

# Sources du projet [inetdoc.LINUX]

Philippe Latu

philippe.latu(at)linux-france.org

<http://www.linux-france.org/prj/inetdoc/>

|  |   |                   |
|--|---|-------------------|
| Historique des versions  |   |                   |
| \$Revision: 1303 \$  | \$Date: 2008-06-03 13:20:29 +0200 (mar, 03 jun 2008) \$ | \$Author: latu \$ |
| Révision installation des paquets Debian Java  |   |                   |
| Résumé   |   |                   |
| Le projet inetdoc.LINUX ne contient pas de code de logiciel ; juste des documents, des images et les Makefiles de traitement. Voici quelques indications sur l'installation d'une copie du projet. Il s'agit juste de générer les pages web et les versions imprimables des documents. |   |                   |

## Table des matières

|   |    |
|---|----|
| 1. Copyright et Licence .....   | 1  |
| 1.1. Méta-information .....   | 2  |
| 2. Installation rapide .....  | 2  |
| 3. Arborescence d'installation .....  | 2  |
| 4. Applications utilisées .....   | 3  |
| 5. Génération des fichiers PDF avec FOP et Java .....   | 4  |
| 5.1. Paquets Java Debian .....  | 4  |
| 5.2. Kit JIMI : traitement des images avec fop 0.20.5 .....                                     | 5  |
| 5.3. Kit JAI : traitement des images avec fop 0.92 beta et plus .....                           | 5  |
| 5.4. Installation de fop à partir du dépôt SVN .....  | 6  |
| 6. Ajout d'un nouveau document .....  | 7  |
| 6.1. Contenu d'un Makefile .....  | 8  |
| 6.2. Types de traitements .....   | 9  |
| 7. Journal des mises à jour : <i>Changelog</i> .....  | 9  |
| 7.1. Génération de la page ChangeLog avec CVS .....   | 10 |
| 7.2. Génération de la page ChangeLog avec SVN .....   | 11 |
| A. Script de transformation du journal XML de Subversion .....                                  | 12 |
| B. Feuille de styles XSL de génération du Changelog à partir du journal XML de Subversion ..... | 14 |

## 1. Copyright et Licence

Copyright (c) 2000,2008 Philippe Latu.  
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2008 Philippe Latu.  
Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.2 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

## 1.1. Méta-information

Cet article est écrit avec *DocBook*<sup>1</sup> XML sur un système *Debian GNU/Linux*<sup>2</sup>. Il est disponible en version imprimable aux formats PDF et Postscript : [sources.pdf](#)<sup>3</sup> | [sources.ps.gz](#)<sup>4</sup>.

## 2. Installation rapide

L'installtion à partir de sources génère les pages web et les versions imprimables des documents du projet [inetdoc.LINUX].

Pour consulter les pages web vous aurez besoin d'un serveur Web configuré pour permettre l'utilisation des pages personnelles. Le répertoire cible par défaut est : `~/public_html/prj/inetdoc`.

Les versions imprimables des documents sont elles aussi placées dans l'arborescence de page web personnelle. Le répertoire cible est : `~/public_html/prj/inetdoc/telechargement/`.

Voici la procédure à suivre :

```
$ wget http://www.linux-france.org/prj/inetdoc/telechargement/inetdoc.src.tar.bz2
$ tar xvjf inetdoc.src.tar.bz2
$ cd lfo/prj/inetdoc
$ make ; make
```

Et voilà, après ... quelques instants, vous disposez d'une copie locale du projet sur votre disque.

Comment ? l'exécution de la commande **make** échoue avec un message d'erreur. Y aurait-il un problème de dépendance ? une application absente ? Vous n'avez qu'à consulter les sections suivantes ;-)).

## 3. Arborescence d'installation

L'installation par défaut utilise la page personnelle de l'utilisateur actif comme racine : `~/public_html/prj/inetdoc`.

Voir la variable `HTML_WRK` dans le fichier `Makefile.Rules` pour modifier ce répertoire par défaut.

```
.
|-- articles
|   |-- adressage.ipv4
|   |-- devmgmt
|   |-- ethernet
|   |-- modelisation
|   |-- reseau.libre
|   |-- rnis
|   |-- segmentation.lan
|   |-- stockage
|   |-- transmission.tcpip
|-- cours
|   |-- admin.reseau.dns
|   |-- admin.reseau.dns-dhcp
|   |-- admin.reseau.fs
|   |-- admin.reseau.neigh
|   |-- admin.reseau.nfs
|   |-- admin.reseau.nis
|   |-- admin.reseau.synthese-dns
|   |-- admin.reseau.synthese-nfs3-nis
|   |-- admin.reseau.tp
```

<sup>1</sup> <http://www.docbook.org>

<sup>2</sup> <http://www.debian.org>

<sup>3</sup> <http://www.linux-france.org/prj/inetdoc/telechargement/sources.pdf>

