DocBook Install mini−HOWTO

Robert B Easter

reaster@comptechnews.com

Revision History

Revision v1.5	2001−01−27	Revised by: rbe

DocBook−Install−mini−HOWTO is a detailed practical guide for novices to quickly getting DocBook installed and processing sgml files into html, ps, and pdf on a GNU/Linux system − other systems may be similar. Since setup of DocBook requires files from several separately distributed packages, it can be confusing for beginners.

Table of Contents

1. Introduction

DocBook−Install−mini−HOWTO is a detailed practical guide for novices to quickly getting DocBook installed and processing sgml files into html, ps, and pdf on a GNU/Linux system − other systems may be similar. Since setup of DocBook requires files from several separately distributed packages, it can be confusing for beginners.

1.1. What is DocBook

DocBook is a Standard Generalized Markup Language (SGML) Document Type Definition (DTD) that defines a set of textual document markup tags that work much like the familiar HTML language used on the web.

DocBook is intended for the authoring of books and articles. As such, it provides tags specifically designed to describe books and articles. For instance, the <book> and <article> DocBook tags are used to create books and articles. Within these documents, the <chapter>, <sect1>, and <para> tags are used. DocBook SGML files are stored in text files with a sgml or gml suffix.

When processed, a single DocBook SGML file can output html, pdf, ps, txt and other formats for both online and printed publication. The processing is governed by stylesheets that can automatically generate a table of contents, page numbering, chapter & section numbering, and other features.

DocBook is also designed for authoring unix manpages using <refentry>.

1.2. Brief Overview

Here are brief descriptions of the packages we will work with in the next sections:

OpenJade. OpenJade is an Standard Generalized Markup Language (SGML) and Document Style Semantics and Specification Language (DSSSL) processor. It processes DocBook sgml source files into html, tex, rtf, txt and others. Openjade is the essential engine for converting a DocBook file into other formats. The tex out format is used mostly as an intermediate format to obtain dvi, pdf, and ps via TeX macros and dvi converters.

DocBook SGML DTD. The Document Type Definition (DTD) files are SGML files that define the DocBook language. Its defines the valid tag set and rules of their use. OpenJade requires access to the DTD files for every document type that it parses.

ISO8879 ENTITY SGML. Entities define how to represent special characters that have either no keyboard key or have special meaning in SGML. Examples familiar from HTML include "&amp;"='&', "&gt;"='>', and "&lt;"='<'.

DocBook DSSSL. Document Style Semantics and Specification Language (DSSSL) files (dsl suffix) for a particular DTD, in this case DocBook, specify how to convert DocBook into html, rtf, tex etc.

SgmlTools−lite. Sgmltools is a frontend wrapper for running openjade and the TeX macros jadetex and pdfjadetex, macros included with openjade. Converting a DocBook file to ps or pdf is a two or three−step process. OpenJade outputs a tex file which is the input of jadetex, to produce a dvi, and pdfjadetex, to produce a pdf. A ps file is obtained by passing the dvi file through dvips. The sgmltools script provides a single command to perform these tasks.

HTMLdoc. HTMLdoc is a free program for converting html files into a pdf or ps file.

SGMLSpm and docbook2X. Together, these two are used to generate manpages. SGMLSpm is a perl5 module library for processing parsed output from `onsgmls`, a program included with OpenJade. SGMLSpm includes an application called `sgmlspl` to use the SGMLSpm library. `Sgmlspl` requires "spec files", which are available from various other sources on the Internet, for each type of document transformation to be performed. DocBook2X is a package that provides the spec files for transforming DocBook files into manpages.

2. Download the Packages

In this section, we will locate and download the software on the Internet.

2.1. OpenJade

OpenJade is an actively maintained open−source software project based on the Jade package by James Clark. Download the lastest stable release (1.3?) at:

http://openjade.sourceforge.net/

OpenJade also includes the OpenSP package and the TeX macros, jadetex and pdfjadetex for converting files to dvi and pdf. The following program are provided by this package:

- openjade
- onsgmls
- osgmlnorm
- ospam
- ospent
- osx

To use the jadetex and pdfjadetex for making dvi, ps, and pdf, you must have a working TeX (tex) installation. If you do not have TeX, check with your Linux distribution for a binary package that can be downloaded and installed. Otherwise, you can download the teTeX distribution of TeX from:

http://www.tug.org/tetex/

2.2. DocBook SGML DTD

The DocBook DTD for SGML and XML are maintained by a technical committee at Oasis−Open.ORG. Download the current version (and any old versions you might need) of DocBook SGML at:

http://www.oasis−open.org/docbook/sgml/index.html

2.3. ISO8879 ENTITY SGML

The entities define representations for special or untypeable symbols or characters, including mathematical symbols, and the entities that you may be familiar with from HTML. These entity files need to be installed for a proper configuration.

- Resources at OASIS:
    - ♦ http://www.oasis−open.org/cover/topics.html#entities
    - ♦ http://www.oasis−open.org/cover/ISOEnts.zip
    - ♦ http://www.oasis−open.org/cover/isoENT−tar.gz

ISOEnts.zip can simply be unzipped into the directory where the DocBook DTD is unzipped without requiring anything else but the files in isoENT−tar.gz are also needed. Again, the files in isoENT−tar.gz are to be unzipped into the DocBook DTD directory (see next section on installing for details), but the filenames end with ".ent" suffix. These will need to be renamed to a ".gml" ending. You can do this manually, or you can download and use the file below, made by this author, which contains the files of both ISOEnts.zip and isoENT−tar.gz:

http://www.comptechnews.com/~reaster/iso8879−entities.tar.gz

2.4. DocBook DSSSL

The Document Style Semantics and Specification Language (DSSSL) files for the DocBook DTD (SGML/XML) is provided by Norm Walsh. These files, called the Modular DocBook Stylesheets, tell `openjade` what to do when converting your DocBook SGML file into other formats. A dsl file specifies things such as the mappings from one DTD's tags to another DTD's tags and other programmatic conversions, programmed in a language called the Core Expression Language which is derived from Scheme. The DocBook DSSSL package and documentation can be downloaded from Norm Walsh:

http://nwalsh.com/docbook/dsssl/

The Linux Documentation Project has a stylesheet customization file that turns on some nice style features. It can be downloaded at:

http://www.linuxdoc.org/authors/tools/ldp.dsl

2.5. Sgmltools−lite

Sgmltools is a frontend for openjade, `jadetex`, `pdfjadex`, `dvips`, and other programs. It provides a single command for generating all the formats possible with these tools. The lastest release, v1.3 as of writing, can be downloaded at:

http://www.sgmltools.org/

http://sourceforge.net/projects/sgmltools−lite/

This package is optional, but does make things easier sometimes.

