

Cyberdocs Documentation

Université Lyon 2 - 2009

Table of contents

1. Introduction.....	3
2. Community.....	4
3. Documentation.....	4
3.1. Install help.....	4
3.1.1. Hardware pre-requisites.....	4
3.1.2. Software pre-requisites.....	5
3.1.3. Underlying components.....	6
3.1.4. Downloading Cyberdocs.....	7
3.1.5. Step by step installation	7
3.1.5.1. The pcd.properties properties file.....	8
3.1.5.2. Compilation.....	11
3.1.5.3. Modules installations.....	11
3.1.5.3.1. Installing conversion module.....	11
3.1.5.3.2. Installing management module.....	12
3.1.5.3.3. Installing publishing module.....	13
3.1.6. Customize Cyberdocs.....	15
3.1.6.1. Create a new institution.....	15
3.1.6.2. Add a new styles code.....	16
3.1.6.3. Other language Conversion	16
3.1.6.4. CSS style sheets.....	16
3.1.6.5. Publishing other documents.....	17
3.1.6.6. Modifying static pages.....	17
3.1.6.7. Modifying the Web application.....	18
3.1.6.8. Customize the management module.....	18
3.1.6.9. Link Apache & Tomcat: install JK module.....	18
3.1.6.10. Use a MySQL database.....	20
3.1.6.11. Integrate the publishing module to your Web site.....	22
3.1.7. Tools.....	22
3.1.7.1. Re-indexing all your documents.....	22
3.1.7.2. Modify your metadata in mass.....	24
3.2. Help to users.....	24
3.2.1. Technological approach.....	24
3.2.2. Convert a document via the management module.....	25
3.2.2.1. Identification under Cyberdocs.....	25
3.2.2.2. Create a space for a thesis.....	26
3.2.2.3. Conversion procedure.....	27
3.2.2.3.1. Styled files and associated files transfer (Files Management tab).....	27
3.2.2.3.2. Metadata input (Metadata tab).....	28
3.2.2.3.3. Conversion (Treatments tab).....	30
3.2.2.4. Verification of the conversion result.....	35
3.2.3. Convert a document without the management module: in command lines	38
3.2.3.1. Operations and conversion steps.....	38
3.2.3.2. Preparing a working space.....	39
3.2.3.3. Prepare a document.....	41
3.2.3.4. Doing a conversion.....	43
3.2.3.5. Conversion result.....	44
3.2.4. Publishing module.....	45
3.2.4.1. Management and users identification.....	45

3.2.4.2. Documents management.....	46
3.2.4.3. Document's metadata use.....	48
3.2.4.4. Consulting a document.....	50
3.2.4.5. Search.....	51
3.2.4.6. Search results.....	52
3.2.4.7. Printable versions.....	52
3.3. Help for programmers.....	52
3.3.1. Bugs reporting.....	53
3.3.2. Contributions to Cyberdocs.....	53
3.4. Downloading Cyberdocs.....	53
4. Downloading.....	53
4.1. Cyberdocs' distributions.....	53
4.2. Getting back the sources with Subversion.....	53
5. Lists.....	55
6. FAQ.....	56
7. Contacts.....	56

1. Introduction

The Cyberdocs platform provides tools, but also a *model*, allowing an institution to **publish scientific literature**, such as thesis, reports, journals, and so on.

Two main principles are associated to the platform:

1. Free software: Cyberdocs is a free platform ([GPL](#)) and uses it free software. This approach allows everyone to use it without coercion, to improve it, to contribute to its development, to share its experience, to translate it, the document, and so on. It is a real project of cooperation between different partners, financiers, developers, users.
2. The structured documents: Cyberdocs's publishing model is based on XML standard structured documents, which has several advantages, including interesting possibilities for consultation, perennial archiving, simple exchanges, etc.

Publication model

The Cyberdocs publishing model is to produce an XML (DTD TEILite) structured document format, which will become the reference document that will be used for archiving and publication.

The aim is to obtain this reference document with the least effort possible, starting from the documents as produced by the authors, most often with a word processor.

Publication process

The publication process of the Cyberdocs chain is composed by those steps:

1. Documents preparation

An operator (it could be the author, a librarian, etc.) will use a word processing document and prepare it adequately. This preparation is to use certain styles to identify parts of the content, ensure that the styles *Title 1*, *Part 2* are used for the table of contents, prepare line drawings for pixelisation, etc. But increasingly, the work has been carried out directly by the author that uses models of documents proposed by the word processor or the scientific institution in which he works.

Example: the [Cyberthèses](#) model document.

2. Metadata preparation

Metadata associated to the document have to be prepared, which consist in the creation of an XML document containing those metadata.

3. XML conversion

The **conversion feature** of the Cyberdocs platform can convert word processing document beforehand prepared in an XML document respecting the DTD TEILite. The **management module** of the Cyberdocs platform can fly most of these operations through a Web interface very easy to use.

4. File production for static publishing

If necessary HTML, XHTML and PDF versions are produced (from the XML format) in order to allow a static diffusion of the documents, on a network or on a CD-ROM, for example.