<sup>4</sup> <http://www.linux-france.org/prj/inetdoc/telechargement/sources.ps.gz>

```

| |-- archi.tp
| |-- config.interface
| |-- explore.gnulinux
| |-- interco.access-control.tp
| |-- interco.noyau
| |-- interco.noyau.synthese
| |-- interco.noyau.tp
| |-- interco.ppp.tp
| |-- interco.rawip.tp
| |-- interconnexion.fonctions.noyau
| |-- interconnexion.perimetres
| |-- intro.analyse
| |-- routage.inter-vlan
| |-- services_internet
| |-- `-- socket.udp.cpp
|-- formations
| |-- explore.gnulinux
| |-- fc.cnes
| |-- m1-stri
| |-- m2-stri
| |-- polycop
| |-- `-- pourquoi.gnulinux
|-- guides
| |-- Advanced-routing-Howto -> lartc
| |-- NAT-HOWTO
| |-- iptables-tutorial
| |-- lartc
| |-- netfilter-hacking-HOWTO
| |-- networking-concepts-HOWTO
| |-- packet-filtering-HOWTO
| |-- rnis
| |-- zebra.ospf
| |-- zebra.rip
| |-- `-- zebra.statique
|-- images
|-- liens
|-- notice.legale
|-- securite
| |-- amavisd-new
| |-- `-- tutoriel
|-- sources
| |-- `-- files
|-- telechargement
| |-- `-- xml

```

## 4. Applications utilisées

Les Makefiles de l'arborescence servent à générer les pages Web et les versions imprimables des documents. Ces documents utilisent plusieurs formats sources différents. Chaque format source nécessite une chaîne de développement spécifique.

Les fichiers sources des documents du projet utilisent les formats : [HTML](#), [DocBook XML](#), [LinuxDoc](#) et [MagicPoint](#). Voici, pour chaque type de source, la liste des paquets/applications associées. Les noms de paquets sont ceux de la distribution *Debian*<sup>5</sup>.

Pour installer le tout :

```

# apt-get install perl sed tidy xsltproc fop \
> libxml2-utils docbook docbook-xml docbook-xsl docbook-utils \
> linuxdoc-tools linuxdoc-tools-latex mgp gs xpdf tetex tetex-extra

```

<sup>5</sup> <http://www.debian.org>

Quelques indications sur les paquets en question :

Documents source de type (X)HTML

- sed : The GNU sed stream editor. Corrections manuelles de balises, (insertion|extraction) de fichiers patrons.
- tidy : outil de validation syntaxique et de mise en forme des fichiers (X)HTML.

Documents source de type DocBook XML

- libxml2-utils : outil **xmllint** de validation syntaxique, de remise en forme des documents XML et de fusion des entités DocBook XML.
- docbook-xml : système de représentation XML standard pour les documents techniques. Ce système est une «collection» de balises qui permet de structurer un document en chapitres, sections, listes, etc.
- docbook-xsl : feuilles de styles XSL standard pour la transformation des fichiers sources DocBook XML vers les formats (X)HTML et FO.

Documents source de type LinuxDoc

- linuxdoc-tools : conversion SGML pour les documents écrits avec le jeu de définitions LinuxDoc DTD. Ce jeu de définition est utilisé pour la publication de documentations «historiques» telles que les *Howtos Netfilter*.

Présentations MagicPoint

- mgp : outil de présentation de vues/transparents basé sur les bibliothèques graphiques du projet *X.Org* (X11).

Formats imprimables PDF et Postscript

- fop ou *fo processor* : transformation des fichiers sources DocBook XML vers le format PDF.
- xpdf : outils de traitement des fichiers Postscript et PDF.

## 5. Génération des fichiers PDF avec FOP et Java

*Apache FOP (Formatting Objects Processor)*<sup>6</sup> est l'outil qui permet de générer les versions imprimables au format PDF à partir des documents sources DocBook XML et d'une feuille de styles XSL-FO. Ce processeur s'appuie sur Java et son développement a un historique un peu mouvementé. Cependant, c'est incontestablement l'outil libre le plus avancé sur la production de versions imprimables.

L'utilisation des outils Java nécessite quelques adaptations locales au système du fait du caractère plus ou moins libre des différentes licences de ces outils. Ces adaptations ressemblent beaucoup à du bricolage misérable.

La version du *Java Runtime Environment* (JRE) influe beaucoup sur le fonctionnement de fop. Par conséquent, il est préférable d'utiliser une version la plus récente possible. Dans ce document on s'appuie sur les paquets construits à partir de la version fournie par Sun microsystems™.

