certifying signature.

- `allowNone`: Document changes invalidate the signature and lock the author's signature. `allowNone` bypasses any custom `legalAttestations` because no document changes can occur and the user does not therefore need to be warned about malicious content. Do not use with `legalAttestations`.
- `default`: Allow form field fill-in if fields are present in the document as well as additional signatures. Other changes to the document invalidates the signature.
- `defaultAndComments`: Allow form field fill-in if fields are present in the document and allows annotations (comments) to be added, deleted, or modified as well as additional signatures. Other changes to the document invalidates the signature. Note that annotations can be used to obscure portions of a document and thereby affect the visual presentation of the document.

The code for the signature field of this document is

```
\begin{docassembly}
\sigInfo{
  cSigFieldName: "sigOfDPS",
  cert: "D_P_Story.pfx", password: "mypassword",
  cLegalAttest: "Trust me and be at ease.",
  oInfo: { location: "Niceville, FL",
    reason: "This is a reason",
    contactInfo: "dpstory@acrotex.net",
    appearance: "My Signature" },
  sv: { mdp: "defaultAndComments" }
};
\signatureSign
\end{docassembly}
```

where I have changed the value of the password key to protect my secrets. The first command, `\sigInfo`, creates an JavaScript object, `oSigInfo`. The command `\signatureSign` uses the information in this object to sign the field designated by the `cSigFieldName` property. When `\signatureSign` is expanded, the JavaScript created tests where there is a `sv` property in the `oSigInfo` object; if so, it gathers the data from the `sv` property, and calls `oSigField.signatureSetSeedValue`, where `oSigField` is the field object of the signature field.

Notice that the value of `mdp` is "defaultAndComments", so form fill-in are available, and comments are allowed without invalidating the certification signature.

Additional 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.

¹Available through the Acrobat Developer Center.