5. Publication

The **publication module** makes available documents, with an easy consultation and exploiting the documents structure, as well as literature searches accurate within a document or a collection of documents.

2. Community

Cyberdocs was made possible by the support of the French-speaking Intergovernmental Agency and its program Cyberthèses, through the " Fonds Francophone des Inforoutes ".

This conversion platform, placed under the GPL, was conducted with the assistance of the AJLSM Company, and is broadcast in Africa, South America, Switzerland and France.

Today, the platform is primarily maintained through the [Cyberthèses](#) project, whose major actors include [University Lumière of Lyon 2](#), the [University of Santiago](#), [Peru universities](#) and [Silecs](#) society.

3. Documentation

3.1. *Install help*

Cyberdocs platform allows diffusing digital documents in an Internet environment. It offers conversion and publishing services of those digital documents.

Different publics are concerned by the platform and therefore by this documentation. We notably distinguish:

- **Users**, for whom the publishing module, with its display and search functionalities constitute the most important element.
- **Person in charge** of the documents preparation and publishing, who will fly the conversion and then the publishing of the digital documents.
- Implementers, as to say, people who will lead up to install, deploy, exploit a Cyberdocs platform, but also to personalize or even adapt it.
- **Programmers**, who will add on new functionalities to the platform or correct the errors.

The Cyberdocs platform is split in three important modules:

- **Conversion module**: allows transforming word processing document into XML format according to the DTD TELLite.
- **Management module**: allows flying, via a web interface, the documents conversions and publishing.
- **Publishing module**: allows diffusing, in dynamic web architecture, documents with search and consultation functions.

The installation of the **management module** requires at least the installation of the **conversion module**. But you do not have to use the management module to run the conversion module in commands line.

Regarding the publishing and conversion modules, they are **independents**.

3.1.1. **Hardware pre-requisites**

The information provided on this page concern first and foremost all the tools of the platform **servers**, that is to say what is necessary to carry out the conversions, install the search engine, and so on. The **user's stations** only need a recent standard Web browser that supports HTML, CSS and JavaScript, such as Internet Explorer, Mozilla, Opera, and so on.

The hardware can be quite varied, its precise characteristics depend largely on the workload you plan and the modules you want to install.

The **disk space** required will be about 200 MB for a full installation, including OpenOffice.org, PHP, Java and SDX. The Cyberdocs tools themselves take relatively little space, about 20 Mo.

Of course, the disk space to store the documents converted is not counted here, and in this matters, there must be about 10 times the original document size, even though this figure may vary depending on the number of external resources such as the images. This space is necessary because the conversion tools produce a lot of documents:

- An XML / TEI lite version of the document,
- Two XHTML static version of the document: one for the static consultation outside any context, and an other one for the printing function in the dynamic web application,
- Two HTML static version of the document: one for the static consultation outside any context, and one for the printing function in the dynamic web application, these HTML versions are mainly destined for the Internet Explorer browser.
- A PDF version of the document, with one PDF file for the complete document and one file per document parts.
- Three intermediate XML version of the document, produced while executing the conversion tools.

Only the reference XML document in TEI Lite format is really necessary, all others are derivatives products that can be preserved or not; to note that it is even possible to do not generate them by the tools. If you do not keep all documents, disk space required decreases as well.

We should also mention that if you use the dynamic Web environment based on SDX to disseminate the documents, you have to count approximately 2.5 times the size of the documents in their TEI Lite in terms of disk space. Indeed, SDX should keep a copy of the complete document, as well as each party. The size of indexes for research varies, but it can be estimated at approximately 50% of the size of the documents in XML format TEI Lite.

The amount of **random access memory (RAM)** on the server is a critical factor. The conversion module uses memory in a proportional manner to the size of the documents that must be processed, as well as the number of images included in these documents. A simple rule to assess the need for memory is to provide 50MB initially, and add 1.25 MB per document page. Thus, for a 500 pages document, it may be necessary to have nearly 700MB of memory for processing, although very often a lot less will be sufficient.

On the other hand, the memory used by the publishing module depends on the number of concurrent users that you want to serve. For an application such as Cyberdocs, a memory of 2GB to 4GB will be generally comfortable.

The power of the **computer processing unit (CPU)** is not a critical factor, given that the more the processor is speedy, the more quickly the conversion is, but also research and display in the publishing module.

3.1.2. Software pre-requisites

The various modules of the Cyberdocs platform were designed to be as portable as possible. However, they are using different technologies, from where some variation in the specification of systems where you can install each module.

It is important to mention that in this page and all documentation in general, the name refers to Windows Server, XP or Vista versions. In particular, it excludes 95, 98 or Millennium versions; some tools can be installed on them, but these operating systems are not designed for this kind of operation and it can be difficult to extricate. One thing is for sure, the procedures are not documented here.

1) Conversion module

The conversion module called upon different technologies:

- Java to fly conversions which are, for most of them, done through the XSLT language.

2) Management module

The management module is a Web-based interface developed with PHP 5. It can therefore be installed on any platform that supports such an environment, which opens a lot of possibilities. Note an excellent PHP support on the Linux and UNIX systems, but also on Windows.