2.6. HTMLdoc

Htmldoc is a free program for converting websites into Portable Document Format (pdf) or Postscript (ps). For pdf, it creates a tree of bookmarks that make navigation easy. Both htmldoc and pdfjadetex output pdf files, but in slightly different formats. Try both and see which turns out best for a particular docbook file. See quick links below for download site.

2.7. DocBook2X

DocBook2X requires perl5 and the SGMLS.pm perl module, available at CPAN. SGMLS.pm provides libraries and a program called `sgmlspl` which translates DocBook files into other formats by using specification files. The specification files are perl files that provide the logic for the translation to a particular format.

http://www.cpan.org/

http://docbook2x.sourceforge.net/

2.8. Quick Download Links

The files below are the latest versions as of this writing:

openjade−1.3.tar.gz. OpenJade, release version 1.3.

docbk41.zip. DocBook SGML DTD, version 4.1.

iso8879−entities.tar.gz. ISO 8879 SGML entities. It's easier to just use this file. You can download the two other files if you want and then rename filename extensions to gml.

db160.zip & db160d.zip. Norm Walsh Modular DocBook DSSSL stylesheets, version 1.60 and its documentation.

sgmltools−lite−3.0.2.tar.gz. Sgmltools−lite release version 3.0.2. Again, this is optional.

ftp://ftp.easysw.com/pub/htmldoc/1.8.9/. Htmldoc 1.8.9. Binaries and source are available. Choose what you need for your platform. Binaries are recommended. To find a binary, you can download it directly from ftp with the link above. If which to choose is not obvious, then try to going to the EasySw website: http://www.easysw.com/software.html

http://www.cpan.org/authors/id/DMEGG/SGMLSpm−1.03ii.tar.gz. SGMLS.pm 1.03ii at CPAN. (sgmlspl)

http://download.sourceforge.net/docbook2x/docbook2X−0.6.0.tar.gz. DocBook2X 0.6.0 (provides docbook2man−spec.pl for use with sgmlspl above)

3. Install the Packages

3.1. Install OpenJade

3.1.1. openjade

Here is what to do, but remember to read the files that come with OpenJade to see if there are any things you want to do special for your platform:

```
                cd /usr/local
                tar −xvzf ~/openjade−1.3.tar.gz
                cd openjade−1.3
                ./configure --prefix=/usr/local/openjade−1.3
                make
                make install

                # Once installed, the objects etc. can be deleted.
                make clean
```

The installation puts libraries in /usr/local/openjade−1.3/lib, so you might like to add it to /etc/ld.so.conf and run ldconfig. Add /usr/local/openjade−1.3/bin to your `$PATH`.

3.1.2. jadetex & pdfjadetex

As mentioned, jadetex and pdfjadetex are TeX macros that are packaged with OpenJade. They can be found in /usr/local/openjade−3.1/dsssl. A handy guide to installing these macros was prepared by Frank Atanassow Christoph and can be found at:

ftp://ftp.dante.de/tex−archive/macros/jadetex/install.pdf

http://www.comptechnews.com/~reaster/installjadetex.pdf

The following, is based on the instructions in install.pdf:

3.1.2.1. Create hugelatex (if needed)

The jadetex and pdfjadetex tex macros require more memory than a regular run of tex. The default tex memory limit configuration is often too limited. The tex configuration file, texmf.cnf, can be edited and variables which limit tex's memory use can be increased. But rather than just editing the texmf.cnf file to allow tex in all instances to have more memory, a custom tex context can be created, called hugelatex. If hugelatex is already configured on your system, you can skip this subsection (which hugelatex).

Verify that a working TeX is installed and find its directory:

```
                bash$ which tex
                /usr/share/texmf/bin/tex
                bash$ kpsewhich −expand−var='$TEXMFMAIN'
                /usr/share/texmf
```

```
bash$
```

Using which should find the location of the tex program. If its not found, then you might need to install teTeX then return here. kpsewhich is a utility that comes with teTeX and finds the main tex directory if all goes well.

Now that the texmf directory is known, installation can begin:

```
cd /usr/share/texmf
cd tex/latex
cp −r config config−temp
cd config−temp
tex −ini −progname=hugelatex latex.ini
mv latex.fmt hugelatex.fmt
mv hugelatex.fmt /usr/share/texmf/web2c
cd ..
rm −r config−temp
cd /usr/share/texmf/bin
ln −s tex hugelatex
cd /usr/share/texmf/web2c
```

The web2c directory contains the texmf.cnf configuration file. Make a backup of this file: cp texmf.cnf texmf.cnf.orig. Edit the file using whatever editor you like, and add the following lines at the end:

```
% hugelatex settings
extra_mem_top.hugelatex = 8000000
extra_mem_bot.hugelatex = 8000000
hash_extra.hugelatex = 15000
pool_size.hugelatex = 5000000
string_vacancies.hugelatex = 45000
max_strings.hugelatex = 55000
pool_free.hugelatex = 47500
nest_size.hugelatex = 500
param_size.hugelatex = 1500
save_size.hugelatex = 5000
stack_size.hugelatex = 15000

% jadetex
extra_mem_top.jadetex = 8000000
extra_mem_bot.jadetex = 8000000
hash_extra.jadetex = 20000
pool_size.jadetex = 5000000
string_vacancies.jadetex = 45000
max_strings.jadetex = 55000
pool_free.jadetex = 47500
nest_size.jadetex = 500
param_size.jadetex = 1500
save_size.jadetex = 5000
stack_size.jadetex = 15000

% pdfjadetex
extra_mem_top.pdfjadetex = 8000000
extra_mem_bot.pdfjadetex = 8000000
hash_extra.pdfjadetex = 20000
pool_size.pdfjadetex = 5000000
string_vacancies.pdfjadetex = 45000
max_strings.pdfjadetex = 55000
```

```
                pool_free.pdfjadetex = 47500
                nest_size.pdfjadetex = 500
                param_size.pdfjadetex = 1500
                save_size.pdfjadetex = 5000
                stack_size.pdfjadetex = 15000
```

Here, we've gone ahead and added entries for jadetex and pdfjadetex, which we'll be setting up below. You can play with these memory settings any way you like if you experience trouble with them.

