Generating PDF for e-reader devices

Abstract
NotuDoc is a commercial Internet application that uses ConTEXt for the on-the-fly generation of PDF documents for, amongst other things, the e-reader devices of iRex technologies. This articles offers a glimpse behind the scenes.

Introduction

This article is about the generation of PDF documents for e-reader devices. Before we delve into that, let's look at the controlling application (NotuDoc) that this process is a part of, and do a short introduction of the e-reader devices that are used.

Generating PDF from within NotuDoc is a fairly small part of the application, but even so there is a fair amount of complexity.

NotuDoc

The Dutch company NotuBiz (http://www.notubiz.nl) focuses on everything related to the recording and publishing of (council) meeting reports using modern media. For example, NotuBiz takes care of live streaming and the publication of digital meeting reports on the Internet. The clients of NotuBiz are local government bodies in The Netherlands, but increasingly also in neighboring countries.

In practical use, it turned out that the information stream *leading up to* the actual meetings was far from optimal. NotuDoc was born out of this realization: NotuDoc is an Internet application that links (preliminary or final) meeting agendas to the corresponding meeting documents like commission reports, presentations and quotations. Afterwards, the resulting combination is made available to the relevant meeting parties via the Internet and/or PDF document export functionality.

By gathering and combining all the necessary documents in one place, it becomes easier for the participants to prepare for the meeting. As a bonus, afterwards the official meeting report can be linked to already existing meeting data easily and therefore everything is set up for near-effortless publication to the members of the community (as is required by law).

NotuDoc is a plug-and-go commercial product with quite extensive configuration possibilities. This is important because the web interface has to integrate nicely with the layout of the client's website, but it goes further than that: not all clients have the same meeting structures and only a very few use the same internal document management system (DMS). NotuDoc comes with pre-installed support for the most commonly used systems in the Netherlands, and the extensibility ensures that support for other DMS-s is easily added.

The NotuDoc code is implemented and maintained by Elvenkind in close cooperation with NotuBiz and is based on Elvenkind's development framework, written using Perl 5.

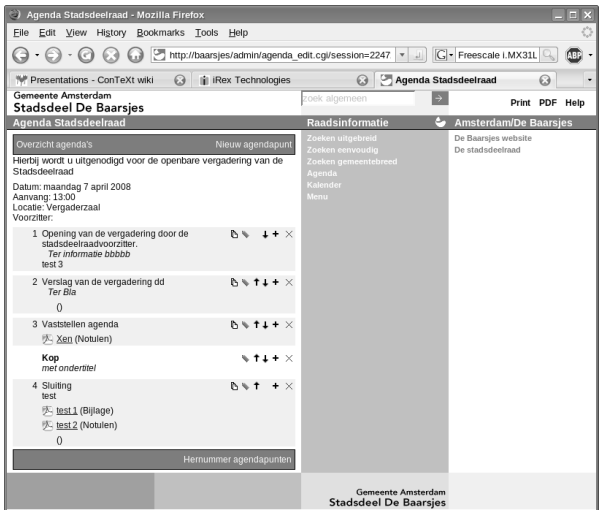


Figure 1. An example main screen of the NotuDoc Internet application

E-reader devices

At present, NotuDoc comes prepared for the generation of PDF documents for two specific e-reader devices, both developed by iRex Technologies (http://www.irextechnologies.com), a spin-off of Philips International. Besides these two specific devices (iLiad and DR1000) it is naturally also possible to generate PDF for printing and for interactive work on a computer / notebook.

