Att skriva dokumentation i DocBook för Proview

Kort introduktion

Revision 0.1	**Revisionshistorik** **2006-04-01** **v0.1 för Proview v4.2 skapad**	**jh**

Att skriva dokumentation i DocBook för Proview: Kort introduktion

av Jonas Haulin

Innehållsförteckning

1. Introduktion 1
    1.1. Om DocBook 1
    1.2. Hello world 1
2. Editeringsverktyg 2
    2.1. Emacs + nXML 2
    2.2. Andra alternativ 2
3. Ett exempeldokument 3
    3.1. Dokumentprolog 3
    3.2. Uppmärkning 3
        3.2.1. Dokumentstruktur 4
        3.2.2. Listor etc. 5
        3.2.3. Kommandon, filnamn... 6
        3.2.4. Proview-uppmärkning 6
4. Generering av utdata 8
    4.1. Bygga DocBook-dokumentation 8
    4.2. Byggkomponenter 8
    4.3. Inställningslager 8
    4.4. Anpassning av utfiler 9
5. Mer om DocBook 10
    5.1. Bakgrund 10
    5.2. Vilka verktyg behövs? 10
        5.2.1. DocBook:s verktygskedja 10
        5.2.2. Editorer för DocBook 11
    5.3. Referenser 11

Kapitel 1. Introduktion

Det här dokumentet är en snabböversikt av dokumentationsstandarden Docbook, och hur den kan användas för dokumentation i Proview [http://www.proview.se]. Texten i dokumentet är för närvarande minimalt uppmärkt/taggad. För ett mer uppmärkt dokument, se Proview Getting Started Guide [../en_us/pwrgsg_index.html].

1.1. Om DocBook

DocBook är en XML-dialekt för teknisk dokumentation. Den används i många stora mjukvaruprojekt, både kommersiella och open source. Från ett källdokument skrivet i DocBook/XML kan utdata i en mängd format genereras, exempelvis html, pdf, ps, man-sidor. Med xml-taggarna anges dokumentets struktur och innehåll, däremot inte dess formatering - detta beror istället av vilken utdata som skapas.

Mer om DocBook i Kapitel 5, *Mer om DocBook*

1.2. Hello world

Ett minimalt DocBook-dokument kan se ut såhär:

```
<?xml version="1.0" encoding="utf-8"?>
<article xmlns="http://docbook.org/ns/docbook" version="5.0" xml:lang="sv">
  <title>Ett minimalt DocBook-dokument</title>
  <para>kan se ut såhär.</para>
</article>
```

Spara filen ovan som exempelvis `minimalt.xml`. Följande kommando skapar en html-fil.

```
bash$ xsltproc -o minimalt.html /usr/local/share/xml/docbook/stylesheet/snapsho
```

För att skapa en pdf behövs två kommandon.

```
bash$ xsltproc -o minimalt.fo /usr/local/share/xml/docbook/stylesheet/snapshot/
bash$ fop minimalt.fo -pdf minimalt.pdf
```

Kapitel 2. Editeringsverktyg

2.1. Emacs + nXML

I princip kan vilken texteditor som helst användas för att skriva xml. En syntaxmedveten editor underlättar dock avsevärt. Emacs 21.3 och uppåt med nXML-läge fungerar mycket bra (finns på pwr42). De viktigaste funktionerna är:

- Autokomplettering av taggar (**C-RET**): man skriver de första bokstäverna och autokompletterar sedan.
- Insättning av sluttag 1 (**C-c C-i**): sätter sluttaggen direkt efter öppningstaggen, och placerar markören mitt emellan.
- Insättning av sluttag 2 (**C-c C-b**): sätter sluttagen på ny rad nedanför, och placerar markören på blank rad mitt emellan.
- Insättning av sluttag 3 (**C-f**): Avslutar närmaste öppna tag uppåt i trädet.
- Autovalidering: Syntaxen kollas mot schemat kontinuerligt. Info om fel. Syntaxfärgning och indentering.

För att ladda nXML-läget, och för att Emacs ska bete sig acceptabelt i övrigt (scroll-hjul, fonter etc), behöver man göra en del inställningar i någon eller några av filerna `.emacs`, `.gnu-emacs` och `.gnu-emacs-custom` i sin hemkatalog. Exempel på hur detta kan göras finns i `/home/jonas_h`. nXML fungerar inte med XEmacs.