After setting up hugelatex, like above, it may not work until the texhash program is called:

```
                root# texhash
                texhash: Updating /usr/share/texmf/ls−R...
                texhash: Updating /var/cache/fonts/ls−R...
                texhash: Done.
                root#
```

3.1.2.2. jadetex & pdfjadetex

Setting up jadetex and pdfjadetex is similar to hugelatex.

```
                cd /usr/local/openjade−1.3/dsssl
                make −f Makefile.jadetex install
                # make creates and installs the .fmt
                # files to /usr/share/texmf/web2c

                # Now create symlinks ...
                cd /usr/share/texmf/bin
                ln −s tex jadetex
                ln −s tex pdfjadetex

                # Finally, run texhash.
                root# texhash
```

This Makefile uses hugelatex, so hugelatex must have been setup already. When tex is run as hugelatex, jadetex, or pdfjadetex, it gets its program name (context) from argv[0] in the environment. Then, it scans texmf.cnf, and uses any context−specific settings it finds. The format (.fmt) files in /usr/share/texmf/web2c are also loaded based on the context.

Jadetex takes a tex file generated from openjade, and outputs a dvi. pdfjadetex takes a tex file generated from openjade, and outputs a pdf. The dvips program takes the dvi and outputs a postscript ps file.

3.2. DocBook SGML DTD

3.2.1. Unpack the DocBook SGML DTD

The DocBook DTD is just some sgml text files, so there is nothing to compile. Just unzip them somewhere:

```
                # DocBook DTD V4.1 in
```

```
# /usr/local/share/sgml/docbook/4.1

cd /usr/local/share
mkdir sgml; cd sgml
mkdir docbook; cd docbook
mkdir 4.1; cd 4.1
unzip −a ~/docbk41.zip
```

If you install doctools−1.2 from the XFree86 distribution, it will put some older versions of DocBook DTD, like 2.4.1/ and 3.0/ in subdirectories of docbook.

There are some differences between the different versions of the DocBook DTD. The xxissues.txt files document those issues. Tags have been added, removed, and renamed between the versions.

If you need to use DocBook DTD V3.1, it is available from the same place where V4.1 is downloaded. V3.1 is used a lot, so its a good idea to get it and install it in a 3.1/ subdirectory.

3.2.2. Unpack the ISO8879 Entities

For each DocBook DTD version unpacked, go into its directory and unpack the iso8879−entities.tar.gz file:

```
cd /usr/local/share/sgml/docbook/4.1
tar −xvzf ~/iso8879−entities.tar.gz
```

In each DocBook directory, there should be a docbook.cat file or a catalog file, or both. If both are present, they are likely to be identical. If only docbook.cat is present, go ahead and make a symlink:

```
# If needed ...
cd /usr/local/share/sgml/docbook/4.1
ln −s docbook.cat catalog
```

3.3. DocBook DSSSL

Installation of the DocBook DSSSL, which works for all versions of DocBook, is just a matter of unzipping it somwhere.

```
cd /usr/local/share/sgml
mkdir dsssl; cd dsssl
unzip −a ~/db160.zip

# If you downloaded the ldp.dsl stylesheet
# customization, copy it to ...
cd docbook
cp ~/ldp.dsl html
cp ~/ldp.dsl print
# Copy into both directories.
```

That's all there is to installing the DSSSL, except for the setup of the `$SGML_CATALOG_PATH` discussed later. Don't forget to straighten out the file modes and owner/group of these unpacked files − often they are scrambled and inappropriate.

3.4. sgmltools−lite

If you like it, you can install the sgmltools−lite, but it is optional. Its installation is the standard:

```
                cd /usr/src
                tar −xvzf ~/sgmltools−lite−3.0.2.tar.gz
                cd sgmltools−lite−3.0.2
                ./configure
                make install
```

This installs the sgmltools python script to /usr/local/bin. Note that it uses python, so if you don't have it, then this package is useless.

One tweak that has to be done to make the sgmltools script work, is you have have to edit it and set the path to openjade: vi `which sgmltools`. Consult its docs to learn more about it.

3.5. htmldoc

3.5.1. binary

Preferrably you downloaded a binary distribution of htmldoc for your platform. The installation is straightforward: just unpack it and run the setup. Read the docs in the package for more info.

3.5.2. source

If you downloaded the source, you will also need the Fast Light Tool Kit or else it will not link:

http://www.fltk.org/

Installation is autoconf style. Just run the configure script, make, make install. If all goes well, it will install in /usr/bin.

3.5.3. ldp_print

The htmldoc program has a few glitches when generating output from html files from openjade. For instance, bullet items are not rendered properly and shaded areas are not always shaded.

To fix this problem, a perl script (ldp_print) is available from LinuxDoc.org. The script processes a nochunks html file from openjade and then runs htmldoc on it to produce correctly rendered pdf and ps.

Tip: Get it!

```
tar −xvzf ldp_print.tar.gz
cd ldp_print
cp fix_print_html.lib /usr/lib/perl5/site_perl

# Copy the lib somewhere where perl looks
cp ldp_print /usr/local/bin
```

Take a look at the script in case there are lines in it you need to change for your system. Perhaps someday htmldoc's bugs will be fixed and this script will not be needed anymore.

3.6. DocBook2X and SGMLS.pm (sgmlspl)

3.6.1. sgmlspl

Before the spec files from DocBook2X are of any use, the SGMLS.pm module for perl5 has to be installed, assuming that perl5 is installed. The installation of this module is not as automated as most perl module installs. It uses a Makefile that has to be edited first before running make.

```
cd /usr/src
tar −xvzf ~/SGMLSpm−1.03ii.tar.gz
cd SGMLSpm

# Edit Makfile
vi Makefile
# In the user options of the Makefile
# set everything correct for
# your system.
# Example:
#       PERL = /usr/bin/perl
#       BINDIR = /usr/local/bin
#       PERL5DIR = /usr/lib/perl5/site_perl
#       MODULEDIR = ${PERL5DIR}/SGMLS
#       SPECDIR = ${PERL5DIR}
#       HTMLDIR= /usr/local/apache/htdocs

make install
```

sgmlspl gets copied to /usr/local/bin.

