DB2LyX Install and Use

B. Guillon

14th October 2001

1 Introduction

The DB2LyX package provides a set of XSL stylesheets to translate an XML DocBook document into a LyX-1.1.6 or LyX-1.2.0 compliant file format. This paper is intended to explain how to install and use it, and is the first example of the translation process.

Besides, while the XML format is not fully supported by LyX, a script is provided by the package to translate easily SGML DocBook documents to XML DocBook documents.

2 Release

This paper is for release 0.1.5.

3 Content

The package contains the following:

***.xsl:** The DB2LyX XSL stylesheets.

format218: XSL stylesheets specific to the LyX-1.16 file format.

format220: XSL stylesheets specific to the LyX-1.2.0 file format.

template: Some templates used to build an entity file during the package configuration.

catresolve: Program to resolve any Public or System ID from the catalogs listed in the SGML_CATALOG_FILES variable. The program is linked to the Gnome xml2 library during the configuration step if xsltproc is available.

doc: Documentation.

script: Some scripts. In particular it contains sgml2xml.pl

4 Installing the package

4.1 Dependencies

The package needs the following features:

- the XML DocBook DTD,
- the Norman Walsh's DocBook XSL stylesheets,
- the MathML ISO entities,
- the DB2LyX stylesheets.

Moreover, you need to install XML/XSLT tools. There are two possibilities: using java tools or using the Gnome project XML/XSLT libraries. The java tools involve to have:

- Java,
- the Arbortext Catalog Classes,
- a java XSLT.

If you choose to use the Gnome XML/XSLT libraries you need to install:

- libxml2,
- libxslt (and its related program, xsltproc).

4.2 Install procedure (java support)

If nothing is installed, you can follow this default install procedure.

1. Install the XML DocBook DTD.
2. Install the Norman Walsh's DocBook XSL stylesheets.
3. Download and install the MathML entities.
4. Download the Arbortext Catalog Classes distribution. The Java catalog classes are used to resolve the file paths during configuration by using the SGML catalogs.
5. Install your favorite XSLT. XT/XP is a translator that should work correctly with the stylesheets. To install XT/XP look at the XT/XP install procedure.
6. Update the CLASSPATH variable. See the CLASSPATH section .
7. Update or create a new catalog file. See the catalog section .
8. Unpack the package in the same directory where LyX is installed. The package is expected to be at the same level than reLyX.

```
# cd /usr/local/share/lyx
# tar xvzf db2lyx-RELEASE.tar.gz
# mv db2lyx-RELEASE db2lyx
```

You then should have the tree:

```
/usr/local/share/lyx
                   |
                   +---- reLyX
                   +---- db2lyx
```

9. Run the config_xml script, from the lyx directory where the lyxrc.defaults and configure files are.

```
# db2lyx/config_xml {1.1.6|1.2.0}
```

4.2.1 Setting the CLASSPATH

The CLASSPATH environment variable must list the Java archives used by the package. In particular, it is expected to have:

- The catalog.jar and catalog-apps.jar paths, used during the configuration step.
- The java archives used by the XSLT, if the XSLT used is a java one.

4.3 Install procedure (Gnome XML/XSLT toolkit)

If nothing is installed, you can follow this default install procedure.

1. Install the XML DocBook DTD.
2. Install the Norman Walsh's DocBook XSL stylesheets.
3. Download and install the MathML entities.
4. Download and install the libxml2/libxslt distributions. It provides libraries and a default XSLT program (xsltproc) linked to them, that supports catalog resolution. The catalog support is used to resolve the file paths during configuration by using the SGML catalogs.
5. Update or create a new catalog file. See the catalog section .
6. Unpack the package in the same directory where LyX is installed. The package is expected to be at the same level than reLyX.

```
# cd /usr/local/share/lyx
# tar xvzf db2lyx-RELEASE.tar.gz
# mv db2lyx-RELEASE db2lyx
```

You then should have the tree:

```
/usr/local/share/lyx
                   |
                   +---- reLyX
                   +---- db2lyx
```

7. Run the config_xml script, from the lyx directory where the lyxrc.defaults and configure files are.

```
# db2lyx/config_xml {1.1.6|1.2.0}
```

4.4 Creating the catalog

4.4.1 Information to give

Configuration works by resolving some file paths through the catalog mechanisms. Thus, a catalog must be provided that defines some expected paths. Here are the public IDs that must be defined:

- The DTD DocBook XML to use. Several DTD versions can be defined here. The configuration script selects the most recent version. This information is used by the `sgml2lyx.pl` script provided by the DB2LyX package.
- `"-//DB2LyX//Norman Walsh's XSL Stylesheet Directory//EN"` . It says where the Norman Walsh's XSL stylesheets are. This information is needed by the DB2LyX package.
- `"-//DB2LyX//MathML ISO Entities Directory//EN"` . I says where the MathML Entities are. This information is needed by the DB2LyX package.