Källkod och dokumentation för nXML finns på pwr42 i `/usr/local/share/emacs/21.4/site-lisp/nxml-mode-20041004/`. I underkatalogen `schema/` finns den senaste versionen av docbook-schemat (filerna `docbook.rnc` och `docbookxi.rnc`) inlagt. nXML använder schemat för validering och autkomplettering, så det är bra om man har rätt schema kopplat till sitt dokument.

En DocBook-meny till Emacs finns i `/usr/local/share/emacs/21.4/site-lisp/docbook-menu-0.92/`. Även den laddas in med inställningar i filerna `.emacs` m fl. Menyn innehåller inga kommandon, men länkar till dokumentation, samt alla element/taggar grupperade alfabetiskt och logiskt. Den beskriver DocBook version 4, men skillnaderna mot version 5 är inte jättestora.

2.2. Andra alternativ

Ett flertal xml-editorer finns, både fria och kommersiella, några med WYSIWYG-funktionalitet. En annan möjlighet som skulle kunna undersökas är att editera via ett wiki/web-gränssnitt.

Kapitel 3. Ett exempeldokument

Här beskrivs hur man kan bygga upp ett större dokument, av typen bok eller artikel. Exempel på lämpliga taggar och uppmärkning för Proview ges.

3.1. Dokumentprolog

Prologen, eller dokumenthuvudet till en docbookfil kan se ut på följande sätt:

```
<?xml version="1.0" encoding="UTF-8"?> ❶
<!DOCTYPE article [ ❷
<!ENTITY % isopub PUBLIC ❸
"ISO 8879:1986//ENTITIES Publishing//EN//XML"
"/usr/share/xml/entities/xml-iso-entities-8879.1986/isopub.ent">
<!-- "http://www.w3.org/2003/entities/iso8879/isopub.ent"> -->
%isopub;
<!ENTITY % pwrent SYSTEM "../src/pwrent.ent"> ❹
%pwrent;
]>
<article version="5.0" xml:lang="en" xmlns="http://docbook.org/ns/docbook" ❺
xmlns:mathml="http://www.w3.org/1998/Math/MathML"
xmlns:svg="http://www.w3.org/2000/svg"
xmlns:xhtml="http://www.w3.org/1999/xhtml"
xmlns:xlink="http://www.w3.org/1999/xlink">
<info>
<title><application>Proview 4.1</application> Getting Started Guide</title>

<subtitle>A step-by-step guide to set up a minimal <application>Proview</ap

<author>
  <personname><firstname>Jonas</firstname>
  <surname>Haulin</surname></personname>
  <affiliation>
<orgname>Proview/R</orgname>
<address> <email>info@proview.se</email> </address>
  </affiliation>
</author>
</info>
```

❶ Det här är ett XML-dokument.

❷ Detta är en artikel. Kan även vara “book”

❸ Hämta standard-entity-koder för särskilda tecken. Lokal fil går snabbast. URL:en bortkommenterad.

❹ Egendefinierade entiteter, se Avsnitt 3.2.4, “Proview-uppmärkning”

❺ Detta är en artikel, DocBook (5.0) namespace. Fler namespace följer. Det här är rotelementet för dokumentet, som kommer att avslutas med `</article>`.

3.2. Uppmärkning

Här ges exempel på uppmärkning av ett dokument. För en fullständig referens, se DocBook 5.0: The Definitive Guide [http://www.docbook.org/tdg5/en/html/docbook.html]. Fler länkar finns i Avsnitt 5.3, “Referenser”.

3.2.1. Dokumentstruktur

För att dela in en article används avsnitt/sections. Texten delas in i stycken. Det kan se ut så här:

```
<section><title>första avsnittet</title>
<para>Ett stycke text</para>
<section><title>första underavsnittet</title>
<para>Ett stycke text</para>
</section>
</section>
```

`<section>` kan alltså innehålla sig själv, och underavsnitt skapas rekursivt. Ett alternativ är att använda `<sect1>`, `<sect2>` etc (upp till 5) för att explicit ange avsnittsstrukturen.