3.6.2. docbook2X (docbook2man−spec.pl)

DocBook2X contains no program to compile or install, though it has some scripts you might want to look at, so all there is to do is unpack it somwhere.

```
cd /usr/local/share/sgml
tar −xvzf ~/docbook2X−0.6.0.tar.gz
cd docbook2X
```

In the unpacked directory is the docbook2man−spec.pl and a patch file for it that corrects a few things. Applying the patch is optional but recommended.

```
                patch docbook2man−spec.pl docbook2man−spec.pl.patch
```

Later, in Using DocBook, you will see how to use sgmlspl and docbook2man−spec.pl to generate a manpage from a refentry docbook document.

3.7. $SGML_CATALOG_FILES

The `$SGML_CATALOG_FILES` environment variable is used by openjade (and other SGML software) to locate DTDs and DSL (stylesheets). SGML software cannot function without finding these files, which have been unpacked to various directories. Given the setup as done so far, here is how `$SGML_CATALOG_FILES` can be set in /etc/profile:

```
##########################################################################################
# SGML DocBook − openjade sgmltools−lite
JADE_HOME=/usr/local/openjade−1.3
SGML_SHARE=/usr/local/share/sgml

PATH=$PATH:$JADE_HOME/bin

# DSSSL stylesheets
#       Norm Walsh's Modular DocBook Stylesheets
SGML_CATALOG_FILES=$SGML_SHARE/dsssl/docbook/catalog
#       OpenJade stylesheets
SGML_CATALOG_FILES=$SGML_CATALOG_FILES:$JADE_HOME/dsssl/catalog
#       sgmltools−lite's stylesheets
SGML_CATALOG_FILES=$SGML_CATALOG_FILES:$SGML_SHARE/stylesheets/sgmltools/sgmltools.cat

# DocBook DTD
#       From OASIS−Open.org
SGML_CATALOG_FILES=$SGML_CATALOG_FILES:$SGML_SHARE/docbook/3.1/catalog
SGML_CATALOG_FILES=$SGML_CATALOG_FILES:$SGML_SHARE/docbook/4.1/catalog
#       These old ones were installed with doctools−1.2 from XFree86.org
SGML_CATALOG_FILES=$SGML_CATALOG_FILES:$SGML_SHARE/docbook/2.4.1/catalog
SGML_CATALOG_FILES=$SGML_CATALOG_FILES:$SGML_SHARE/docbook/3.0/catalog

# sgmltools−lite catalogs for LinuxDoc
SGML_CATALOG_FILES=$SGML_CATALOG_FILES:$SGML_SHARE/dtd/sgmltools/catalog

export JADE_HOME SGML_SHARE PATH SGML_CATALOG_FILES
##########################################################################################
```

Save your profile, logout and then log back in to take effect.

Installation is complete! In the next section, we'll test the installation and convert some test DocBook files.

4. Using DocBook

Now that everything is installed, it's time to test it out and see how to use openjade and the other tools.

Figure 1. Example DocBook SGML file − test.sgml

```
<!DOCTYPE article PUBLIC "−//OASIS//DTD DocBook V4.1//EN">

<article lang="en">
<articleinfo>
        <title>This is a Test</title>

        <author>
                <firstname>John</firstname>
                <surname>Doe</surname>
                <othername role="mi">L</othername>
                <affiliation>
                        <address>
                        <email>j.doe@jdoe dot com</email>
                        </address>
                </affiliation>
        </author>

        <revhistory>
                <revision>
                <revnumber>v1.0</revnumber>
                <date>2000−12−30</date>
                <authorinitials>jld</authorinitials>
                </revision>
        </revhistory>

        <abstract>
        <para>
        This is a test DocBook document.
        </para>
        </abstract>

</articleinfo>

<sect1 id="test1">
<title>Test 1</title>
<para>
Test section 1.
</para>
        <sect2>
        <title>Test 1.1</title>
        <para>
        Test section 1.1
        </para>
        </sect2>

        <sect2>
        <title>Test 1.2</title>
        <para>
        <screen>
                −− Test section 1.2
                openjade −t sgml −d $DSLFILE test.sgml
        </screen>
        </para>
        </sect2>

</sect1>

<sect1 id="test2">
<title>Test 2</title>
<para>
Test section 2.
</para>

        <sect2>
        <title>Test 2.1</title>
        <para>
        Test section 2.1
        </para>
        </sect2>

        <sect2>
        <title>Test 2.2</title>
        <para>
        Test section 2.2
        </para>
        </sect2>

</sect1>
</article>
```

For a guide to DocBook and a reference of DocBook elements, go to:

DocBook: The Definitive Reference. http://www.docbook.org/tdg/html/docbook.html

4.1. Generating HTML

4.1.1. docbook.dsl

Figure 2. Generating HTML output using docbook.dsl

```
bash$ ls −l
total 4
−rw−r−−r−−   1 reaster  users        1077 Dec 31 16:25 test.sgml
bash$ echo $SGML_SHARE
/usr/local/share/sgml
bash$ openjade −t sgml −d $SGML_SHARE/dsssl/docbook/html/docbook.dsl test.sgml
openjade:/usr/local/share/sgml/docbook/4.1/catalog:22:0:W: DTDDECL catalog entries are not suppor
[snip − DTDDECL catalog entries are not supported, repeats]
openjade:/usr/local/share/sgml/docbook/4.1/catalog:22:0:W: DTDDECL catalog entries are not suppor
bash$ ls −l
total 12
−rw−r−−r−−   1 reaster  users        1885 Dec 31 17:34 t1.htm
−rw−r−−r−−   1 reaster  users        1077 Dec 31 16:25 test.sgml
−rw−r−−r−−   1 reaster  users        1544 Dec 31 17:34 x27.htm
bash$
```

The warnings about DTDDECL can be ignored. They might be a little annoying, but these warnings are normal when using jade. Other warnings and errors should be looked at and often indicate syntax errors that you should fix.