### 5.1. Paquets Java Debian

Pour une installation Java «standard» comprenant la chaîne de développement, on obtient la liste suivante :

```
# dpkg -l sun-java5* | grep ^ii
ii sun-java5-bin      1.5.0-13-1 Sun Java(TM) Runtime Environment (JRE) 5.0 (architecture depen
ii sun-java5-demo    1.5.0-13-1 Sun Java(TM) Development Kit (JDK) 5.0 demos and examples
ii sun-java5-fonts   1.5.0-13-1 Lucida TrueType fonts (from the Sun JRE)
ii sun-java5-jdk     1.5.0-13-1 Sun Java(TM) Development Kit (JDK) 5.0
ii sun-java5-jre     1.5.0-13-1 Sun Java(TM) Runtime Environment (JRE) 5.0 (architecture indep
ii sun-java5-plugin  1.5.0-13-1 The Java(TM) Plug-in, Java SE 5.0
```

Comme il est possible de faire coexister plusieurs versions Java sur le même système, on doit s'assurer que la version voulue sera bien utilisée par les autres outils.

```
# update-java-alternatives --jre --plugin -v -s java-1.5.0-sun
resetting java alternatives
```

<sup>6</sup> <http://xmlgraphics.apache.org/fop/>

```

Pas d'alternatives pour mozilla-snapshot-javaplugin.so.
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/ControlPanel » pour fournir « ControlPanel ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/java » pour fournir « java ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/java_vm » pour fournir « java_vm ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/javaws » pour fournir « javaws ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/keytool » pour fournir « keytool ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/orbd » pour fournir « orbd ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/pack200 » pour fournir « pack200 ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/policytool » pour fournir « policytool ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/rmid » pour fournir « rmid ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/rmiregistry » pour fournir « rmiregistry ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/servertool » pour fournir « servertool ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/tnameserv » pour fournir « tnameserv ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/bin/unpack200 » pour fournir « unpack200 ».
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/plugin/i386/ns7/libjavaplugin_oji.so » pour fou
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/plugin/i386/ns7/libjavaplugin_oji.so » pour fou
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/plugin/i386/ns7/libjavaplugin_oji.so » pour fou
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/plugin/i386/ns7/libjavaplugin_oji.so » pour fou
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/plugin/i386/ns7/libjavaplugin_oji.so » pour fou
Utilisation de « /usr/lib/jvm/java-1.5.0-sun/jre/plugin/i386/ns7/libjavaplugin_oji.so » pour fou

```

Il reste ensuite à vérifier que le paquet correspondant au processeur fop est bien installé :

```

# dpkg -l fop* | grep ^ii
ii fop                1:0.93.dfsg.1-2      XML to PDF Translator

```

## 5.2. Kit JIMI : traitement des images avec fop 0.20.5

L'un des principaux intérêts de l'utilisation de fop tient au traitement simplifié des images. En effet, avec les autres outils de traitement tels que jade et openjade, il était nécessaire de créer des fichiers sous deux formats distincts : un pour les pages web (jpeg, png ou gif) et un pour les documents imprimables (eps). Avec fop on ne spécifie plus qu'un seul format de fichier image quelque soit le traitement. La simplification des sources et des manipulations de format est donc significative.

Cette simplification a un coût l'installation du *JIMI Software Development Kit*<sup>7</sup>.

Une fois le kit téléchargé les manipulations sont les suivantes :

```

$ cd ~/tmp
$ unzip jimil_0.zip
$ find . -type f -name JimiProClasses.zip
$ cd /usr/share/java/
$ su
# cp /home/phil/tmp/Jimi/JimiProClasses.zip jimil-1.0.jar

```

Ouf ! A partir de là, le paquet Debian fop 0.20.5 pourra traiter les images au format PNG.

## 5.3. Kit JAI : traitement des images avec fop 0.92 beta et plus

Comme on l'a déjà indiqué, un des principaux intérêts de l'utilisation de fop tient au traitement simplifié des images. Là encore, cette simplification a un coût l'installation du kit *Java Advanced Imaging (JAI) API*<sup>8</sup>.

À partir du site, on télécharge un fichier du type : jai-1\_1\_4-pre-dr-b03-lib-linux-i586-08\_Oct\_2007.zip. Le nom de ce fichier dépend de la cible sur laquelle le kit sera utilisé. Une fois le téléchargement effectué, les manipulations sont les suivantes :

```

:~/src$ rm -rf jai
:~/src$ mkdir jai ; cd jai
:~/src/jai$ unzip ../jai-1_1_4-pre-dr-b03-lib-linux-i586-08_Oct_2007.zip
Archive:  ../jai-1_1_4-pre-dr-b03-lib-linux-i586-08_Oct_2007.zip

```

<sup>7</sup> <http://java.sun.com/products/jimi/>

<sup>8</sup> <https://jai.dev.java.net/>

```

inflating: COPYRIGHT.txt
inflating: LICENSE.txt
inflating: LICENSE-JRL.txt
inflating: LICENSE-JDL.txt
inflating: LICENSE-mediaLibJAI.txt
inflating: lib/ext/jai_codec.jar
inflating: lib/ext/jai_core.jar
inflating: lib/ext/mlibwrapper_jai.jar
inflating: lib/i386/libmlib_jai.so
~/src/jai$ su
Mot de passe :
/home/phil/src/jai# cp -r lib/* /usr/lib/jvm/java-1.5.0-sun/jre/lib/
/home/phil/src/jai# exit
~/src/jai$ cd ..
~/src$ rm -rf jai