Both the iLiad and the newer DR1000 are based on the same core technology. The iLiad has been around for a few years now and amongst other things it is the publishing platform for the digital version of 'NRC Handelsblad' (a Dutch newspaper comparable to the Wall Street Journal). The DR1000 has a new design and it

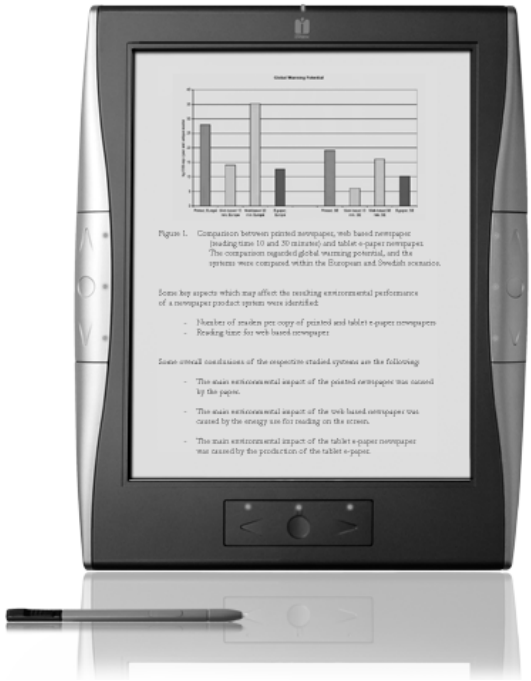


Figure 2. iLiad (left) and DR1000 (right).

offers a larger screen and somewhat faster hardware, but technologically there are very few differences between the two devices.

Both devices are based on 'digital paper', a technology whereby the displayed data stays visible without the need to refresh the screen many times a second.

The key advantage of this technology is that it uses far less power, resulting in much longer battery life when compared to traditional TFT or LCD screens. Another good thing is that because there is no need for a back light, actually looking at the screen is a lot easier on the eyes.

On the other hand, there are downsides to electronic paper. The two largest of these: the reaction time, which is much slower than for conventional computer displays, and the output is grayscale only. Hopefully, future developments will remove both limitations.

Both devices make use of 'Wacom Penabled' technology (http://wacom.com/tabletpc/what_is_penabled.cfm) that makes it possible to write and sketch directly on the display, so that you can create notes directly into the PDF. The companion software (for Windows) is able to merge these notes with the original PDF document into a new PDF document for later use.

The supported formats for both devices are the same as well: PDF, HTML, mobipocket, and a few bitmap image formats. All software used and developed by iReX is open source, based on a Linux distribution for embedded devices. Connection to the PC for exchanging documents is done via USB or optionally (for iLiad) using a wireless network. Both iLiad and DR1000 use removable memory cards as storage medium.