3) Publishing module

The publishing module is a dynamic Web application developed using the SDX platform, itself built on the Cocoon infrastructure. It is a sort of dynamic Web application according to the API Java Servlets.

To host it, you should have a system that can accommodate a Java virtual machine, and such a virtual machine exists for all recent operating systems, from Windows to Mac OSX through many Linux and UNIX.

3.1.3. Underlying components

Depending on the Cyberdocs's modules to install, you must plan the installation of the underlying components, needed to use the Cyberdocs's functions. These components are: Java, a Web server, PHP, a servlet engine and SDX.

1) Java

A Java Virtual Machine is required for conversion modules and publishing. It is even necessary to proceed with the installation of these modules, which is why it is usually the first installation to be done. Cyberdocs's modules work with Java 1.4, but if you perform a new installation, we strongly advise you to use the latest version of Java, 1.6.

Below you will find links to all Java's distributions for the most popular platforms. Note that you should use a **Java Development Kit (JDK)**, and not a **Java Runtime Environment (JRE)**.

[Sun's distribution](#) (for Windows, Linux and Solaris)

[HP-UX](#) (UNIX for Hewlett-Packard stations)

[IBM](#) (for AIX, Windows, Linux, OS/2, z/OS)

[Apple](#) (for Mac OS X)

These distributions are generally very easy to install, simply follow the instructions or consult their documentation. The installation of Java is expected to take about 40MB.

Important

Once Java installed, you must manually define an environment variable named **JAVA_HOME**, and give it the value of the path where you installed Java. On UNIX and Linux, you can set this value in one of the startup scripts located in the home directory of the user used by Java (eg in the file `.bashrc`). On Windows, you can do this by opening the **System** control panel and selecting the Advanced tab, where you will find a button that allows you to set environment variables.

2) Web Server

A Web server is required to use the publishing and management modules of Cyberdocs. Note that if you only want to use the publishing module for demonstrations or tests, you can avoid this facility and move directly to the installation of servlet engine.

But for a production site, or to install the management module, we strongly encourage you to install the [Apache web server](#). On some systems (Linux, UNIX) that server is probably already installed otherwise go to the [download site](#) and take the version that suits your environment. The Apache [documentation](#) is very complete, including a section on the [compilation and installation](#), but also on the [configuration](#).

If you want to use the management module, the Rewrite module (or equivalent) must be installed and enabled for your web server.

3) PHP

The PHP environment's script must be installed to use the management module of Cyberdocs. PHP is based on a web server, usually Apache, which must also be installed. It should be noted that very often the PHP support is already provided by Apache, and that version 5.x PHP is required.

The [main site of PHP](#) will provide you with all the necessary information about this tool. There is a specific page for [downloads](#).

Important:

As a management process question, the management module requires that PHP is installed as a module on the web server and not as a CGI ([Common Gateway Interface](#)). Refer to the [PHP installation documentation](#).

4) Servlets engine

The servlet engine is an execution environment dynamic websites written in Java. The publishing module needs such an environment to operate. [Tomcat](#) of the Apache foundation is the servlet engine more widely used. You can download and install it easily, if you have previously installed a Java virtual machine. Tomcat includes its own Web server, which is generally sufficient for tests or demonstrations and prevent the installation of the web server. If you want to use Tomcat with the Apache Web server, you must install and configure the module JK to make the link between the two tools. To simplify installation, we offer a quick installation guide for Linux.

5) SDX

The SDX platform is a research and consultation tool of XML documents. It is used as a basic tool for the publishing module of Cyberdocs. **The version 2.3 or higher is required.** You can [download SDX](#) and its sources, but if you just want to install quickly we suggest using the file sdx.war and place it in the dossier of webapps of the servlets engine. Once your engine restarted, the WAR file will be unpacked and SDX will be available.

3.1.4. Downloading Cyberdocs

Find all the instructions on the page "downloads".

3.1.5. Step by step installation

When [downloading](#) Cyberdocs platform, it is necessary to perform a number of operations before it can be used; in this documentation, we will call these operations the **installation process** of the platform.

It is important to realize that the installation process affects only the specific operations to the Cyberdocs platform. To install the underlying components (Web server, PHP, servlets engine, SDX, Java), you must follow the instructions provided with these tools. On the [download page of the components](#), you will find links to the tools Web sites which will help you achieve your installation.

The installation process consists essentially of two types of operations: editing files properties and execution of different scripts, depending on the modules to be installed.

In order to not forget anything, respect the following order:

- Modification of the pcd-properties file
- Compilation
- Modules installation

3.1.5.1. The pcd.properties properties file

In the Cyberdocs's root folder sources is a file named pcd.properties. This file contains some information that helps determine the installation of the platform. It's important to understand these parameters and their appropriate values for your environment.

The format of this file is the standard properties of the Java environment. It is a text file, and to be sure it is correctly interpreted in all Java environments, it contains only the ASCII characters. The lines that begin with the character # are comments, so they are ignored, you can add if you find it necessary. Ditto for the white lines, they have no meaning except that make the file more readable. Finally, the significant lines are those where we find the name of a property, followed by a = character, followed by its value.

