DocBook-opsætning i Mikro Værkstedet

Rune Enggaard Lausen

Revision 1.7, 16. august 2004, Rune Enggaard Lausen

Revisionshistorie

Revision 1.7	2004-08-16 Corrected typo	Rune Enggaard Lausen
Revision 1.6	2004-08-16 Documentation update	Rune Enggaard Lausen
Revision 1.5	2004-08-16 Ooops! setup.dbx didn't validate!	Rune Enggaard Lausen
Revision 1.4	2004-08-16 Documentation updates	Rune Enggaard Lausen

I dette dokument beskrives hvorledes DocBook-miljøet i Mikro Værkstedet sættes op. Der er foreløbig opsætningsvejledninger for kommandoliniekørsler, GNU Emacs og `<oXygen/>`-editoren.

Når der skal indtastes stier, skal man bruge `/` i stedet for `\`.

Indholdsfortegnelse

1. Indledning 1
    1.1. Forudsætninger 2
    1.2. Konventioner 2
2. Første skridt 2
    2.1. Installer/opdater nødvendige Cygwin-pakker 2
    2.2. Sæt environment variabel 2
3. Makefile 3
4. oXygen 4
    4.1. Filtyper 4
    4.2. Pretty-printing 4
    4.3. xmlcatalog 5
    4.4. make 5
    4.5. Transformation Scenarios 5
5. Emacs 6

1. Indledning

Der er lavet en komplet DocBook-opsætning for Mikro Værkstedet. Den ligger som modul i CVS under navnet `docbook`.

Der er lavet en referenceinstallation på …. Den ligger i et share kaldet `\\…\udvikling`. Hvis du har lyst kan du selvfølgelig "mappe" et drev til dette og sætte.

1.1. Forudsætninger

I det følgende forudsættes det, at man har Cygwin installeret og at alle kommandolinier udføres i en Unix-skal, fx bash eller tcsh. Derfor vil alle referencer til environment variabler blive skrevet som `$variabelnavn`.

I opsætningen af `<oXygen/>` forudsættes det, at man har version 4.2.

1.2. Konventioner

Refencer til værdier af environment variabler
Skrives som `$variabelnavn`, fx `$DOCBOOK_ROOT`.

Referencer til valgfri dele af kommandolinier og opsætningsparametre
Disse skrives som *`valgfri`*, fx *`DOCBOOK_ROOT`*. Forskellen fra dette til egentlige variabel-referencer er at man i disse tilfælde selv skal skrive den relevante værdi.

2. Første skridt

2.1. Installer/opdater nødvendige Cygwin-pakker

Sørg for at have pakkerne `libxml2` og `libxslt` i de nyeste versioner.

Hvis du ikke allerede har make installeret, så installer den.

Vigtigt

DocBook-opsætningen forudsætter, at man har make i ver. 3.80 eller højere.

Vink

Når du installerer/opdaterer Cygwin anbefales det *ikke*, at man bruger `sunsite.dk`. Den er godt nok hurtig, men til gengæld er den ikke særligt godt opdateret med Cygwin.

I stedet kan du bruge fx `ftp.funet.fi`, hvor der opdateres oftere og som også har en god forbindelse.

2.2. Sæt environment variabel

Sæt variablen `DOCBOOK_ROOT` til det sted du har installeret DocBook, hvis du selv har checket det ud af CVS. Hvis du vil bruge standardinstallationen, sætter du den blot til `//…/udvikling/docbook`.

Vigtigt

Brug `/` i stedet for `\` i stien, fx `c:/docbook` i stedet for `c:\docbook`. Systemet konverterer selv til Windoze™-stier, hvor det er relevant.

3. Makefile

Der er lavet en generel Makefile, som kan bruges ved alle transformationer. Den er placeret i

```
DOCBOOK_ROOT/makeinclude/Makefile
```

I denne defineres targets til generering af PDF, HTML og CHM (HTMLHelp) samt til validering. Alle targets forudsætter at extension på DocBook-filer er `.dbx` eller `.xml`.