Two htm files are generated, one for each <SECT1>. The filenames are not very descriptive. Section one appears on the same page as the article information. These are the results of using the default stylesheet that comes with the Modular DocBook Stylesheets, docbook.dsl.

Stylesheets can be customized to improve on these defaults. If you downloaded the Linux Documentation Project's ldp.dsl file and installed it as shown in Section 3.3, then you already have a customized style available.

4.1.2. ldp.dsl

Figure 3. Generating HTML output using ldp.dsl

```
bash$ openjade -t sgml -d $SGML_SHARE/dsssl/docbook/html/ldp.dsl#html test.sgml
bash$ ls -l
total 16
-rw-r--r--   1 reaster  users        2006 Dec 31 18:00 index.html
-rw-r--r--   1 reaster  users        1077 Dec 31 16:25 test.sgml
-rw-r--r--   1 reaster  users        1677 Dec 31 18:00 test1.html
-rw-r--r--   1 reaster  users        1598 Dec 31 18:00 test2.html
bash$
```

Using ldp.dsl, the output looks better:

- An index file has been created that contains the article information.
- A table of contents has been automatically generated.
- Each <SECT1> is in its own file.
- Filenames are derived from ID attributes of the <SECT1> elements.
- The file extension has changed to html.
- The <SCREEN> elements are shaded.

Note how the ldp.dsl file is written in the command line: it has "#html" appended. Ldp.dsl contains two <STYLE−SPECIFICATION> elements, one with ID="html" and another with ID="print". This selects the html style from within ldp.dsl. The DocBook DSSSL contains support for converting DocBook files into html and print formats. In Section 3.3, we copied ldp.dsl into both the print and html directories. When generating html output, the html style should be selected like above. When generating other types of files, such as rtf and tex, they fall under the print style and so the print style should be selected from ldp.dsl. The alternative is to comment out or delete the print or html style in the copy of ldp.dsl in the respective directory. If a dsl file has more than one style−spec in it and none is selected like in the example above, then the first style encountered in the file is selected. For ldp.dsl, the print style−spec is first in the file, so it gets selected by default. So in the example above, without appending "#html" when specifying ldp.dsl as the dsssl stylesheet, the "print" style−spec would be selected and used when generating the html output. It will work, but is intended for when selecting the "print/ldp.dsl" and the formatting will be different.

To learn more about how the stylesheet customization files are made, read the documentation for the Modular DocBook Stylesheets. Customization mainly involves setting boolean option parameters to toggle style features on and off. Completely new style logic can be programmed using DSSSL's Core Programming Language, as mentioned in Section 2.4.

The openjade option "−t output_type" specifies the output type. The "−d dsssl_spec" option is the path to the dsssl stylesheet to use. In the example above, the output type specified is sgml, which is for SGML to SGML transformations. HTML, defined by the HTML Document Type Definition (DTD), is an SGML document type just as DocBook is, so "sgml" is the correct output_type option. The other two output types commonly used are "rtf" and "tex". The tex output_type will be used later as an intermediate format for the generation of pdf and ps formats. The dsssl_spec must specify a dsl file, not a directory.

4.2. Generating rtf and tex

```
bash$ ls −l
-rw-r--r--   1 reaster  users        1143 Dec 31 18:18 test.sgml
bash$ openjade −t rtf −d $SGML_SHARE/dsssl/docbook/print/ldp.dsl#print test.sgml
bash$ openjade −t tex −d $SGML_SHARE/dsssl/docbook/print/ldp.dsl#print test.sgml
bash$ ls −l
-rw-r--r--   1 reaster  users        4584 Dec 31 20:51 test.rtf
-rw-r--r--   1 reaster  users        1143 Dec 31 18:18 test.sgml
-rw-r--r--   1 reaster  users       18719 Dec 31 20:51 test.tex
```

4.3. Generating dvi and ps

Figure 4. Running jadetex to generate a Device Independent (dvi) file.

```
-rw-r--r--   1 reaster  users        4584 Dec 31 20:51 test.rtf
-rw-r--r--   1 reaster  users        1143 Dec 31 18:18 test.sgml
-rw-r--r--   1 reaster  users       18719 Dec 31 20:51 test.tex
bash$ jadetex test.tex
This is TeX, Version 3.14159 (Web2C 7.3.1)
(test.tex
JadeTeX 1999/06/29: 2.7
(/usr/share/texmf/tex/latex/psnfss/t1ptm.fd)
(/usr/share/texmf/tex/jadetex/isoents.tex)
Elements will be labelled
Jade begin document sequence at 19
No file test.aux.
(/usr/share/texmf/tex/latex/cyrillic/ot2cmr.fd)
(/usr/share/texmf/tex/latex/base/ts1cmr.fd)
(/usr/share/texmf/tex/latex/lucidabr/lmrhlcm.fd)
(/usr/share/texmf/tex/latex/hyperref/nameref.sty)
(/usr/share/texmf/tex/latex/psnfss/t1phv.fd)

LaTeX Warning: Reference `TEST1' on page 1 undefined on input line 238.


LaTeX Warning: Reference `20' on page 1 undefined on input line 262.


LaTeX Warning: Reference `23' on page 1 undefined on input line 285.
LaTeX Warning: Reference `TEST2' on page 1 undefined on input line 316.


LaTeX Warning: Reference `30' on page 1 undefined on input line 340.


LaTeX Warning: Reference `33' on page 1 undefined on input line 363.

[1.0.46] (/usr/share/texmf/tex/latex/psnfss/t1pcr.fd) [2.0.46] [3.0.46]
(test.aux)

LaTeX Warning: There were undefined references.

 )
Output written on test.dvi (3 pages, 34984 bytes).
Transcript written on test.log.
bash$ ls −l
total 80
−rw−r−−r−−   1 reaster  users         771 Dec 31 20:55 test.aux
−rw−r−−r−−   1 reaster  users       34984 Dec 31 20:55 test.dvi
−rw−r−−r−−   1 reaster  users        5072 Dec 31 20:55 test.log
−rw−r−−r−−   1 reaster  users        4584 Dec 31 20:51 test.rtf
−rw−r−−r−−   1 reaster  users        1143 Dec 31 18:18 test.sgml
−rw−r−−r−−   1 reaster  users       18719 Dec 31 20:51 test.tex
bash$ jadetex test.tex
This is TeX, Version 3.14159 (Web2C 7.3.1)
(test.tex
JadeTeX 1999/06/29: 2.7
(/usr/share/texmf/tex/latex/psnfss/t1ptm.fd)
(/usr/share/texmf/tex/jadetex/isoents.tex)
Elements will be labelled
Jade begin document sequence at 19
(test.aux) (/usr/share/texmf/tex/latex/cyrillic/ot2cmr.fd)
(/usr/share/texmf/tex/latex/base/ts1cmr.fd)
(/usr/share/texmf/tex/latex/lucidabr/lmrhlcm.fd)
(/usr/share/texmf/tex/latex/hyperref/nameref.sty)
(/usr/share/texmf/tex/latex/psnfss/t1phv.fd) [1.0.46]
(/usr/share/texmf/tex/latex/psnfss/t1pcr.fd) [2.0.46] [3.0.46] (test.aux) )
Output written on test.dvi (3 pages, 34148 bytes).
Transcript written on test.log.
You have new mail in /var/spool/mail/reaster
bash$ ls −l
total 80
−rw−r−−r−−   1 reaster  users         753 Dec 31 20:58 test.aux
−rw−r−−r−−   1 reaster  users       34148 Dec 31 20:58 test.dvi
−rw−r−−r−−   1 reaster  users        4433 Dec 31 20:58 test.log
−rw−r−−r−−   1 reaster  users        4584 Dec 31 20:51 test.rtf
−rw−r−−r−−   1 reaster  users        1143 Dec 31 18:18 test.sgml
−rw−r−−r−−   1 reaster  users       18719 Dec 31 20:51 test.tex
bash$
```