```

Pour s'assurer que le kit a bien été installé au bon endroit dans l'arborescence du système de fichiers, on peut utiliser la commande suivante :

```

$ find /usr/lib/jvm/java-1.5.0-sun/jre/lib/ -type f -name "*jai*"
/usr/lib/jvm/java-1.5.0-sun/jre/lib/i386/libmlib_jai.so
/usr/lib/jvm/java-1.5.0-sun/jre/lib/ext/jai_codec.jar
/usr/lib/jvm/java-1.5.0-sun/jre/lib/ext/jai_core.jar
/usr/lib/jvm/java-1.5.0-sun/jre/lib/ext/mlibwrapper_jai.jar

```

## 5.4. Installation de fop à partir du dépôt SVN

Pour bénéficier des toutes dernières avancées sur le développement de fop, il est nécessaire de construire l'outil à partir de ses sources. Pour ce faire, il faut télécharger les sources à partir du dépôt du serveur Subversion du projet. On passe ensuite aux opérations de construction avec `ant`, l'équivalent de `make` dans l'environnement Java.

La récupération de l'arborescence des sources du projet fop se fait avec `svn` la commande principale du paquet subversion :

```

~/SVN$ mkdir ~/SVN/fop
~/SVN$ cd ~/SVN/fop/
~/SVN/fop$ svn co https://svn.apache.org/repos/asf/xmlgraphics/fop/trunk
<snip/>
A trunk/examples/embedding/README
A trunk/examples/embedding/build.xml
U trunk
Révision 410929 extraite.

```

Après ce téléchargement initial, toutes les opérations de mises à jour se font à l'aide de la commande `svn update`.

On construit ensuite le processeur à l'aide des directives données à la commande `ant`. La première instruction `ant clean` élimine toute trace de fichiers binaires issus de la construction précédente et la seconde `ant package` construit les paquets jar nécessaires à l'utilisation de fop.

```

$ cd ~/SVN/fop/trunk/
~/SVN/fop/trunk$ ant clean
Buildfile: build.xml

clean:
[delete] Deleting directory /home/phil/SVN/fop/trunk/build

BUILD SUCCESSFUL
Total time: 6 seconds

~/SVN/fop/trunk$ ant package
Buildfile: build.xml

init-avail:

```

```

[echo] ----- Apache FOP svn-trunk [1999-2007] -----
[echo] See build.properties and build-local.properties for additional build settings
[echo] Apache Ant version 1.7.0 compiled on August 24 2007
[echo] VM: 1.5.0_13-b05, Sun Microsystems Inc.
[echo] JAVA_HOME: ${env.JAVA_HOME}
[echo] Jimi Support NOT Present
[echo] JAI Support PRESENT
[echo] JCE Support PRESENT
[echo] JUnit Support PRESENT
[echo] XMLUnit Support PRESENT

init-filters-jdk14:
[echo] Use GraphicsConfiguration adapter for JDK 1.4.

init-filters-jdk13:

init:
<snip/>

uptodate-jar-main:

jar-main:
[jar] Building jar: /home/phil/SVN/fop/trunk/build/fop.jar

compile-hyphenation:
[mkdir] Created dir: /home/phil/SVN/fop/trunk/build/classes/hyph

uptodate-jar-hyphenation:

jar-hyphenation:
[jar] Warning: skipping jar archive /home/phil/SVN/fop/trunk/build/fop-hyph.jar because no file
[jar] Building MANIFEST-only jar: /home/phil/SVN/fop/trunk/build/fop-hyph.jar

uptodate-jar-sandbox:

jar-sandbox:
[jar] Building jar: /home/phil/SVN/fop/trunk/build/fop-sandbox.jar

package:

BUILD SUCCESSFUL
Total time: 29 seconds

```

On peut maintenant vérifier que l'outil est disponible.

```

~/SVN/fop/trunk$ ./fop -v
FOP Version svn-trunk
<snip/>

```

Dernière étape essentielle, il faut positionner la variable FOP fichier principal des règles de génération des pages XHTML et des fichiers imprimables : Makefile.Rules.

```

# fop -> Génération FO-PDF
FOP=~ /SVN/fop/trunk/fop

```

Voilà, c'est prêt !. On dispose maintenant d'un processeur qui va permettre de générer des documents PDF en quantités industrielles. Il ne reste plus qu'à rédiger ;)).

## 6. Ajout d'un nouveau document

Pour ajouter un nouveau document, il faut créer un répertoire dans lequel on place le fichier source et un Makefile pour automatiser les traitements.