4.4.2 Catalog example

Here is a catalog example, that gives the expected information with several DocBook DTD versions.

```
PUBLIC "-//OASIS//DTD DocBook XML V4.0//EN"
  "file:///usr/local/share/xml/dtd/docbook/docbkx40/docbookx.dtd"

PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
  "file:///usr/local/share/xml/dtd/docbook/docbkx412/docbookx.dtd"

PUBLIC "-//DB2LyX//Norman Walsh's XSL Stylesheet Directory//EN"
  "file:///usr/local/share/xml/stylesheet/docbook"

PUBLIC "-//DB2LyX//MathML ISO Entities Directory//EN"
  "file:///home/local/share/xml/mmlents"
```

4.4.3 Catalog location

Your new catalog path must be added to the SGML_CATALOG_FILES variable. Personnaly I suggest to put the catalog under the personnal .lyx directory, i.e. its full path would be `$HOME/.lyx/db2lyx/catalog` .

And the SGML_CATALOG_FILES setting would look like this:

```
export SGML_CATALOG_FILES=$SGML_CATALOG_FILES:$HOME/.lyx/db2lyx/catalog
```

5 Installing XT/XP

XT is the translator used to test the stylesheets, and so it should work correctly on your platform. XT is a set of java libraries, and it needs an XML parser such as XP. Here is an install procedure example:

1. First, check that java is installed.
2. Download the XT distribution.
3. Download an XML parser that supports SAX, let's say XP.
4. Unzip de XT/XP packages:

```
> mkdir /usr/local/xt /usr/local/xp
> unzip xt.zip -d /usr/local/xt
> unzip xp.zip -d /usr/local/xp
```

5. Put the XT libraries (xt.jar and sax.jar) and the XP library (xp.jar) in your CLASSPATH:

```
> CLASSPATH=$CLASSPATH:/usr/local/xt/xt.jar:/usr/local/xt/sax.jar
> CLASSPATH=$CLASSPATH:/usr/local/xp/xp.jar
> export CLASSPATH
```

6 What is the configuration process?

The `config_xml` script does the following:

- If it is not already done, it patches the LyX configure file so that it will call config_xml during the next reconfiguration.
- It looks for catalog resolvers. To find the java catalog classes, it checks that catalog.jar and catalog-apps.jar are listed in the CLASSPATH variable. The other catalog resolver tested comes with libxml2, available if xsltproc is found. If xsltproc is found, the catresolve program is build.

- If no catalog resolver is found the script exits because no configuration can be done.
- It looks for a default user catalog. If it is not defined, configuration can be ok, provided that the actual catalog to use is defined in the SGML_CATALOG_FILES variable.
- It detects the available XML DocBook DTDs, from the most recent version (4.1.2) to the last recent one. It stops once a DTD is found, and then prepare the DocBook->XML converter.
- It detects the available XSLTs. The java XSLTs are detected by scanning the CLASSPATH variable. xsltproc is detected by looking up the PATH variable.
- If several XSLTs are available, it selects the first one found. If no XSLT is found, the XML->LyX converter will not be available.
- It sets the LyX format to use.
- If an XSLT is found, it tries to resolve the needed Public IDs and, if it is a success, it configures the DB2LyX stylesheets and prepare the XML->LyX converter.
- It adds the converters and the XML format to the `lyxrc.defaults` file.