It is important to note that you can define these properties other than in the pcd.properties file. In order to facilitate upgrades, it is advisable not to touch this file, but rather place the properties in either of these files:

- A file called pcd.properties in the installation user's "home" folder.
- A file called .ant.properties in the installation user's "home" folder.
- A file called .ant.properties in the Cyberdocs's sources root folder which will be used to the installation.

These properties files are read in this order, and the first time a property is defined, it retains its value until the end of the process. **The last file read is pcd.properties located in the root directory of Cyberdocs's sources, properties and values found there are therefore defaults.**

After installation, the file pcd.properties is located at the following location: tools/bin/pcd.properties

1. The conversion tools installation folder

The property "dossier.installation.up" can define the location of the installation directory conversion module. Note that the directory is also the working directory where documents are translated.

The value of this property is the path which can be either absolute (for example: /usr/local/cyberdocs/pcd-install or C:/programs/cyberdocs/pcd-install) or relative (eg .././pcd-install). If it is relative, the resolution of the path is relative to the Cyberdocs's root sources. Please also note that even under Windows, it is advisable to use the character / to separate directories in a path, otherwise you must double separators: \ \.

The default value is ../pcd-install, the working directory will be the folder pcd-install located in the hierarchy of your workstation at the same level as the directory containing all sources of the platform.

If you do not plan to use the conversion module nor the management module of Cyberdocs, this property is of course useless.

2. OpenOffice installation folder and version

Important note: opposite to earlier versions of Cyberdocs, OpenOffice isn't used to convert DOC files, and this for stability reasons.

You can ignore this property.

3. Default used style during the conversion

This is the styles identifier used by default during the conversion. The list of styles and codes are in the file: src/oo-vers-tei/xslt/utiles/styles.xml. At the time of installation, this file is avail-

able here: tools/xslt/utiles/styles.xml.

Refer to the "list of supported styles by Cyberdocs" for more information.

The default value "lyon2" is functional for testing.

4. Publishing module path under SDX

The publishing module of Cyberdocs uses SDX as a dynamic web site infrastructure, and once installed, the module becomes an *SDX application*. Each SDX application must be installed in a folder immediately into the installation of SDX itself.

The property `sdx.application.path` indicates the name of this file. By default, this is the `pcd` value that is used.

This property affects only those installations where it is planned to use the publishing module.

5. Complementary Information for SDX

For `sdx.logicsheet.path`, `cocoon.class.cinclud` and `cocoon.class.wildcard` properties, see the comments of the `pcd.properties` file provided by default, because these properties will evolve very quickly.

6. Skin of interface consultation

The module publication can have different skins for documents consultation on the Web. The skin property lets you specify which to choose. The value of this property must match the name of a file that is in the file `src/web/skins` from the root of the sources used in the installation.

In the normal distribution of Cyberdocs, only dressing `pcd` is proposed. If you download the source code, you will also have the opportunity to use *cybertheses* skin, but it should be reserved for a specific use in the *Cyberthèses* project.

This property affects only those installations where it is planned to use the publishing module.

7. L'identifiant du module de publication sous SDX

The SDX platform offers a range of features, some of which may be distributed on different servers or different applications. Therefore, all applications under SDX must have an ID, and the person in charge of the applications must ensure that this identifier is unique among all existing SDX applications.

A good way to achieve this, is to use the principle of *domain names reversed*. For example, if you are considering installing the publishing module to an address like `http://www.monsite.org/sdx/pcd/`, you are normally responsible for the domain name `monsite.org`. In this case, you should choose an identifier that begins by `org.monsite` and is supplemented from different parts separated by a "." and you are responsible to ensure that this supplement is unique among SDX applications, which are under the responsibility of *monsite.org*.

The `application.sdx` property identifies has this value, and it is **necessary to change it** in order to avoid that everyone use the same value.

This property affects only those installations where it is planned to use the publishing module.

8. SDX installation folder

To install the publishing module, it is necessary to know the directory where the SDX platform is installed. The property `dossier.installation.consultation` should contain the name of this folder.

This property affects only those installations where it is planned to use the publishing module.

9. SDX server address

Specify here the URL to access SDX. For example: <http://www.monserveur.fr:8080/sdx>.

The port 8080 is necessary only if you use Tomcat in its default configuration. For a use in Cyberdocs's production, we suggest using [JK module](#) for linking Tomcat to Apache.

This property will serve as a basis for the URL of the documents available via the publishing module. Change this value by the external address of your server, do not let "localhost" except for use as part of an experiment.

This property affects only those installations where it is planned to use the publishing module.

10. Consultation application opening

In an SDX installation, we can recover a folder containing the source of an application without that this one is really visible publicly. For it to be, it must be *opened* in SDX, which means that it must be ready to use.

The installation of the Cyberdocs platform can automatically perform this opening, if the `sdx.application.open` property has a value of *1*, which is the default that we advise you to keep.

This property affects only those installations where it is planned to use the publishing module.