## 6.1. Contenu d'un Makefile

Voici un patron de Makefile et quelques indications sur ses variables et fonctions :

```
# Support TP routage Inter-VLAN
# $Id: Makefile 1191 2007-10-31 10:01:20Z latu $
# - philippe.latu(at)linux-france.org

# Répertoire racine
ifndef $(MAIN_DIR)
MAIN_DIR = prj/inetdoc
endif

# Répertoire courant
ifndef $(DIR)
DIR = $(MAIN_DIR)/cours/routage.inter-vlan
endif

ifndef $(DEPTH)
DEPTH = $(shell echo $(DIR) | sed 's/[-._a-zA-Z0-9]*\/\.\./g')
endif

# Liste des répertoires à traiter à partir de ce niveau
SUBDIRS = \
  images

# Type(s) de traitement
PROCESS_TYPE = dirs subdirs symlink _xml2xhtml _xml2print

XML_FILES = \
  routage.inter-vlan.xml \
  $(wildcard files/*.txt)

SYMLINKS = \
  $(DEPTH)/common/inetdoc.urls.xml \
  $(DEPTH)/common/author.xml \
  $(DEPTH)/common/legal.xml

all: $(PROCESS_TYPE)

include $(DEPTH)/common/Makefile.Rules
```

DIR

Nom du répertoire de travail courant dans lequel on place le(s) fichier(s) source(s) à traiter.

DEPTH

Profondeur du répertoire de travail relativement à la racine de l'arborescence de travail.

SUBDIRS

Liste des sous-répertoires à traiter. Dans le cas du projet *inetdoc.linux*, un document DocBook de type «bibliographie» pointe vers plusieurs autres documents. Chacun de ces documents est placé dans un sous-répertoire différent.

PROCESS\_TYPE

Liste des traitements à effectuer dans le répertoire de travail. Les différents types de traitements sont détaillés dans [Section 6.2, « Types de traitements »](#).

XML\_FILES

Dans le cas d'un document DocBook ; liste des fichiers source. Dès qu'un document de cette liste est modifié, il faut reprendre le traitement

**SYMLINKS**

Liste des liens symboliques à créer. Un document peut reprendre des éléments contenus dans d'autres répertoires. Les liens symboliques sont alors beaucoup plus utiles que les copies.

`all: dirs subdirs $(PROCESS_TYPE) snapshot_instructions snapshot_build`

Somme des traitements à effectuer :

- `dirs` : création des répertoires contenus dans la liste `DIR`.
- `subdirs` : création des sous-répertoires contenus dans la liste `SUBDIRS`.
- `$(PROCESS_TYPE)` : traitements spécifiques au(x) document(s) donnés dans la liste `PROCESS_TYPE`.
- autres traitements : dans le cas du répertoire `sources`, il s'agit de générer le fichier de distribution des sources du projet.

**6.2. Types de traitements**

L'ensemble des traitements existants sont définis dans le fichiers de règles : `Makefile.Rules`. C'est ce fichier qu'il faut consulter pour connaître de détail des opérations effectuées.

**dirs**

Création des répertoires dans l'arborescence cible à partir de la racine définie par `HTML_WRK`.

**subdirs**

Liste des sous-répertoires dans lesquels d'autres documents sont à traiter.

**clean**

Nettoyage de tous les fichiers intermédiaires générés lors des traitements. La liste des fichiers à effacer est extraite des propriétés `svn:ignore` des répertoires du dépôt Subversion (SVN).

**symlink**

Création des liens symboliques dans le répertoire de travail.

**file**

Copie simple de fichiers dans l'arborescence cible.

**html**

Passage à la lessiveuse `tidy` et copie du résultat dans l'arborescence cible.

**ld2www**

Génération des pages web à partir d'une source LinuxDoc sans oublier le passage par la lessiveuse.

**ld2print**

Génération des formats imprimables Postscript et PDF à partir d'une source LinuxDoc.

**\_xml2xhtml**

Génération des pages web à partir d'une source DocBook sans oublier le passage par la lessiveuse.

**\_xml2print**

Génération des formats imprimables Postscript et PDF à partir d'une source DocBook.

**mgp**

Génération des pages Web et des formats imprimables Postscript et PDF à partir d'une source MagicPoint.

**7. Journal des mises à jour : *Changelog***

La documentation se gère comme le code source d'un logiciel. Les fichiers XML et les `Makefiles` représentent de code source du projet. Pour gérer les évolutions d'un volume important de code source dans le temps, il est essentiel d'utiliser un gestionnaire de contrôle de version.

De façon à ce que les (lecteurs|utilisateurs) des documentations du projet puissent suivre les évolutions «en temps réel», il est très utile de générer une page Web contenant le journal des mises à jour de l'ensemble du projet. Via un artifice de construction de ma page Web personnelle sur le site Linux France (AKA *lfo*), ce journal est placé dans le fichier `README.html` à l'adresse : [ChangeLog \[inetdoc.LINUX\]](http://www.linux-france.org/~platu/README.html)<sup>9</sup>.

