The History of pdfChip

pdfChip was a long time in the making; it originated from a range of feature requests in other callas software products such as pdfToolbox and pdfaPilot. This chapter explains briefly where the idea for a HTML to PDF conversion tool comes from and where you may have seen (parts of it) before.

A customizable Preflight Report

For a very long time, the pdfToolbox and pdfaPilot products from callas software have allowed quality control and fixing of PDF documents for various purposes. In such a scenario you want to be able to communicate what exactly has been fixed or what errors and warnings have been found, so both products have the notion of a preflight report that details all of that information.

Preflight reports come in various flavors; some are better suited for automated processing (such as text based or XML based preflight reports), others are better suited for human consumption such as HTML or PDF preflight report. About 2005, callas software came up with three different versions of a PDF preflight report that show information either using annotations, transparent overlays or PDF layers.

In essence though, all three reports were static; customers were not able to modify the information on them nor the branding of the preflight report. Having a customisable preflight report that could be adjusted to the needs of a customer became one of the most frequent feature requests for both pdfToolbox and pdfaPilot.

Not another report language...

The *customary* way to implement a custom preflight report would be to come up with some sort of "report language" or "report template system" in order to allow customers to modify the look and feel of the generated reports. The problem with this of course is that it is yet another "language" system integrators and customers must master and that such languages are typically either very complex or very limited.

Rather than going down that path, it was decided to develop an HTML template and convert that template on the fly to good PDF. HTML is a well-known and stable standard and lots of people know how to create HTML files. The tools for it are ubiquitous, and the cascading style sheet (CSS) system provides ample branding capabilities. On top of that there are a number of very fast, stable and open source HTML engines that can be used to handle the heavy lifting around interpreting the HTML and CSS. In this case WebKit was selected as the engine.

The report engine could also take advantage of WebKit's support for Javascript and the HTML templates for the report use Javascript to integrate with the preflight engine and insert the actual results from the preflight into the generated PDF file.

This new preflight report was introduced with pdfToolbox 6 and pdfaPilot 4 and proved to be very flexible, more powerful than originally thought and very popular.

Email Archival done Right

When pdfaPilot 5 was discussed a similar problem raised its head; the principal new feature for pdfaPilot 5 was to allow conversion of emails into archivable PDF documents. This posed two challenges:

- Different companies want different "views" or "representations" for their archived emails. Again this could have been solved with a custom "template" language, but the previously developed HTML template strategy was again easier and more flexible.
- The more severe challenge rested with the fact that archiving emails is not trivial. To be compliant with PDF/A, a whole list of demands had to be placed on the generated PDF file and the HTML to PDF engine had to flexibily support all of those. Emails also came with hyperlinks, metadata, attachements and other more advanced features that had not been needed to generate simple preflight reports.

In the end the same HTML to PDF engine was used, but it of course received a significant update to allow it to support first class email to archivable PDF conversion. And that up-

date started the thinking that this engine had a reason to exist by itself as a separate product; the birth of pdfChip.