

To: Honorable Barrister Maxwell Frimpong
From: D. P. Speaker
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 your 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,

Dr. D. P. Speaker
Department of Rhetoric,
Talking University
Talkville, FL 12345
dpspeaker@talking.edu

1 Creating and Signing a 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 \LaTeX source.

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

```
Best regards,\  
\sigField{sigOfDPS}{2.5in}{4\baselineskip}\[3pt]  
Dr.\ D. P. Speaker\  
Department of Rhetoric,\  
Talking University\  
Talkville, FL 12345\  
\texttt{dpspeaker@talking.edu}
```

The `\sigField` command appears in the second line, and uses the usual syntax for form fields, as defined in the `eforms` package. Here's what the field looks like when it is unsigned.

Once the field is created, it can be signed using the Acrobat or Adobe Reader (version 10.0.1 or later) user interface.

A signature field may also be signed programmatically from the \LaTeX source file using AeB Pro. The first signature field of this document was automatically signed when the newly created PDF is brought into Acrobat.

```
\begin{docassembly}  
\sigInfo{  
  cSigFieldName: "sigOfDPS",  
  cert: "<name>.pfx", password: "<password>",  
  oInfo: {  
    location: "AcroTeX Central, FL",  
    reason: "I am approving this document for distribution",  
    contactInfo: "dpspeaker@talking.edu",  
    appearance: "My Signature" }  
};  
\signatureSign  
\end{docassembly}
```

The script above was used to programmatically sign the signature field with field name of `sigOfDPS`. 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.

The general syntax of the argument of `\sigInfo` is displayed below.

```
\sigInfo{
  cSigFieldName: "<sigFldName>",
  oHandler: <securityHandlerObj>,
  path2Cert: <path>,
  cert: "<certName>.pfx",
  password: "<password>",
  bUI: true|false;
  oInfo: {
    <various properties>
  }
}
```

The first five keys are used by AeB Pro to build the proper information needed sign the field.

cSigFieldName (required) The name of the signature field to be signed.

oHandler (optional) The security handler to be used. If this key is not present, `security.PPKLiteHandler` is used.

path2Cert (optional) The device-independent path to the PFX file to be used. If not present, AeB Pro uses the name provided by the `cert` key and builds the path to where Acrobat normally saves digital IDs.

cert (optional) The name of the digital ID to be used for signing. Optional if `path2Cert` is provided, required otherwise.

password (required) The password for the PFX file provided, necessary to login to the security handler (`oHandler`).

bUI (optional) A Boolean, if `true` the security handler displays the user interface when signing. The default is `false`.

`oInfo` (optional) The `oInfo` property key is one that appears in the argument property list of the `Field.signatureSign()` method. It is passed to the argument of this JavaScript method. See the documentation of `Field.signatureSign()` in the *JavaScript for Acrobat API Reference* for more information.

Additional information on signatures is found at the **Acrobat Developer Center**.¹ Refer to the *JavaScript for Acrobat API Reference*, also found at the **Acrobat Developer Center**, for details on these methods and their parameters. Adobe is notorious for moving its reference documents and renaming them, year after year. Good luck searching the Adobe web site for the references you need.

2 Compiling this file

In the preamble of this document, the `docassembly` environment is found:

```
\begin{docassembly}
\sigInfo{
  cert: "<name>.pfx", password: "<password>",
  oInfo: {
    location: "AcroTeX Central, FL",
    reason: "I am certifying this document",
    mdp: "defaultAndComments",
    contactInfo: "dpspeaker@talking.edu"
  }
};
\certifyInvisibleSign
\end{docassembly}
```

To compile this document yourself, you need to create a digital ID using Acrobat. Replace `<name>` with the file name of your digital ID. and of course replace `<password>` with the password you selected when you created your digital ID. Modify the `oInfo` property as designed.

Now, back to my retirement.

¹http://www.adobe.com/go/acrobat_developer