11. Conversion errors identification

When the conversion module handles a word processing document, it may show some errors (missing style, etc.) proper to a certain context of use. For now, the only context in which such errors are defined is the project Cyberthèses. These errors are indicated by specific attributes in the XML document format TEILite, which will not be valid and should be modified.

To enable the errors identification, should be given the value *1* to `erreurs.cyberdocs` parameter. This feature is disabled for any other value.

12. OAI parameters

The Cyberdocs platform makes available metadata associated with the documents via the [OAI-PMH](#). To make this feature work, you must have version 2.2 or higher of SDX. To enable this feature, you need to change three properties in the `pcd.properties` file:

- `entrepot.oai`: give the value *true* to enable OAI.
- `entrepot.oai.nom`: the name of your OAI warehouse, required by the protocol.
- `admin.oai.email`: the e-mail address of the administrator of this OAI warehouse, required by the protocol.

13. Binary PHP location

This property affects only those installations where it is planned to use the management module.

This is the directory where is the binary PHP (command line interface). The path must be absolute.

For example: `/usr/local/bin`.

If you do not know where to find the file, help yourself with the command: `which php`

If PHP is active and that you do not find the executable php, try installing "php-cli" ("cli" for

Command Line Interface) with your usual driver installation.

For example, Debian or Ubuntu: apt-get install php-cli.

Warning:

- Ensure that the binary bears the name php or php.exe within the specified directory. Create a symbolic link in the opposite case.
- Your version of php has to be greater than or equal to 4.3 (version 5 recommended). You can check by using the command: php -- version.

14.And after?

Check one last time your properties, in particular that there is no space in the access paths, etc....

Your configuration file is now ready: you can move on to the next steps (compilation and installation of modules).

3.1.5.2. **Compilation**

Warning:

This is required only if you install from sources obtained from the CVS Cyberdocs's warehouse. If you are installing from the distribution, this operation has already been done for you. It may be necessary if you have a version of Java prior to 1.6 (in this case delete the "build" directory in the root sources).

Several Java classes were written for the conversion module, and they are necessary to implement the tools. Thus their compilation is the first thing to do when installing Cyberdocs.

To do this, simply execute the script build-ant.bat or build-ant.sh depending on the operating system you use.

If the compilation returns an error that you do not understand, check your pcd.properties file and see the FAQ.

If the compilation completes successfully, you can now proceed to the modules installation.

3.1.5.3. **Modules installations**

Important: before you install modules, we remind you that it is necessary to have read and respected the hardware and software "pre-requisite", as well as the three previous stages of this installation manual: download, configuration of the pcd.properties file and compilation.

The complete installation of Cyberdocs takes place in three stages:

- Installing the conversion module.
- Installing the management module.
- Installing the publishing module.

The installation of the **management module** requires at least installing the **conversion module**. But you do not have to use the management module to run the conversion module from the command line.

Regarding the publishing and conversion modules, they are independent

3.1.5.3.1. **Installing conversion module**

This step will result in the production of the conversion module folder, containing all the sources needed for the documents conversion and examples. You can do this simply by running the installation script *installation-oo2xml.bat* or *installation-oo2xml.sh*. At the same time, the conversion tools, as well as files in the management module, will be copied.

This script will create a folder that will be the basic folder for conversion and management modules. The location of this folder is specified in the pcd.properties file. In this folder, you will find three subdirectories:

- Tools: contains the tools of the conversion module, as to say, XSLT processors or XSL-FO, XSLT transformation, the environment ANT and its guidelines file, and so on.
- php: contains the PHP scripts of the management module and certain binaries for the call to conversion module.
- Production is the basic directory for the documents production using the conversion module.

3.1.5.3.2. Installing management module

Warning:

This part relates only to users who want to use the management module in php. If you want to use the Cyberdocs platform in command line, skip this step.

Files of the management module will be copied automatically when installing the conversion module. If you choose not to use the management module, you can delete the entire folder "gestion".

Before using the management module, make examinations and following modifications:

1. Files's access rights

The management module is a web site piloting the conversion module. To achieve this goal, php scripts must be able to execute certain commands within Cyberdocs platform's installation directories.

So you should allow your web server to access the conversion commands of the platform. The simplest way is to take the following actions:

Under Linux and Unix-like: make sure you're in the directory containing the pcd-install file and assign all the files on the platform to the user and group account used by the web server.

For example: `chown -Rh apache:apache pcd-install.`

To find out the name of the user account and its group, you can edit your httpd.conf and watch settings of *User* and *Group*.

Under Windows: Verify that the files are available in reading, writing and executing for the user account used by your web server. This is normally the case by default.

2. PHP/Apache parameters

Edit php.ini file from /etc on Linux, c:/windows or install directory of Apache under Windows, and check the following settings (these values are given as an indication, adapt them to your needs):

- Maximum execution time of each script, in seconds: (possibility to modify only for php-cli-cf doc of php-cli and its own php.ini file)
`max_execution_time = 300000`
- Maximum amount of time each script may spend parsing request data :
`max_input_time = 6000`
- Maximum amount of memory a script may consume
`memory_limit = 1600M`

- Maximum size of POST data that PHP will accept.
post_max_size = 1500M
- Show all errors, except coding standards warnings (you can also disactivated the errors!)
error_reporting=E_ALL & ~E_NOTICE
- Whether to allow HTTP file uploads.
file_uploads = On
- Maximum allowed size for uploaded files.
upload_max_filesize = 1400M

Note: For users of Redhat Linux 8.0/9.0 with Apache 2.xx, it may be necessary to modify the /etc/httpd/conf.d/php.conf and change the value of LimitRequestBody.