Ett dokument av typen `<book>` kan ha en kapitelnivå, `<chapter>`, över avsnitten, och däröver en delnivå, `<part>`. Ofta vill man dela upp en "book" så att varje kapitel utgör en egen fil. Detta är särskilt användbart för större dokument, som då blir mer hanterliga, kan editeras av flera personer samtidigt, och kan sammanfogas modulärt. Man infogar delarna i boken med hjälp av xinclude:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE book [
<!ENTITY % isopub PUBLIC
"ISO 8879:1986//ENTITIES Publishing//EN//XML"
"/usr/share/xml/entities/xml-iso-entities-8879.1986/isopub.ent">
%isopub;
<!ENTITY % pwrent SYSTEM "../src/pwrent.ent">
%pwrent;
]>
<book version="5.0b5" xml:lang="sv" xmlns="http://docbook.org/ns/docbook"
xmlns:mathml="http://www.w3.org/1998/Math/MathML"
xmlns:svg="http://www.w3.org/2000/svg"
xmlns:xhtml="http://www.w3.org/1999/xhtml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xi="http://www.w3.org/2001/XInclude"> (1)
<info>
  ...
</info>

<xi:include href="dbpwr_intro.xml"/> (2)
<xi:include href="dbpwr_skapadok.xml"/>
<xi:include href="dbpwr_utdata.xml"/>
<xi:include href="dbpwr_omdocbook.xml"/>

</book>
```

(1) Namespace för XInclude måste läggas till.

(2) Här inkluderas bokens kapitel. Dessa skrivs som enskilda och fullständiga DocBook-dokument, men av dokumenttypen "chapter" istället för "book" eller "article".

Notera

Om man vill autovalidera ett dokument med xincludes i nXML måste man koppla schemat `docbookxi.rnc`, till dokumentet. Detta görs i XML → Set Schema → File.... Default är annars `docbook.rnc`.

3.2.2. Listor etc.

```
itemizedlist>
<listitem><para>xml-grammatik för dokumentation, ssk teknisk, ssk hård- och mj
<listitem><para>Strukturerad uppmärkning baserat på innehåll, ej formatering (
Möjligt att skapa mer sökbara databaser.</para></listitem>
</itemizedlist>
```

ger

- xml-grammatik för dokumentation, ssk teknisk, ssk hård- och mjukvara
- Strukturerad uppmärkning baserat på innehåll, ej formatering (som ex.vis html). Möjligt att skapa mer sökbara databaser.

`<orderedlist>` skapar en numrerad lista istället för en punktlista.

```
<variablelist>
  <varlistentry><term><filename>pwr_xsl/common-customizations.xsl</filename></
  <listitem><para>Gemensamma parameterinställningar och template-definitioner.<
  <varlistentry><term><filename>pwr_xsl/html-common.xsl</filename></term>
  <listitem><para>Html-specifika parameterinställningar och template-definitio
  </variablelist>
```

ger

`pwr_xsl/common-customizations.xsl`	Gemensamma parameterinställningar och template-definitioner.
`pwr_xsl/html-common.xsl`	Html-specifika parameterinställningar och template-definitioner.

```
<programlisting>
  ### Document filename without suffix (.xml):
  ###FILE = pwr_gsg
  FILE = dbpwr                                          <co xml:id="co.makefil.f
</programlisting>
<calloutlist>
  <callout arearefs="co.makefil.file2"><para>Här anges filnamnet utan ändelse
  </calloutlist>
```

ger

```
### Document filename without suffix (.xml):
###FILE = pwr_gsg
FILE = dbpwr                                                    ❶
```

❶ Här anges filnamnet utan ändelse för infil (.xml) och utfil (.html, .fo, .pdf)

```
<qandaset>
```

```
<qandaentry><question><para>Hur gör man en FAQ-avdelning?</para></question>
<answer><para>Med en <tag>&lt;qandaset&gt;</tag></para></answer> </qandaentry
</qandaset>
```

ger

3.2.2.1. Hur gör man en FAQ-avdelning?

Med en `<qandaset>`

3.2.3. Kommandon, filnamn...

Detta är lånat från KDE, och är kanske litet övertaggat..:

```
    <screen>
<prompt>bash$</prompt> <userinput><command>mkdir</command> <parameter><filenam
<prompt>bash$</prompt> <userinput><command>mount</command> <option>-t</option>
```

ger

```
  bash$ mkdir /dev/mqueue
bash$ mount -t mqueue none /dev/mqueue
```

Annan uppmärkning som kan vara värda att använda är:

```
<menuchoice>
  <shortcut><keycombo><keycap>Ctrl</keycap><keycap>Q</keycap></keycombo></shor
  <guimenu>Edit</guimenu>
  <guimenuitem>Change value</guimenuitem>
  </menuchoice>
```

ger

Edit → Change value (**Ctrl-Q**)

Tips

Noteringar, tips, viktigt, uppmaning till försiktighet och varningar skapas med taggarna `<note>`, `<tip>`, `<important>`, `<caution>` och `<warning>`.

Viktigt

Texten i ex.vis en viktigt måste skrivas inom en `<para>`

Notera

Andra taggar som använts är `pwrp` (`<systemitem role="user">pwrp</systemitem>`) och `PWR_BUS_ID` (`<envar>PWR_BUS_ID</envar>`).

3.2.4. Proview-uppmärkning

- Klasser: Taggen `<classname>` har använts. Exempel **ProjectReg**. Den är kanske framför allt avsedd för objektorienterade programmeringsspråk, iofs.

- Attribut: Taggen `<property>` har använts. Exempel *ObjectName*.
- Miljöer/applikationer: Entiteter har använts för ProjectList, VolumeList, Directory, Volume, Xtt. De översätts till `<application role='pwrenv'>VolumeList</application>` etc. Entiteterna är definierade i filen `pwrent.ent`.

Kapitel 4. Generering av utdata

4.1. Bygga DocBook-dokumentation

Generering av dokument från docbook-filer är integrerad i byggrutinen för Proview. Dessa byggs med kommandot