Here is an example of the console ouput of the XML configuration:

```
+checking for catalog.jar... yes
+checking for /home/ben/.lyx-1.1.6/db2lyx/catalog... yes
+checking for xsltproc... yes
make: Entering directory ‘catresolve’
make: Nothing to be done for ‘all’.
make: Leaving directory ‘catresolve’
+checking for catalog.jar... yes
+checking for resolvers... (gnome_resolve java_resolve)
+checking for DTD DocBook XML V4.1.2... no
+checking for DTD DocBook XML V4.0... yes
+checking for XSLT... (xsltproc xt.jar xalan.jar)
LyX format: format220
xsltproc --catalogs \
--params base-dbcommon \"/usr/lib/xml/docbook/common\" \
--params base-mmlents \"/usr/lib/xml/docbook/mmlents\" \
/usr/local/share/lyx/db2lyx/template/xslpath.xml \
/usr/local/share/lyx/db2lyx/template/xslpath.xsl > \
/usr/local/share/lyx/db2lyx/xslpath.ent
```

7 Using the XML support in LyX

Once the XML configuration is done, you can check that XML support is now available in LyX. You should see the following things added in the menu:

- ”DocBook (XML)” must be available in the File->Import menu list,
- ”DocBook (XML)” must be available in the File->Export as menu list,

More precisely, looking at the Edit->Preferences->Formats/Converters menu should give the following information:

1. The XML format is available, defined as:
   - Format: xml
   - GUI name: DocBook (XML)
   - Extension: xml
2. The XML -> LyX converter is added, defined as follow:
   - From: XML
   - To: LyX
   - Converter: `java com.jclark.xsl.sax.Driver $$i (path)/db2lyx/format220/docbook.xsl > $$o`
3. The DocBook -> XML converter is added, defined as follow:
   - From: DocBook
   - To: XML
   - Converter: `(path)/db2lyx/script/sgml2xml.pl <XML DocBook DTD path> $$i`

If everything is ok, you can directly import an XML file through the menu File->Import->DocBook (XML). Of course, instead of the DB2LyX docbook stylesheet you can define the converter command such that it points to a customized stylesheet.

Besides, you can also export SGML DocBook file to XML through the menu File->Export as->DocBook (XML).

8 Using the stylesheets out of LyX

The package can be used out of LyX. The main stylesheet to use is `docbook.xsl` provided by the package. For instance, to translate the file `mydoc.xml` to `mydoc.lyx` with XT run the command:

```
java com.jclark.xsl.sax.Driver mydoc.xml (path)/db2lyx/docbook.xsl > mydoc.lyx
```

9 Customizing the stylesheets

Some XSL variables can be set by the user to customize the XSLT output. Redefining some variables needs to define your own stylesheet. Read the Norman Walsh documentation if you want to have more details about extending the stylesheets.

9.1 XSL variables

The following variables are defined to customize the output, detailed in the Parameters and Variables reference. These variables are related to some LyX parameters:

Name	Possible values	Default value
lyx.doc.font	One of the Layout->Document->Font list	default
lyx.doc.separation	skip or indent	skip
lyx.lang.lang	One from the Layout->Document->Lang->Language choice	english
lyx.lang.encoding	One from the Layout->Document->Lang->Encoding choice	auto
num.section.depth	1 <= number < 5	4
toc.section.depth	1 <= number < 5	4

9.2 Extending the stylesheets

You can extend the stylesheets by defining your own stylesheet to use. To modify the XSL variable or overload an existing template, or add some new behaviour, you need to write your personal stylesheet. Here is a simple example:

```
<?xml version='1.0'?>
<!-- mystyle.xsl -->

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

<xsl:import href="/usr/lib/sgml/xsl/docbook/lyx/docbook.xsl"/>
<xsl:variable name="lyx.doc.font">default</xsl:variable>
<xsl:variable name="lyx.doc.separation">skip</xsl:variable>
<xsl:variable name="lyx.lang.lang">spanish</xsl:variable>
<xsl:variable name="lyx.lang.encoding">default</xsl:variable>
<xsl:variable name="toc.section.depth">5</xsl:variable>
</xsl:stylesheet>
```

Once defined, run the XSL translator on this stylesheet:

```
java com.jclark.xsl.sax.Driver mydoc.xml mystyle.xsl > mydoc.xml
```

10 Building this document

You can test your configuration and the stylesheets by building the document you are reading now (manual.lyx). You can either build only the output LyX file from manual.xml, or building the whole HTML document including the stylesheets references. The subsections consider that the current directory is `docbook/lyx/doc` .

10.1 Documentation structure

The documentation directories are the following:

html: contains the output HTML files.

outlyx: contains the output LyX file made by using the package stylesheets on the XML source file.

references: contains the XML reference files, made from the package stylesheets.

src: main document source files (manual.lyx and manual.xml).

xsl: some stylesheets used to produce the XML reference files and HTML files.