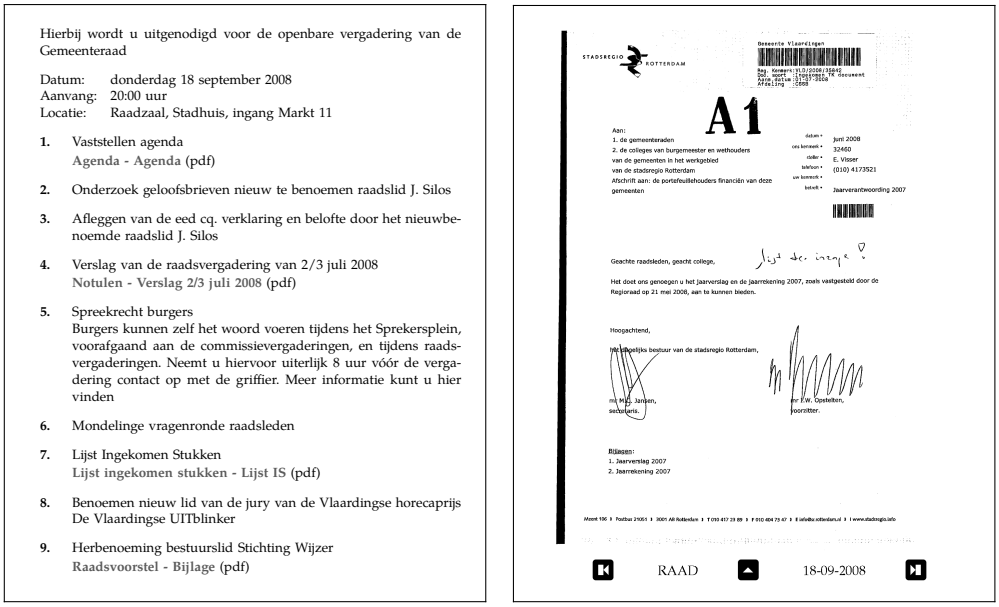
PDF generation

The PDF generation in NotuDoc is handled by a Perl script that is completely template driven. It uses near-identical code both for the generation of TEX input files and for HTML pages. Only the character escape functions and filenames are adjusted specifically for TEX. Just like the web pages, the PDF documents are generated at runtime by calling `texexec`. The TEX subsystem uses ConTEXt supplied by the 'contextgarden' distribution (http://minimals.contextgarden.net/).

ConTEXt templates

For each client, the application stores a setting for the desired PDF output type. The following example uses `iliad`, but it can also be something else like `dr1000` or just `a4`. Separate from this global preference, it is possible to define a client-specific layout that creates a document layout that corresponds to the desired house style.

Layout definitions are implemented via ConTEXt macros that are separated out from the Internet application. The application only takes care of converting the database records of a meeting agenda into a ConTEXt input source and exporting the required meeting doc-



Hierbij wordt u uitgenodigd voor de openbare vergadering van de Gemeenteraad

Datum: donderdag 18 september 2008
Aanvang: 20:00 uur
Locatie: Raadzaal, Stadhuis, ingang Markt 11

1. Vaststellen agenda
   Agenda - Agenda (pdf)
2. Onderzoek geloofsbrieven nieuw te benoemen raadslid J. Silos
3. Afleggen van de eed cq. verklaring en belofte door het nieuwbenoemde raadslid J. Silos
4. Verslag van de raadsvergadering van 2/3 juli 2008
   Notulen - Verslag 2/3 juli 2008 (pdf)
5. Spreekrecht burgers
   Burgers kunnen zelf het woord voeren tijdens het Sprekersplein, voorafgaand aan de commissievergaderingen, en tijdens raadsvergaderingen. Neemt u hiervoor uiterlijk 8 uur vóór de vergadering contact op met de griffier. Meer informatie kunt u hier vinden
6. Mondelinge vragenronde raadsleden
7. Lijst Ingekomen Stukken
   Lijst ingekomen stukken - Lijst IS (pdf)
8. Benoemen nieuw lid van de jury van de Vlaardingse horecaprijs De Vlaardingse UITblinker
9. Herbenoeming bestuurslid Stichting Wijzer
   Raadsvoorstel - Bijlage (pdf)

STADSREGIO ROTTERDAM

Gemeente Vlaardingen

A1

Aan:
1. de gemeenteraden
2. de colleges van burgemeester en wethouders
van de gemeenten in het werkgebied
van de stadsregio Rotterdam
Afschrift aan: de portefeuillehouders financiën van deze gemeenten

datum: juni 2008
ons kenmerk: 32460
datum: E. Visser
telefoon: (010) 4173521
uw kenmerk:
betreft: Jaarverantwoording 2007

Geachte raadsleden, geacht college,

Het doet ons genoegen u het jaarverslag en de jaarrekening 2007, zoals vastgesteld door de Regioraad op 21 mei 2008, aan te kunnen bieden.

Hoogachtend,

het dagelijks bestuur van de stadsregio Rotterdam,

mr. M.G. Jansen, secretaris.

mr. E.W. Oppelaar, voorzitter.

Bijlagen:
1. Jaarverslag 2007
2. Jaarrekening 2007

Meent 106 | Postbus 21051 | 3001 AB Rotterdam | T 010 417 28 88 | F 010 404 75 47 | E info@sr.rotterdam.nl | I www.stadsregio.info

RAAD 18-09-2008

Figure 3. The first and one of the following pages of a PDF generated for the iLiad

uments to PDF files. Everything else is handled by ConTEXt macros; the application only copies from a template include file into the TEX file. The used files are:

agenda-iliad.tex

This is the main TEX file, and in this file two different types of replacements take place.