```
bash$ pwre build doc man docbook
```

vilket ger utdata i form av html och pdf. Html-dokumentet skapas i två versioner: i en respektive flera filer. I ett sista steg kopieras css- och mediafiler till målkatalogen.

Nya xml-dokument kan läggas till genom att placera xml-filerna i `$pwre_sroot/doc/man/en_us/`, eller `$pwre_sroot/doc/man/sv_se/`. De grafik- eller mediafiler som dokumenten använder läggs i `$pwre_sroot/doc/man/src/`. För att utdatafiler ska genereras måste man lägga till filnamnet (endast namnet på huvudfilen, om flera) i variablerna `sv_se_xmlsources` och / eller `en_us_xmlsources` i makefilen. Huvudkällfilerna måste anges, eftersom det finns xml-filer i katalogen som inte är fristående källfiler (ex.vis kapitelfiler). Mediafiler kopieras automatiskt dit de behövs, och behöver inte anges någonstans.

4.2. Byggkomponenter

För att generera dokumentationen behövs följande:

- Xsltproc. Standardkomponent i de flesta linuxdistributioner.
- DocBook-xsl-stylesheets. Går att hämta med apt-get, eller från sourceforge. Den senaste versionen finns på pwr42 i `/usr/local/share/xml/docbook/stylesheet/docbook-xsl-1.70.1/`. När man uppdaterar till en ny release av dessa måste man köra installationsscriptet `install.sh` i distributionen. Detta uppdaterar användarens `XML_CATALOG_FILES` att länka till den nya distributionen.
- Fop. Version 0.92b av Apaches fo-processor är installerad på pwr42 i `/usr/local/fop-0.92beta/`. En symlänk finns i `/usr/local/bin`.

Notera

Ingen kontroll görs i nuläget för att dessa komponenter finns på systemet. Detta bör antagligen läggas till i makefilen om möjligt.

4.3. Inställningslager

Ett stort antal parametrar går att ställa in för xsl-mallarna. I de fall man inte vill ha defaultvärdena (sätts i `param.xsl`) kan man ange parametervärden i xsltproc-anropet, eller i inställningslager (customization layers). Där kan också modifierade versioner av olika templates i mallarna definieras. Dokumentation över vilka parameterinställningar som kan göras finns på DocBook XSL Stylesheet Reference Documentation [http://docbook.xml-doc.org/snapshots/xsl/doc/]. Följande inställningslager ligger under `$pwre_sroot/doc/man/src/`.

`pwrxsl-common.xsl`	Gemensamma parameterinställningar och template-definitioner.
`pwrxsl-html-common.xsl`	Html-specifika parameterinställningar och template-definitioner.
`pwrxsl-html.xsl`	1. Importerar `/docbook-xsl-n.nn.n/html/docbook.xsl` som är grundmallen för html-processning - en utfil.

	2. Hämtar från `pwrxsl-common.xsl`
	3. Hämtar från `pwrxsl-html-common.xsl`
	4. Sätter parametrar och templates specifika för html - en utfil.
`pwrxsl-chunk.xsl`	1. Importerar `/docbook-xsl-n.nn.n/html/chunk.xsl` som är grundmallen för html-processning - flera utfiler.
	2. Hämtar från `pwrxsl-common.xsl`
	3. Hämtar från `pwrxsl-html-common.xsl`
	4. Sätter parametrar och templates specifika för html - flera utfiler.
`pwrxsl-fo.xsl`	1. Importerar `/docbook-xsl-n.nn.n/fo/docbook.xsl` som är grundmallen för fo-processning.
	2. Hämtar från `pwrxsl-common.xsl`
	3. Sätter parametrar och templates specifika för fo.

4.4. Anpassning av utfiler

- FO/PDF: Justering av utseende, fonter, textstorlek etc. görs helt i inställningslagret, och behandlas av xslt-processorn. FO-processorn tar inga sådana argument eller parametrar.
- HTML: Slutgiltig formatering bestäms av css. Man kopplar de element/taggar man använt till formateringsinstruktioner i css:en. Ger önskad grafisk profil åt dokumentet. För att det ska fungera krävs dock att uppmärkningen är konsekvent.

Kapitel 5. Mer om DocBook

Detta kapitel är framför allt av orienterande karaktär.

5.1. Bakgrund

Docbook utvecklades ursprungligen av HaL och O'Reilly med början 1991. 1998 blev det en del av SGML Open Consortium, vilket senare blev OASIS (Organization for the Advancement of Structured Information Standards).

Några punkter..

- xml-grammatik för dokumentation, ssk teknisk, ssk hård- och mjukvara
- Strukturerad uppmärkning baserat på innehåll, ej formatering (som ex.vis html). Möjligt att skapa mer sökbara databaser.
- Kan generera utdata i många olika format från en källa: html, pdf, manpages, rtf, txt, htmlhelp, Utdata för olika plattformar från samma källa (t ex SuSE / RedHat / Windows) - profiling. Utdata på olika språk från samma källa (med fallback).
- Enkelt att integrera i byggrutinen för ett projekt. Make-filer, versionshantering.
- OASIS-standard. Används av bl a: Sun, Microsoft, HP, Novell, Red Hat, och open source projekt som Linux kernel, KDE, Gnome, debian, Ubuntu, Fedora, Free-BSD, LDP, m fl.
- Möjligt att skapa stora korsrefererande dokumentationssystem från fristående komponenter.