Add the environment variable in httpd.conf as following:

```
SetEnv          JAVA_HOME          répertoire_installation_java.
```

For example: *SetEnv JAVA_HOME /usr/java/j2sdk1.4.2_01*

Restart Apache after files modifications.

3. HTTP access to the management module

Create a web access to the platform in your web server configuration (this is the httpd.conf file for Apache), pointing in the folder pcd-install/gestion/. For security reasons, do NOT give HTTP access to the "production" directory.

If some documents are confidential, we recommend that you use secure connections (HTTPS) to access the management module.

If necessary, edit the .htaccess file at the root of the pcd-install/gestion/ directory. Configure automatic re-directions to allow downloading files from the "production" directory: eg, uncommment and modify "RewriteBase" with the URL base path of the management module - especially if you have an "alias" in your Apache configuration.

Go to the management module home page (http://nom_host/php/), and identify with the username *root*, without a password. The first action to achieve will be to set a password by going to the heading *users*. See the [management module documentation](#) to learn more.

4. Advanced configuration of management module.

The management module can use an LDAP server for authentication, be easily translated into the language of your choice and be configured differently. To do this, see the section entitled: [3.1.6.8. Customize the management module](#)

5. Warning

We strongly recommend restricting access to the management module to your intranet network. You can also use https connexions.

3.1.5.3.3. Installing publishing module

To install the publishing module, some steps may be necessary. First, we must run the script installation-web.bat or installation-web.sh. This will result to copy the source files in the appropriate folder under SDX, while filtering certain parameters to contextualize your installation.

Next, it is also preferable to copy the configuration files of the institutions for which you want to publish the documents. The script copie-institutions.{bat | sh} performs this operation. In particular, it will copy the example file included with the Cyberdocs platform, for the university

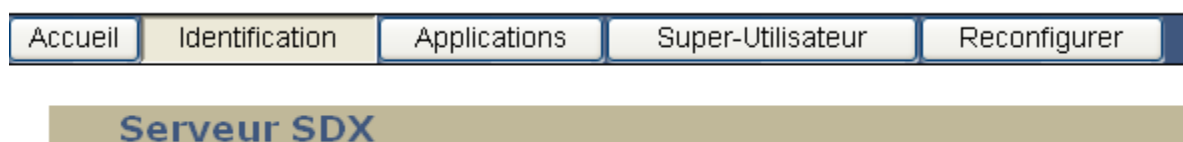
Lyon 2. If you want to publish documents from other institutions, you must create similar files to those found in the folder `exemples/configurations/web/lyon2` in Cyberdocs sources.

Once all sources of Linux 8.0/9.0 with Apache 2.xx: modify the *open* application in SDX. This means that we must tell our server SDX there is a new application installed, and we want to make that application visible and usable.

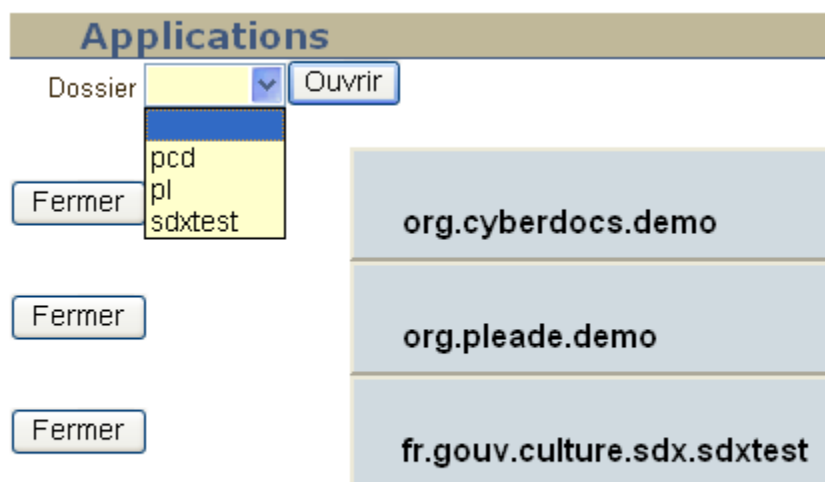
To do so, simply go to the address of the SDX administration page server (eg `http://localhost:8080/sdx/sdx/admin/loginsu.xsp`), the identification form as presented below, and identify yourself as root:



This step done, you next need to click on the **applications** button on top of the screen:



On this page, a drop list offers all the applications available on the server:



By selecting the identifier of your application (value `sdx.application.path` in the `pcd.properties` file), a link to the application shows. It is now open, you only need to click on that link.

This opening operation needs to be done only once.