Denne Makefile bruger `DOCBOOK_ROOT` til at bestemme placeringer af diverse filer, der bruges ved validering og transformering.

Brug denne Makefile ved at referere direkte til den i **make**-kommandoen med `-f` eller lav en Makefile, der inkluderer den. Altså: Enten køres kommandoen a la

```
make -f $DOCBOOK_ROOT/makeinclude/Makefile …
```

eller der laves en fil kaldet `Makefile`, der indeholder

```
include $(DOCBOOK_ROOT)/makeinclude/Makefile
```

hvorefter kommandoen køres a la

```
make …
```

Du kan nu validere samt generere PDF, HTML og HTMLHelp ved at køre make med forskellige targets. I det følgende antages det, at du har lavet en `Makefile`, der inkluderer den generelle, som beskrevet ovenfor. Kør make på følgende måder:

PDF

```
make ditdok.pdf
```

HTML – én stor fil

```
make ditdok.html
```

HTML – flere filer med `index.html`

```
make CHUNK=1 ditdok.html
```

Filer placeres i et subdirectory kaldet `html`

HTMLHelp (`.chm`)

```
make ditdok.chm
```

Midlertidige filer placeres i et subdirectory kaldet `htmlhelp`

Validering

```
make ditdok.val
```

Alle transformationer er som standard sat til at validere dokumentet først. Hvis du af en eller anden obskur grund ikke vil have det, kan du sætte variablen `NO_VALIDATE` til en tilfældig værdi. Eksempel:

```
make NO_VALIDATE=1 ditdok.pdf
```

Alle transformationer er som standard sat til at bruge XEP som PDF-generator. Hvis du af en eller anden obskur grund vil bruge FOP i stedet, kan du sætte variablen `FOP` til en tilfældig værdi. Eksempel:

```
make FOP=1 ditdok.pdf
```

4. <oXygen/>

I det følgende beskrives hvorledes `<oXygen/>` sættes op til at finde vores filer samt udføre transformationer via make og via *transformation scenarios*.

Det bekrives kun, hvordan man sætter transformation scenarios op til HTML samt PDF ved brug af XEP.

Ved validering kan man altid bruge `<oXygen/>`s indbyggede valideringsmekanisme.

4.1. Filtyper

Start med at fortælle `<oXygen/>`, at filer med endelsen `.dbx` er XML-filer:

1. Åbn Options->Preferences
2. Vælg File types.
   a. Klik New.
   b. Under Extension skrives `dbx`. Under Editor vælges `XML Editor`.

4.2. Pretty-printing

`<oXygen/>` kan formatere XML-dokumentet pænt op. Dette gøres ved menuvalget Document->Format and indent (**Ctrl-Shift-P**). For at den ikke skal lave kaos i programlistninger og andre *verbatim* elementer, skal den have at vide, hvad den skal holde fingrene fra:

1. Åbn Options->Preferences
2. Vælg Editor->Format
3. Under Preserve space elements bruges knappen Add nu til at tilføje navnene

   literallayout
   programlisting
   synopsis
   screen

4.3. `xmlcatalog`

1. Åbn Options->Preferences
2. Vælg XML->XML Catalog
   a. Slet de eksisterende refencer til kataloger.
   b. Opret en ny reference ved klik på New.
   c. I Open-dialogboksen findes og vælges *`DOCBOOK_ROOT`*`/xmlcatalog`.

      Hvis du bruger den "officielle" DocBook-installation på ..., er det nemmeste blot at skrive `//…/udvikling/docbook/xmlcatalog` direkte i File name.
   d. Kontroller, at der under Prefer er valgt public.

4.4. make

Her beskrives hvorledes man sætter `<oXygen/>` op til at bruge make til generering af PDF. Metoden til at generere fx HTML er fuldstændig analog – det eneste man skal ændre er extension på navnet på den genererede fil.