## 7.1. Génération de la page ChangeLog avec CVS

Depuis le début du projet [inetdoc.LINUX] en 2001 jusqu'en Novembre 2005, le gestionnaire de contrôle de version était CVS. Pour générer le journal au format XML, il est nécessaire d'utiliser un script fourni dans le paquet du même nom `cvs2cl`. Une fois le journal créé au format XML, on lui applique une feuille de styles XSL pour produire la page Web «lisible» au format XHTML.

Les règles de génération de la page Web contenues dans le Makefile sont :

```
# Date de départ de collecte Changelog.
# Construction du fichier README.html
START_LOG_DATE = 2005-01-01

<snipped/>

$(TARGET_DIR)/README.html:
    cvs2cl --prune --lines-modified --gmt --xml --xml-encoding utf-8 \
    --noxmlns -f changes.xml -l -d '$(START_LOG_DATE)'
    xsltproc -o $@ cl2html.xsl changes.xml
    rm -f changes.xml*
```

La feuille de styles XSL utilisée pour produire la page Web se présente comme suit :

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" >
  <xsl:output encoding="iso-8859-15" method="html" indent="yes"
    doctype-public="-//W3C//DTD HTML 4.01 Transitional//EN" />
  <xsl:template match="/changelog">
    <html>
    <head>
      <title>CVS-ChangeLog</title>
      <link rel="stylesheet" type="text/css" href="inetdoc.css" media="all" />
    </head>
    <body>
      <div id='container'>
        <h1 id='changelog'>Changelog - <xsl:value-of select="entry/date"/></h1>
        <xsl:for-each select="entry">
          <h4>
            <xsl:text>Fichiers édités par </xsl:text>
            <xsl:value-of select="concat(author, ': ', date, ' (', time, ')')"/>
          </h4>
          <pre>
            <xsl:value-of select="msg"/>
          </pre>
          <ul>
            <xsl:for-each select="file">
              <li>
                <strong><xsl:value-of select="name"/></strong>
                <br />
                <span class="rev">
                  <xsl:text>Revision : </xsl:text>
                  <xsl:value-of select="revision"/>
                  <xsl:text>, status : </xsl:text>
                  <xsl:value-of select="cvsstate"/>
                  <xsl:text>, lignes ajoutées : +</xsl:text>
                  <xsl:value-of select="linesadded"/>
                </span>
              </li>
            </xsl:for-each>
          </ul>
        </xsl:for-each>
      </div>
    </body>
  </template>
</xsl:stylesheet>
```

<sup>9</sup> <http://www.linux-france.org/~platu/README.html>

```

<xsl:text>, supprimées : -</xsl:text>
<xsl:value-of select="linesremoved"/>
</span>
</li>
                </xsl:for-each>
            </ul>
        </xsl:for-each>
</div>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

## 7.2. Génération de la page ChangeLog avec SVN

Depuis Novembre 2005, le gestionnaire de contrôle de version des fichiers sources utilisé pour le projet [inetdoc.LINUX] est subversion (AKA SVN). Ce gestionnaire présente de nombreux avantages relativement au précédent (CVS). Il est possible de générer directement le journal des mises à jour au format XML sans avoir recours à un paquet supplémentaire. Il est cependant toujours nécessaire d'utiliser une feuille de styles XSL pour produire la page Web au format XHTML.

Depuis Avril 2007, le journal des mises à jour au format XML est modifié à l'aide d'un script perl de façon à ajouter les statistiques produites par la commande **diffstat** pour chaque fichier modifié. L'objectif de cette modification est de fournir une information sur l'importance des éditions sur les fichiers sources.

Les règles de génération de la page Web contenues dans le Makefile de génération et de publication du journal des éditions de fichiers source (*Changelog*) sont :

```

.PHONY: $(TARGET_DIR)/README.html
$(TARGET_DIR)/README.html: common/asvn2cl.xsl
    svn log --xml --verbose \
    -r{`date --date 'tomorrow' +%Y-%m-%d`}:{`date --date '9 months ago' +%Y-%m-%d`} \
    >changes.xml
    ./common/extract_changelog_diffstat.pl <changes.xml >changes.xml.new
    xsltproc -o $@ common/asvn2cl.xsl changes.xml.new
    rm -f changes.xml*

```

Le script Perl de modification du journal XML effectue les transformations de balises suivantes :

- Résultat produit par la commande **svn log --xml**.

```

<paths>
  <path
    action="M">/trunk/prj/inetdoc/sources/sources.xml<path>
</paths>

```

- Résultat après transformation par le script `extract_changelog_diffstat.pl`.

```

<objects>
<object>
<path
  action="M">/trunk/prj/inetdoc/sources/sources.xml<path>
  <stats>14 ++++++-----<stats>
  <object>
</objects>

```

Le script Perl `extract_changelog_diffstat.pl` de modification du journal est donné dans l'*Annexe A, Script de transformation du journal XML de Subversion*.