3.1.6. Customize Cyberdocs

3.1.6.1. Create a new institution

The publishing module web interface assumes that each document is published by an institution. Information on this institution must be provided in order to make the interface clear and effective. If a document of a new institution must be added, it is necessary to configure the module to include this institution. The simplest way is to start from the example provided with the institution *lyon2*.

The information on the institutions are placed in the institutions folder in the web application. In this folder, each institution has its own folder and the name of this folder should be given the code of the institution. In this case, there are three important pieces of information:

1. A configuration file for the institution, absolutely essential.
2. A CSS sheet for the documents issued from the institution, essential but may be empty.
3. A logo, which is displayed in the top frame of the consultation window, when a document published by this institution is consulted.

This is the configuration document, to be named config.xml, which contains important information. The example below can document this file.

Example 1: configuration file of an institution example

```
<!-- institution code must be in the "id" attribute -->
<ins:institution id="lyon2"
  xmlns:ins="http://cyberdocs.org/institutions">
  <ins:informations>

    <!-- institution name-->
    <ins:nom>Université Lumière Lyon 2</ins:nom>
    <ins:logos>
      <!-- logo-->
      <ins:logo role="consultation" height="20" width="10" src="logos/logo.png"/
    >
    </ins:logos>
    <ins:description/>

    <!-- ip addresses included in the Intranet-->
      <ins:intranet>
        <ins:masque ip="192.168.123.*"/>
        <ins:masque ip="127.0.0.*"/>
      </ins:intranet>
    </ins:informations>

    <!-- metadata fields name -->
    <ins:metadonnees>
      <ins:champ nom="dc:creator">
        <ins:libelle xml:lang="fr">Auteur</ins:libelle>
      </ins:champ>
      <ins:champ nom="dc:title">
        <ins:libelle xml:lang="fr">Titre</ins:libelle>
      </ins:champ>
      ...
    </ins:metadonnees>
    <ins:messages>
      <!-- dedication name-->
      <ins:intitule code="dedicace" xml:lang="fr">Dedicace</ins:intitule>
    </ins:messages>
  </ins:institution>
```

3.1.6.2. Add a new styles code

Either the new style author which allow to stylage the author in the styles code "new-code".

1. First, you have to put "new-code" in parameter in the conversion script. (eg : [conversion information](#))
2. In the file production_folder/tools/xslt/utiles/styles.xml, it is necessary to add an element: <institutions> <institution code="new-code"/> </institutions> (as the one existing for lyon2)
3. For **each** styles, it is necessary to declare an element name, indicating the style name in the styles code 'new-code', for example:

```
<style code="auteur"> <name code="lyon2" xml:lang="en">1|Author</nom> <name code="new-code" xml:lang="en">new-author </nom> </style>
```

or also :

```
<style code="text"> <nom code="lyon2" xml:lang="en">Text</nom> <nom code="new-code" xml:lang="en">Text</nom> </style>
```

3.1.6.3. Other language Conversion

1. To use English, you first have to enter the 'en' parameter in the [conversion script](#)
2. In each xml file from production_folder/tools/xslt/utiles elements of type: <element xml :lang= 'fr'>----</element> must have their equivalence in the wished language: <element xml :lang= 'en'>----</element>
3. It's the same for the production_folder/tools/xslt/metadonnees/metadonnees.xml file.

3.1.6.4. CSS style sheets

The documents consultation, in the publishing module or in static, passes through the HTML or XHTML, and it allows you to specify the display by using CSS style sheets. In Cyberdocs, we use much these style sheets, allowing use several of them:

1. A general style sheet applied to all documents.
2. A style sheet by institution, applied to all documents issued by that institution.
3. A style sheet specific to a document.

These three CSS are presented in reverse order of priority, in fact, if two selectors apply CSS, it is the one found in the specific style sheet, and then in the style sheet for the institution, and finally the general.

To add a CSS style sheet to a specific document, simply rename the file [Document code].css and place it in the source folder of the document if the module conversion is used, or place it directly to the root folder of the document.

To add a CSS proper to an institution, you must create a file [code of the institution].css, and place it in the file of the institution conversion module or publishing module.

3.1.6.5. Publishing other documents

It is possible to use the publishing module without having the document in XML format TEILite. For example, it could be a PDF document.

These documents must also have a unique code and be organized by institution and by publishing year. To treat them, simply place the source file into [Document code]/sources and rename the file with the document's code and an extension representative of its format.

You absolutely must have a metadata file, always named [code of document]-md.xml. The element dc:identifier at this time should contain the complete URL of the document.