1. Åbn Options->Preferences
2. Vælg External Tools
   a. Klik New
   b. I dialogboksen Command line skriver du et passende navn, fx `Make PDF`.
   c. Hvis du vil have en genvejstast (anbefales!), sættes denne ved at klikke på knappen Choose og følge anvisningerne.
   d. Under Command line skrives

```
make -C ${cfd} ${cfn}.pdf
```

   e. Klik OK.

4.5. *Transformation Scenarios*

1. Vælg XSLT/FO->FO processors
   a. Klik knappen New for at lave en ny opsætning af XEP.
   b. Kald den `XEP`[1].
   c. I Command line skrives:

[1] Det er vigtigt at kalde den `XEP`, da der refereres til dette i transformationsscenarierne senere.

```
DOCBOOK_ROOT/bin/win32/xep.bat ${fo} ${method} ${out}
```

Vigtigt

Ovenstående batchfil forudsætter, at du har en environment variabel kaldet `TEMP`. Denne er som standard sat i Windoze™, så det skulle ikke give problemer. Hvis du alligevel ikke har den sat, så opret den og sæt værdien til et directory, hvor XEP må gemme midlertidige filer.

Hvis du ikke gider at se alt det information, der kommer fra XEP, tilføjer du blot `-quiet` i slutningen af ovenstående kommandolinie. Så siger XEP kun noget ved advarsler og fejl.

d. Klik OK.

2. Klik OK for at lukke Preferences.
3. Vælg menupunktet Options->Import transformation scenarios
4. I dialogboksen findes og vælges

   ```
   DOCBOOK_ROOT/editor/oxygen/trans-scenarios.properties
   ```

5. Du skulle nu gerne have to nye transformationsscenarier kaldet "`MV PDF (XEP)`" og "`MV HTML`" til hhv. PDF- og HTML-generering.

   Der er ikke lavet transformationsscenarier til *chunked* HTML, da dette ikke er særligt velunderstøttet i `<oXygen/>`. Brug i stedet make til dette!

5. Emacs

I det følgende beskrives opsætning af Emacs-pakken PSGML.
Forfatteren til indeværende dokument har i skrivende stund kun testet opsætningen i GNU Emacs. De, der bruger XEmacs opfordres til at afprøve opsætningen og melde tilbage, så vi kan få dokumenteret resultatet.

- Hvis du ikke allerede har gjort det i forbindelse med det foregående afsnit, så lav en ny environment variable med navnet `DOCBOOK_ROOT.`

  Giv den værdien `//…/udvikling/docbook`.

- Lav en ny environment variable med navnet `SGML_CATALOG_FILES`

  Giv den værdien `$DOCBOOK_ROOT/sgmlcatalog`

- Føj følgende linier til din `.emacs`:

```
(setq load-path
      (cons (concat (getenv "DOCBOOK_ROOT") "/editor/emacs/site-lisp/")
            load-path))
(load "docbook-config")
```

- Næste gang du starter Emacs har du nu automatisk adgang til DTD'erne “Mikrov Article”, “Mikrov Book”, “DocBook Article”, “DocBook Book” og “Slides v. 3.3.1”.

  “Mikrov Article” og “Mikrov Book” er vores egen tilpassede udgave af DocBook-DTD'en, hovedsageligt bestående af et sæt prædefinerede entiteter.

  “DocBook Article” og “DocBook Book” er *out of the box* DocBook, i skrivende stund i version 4.3.

  “Slides” er en DocBook-baseret DTD til at lave præsentationer med.

  `DOCTYPE` for disse dokumenttyper indsættes nemt med DTD->Insert DTD->*`Ønsket DTD`* (**C-c C-u C-d**)

Det vil selvsagt føre for vidt at beskrive hele PSGML her, men et par nyttige tastetryk skal da lige nævnes:

Indsæt element
C-c C-e

Tilføj attribut
C-c +

Valider dokument
C-c C-v

Byg PDF, HTML etc.
Lav en Makefile som beskrevet i Afsnit 3 og brug standard Emacs-funktionalitet:

```
M-x compile
```