- Aktiv open source-utveckling av hela verktygskedjan.

Fördelar: Alla ovan.

Nackdelar: många taggar, ej entydigt vilka man ska använda, inlärningskurva, sammansatt verktygskedja, XML (om man inte gillar XML...).

5.2. Vilka verktyg behövs?

Detta avsnitt beskriver verktygskedjan för Docbook, och hur man editerar och processar docbook-dokument.

5.2.1. DocBook:s verktygskedja

- Vokabulären specificeras i ett RELAX NG [1] schema fr o m Docbook v 5.0, tidigare användes en DTD (Document Type Definition). Den har bantats ned och stramats upp fr o m 5.0, möjligen ej helt fryst ännu. För detta dokument används v 5.05b. Schemat behövs egentligen bara för validering och för editorer, och om man ska skriva egna stylesheets.
- Stylesheets definierar hur docbookvokabulär transformeras till andra taggar för olika utformat. Mallarna är skrivna i xsl (extensible stylesheet language) som också är en xml-vokabulär. Tidigare användes dssl. Senaste släppta versionen är docbook-xsl-1.69.1 (för detta dokument används en snapshot-version från sourceforge). Huvudstylesheets för html, chunk, fo (formatted output), manpages etc. anropas. Dessa använder sig av ett flertal gemensamma stylesheets.
- En xslt-processor används för att generera utdata (.html, .fo, etc). Man anropar den med indata och stylesheet som argument, enklast via en make-fil. xsltproc är standard i många linuxdistributioner, ingår i libxslt. Andra processorer finns, ex.vis saxon, xalan. För detta dokument används en snapshot-version av libxslt från xmlsoft.org.

- För att skapa utdata för papper (pdf, ps, rtf, txt) får man processa den skapade .fo-filen ett steg till, med en xsl-fo-processor. xsl-fo är en xml-vokabulär för att beskriva formaterad text. Den innehåller information om hur texten ska se ut på pappret med fonter, textstorlek, pappersformat, styckeindelningar etc. Den innehåller däremot ingen information om typsättning, radbrytningar, sidbrytningar och avstavning. För att generera sådan information och utdatafiler i ex.vis pdf-format använder man en xsl-fo-processor. xsl-fo-processorer är typsättningsmotorer, och det finns ett flertal kommersiella och öppna. Många av dem fungerar bra för de flesta tillämpningar, men ingen implementerar xsl-fo-standarden fullständigt. Detta uppges bero på tre saker:

  1. Xsl-fo-standarden är ganska ny och omfattande.

  2. Den har visat sig svårimplementerad i sin helhet.

  3. Typsättning är svårt, generellt.

  Den mest använda open source-processorn är Apache's javabaserade FOP. Senaste stabila releasen var 0.20.5. En helt omarbetad version 0.90 alfa gjordes tillgänglig nyligen. För pdf-versionen av detta dokument används 0.92 beta. En 1.0-version uppges kunna släppas snart.

  Andra open source fop:ar: PassiveTeX och xmlroff. Kommersiella: E3, XEP, XPP, XSL Formatter m fl.

5.2.2. Editorer för DocBook

I princip kan vilken texteditor som helst användas. Eftersom taggarna ofta är långa (inga förkortningar) blir det dock lätt omständligt utan xml-medveten editor. Två alternativ:

- Wysiwyg-editorer: Gömmer xml-taggarna. Exempel: oXygen (kommersiellt), XXE (halvt kommersiellt)

- Emacs-lägen, etc.: psgml-mode, nXML-mode

För detta dokument har Emacs med nXML-läge använts (Emacs version 21.x krävs). nXML är skrivet av xml-gurun James Clark (groff, expat, xml-standarden, relax ng mm). Man kopplar nXML till sitt schema (docbook.rng i detta fall), och sedan får man bl a:

- auto-komplettering av taggar (**C-RET**): man skriver de första bokstäverna och autokompletterar sedan.

- Insättning av sluttag 1 (**C-c C-i**): sätter sluttaggen direkt efter öppningstaggen, och placerar markören mitt emellan.

- Insättning av sluttag 2 (**C-c C-b**): sätter sluttagen på ny rad nedanför, och placerar markören på blank rad mitt emellan.

- Insättning av sluttag 3 (**C-f**): Avslutar närmaste öppna tag uppåt i trädet.

- Autovalidering: Syntaxen kollas mot schemat kontinuerligt. Info om fel. Syntaxfärgning och indentering.

Till Emacs finns också en DocBook-meny att ladda ned. Den innehåller inga kommandon, men länkar till dokumentation, samt alla element/taggar grupperade alfabetiskt och logiskt. Användbart.

5.3. Referenser

- DocBook 5.0: The Definitive Guide [http://www.docbook.org/tdg5/en/html/docbook.html]. Komplett referens över alla element. Innehåller även en Quick Reference [http://www.docbook.org/tdg5/en/html/quickref.html].

- DocBook XSL: The Complete Guide [http://www.sagehill.net/docbookxsl/index.html]. Använda och sätta upp ett DocBook-system.

- DocBook XSL Stylesheet Reference Documentation [http://docbook.xml-doc.org/snapshots/xsl/doc/]. Referensdokumentation över inställningsparametrar för html och fo.