The first time jadetex is run, warnings are printed. They can be ignored. Running it a second time, they do not appear again.

Figure 5. Running dvips to generate a Postscript (ps) file.

```
bash$ ls −l
total 80
```

```
-rw-r--r--   1 reaster  users         753 Dec 31 20:58 test.aux
-rw-r--r--   1 reaster  users       34148 Dec 31 20:58 test.dvi
-rw-r--r--   1 reaster  users        4433 Dec 31 20:58 test.log
-rw-r--r--   1 reaster  users        4584 Dec 31 20:51 test.rtf
-rw-r--r--   1 reaster  users        1143 Dec 31 18:18 test.sgml
-rw-r--r--   1 reaster  users       18719 Dec 31 20:51 test.tex
bash$ dvips test.dvi
This is dvips(k) 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 2000.12.31:2058' -> test.ps
<texc.pro><8r.enc><texps.pro><special.pro><color.pro>. [1] [2] [3]
bash$ ls -l
total 116
-rw-r--r--   1 reaster  users         753 Dec 31 20:58 test.aux
-rw-r--r--   1 reaster  users       34148 Dec 31 20:58 test.dvi
-rw-r--r--   1 reaster  users        4433 Dec 31 20:58 test.log
-rw-r--r--   1 reaster  users       34817 Dec 31 21:06 test.ps
-rw-r--r--   1 reaster  users        4584 Dec 31 20:51 test.rtf
-rw-r--r--   1 reaster  users        1143 Dec 31 18:18 test.sgml
-rw-r--r--   1 reaster  users       18719 Dec 31 20:51 test.tex
bash$
```

Figure 6. Running htmldoc to generate a Postscript (ps) file.

```
bash$ ls -l
-rw-r--r--   1 reaster  users        1143 Dec 31 18:18 test.sgml
bash$ export DSL_HTML=$SGML_SHARE/dsssl/docbook/html/ldp.dsl\#html
bash$ openjade -t sgml -V nochunks -d $DSL_HTML test.sgml | htmldoc -f test-htmldoc.ps -
bash$ ls -l
-rw-r--r--   1 reaster  users        9050 Jan  1 00:44 test-htmldoc.ps
-rw-r--r--   1 reaster  users        1143 Dec 31 18:18 test.sgml
bash$
```

If the ps doesn't appear as expected, it is due to bugs in htmldoc. Look inside the ldp_print script if you want to use it to make ps.

4.4. Generating pdf

Figure 7. Running pdfjadetex to generate a Portable Document Format (pdf) file.

```
bash$ ls -l
-rw-r--r--   1 reaster  users         753 Dec 31 20:58 test.aux
-rw-r--r--   1 reaster  users       34148 Dec 31 20:58 test.dvi
-rw-r--r--   1 reaster  users        4433 Dec 31 20:58 test.log
-rw-r--r--   1 reaster  users       34817 Dec 31 21:06 test.ps
-rw-r--r--   1 reaster  users        4584 Dec 31 20:51 test.rtf
-rw-r--r--   1 reaster  users        1143 Dec 31 18:18 test.sgml
-rw-r--r--   1 reaster  users       18719 Dec 31 20:51 test.tex
bash$ pdfjadetex test.tex
This is pdfTeX, Version 3.14159-13d (Web2C 7.3.1)
(test.tex[/usr/share/texmf/pdftex/config/pdftex.cfg]
JadeTeX 1999/06/29: 2.7
(/usr/share/texmf/tex/latex/psnfss/t1ptm.fd)
(/usr/share/texmf/tex/jadetex/isoents.tex)
Elements will be labelled
Jade begin document sequence at 19
(test.aux) (/usr/share/texmf/tex/latex/cyrillic/ot2cmr.fd)
(/usr/share/texmf/tex/latex/base/ts1cmr.fd)
(/usr/share/texmf/tex/latex/lucidabr/lmrhlcm.fd)
(/usr/share/texmf/tex/context/base/supp−pdf.tex
(/usr/share/texmf/tex/context/base/supp−mis.tex
loading : Context Support Macros / Missing
)
loading : Context Support Macros / PDF
) (/usr/share/texmf/tex/latex/hyperref/nameref.sty)
(/usr/share/texmf/tex/latex/psnfss/t1phv.fd) [1.0.46[/usr/share/texmf/dvips/con
fig/pdftex.map]] (/usr/share/texmf/tex/latex/psnfss/t1pcr.fd) [2.0.46] [3.0.46]
 (test.aux) )<8r.enc>
Output written on test.pdf (3 pages, 9912 bytes).
Transcript written on test.log.
bash$ ls −l
total 128
−rw−r−−r−−   1 reaster  users         753 Dec 31 21:13 test.aux
−rw−r−−r−−   1 reaster  users       34148 Dec 31 20:58 test.dvi
−rw−r−−r−−   1 reaster  users        5075 Dec 31 21:13 test.log
−rw−r−−r−−   1 reaster  users        9912 Dec 31 21:13 test.pdf
−rw−r−−r−−   1 reaster  users       34817 Dec 31 21:06 test.ps
−rw−r−−r−−   1 reaster  users        4584 Dec 31 20:51 test.rtf
−rw−r−−r−−   1 reaster  users        1143 Dec 31 18:18 test.sgml
−rw−r−−r−−   1 reaster  users       18719 Dec 31 20:51 test.tex
bash$
bash$ pdfjadetex test.tex
[snip]
bash$ pdfjadetex test.tex
[snip]
```