La feuille de styles XSL `asvn2cl.xsl` utilisée pour produire la page Web *ChangeLog [inetdoc.LINUX]*<sup>10</sup> est donnée dans l'*Annexe B, Feuille de styles XSL de génération du Changelog à partir du journal XML de Subversion*.

<sup>10</sup> <http://www.linux-france.org/~platu/README.html>

## A. Script de transformation du journal XML de Subversion

Ce script Perl modifie les balises XML du journal produit par la commande `svn log --xml` en y ajoutant le résultat de la commande `diffstat`. Le nouveau jeu de balises est donné dans la [Section 7.2, « Génération de la page ChangeLog avec SVN »](#).

```
#!/usr/bin/perl -w
# $Id: extract_changelog_diffstat.pl 1174 2007-10-09 14:33:58Z latu $
# SVN XML log is edited to insert 'diffstat' result for every
# 'M'odified source file
#
# SVN XML log tag structure:
# <paths>
#   <path action="(A|D|M)"> source file path </path>
#   <path action="A"> source file path </path>
#   ...
# </paths>
#
# New tag structure:
# <objects>
#   <object>
#     <path action="M"> source file path </path>
#     <stats> diffstat information </stats>
#   </object>
#   <object>
#     <path action="D"> source file path </path>
#   </object>
#   ...
# </objects>
#
# Important !
# . This is definitely not the best Perl script you've ever read.
# . Calling svn diff for every single file is inefficient but
#   sorting of file entries is needed to call svn diff for multiple
#   files. Maybe for the next release.
#
# Copyright (c) 2007 Philippe Latu.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
# 1. Redistributions of source code must retain the above copyright
#   notice, this list of conditions and the following disclaimer.
# 2. Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in
#   the documentation and/or other materials provided with the
#   distribution.
# 3. The name of the author may not be used to endorse or promote
#   products derived from this software without specific prior
#   written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
# IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
# ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
# GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
# INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
# IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
# OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
# IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```

use strict;
use Symbol;
use POSIX;

my $infile=shift;

if (!defined $infile) {
    open(IN,"<STDIN") or die "Could not open an input stream";
}
else {
    open(IN,"<$infile") or die "Could not open an input stream";
}

my $filename = "";
my $s = "";
my $revision_delta = "";
my @pipe_text = ();

while (<IN>) {

    chomp;

    # Différence entre la version courante et la précédente
    if ( /revision=/ ) {
        $s = $_;
        $s =~ s/[^0-9]//g;
        $revision_delta = $s . ":" . --($s);
    }

    # Substitution balise paths -> objects
    if ( /paths/ ) {
        $_ =~ s/paths/objects/;
    }

    # Ajout balise object
    if ( /<path/ ) {
        $_ = "<object>\n" . $_;
    }

    # Traitement sur les fichiers modifiés uniquement
    if ( /action="\M"/ ) {
        # extraction nom de fichier
        $filename = $_;
        $filename =~ s/action="\M">\/trunk\/\///;
        $filename =~ s/<\/path>\/\///;
        $filename =~ s/^ *\/\///;
        # diff si le fichier existe et si ce n'est pas une image
        if ( -f $filename && $filename !~/(\gif|png|jpg)/ ) {
            # fork appels commandes
            my $pid = open CMD, "svn diff -r $revision_delta $filename | diffstat -p0 -w60 2>/dev/null";
            if (!defined $pid) {
                die "$0: can't fork svn: $!\n";
            }
            while ($s = <CMD>) {
                if ( $s !~/(\file|files) changed/ ) {
                    push(@pipe_text, $s);
                }
            }
            close CMD;
            # modification sur les propriétés de l'objet uniquement
            if ( (@pipe_text) == 0 ) {

```

```

    $_ .= "<stats>propriétés subversion</stats>";
  }
  else {
    # nouvelles balises pour les fichiers modifiés
    foreach $s ( @pipe_text ) {
      chomp ($s);
      my @parts = split /\|/, $s;
      $parts[0] =~ s/ //g;
      $_ = "  action=\"M\">\|trunk\|\" . $parts[0] . "</path>\n";
      $parts[1] =~ s/^ *//;
      $_ .= "<stats>" . $parts[1] . "</stats>";
    }
    @pipe_text = ();
  }
  $_ .= "\n</object>";
}
else {
  $_ .= "\n</object>";
}
}
elsif ( /action="(A|D)\|"/ ) {
  $_ .= "\n</object>";
}
# recopie de l'entrée
print $_ . "\n";
}

close(IN);