In the listing below you see two lines that look like HTML syntax for so-called 'server side includes'. That is not a coincidence: as mentioned above, the application uses the same code for TEX files and HTML page generation.

The two `#include` files are read by the Perl script, and inserted in that place in the TEX output. The contents of those files are explained in the next paragraph.

The second type of replacement deals with the words in all caps between # markers. These keywords are replaced by the actual content (and meta-data) of the meeting. The keyword `#LIST#` is the most important of those because effectively that contains the whole content of the agenda, which is built up recursively.

A meeting agenda consists of meta-information like place and time, and a variable number of meeting items. Items can be organized into categories, and for each item there can be an optional number of related meeting documents. All of this is controlled by small template files that are inserted at various levels. Their names are predefined system constants.

```
\unprotect
<!--#include src='agenda-macros-00.tex' -->
<!--#include src='agenda-macros-iliad.tex' -->
\protect

\starttext
\startagenda[Gremium={#GREMIUM#},
             Datum={#DATUM#},
             Datumkort={#DATUMKORT#},
             Categorie={#CATEGORIE#},
             Aanvang={#AANVANG#},
             Locatie={#LOCATIE#},
             Aanhef={#AANHEF#},
             Koptitel={#TITLE2#},
             Titel={#TITLE#}]

\startpunten
#LIST#
\stoppunten

\stopagenda
\stoptext
```

header_line.tinc

This template is used for any category section headings.

```
\startheaderline
 [Titel={#TEXT#},Pagina=#PAGE#,Aard={#AARD#}]
\startheaderbody
#BODY#
\stopheaderbody
\stopheaderline
```

puntnr_line.tinc
This is the template for each of the separate agenda items.
#BODY# contains the explanatory text for this item, #DOCS# is a placeholder for the list of relevant meeting documents. The latter is itself built up programmatically because there can be any number of relevant documents per item.

```
\startpunt
 [Nummer={#NR#},Titel={#PUNT#},Aard={#AARD#}]
\startpuntbody
#BODY#
\stoppuntbody
\startpuntdocs
#DOCS#
\stoppuntdocs
\stoppunt
```

puntdoc_line.tinc
This is the first of three possible templates for a meeting document. It is used for meeting documents (i.e. exported PDF files) that will be included in the generated PDF as appendices.

```
\agendadocument[#ICON#]{#LINK#}{#LABEL#}
```

puntnodoc_line.tinc
This template will be used for meeting documents that should be included as appendices, but for which it is decided (based on a configuration parameter) that they are too large for actual inclusion. A separate macro is used so that an explanatory text can be printed in the output.

```
\agendanodocument[#ICON#]{#LABEL#}
```

puntdoc_line_noembed.tinc
This is the third possibility, intended for non-PDF meeting documents like Microsoft Word documents and PowerPoint presentations. Because files in these formats cannot be handled in the PDF output, no hyperlink is possible; thus the keyword #LINK# is not present in this case.

```
\agendadocument[#ICON#]{}{#LABEL#}
```

ConTEXt macros

As mentioned above, the used ConTEXt macros are split over two separate files.

The first has the name agenda-macros-00.tex, and is used unaltered for all clients and all PDF layout. It contains a generic implementation of the macros we saw earlier in the template files. These macros only take care of the infrastructure; they don't do any layout themselves. Handling the layout is passed on to other macros via the ConTEXt command \directsetup.

Typical for the content of this file are macro definitions like this:

```
\def\dostartagenda[#1]%
  {\getparameters
      [Agenda]
      [Gremium=,Datum=,Datumkort=,
       Categorie=,Aanvang=,Locatie=,
       Aanhef=,Titel=,Koptitel=,
       Voorzitter=,
       #1]%
   \pagereference[firstpage]
   \directsetup{agenda:start}}

\def\stopagenda
  {\directsetup{agenda:stop}}
```

and this:

```
\def\agendadocument[#1]#2#3%
  {\doifnotempty {#2}
     {\doglobal \appendtoks
        \addimage{#2}{#3}\to \everyendagenda }%
   \def\DocumentType{#1}%
   \def\DocumentFile{#2}%
   \def\DocumentBody{#3}%
   \pagereference[#2-referer]
   \directsetup{agenda:document}}
```