Pdfjadetex must be run up to three times to resolve all internal references for things such as TOC page numbers.

Figure 8. Running htmldoc to generate a Portable Document Format (pdf) file.

```
bash$ ls −l
−rw−r−−r−−   1 reaster  users        1143 Dec 31 18:18 test.sgml
bash$ export DSL_HTML=$SGML_SHARE/dsssl/docbook/html/ldp.dsl\#html
bash$ openjade −t sgml −V nochunks −d $DSL_HTML test.sgml > test−htmldoc.htm
bash$ ldp_print test−htmldoc.htm
bash$ ls −l
−rw−r−−r−−   1 reaster  users        9050 Jan  1 01:17 test−htmldoc.pdf
−rw−r−−r−−   1 reaster  users        1143 Dec 31 18:18 test.sgml
bash$
```

If enabled in the ldp_print script, this would generate a ps file also.

4.5. Using make

Repeating the commands to generate the output files is tedious. The make command works perfectly to automate the process.

Figure 9. Filename: Makefile − automates document generation.

```
# Generates online and print versions of SGML source file.

BASENAME=DocBook−Install

# SGML source file.
SGML_FILE=$(BASENAME).sgml

# Stylesheets
DSL_PRINT=$(SGML_SHARE)/dsssl/docbook/print/ldp.dsl\#print
DSL_HTML=$(SGML_SHARE)/dsssl/docbook/html/ldp.dsl\#html

# Generated files.
HTML_FILE=index.html
HTM_FILE=$(BASENAME).htm
TEX_FILE=$(BASENAME).tex
RTF_FILE=$(BASENAME).rtf
PDF_FILE=$(BASENAME).pdf
DVI_FILE=$(BASENAME).dvi
PS_FILE=$(BASENAME).ps


# Build rules.

html: $(HTML_FILE)

htm: $(HTM_FILE)

tex: $(TEX_FILE)

rtf: $(RTF_FILE)

pdf: $(PDF_FILE)

dvi: $(DVI_FILE)

ps: $(PS_FILE)

all: html htm tex rtf pdf dvi ps

clean:
        rm −f $(BASENAME).{htm,log,aux,ps,pdf,tex,dvi,rtf,fot}
        rm −f *.html

distclean: clean
        rm −f $(BASENAME).tgz

package:
        rm −f $(BASENAME).tgz
        tar −C .. −czf /tmp/$(BASENAME).tgz $(BASENAME)
        mv /tmp/$(BASENAME).tgz .

dist: clean package

distall: all package


# Compile rules.

$(HTML_FILE): $(SGML_FILE)
```

```
        openjade −t sgml −d $(DSL_HTML) $(SGML_FILE)

$(HTM_FILE): $(SGML_FILE)
        openjade −t sgml −V nochunks −d $(DSL_HTML) \
        $(SGML_FILE) > $(HTM_FILE)

$(TEX_FILE): $(SGML_FILE)
        openjade −t tex −d $(DSL_PRINT) $(SGML_FILE)

$(RTF_FILE): $(SGML_FILE)
        openjade −t rtf −d $(DSL_PRINT) $(SGML_FILE)

# [pdf]jadetex is run 3 times to resolve references.
#$(PDF_FILE): $(TEX_FILE)
#       pdfjadetex $(TEX_FILE)
#       pdfjadetex $(TEX_FILE)
#       pdfjadetex $(TEX_FILE)

# This *should* work, but htmldoc has bugs ...
#$(PDF_FILE): $(SGML_FILE)
#       openjade −t sgml −V nochunks −d $(DSL_HTML) \
#       $(SGML_FILE) | htmldoc −f $(PDF_FILE) −

# Have to use ldp_print to work around htmldoc bugs
# ldp_print can also do the ps file − see script
$(PDF_FILE): $(HTM_FILE)
        ldp_print $(HTM_FILE)

$(DVI_FILE): $(TEX_FILE)
        jadetex $(TEX_FILE)
        jadetex $(TEX_FILE)
        jadetex $(TEX_FILE)

$(PS_FILE): $(DVI_FILE)
        dvips $(DVI_FILE)

#$(PS_FILE): $(SGML_FILE)
#       openjade −t sgml −V nochunks −d $(DSL_HTML) \
#       $(SGML_FILE) | htmldoc −f $(PS_FILE) −
```

Usage is just like for most projects:

Figure 10. Invoking make to run Makefile

```
                −− generate html (default)
                make
                −− generate just pdf
                make pdf
                −− generate all files
                make all
                −− delete all generated files
                make clean
                −− create tgz distribution
                −− with no generated files
                make dist
                −− create tgz distribution
                −− containing all generated files
                make distall
```

Notice the commented compile rules for pdf and ps which provide alternative means of generating those files.

4.6. Generating a manpage

During the section on installing everything, we installed the perl5 module SGMLSpm. Then we installed docbook2X which provides the spec.pl files for transforming DocBook RefEntry documents into nroff (manpage) format with sgmlspl.

An example Docbook RefEntry document, for the `foo` command, is given below.