```

## B. Feuille de styles XSL de génération du Changelog à partir du journal XML de Subversion

Cette feuille de styles XSL est utilisée avec **xsltproc** pour transformer le journal XML de Subversion en une page Web XHTML. Elle prend en compte les balises XML insérées pour la publication des statistiques **diffstat**.

```

<?xml version='1.0'?>

<!--
  Another SVN 2 Changelog conversion

  asvn2cl.xsl - xslt stylesheet for converting svn log to a
               XHTML changelog

  Cette feuille de styles XSL est basée sur la génération de
  changelog GNU au format texte. Elle est adaptée pour produire
  une page Web au format XHTML. L'introduction ci-dessous reste
  parfaitement valable ;)).

  This file is based on several implementations of this conversion
  that I was not completely happy with and some other common
  xslt constructs found on the web.

  Copyright (C) 2005, 2007 Philippe Latu.
  Copyright (C) 2004 Arthur de Jong.

  Redistribution and use in source and binary forms, with or without
  modification, are permitted provided that the following conditions
  are met:
  1. Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.

```

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-->

```
<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/1999/xhtml">

  <xsl:output
    method="html"
    encoding="utf-8"
    standalone="yes"
    indent="yes" />

  <xsl:strip-space elements="*" />

  <!-- the prefix of pathnames to strip -->
  <xsl:param name="strip-prefix" select="'/trunk/'" />

  <!-- format date -->
  <xsl:template match="date">
    <xsl:variable name="date" select="normalize-space(.)" />
    <xsl:value-of select="substring($date,1,10)" />
    <xsl:text> </xsl:text>
    <xsl:value-of select="substring($date,12,5)" />
  </xsl:template>

  <!-- format author -->
  <xsl:template match="author">
    <xsl:value-of select="normalize-space(.)" />
  </xsl:template>

  <!-- format de la cellule stats -->
  <xsl:template match="stats">
    <xsl:if test="string(text())">
      <xsl:text> </xsl:text>
    </xsl:if>
    <xsl:value-of select="normalize-space(.)" />
  </xsl:template>

  <!-- extraction du nom de répertoire -->
  <xsl:template name="dirname">
    <xsl:param name="path" />
```

```

<xsl:if test="contains($path, '/')">
  <xsl:value-of select="concat(substring-before($path, '/'), '/')" />
  <xsl:call-template name="dirname">
    <xsl:with-param name="path"
      select="substring-after($path, '/')" />
  </xsl:call-template>
</xsl:if>
</xsl:template>

<xsl:template match="path">
  <td>
    <xsl:value-of select="@action"/>
  </td>
  <td>
    <xsl:choose>
      <xsl:when test="contains(text(), 'prj')">
        <xsl:choose>
          <!-- pas de lien pour une image -->
          <xsl:when test="contains(text(), 'image')">
            <xsl:value-of select="substring(text(),1+string-length($strip-prefix))" />
          </xsl:when>
          <!-- lien sur le répertoire de l'objet modifié -->
          <xsl:otherwise>
            <xsl:element name="a">
              <xsl:attribute name="href">
                <xsl:call-template name="dirname">
                  <xsl:with-param name="path"
                    select="substring(text(),string-length($strip-prefix))"/>
                </xsl:call-template>
              </xsl:attribute>
              <xsl:value-of select="substring(text(),1+string-length($strip-prefix))"/>
            </xsl:element>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="substring(text(),1+string-length($strip-prefix))" />
      </xsl:otherwise>
    </xsl:choose>
  </td>
</xsl:template>

<!-- format de l'objet courant -->
<xsl:template match="object">
  <tr>
    <xsl:apply-templates select="path"/>
    <td>
      <xsl:apply-templates select="stats"/>
    </td>
  </tr>
</xsl:template>

<xsl:template match="/log">
  <html>
  <head>
    <title>SVN ChangeLog [inetdoc.LINUX]</title>
  </head>
  <body>
    <div id='sitemast' style='margin: 0px; padding: 0px;'>
      <h1 style='font-size: 150%;'>SVN ChangeLog [inetdoc.LINUX]</h1>
    </div>
    <xsl:for-each select="logentry">

```

```
<h4>
  <!-- revision -->
  <xsl:text>Révision : </xsl:text>
  <xsl:value-of select="@revision"/>
  <xsl:text> / </xsl:text>
  <!-- date -->
  <xsl:text>Date : </xsl:text>
  <xsl:apply-templates select="date" />
  <xsl:text> / </xsl:text>
  <!-- author's name -->
  <xsl:text>Auteur : </xsl:text>
  <xsl:apply-templates select="author" />
</h4>
<div class='revhistory'>
<table>
  <tr>
    <th>Action</th>
    <th>Fichier</th>
    <th>Statistiques</th>
  </tr>
  <xsl:for-each select="objects">
    <xsl:apply-templates select="object"/>
  </xsl:for-each>
</table>
</div>
<pre>
  <xsl:value-of select="msg" />
</pre>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```