The macro \addimage is the most interesting macro in this file. It receives the id and file name of an exported PDF document as arguments, and ensures that that PDF document is added page by page via \externalfigure. In slightly simplified form it looks like this:

```
\unexpanded\def\addimage#1#2{%
  \pagereference[#1]
  \xdef\previouspdf{\currentpdf}%
  \gdef\currentpdf{#1}%i
  \getfiguredimensions[#1.pdf]%
  \imgcount=\noffigurepages
  \dorecurse
      {\the\imgcount}
      {\externalfigure
          [#1.pdf]
          [page=\recurselevel,
           factor=max,
           size=cropbox]%
       \page}%
    \pagereference[#1-last]
}
```

In the appendices of the generated PDF (see figure 3) there is an extra interaction line at the bottom of the page containing three buttons that jump to the first page of the current appendix, the first page of the next appendix, and to the reference to this appendix in the meeting agenda itself. These hyperlinks use the values of `\currentpdf` and `\previouspdf`.

The needed setups and the general layout definitions are in the file `agenda-macros-iliad.tex`. This file can be instantiated with a specific version for a single client, but otherwise a generic version will be used: there is a default implementation file for each of the predefined PDF output types.

The PDF output layout for the e-reader devices differ from the layouts for PC screens or printer, but most of those differences are obvious. Of course there is a different (smaller) paper format. The room on an e-reader screen is limited, so it must be used optimally and therefore very small margins are used. PDF object compression is turned off because the e-reader hardware is very limited compared to a PC. The color support in ConTEXt is turned on, but only using grayscale.

The biggest difference is that the meeting documents that would normally remain separate files are included into the main output document. This makes moving the result file to the e-reader easier, but most important is that it improves the user friendliness of the result: external PDF links on the e-reader are either not supported at all (on the iLiad) or extremely slow (on the DR1000).

Parts of the content of `agenda-macros-iliad.tex`:

```
\definepapersize [iliad]
                 [width=124mm,height=152mm]

\setuppapersize [iliad] [iliad]

\enableregime[utf8]

\pdfminorversion = 4

\setuplayout[height=14.5cm,
             footer=12pt,
             footerdistance=6pt,
             width=11cm,
             topspace=12pt,
             header=0pt,
             backspace=24pt,
             leftmargin=12pt,
             rightmargin=12pt]

\setupcolors[state=start,conversion=yes,
             reduction=yes,rgb=no,cmyk=no]

\definecolor[papercolor][r=1,b=1,g=1]

...

\setupbackgrounds[page]
                 [state=repeat,
                  background=color,
                  backgroundcolor=papercolor]

...

\startsetups agenda:start
  \blank
  \setupfootertexts[\dofooteragenda]
  \setupfooter[state=high]
  \AgendaGremium
  \blank
  \starttabulate[|l|p|]
  \NC Datum:   \NC \ss\AgendaDatum\NC \NR
  \NC Aanvang: \NC \ss\AgendaAanvang\NC\NR
  \NC Locatie: \NC \ss\AgendaLocatie \NC\NR
  \stoptabulate
  \blank
\stopsetups

\startsetups agenda:stop
  \page
  \the\everyendagenda
  \everyendagenda={}
\stopsetups

\startsetups punten:start
  \startitemize[width=24pt]
\stopsetups

\startsetups punten:stop
  \stopitemize
\stopsetups

....

\startsetups agenda:nodocument
  {\DocumentBody
    {\tfx bestand te groot voor inclusie}\par }%
\stopsetups
```

Summary

The generation of PDF documents is a small but important part of NotuDoc. We chose to use TEX for the high quality of the output and ConTEXt in particular because of the simple methods it offers to separate the layout definitions from the actual data.

Taco Hoekwater
Elvenkind BV
taco (at) elvenkind dot com