Figure 11. foo manpage, docbook refentry source (foo−ref.sgml)

```
<!DOCTYPE refentry PUBLIC "−//OASIS//DTD DocBook V4.1//EN">
<refentry>
<refentryinfo>
        <date>2001−01−01</date>
</refentryinfo>
<refmeta>
        <refentrytitle>
                <application>foo</application>
        </refentrytitle>
        <manvolnum>1</manvolnum>
        <refmiscinfo>foo 1.0</refmiscinfo>
</refmeta>
<refnamediv>
        <refname>
                <application>foo</application>
        </refname>
        <refpurpose>
        Does nothing useful.
        </refpurpose>
</refnamediv>
<refsynopsisdiv>
        <refsynopsisdivinfo>
                <date>2001−01−01</date>
        </refsynopsisdivinfo>
        <cmdsynopsis>
        <command>foo</command>
<arg><option>−f </option><replaceable class="parameter">bar</replaceable></arg>
<arg><option>−d<replaceable class="parameter">n</replaceable></option></arg>
<arg rep="repeat"><replaceable class="parameter">file</replaceable></arg>
        </cmdsynopsis>
</refsynopsisdiv>
<refsect1>
        <refsect1info>
                <date>2001−01−01</date>
        </refsect1info>
        <title>DESCRIPTION</title>
        <para>
        <command>foo</command> does nothing useful.
        </para>
</refsect1>
<refsect1>
        <title>OPTIONS</title>
        <variablelist>
        <varlistentry>
```

```
                <term>−f <replaceable class="parameter">bar</replaceable></term>
                <listitem>
                        <para>
                        Takes <filename>bar</filename> as it's run
                        control file. If this were a real program,
                        there might be more to say here about what
                        bar is and how it will be used.
                        </para>
                </listitem>
        </varlistentry>
        <varlistentry>
                <term>−d<replaceable class="parameter">n</replaceable></term>
                <listitem>
                        <para>
                        Do something, where integer
                        <replaceable class="parameter">n</replaceable>
                        specifies how many times.
                        </para>
                </listitem>
        </varlistentry>
        <varlistentry>
                <term><replaceable class="parameter">file...</replaceable></term>
                <listitem>
                        <para>
                        Processes the files in the order listed,
                        sending all output to stdout.
                        </para>
                </listitem>
        </varlistentry>
        </variablelist>
</refsect1>
<refsect1>
        <title>USAGE</title>
        <para>
        <command>foo</command> −f foo.conf −d2 foodata.foo
        </para>
</refsect1>
<refsect1>
        <title>CAVEATS</title>
        <para>
        Other programs named <command>foo</command> may exist and actually
        do something!
        </para>
</refsect1>
<refsect1>
        <title>BUGS</title>
        <para>
        None.  Program does nothing.
        </para>
</refsect1>
<refsect1>
        <title>AUTHOR</title>
        <para>
        <author>
                <firstname>Foo</firstname>
                <othername role="mi">E</othername>
                <surname>Bar</surname>
                <contrib>Original author</contrib>
        </author>
        </para>
</refsect1>
</refentry>
```

Figure 12. Generating a manpage with onsgmls, sgmlspl, and docbook2man−spec.pl

```
bash$ ls −l
−rw−r−−r−−   1 reaster  users        2434 Jan  3 03:51 foo−ref.sgml
bash$ onsgmls foo−ref.sgml | sgmlspl $SGML_SHARE/docbook2X/docbook2man−spec.pl
onsgmls:/usr/local/share/sgml/docbook/4.1/catalog:22:0:W: DTDDECL catalog entries are not support
bash$ ls −l
−rw−r−−r−−   1 reaster  users        2434 Jan  3 03:51 foo−ref.sgml
−rw−r−−r−−   1 reaster  users        1129 Jan  3 04:03 foo.1
−rw−r−−r−−   1 reaster  users           0 Jan  3 04:03 manpage.links
−rw−r−−r−−   1 reaster  users           0 Jan  3 04:03 manpage.log
−rw−r−−r−−   1 reaster  users          15 Jan  3 04:03 manpage.refs
bash$ groff −mandoc −Tascii foo.1

FOO(1)                                                     FOO(1)


NAME
       foo − Does nothing useful.

SYNOPSIS
       foo [ −f bar ]  [ −dn ]  [ file... ]

DESCRIPTION
       foo does nothing useful.

OPTIONS
       −f bar Takes  bar as its run control file. If this were a
              real program, there might be more to say here about
              what bar is and how it will be used.

       −dn    Do  something,  where  integer n specifies how many
              times.

       file...
              Processes the files in the  order  listed,  sending
              all output to stdout.

USAGE
       foo −f foo.conf −d2 foodata.foo

CAVEATS
       Other  programs  named foo may exist and actually do some−
       thing!

BUGS
       None. Program does nothing.

AUTHOR
       Foo E Bar (Original author)

[snip − several extra blank lines that man would not show]
foo 1.0                    2001−01−01                        1
bash$ groff −mandoc −Tascii foo.1 | less
bash$ less foo.1
```

The manpage, `foo.1`, is generated as a Section 1 page. The groff command is used to give a quick look at its formatted appearance.

To install this manpage, it belongs in any man/man1 directory, where the directory man/ is added to `$MANPATH` in the environment. The standard location is `/usr/local/man/man1`. The standard sections in the manpages system are 1 though 9. Each is for holding specific catagories of documentation.

Table 1. Manpage Sections

Section	Purpose
man1	User programs
man2	System calls
man3	Library functions and subroutines
man4	Devices
man5	File formats
man6	Games
man7	Miscellaneous
man8	System administration
man9	Kernel internal variables and functions

> **Tip:** The source file for a manpage, like `foo−ref.sgml`, can be processed into all the other formats just like any other DocBook file. So using the same commands discussed earlier to generate html and print output types, a manpage can be made into html and rtf, tex, pdf, dvi, and ps. This can really save a lot of conversion work!

Have fun!

---

5. Copyright

5.1. New Versions of this Document

The lastest version of this mini−HOWTO can be found at:

http://www.linuxdoc.org/HOWTO/mini/DocBook−Install/

5.2. Copyright Information

If you have any questions, please contact <`linux−howto@metalab.unc.edu`>

5.3. Disclaimer

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.