

AeB and AeB Pro

Including Acrobat in the L^AT_EX workflow

D. P. Story

1. Introduction

Adobe Acrobat (AA) is a desktop application that has many powerful features beyond the functionality of a PDF viewer. In this article the focus is a L^AT_EX workflow that includes AA in its workflow.

AeB and AeB Pro, as well as other packages written under the AeB banner, use FDF files in various situations (as described in Section 4).¹ These packages use Acrobat to import the FDF files to perform designated JavaScript actions as required by the package or document author. The importing of FDF files is only needed during the PDF creation stage of document development; once the PDF file is created, the document can be viewed by Adobe Reader or any other PDF viewer.

For FDF files to be imported, the Acrobat application may need to be configured.

2. Configuration of Acrobat

The current version of AA is Adobe Acrobat DC. In this and previous versions of AA no special configuration is needed; however if you purchased AA after December 2020, or you updated your AA after December 2020, you need to make one change in the Preferences.

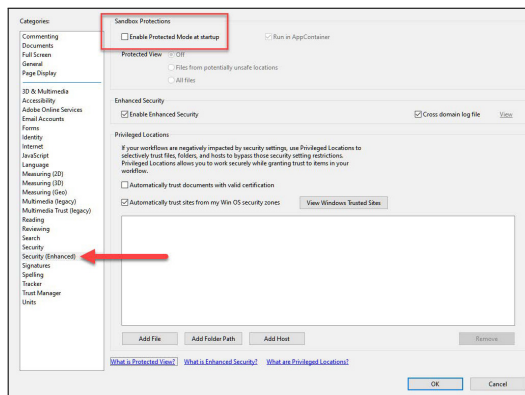


Figure 1: Security (Enhanced) Preferences

Acrobat DC users, action required!

Start Acrobat and open the menu Edit > Preferences (Ctrl+K). In the Preferences dialog box, select Security (Enhanced) category from the left panel; the top most line of the right panel is a checkbox labeled Enable Protected Mode at startup. *Clear this checkbox.*² Refer to Figure ?? for a visual. If this checkbox is not present in your Acrobat, there is nothing to do until this checkbox appears in a future updated version of your AA application.

¹An FDF file is a file type that was created by Adobe. The Acrobat and Adobe Reader applications can use FDF files in various ways.

²When the box is checked, importing FDF files, as described in the next two sections, is *prohibited*. We want FDF files imported, so we clear the checkbox.

3. When FDF files are imported

It is important to be aware of when and how FDF files are imported, and to understand the actions required of the document author in response to this “import FDF event.”

For a document that uses FDF files to perform tasks, the files are imported *after* PDF creation when the newly created PDF document is *first opened* in Acrobat, and *before the PDF file is moved to another location and before any auxiliary files (the FDF files) are deleted*. Once the FDF files are imported, it is important *to save* the PDF file, this incorporates the results of the actions taken by the imported FDF files into the document, actions, such as the embedding of document JavaScript.

Another important point to realize is that the use of FDF files is not restricted to the dvips->Distiller->Acrobat method of creating a PDF; indeed, any PDF creator used by the \LaTeX community may be used, for example, pdftex->Acrobat. The only prerequisite is to use \LaTeX markup that initiates the “import FDF event.” This markup is available through packages that support it; this includes, but not restricted to, AeB and AeB Pro.

4. Applications of FDF files to a \LaTeX workflow

With regard to producing a PDF file through a \LaTeX workflow, there are two important applications to importing FDF files: importing document JavaScript and post-PDF creation actions. These two applications to FDF files are covered in the next two subsections.

4.1. Document JavaScript

In this case, FDF files are required only for the dvips->Distiller->Acrobat workflow; the other PDF creators (pdflatex, lualatex, and xelatex) have primitives that facilitate the creation of document JavaScript without FDF files or Acrobat.

For the exerquiz package, part of the Acrobat \LaTeX eEducation Bundle (AeB), creation of Document JavaScript is automatic and invisible for all PDF creators; it is only in the tex->dv \rightarrow dvips->Distiller->Acrobat workflow that FDF files are used.

Document JavaScript, by the way, is code that is embedded in the PDF and is available to be called throughout the document. Usually, Document JavaScript is general code that is repeatedly used.

4.2. Post PDF Creation actions

The AeB Pro Bundle has many “post PDF creation” actions, such as auto saving the document, importing sounds, attaching documents, inserting pages, extracting pages, and many more. With proper \LaTeX markup, when the document is compiled, FDF files are written to the hard drive, when the document is then first opened in Acrobat, the FDF files are imported and the actions are executed. Again, for the FDF files to be imported, they must not have already been deleted, and Acrobat must be correctly configured (Section ??). One very simple example, is automatic saving of the document when Acrobat is first opened. For example, an aeb_pro document has a docassembly environment that is placed in the preamble. Within this environment, place any post PDF creation JavaScript actions:

Save the PDF

```
\begin{docassembly}
console.show();
console.println("Saving document");
\executeSave();
\end{docassembly}
```

The `\executeSave` command is defined by `aeb_pro` and its action is to save the document. It is the `docassembly` environment that writes the FDF file that contains its contents. Acrobat inputs this command on opening and the PDF is saved.³

The `docassembly` package. This package uses the post PDF creation actions described above, but is a stand alone package and is available to all \LaTeX workflows who own the Acrobat application. The `docassembly` package defines the `docassembly` environment and supports the same functionality of as `AeB Pro`. The `docassembly` package may be downloaded from CTAN at <https://ctan.org/pkg/docassembly>.

³`\executeSave` must be the last post PDF creation action; otherwise, as later actions may not be executed.