10.2 Building the LyX file

10.2.1 Content

The files involved in building the ouput LyX file are:

src/manual.lyx: the original LyX file.

src/manual.xml: the XML document from the LyX one.

outlyx/Makefile: a very simple makefile to make manual.out.lyx from manual.xml. It expects that the XSL translator to use is XT.

outlyx/manual-lyx.xsl: the stylesheet used to create the LyX document. It points to the lyx/docbook.xsl stylesheets.

outlyx/manual.out.lyx: the output LyX file produced once make is done.

10.2.2 Changing the DTD path

To build the document, you need to set the SYSTEM path to the XML DocBook DTD location of your system, at the top of the `manual.xml` file:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE article SYSTEM "file:///[your path here]/docbookx.dtd">
```

10.2.3 Creating the document

If everything is ok:

```
cd outlyx
make
```

That’s it!

10.3 Building the HTML files

The document is composed of:

1. The HTML manual file. This file depends on src/manual.xml.
2. The HTML references files. These files depend on the XML files under the directory `references` . The XML reference files depend on the XSL stylesheets of the package, since the reference documentation is included in the stylesheets. The XML files production uses the stylesheets of the directory `xsl` . Building the HTML files from the XML files need the jrefentry DTD in the `references/dtd` directory.

To build the HTML document you can simply use the the upper level Makefile (under docbook/lyx/doc). Its operations are:

1. Reference XML files production (under references)
2. HTML files production (under html)
3. LyX file from the DocBook XML file (under outlyx). This step has no relationship with the preceding steps. It is done because the Makefile tries to update all the document files.

11 MathML support

A basic MathML support is provided. Only the most common presentation elements are translated.

12 References

The references are generated automatically from the DB2LyX stylesheets. The following reference entries are available:

- DB2LyX Parameters and Variables
- Unsupported DocBook Elements

13 Limitations – Bugs

13.1 Loss of information

When you write a LyX document, export it to XML and use the stylesheets on this XML output to have a LyX document again, some information from the original LyX file can be lost. The reason is that one can define more information in a LyX document than in an XML one. Here are some information losses examples:

- A
  exported to XML becomes a simple space. So the reverse process cannot translate it into a newline again.

- In XML documents exported by LyX, figures do not contain their size, nor their scale. This information cannot be set when translating back to the LyX format.
- SGML Style is written as is when LyX exports to XML. The reverse translation cannot know that it was written in an SGML layout paragraph.
- The table cell border lines information defined in LyX does not appear in the XML output file. So the reverse translation looses these characteristics.

13.2 Bugs, problems

Here are some known bugs. The bug list is not intended to be complete:

- The file produced is not clean: it contains sometimes too many empty lines, and sometimes some empty lines should be added.
- Some translations are understood by LyX but it is not really what LyX would do.
- Problems with quotes and the backslash character.

13.3 Features not yet supported

- Preamble information is not translated. But how to translate it?
- Many XML tags are not translated.
- Many XML elements cannot be translated to something known by LyX. Some of these elements are traced in the Unsupported DocBook Elements reference.

14 Changes

Differences between the current release and version 0.1.4:

- Possibility to support two LyX formats:
  - Format 218 (LyX 1.1.6),
  - Format 220 (LyX 1.2.0).
- Bug fixes.

Differences between release 0.1.4 release and release 0.1.3:

- Possibility to select the XSLT used by the Makefiles, by setting the variable named XSLT. The variable can take three values: ’xt’, ’xsltproc’ or ’xalan’. By default it is set to ’xsltproc’.
- Changes in the reference building stylesheets so that it works with xsltproc.

Differences between release 0.1.3 and release 0.1.2:

- Support for using xsltproc (catalog resolution program catresolve + configuration step).

- XML Configuration script added.
- SGML to XML translation script added.
- Manual change (new install procedure).
- Bug fixes.

Differences between release 0.1.1 and release 0.1.2:

- MathML support improved.
- Paths of MathML entities and Norman Walsh stylesheets are not hard coded anymore in the stylesheets.
- Some References are added, build from the stylesheets.
- Manual production improvement (more targets).
- Bug fixes.

Differences between release 0.1.1 and release 0.1.0:

- More elements are supported.
- Deep cleanup (from the original package).

15 Thanks

Thanks to Ramon Casellas who kindly accepted that I use his DB2LaTeX stylesheets as a basis for this package. And thanks to Jose Abilio Oliveira Matos for his support and his patience!

16 Feedback

Please give your comments! This is the only way to efficiently improve the package and fix the bugs. To report anything, you can send me an email.