3.1.6.6. Modifying static pages

The dynamic Web application provides users with an easy interface, it is preferable to present in their own language. The content generated by this application are associated with a specific display, which offers several languages for the same appearance and a same static content. These generated contents are stored in following XML documents:

- `src/web/habillages/pcd/messages/global.xml`: common messages to most of the application's pages
- `src/web/habillages/pcd/messages/menu-general.xml`: definition of the general navigation menu
- `src/web/habillages/pcd/messages/messages-avancee.xml`: messages from the advanced search form
- `src/web/habillages/pcd/messages/messages-cadre.xml`: messages from the upper frame consultation in a document
- `src/web/habillages/pcd/messages/messages-chargement.xml`: messages linked to the indexation form of a document
- `src/web/habillages/pcd/messages/messages-contenu.xml`: messages linked to the main consultation frame of a document
- `src/web/habillages/pcd/messages/messages-fiche.xml`: messages linked to a metadata window of a document
- `src/web/habillages/pcd/messages/messages-id.xml`: messages linked to the identification form
- `src/web/habillages/pcd/messages/messages-imprimer.xml`: messages linked to the impression format choice window
- `src/web/habillages/pcd/messages/messages-rchargement.xml`: messages about the indexation result page
- `src/web/habillages/pcd/messages/messages-resultats.xml`: messages linked to search results
- `src/web/habillages/pcd/messages/messages-termes.xml`: messages linked to the index term's lists.

In order to modify the static content (notably the index page), edit and create files in the `src/web/skins/pcd/static`.

You will note that display default is pcd: it is of course possible to create a new looking totally different based on the basic model. For that, in the sources, create a new directory in the layout directory, lay your custom files and change the parameter "display" of the `pcd.properties` file in order to specify the name of your new look. Then run the script "installation-web" for your new files to be copied to the SDX working folder.

3.1.6.7. Modifying the Web application

In order to modify the Web application, it is possible to distinguish four modifying levels:

- **Modifying the src/web/css/structure.css style sheet:** This is the simplest solution that allows you to change the layout of the documents. The style sheet contains sections for the different parts of the site, remember to read the comments of the file for more information.
- **Modifying the static pages of the site:** consult the specific part of this documentation about those modifications.
- **Modifying XSL files from the folder src/web/xsl:** modify those files only if you can't obtain what you want with the styles sheets and static pages. It is deep modifications, notably to change the automatic generated HTML code.
- **Modifying the XSP files from the folder src/web:** written in JAVA, those pages are the articulation of the Web application. Edit those pages if you want to modify the site compartment (search, filters, new functionalities, etc.).

To note: you have to restart the installation script "installation-web" after each source modification. Otherwise the modified files won't be copied in the SDX folder.

3.1.6.8. Customize the management module

The management module is a dynamic Web application written in PHP (<http://www.php.net>). You can easily modify its behavior by editing the PHP files. However, some configuration files have been planned so as to avoid deep changes.

Here's a quick description of the management module architecture:

- **General configuration file:** it's the "pcd.properties" file global to Cyberdocs. Consult the installation documentation for more information.
- **The "configuration" file src/gestion/gestion/config.inc.php:** available languages, optional configuration of an LDAP server.
- **The "constant" file src/gestion/php/inc/const.php:** find in this file all the constant of the management module (document folder, languages, conversion step, etc.).
- **"Display" files in the "src/gestion/php/inc" folder:** it's the skin for the management module. It's here you can customize pages or translate them in an other language.
- **"applicative" file at the root of the folder src/gestion/php:** it is treatment files for the data conversion management. To modify only if you wish to change the compartment of the management module.

To note: you have to restart the installation script "installation-oo2xml" after each sources modification. Otherwise the modified files won't be copied in the installation folder.

Translation: you can translate the management module without changing the PHP code. To do this:

- In the "configuration" part of the management module, click on "Extract File translation"
- The extracted file is a ".po" file, editable with a "po editor", such as "[poEdit](#)" "[Lokalize](#)", "[Kbabel](#)"...
- Once translated, rename the file with the suffix of the language (eg "en.po" for English)
- Copy the file in the "pcd-install/gestion/po/" directory.
- Add the language in the PHP table "GLOBALS["TRANSLATIONS"]" of the configuration file : "pcd-install/gestion/config.inc.php".

3.1.6.9. Link Apache & Tomcat: install JK module

The module allows JK to make communicate Apache and Tomcat: The main advantage lies in the possibility of using the two servers at the same time as if they were both on port 80. Indeed, the port 8080 (default for Tomcat) is not recommended for a Web server and some network administrators block the access for security reasons. In addition, the URL of your documents will be less complicated to remember and more "conform" to what Internet users are fa-

miliar with. Finally, your Apache statistics consultations must also contain the calls to documents and research via SDX.

Warning: this help page is not exhaustive on the action to be taken in order to install the JK module. We advise you to read the official documentation available on the [website of JK module](#).

This page is an example of installation: we based it upon the using of a Linux Fedora 7 distribution.

You must have installed and configured your Apache and Tomcat servers. We advise you to install SDX beforehand so that you can more easily test your new configuration.

1) Dependences

Verify that you have install the following packages:

- http-devel (yum install http-devel)

2) Download JK module sources

You will fin the source of the last JK module on the official Web site: <http://tomcat.apache.org/connectors-doc/>

Or, for example, directly in command lines:

```
wget http://www.apache.org/dist/tomcat/tomcat-connectors/jk/source/jk-1.2.25/tomcat-connectors-1.2.25-src.tar.gz
```

Extract the archive in /usr/loca/src/

3) Compilation and installation

Find the APX installation folder:

```
# which apxs
/usr/sbin/apxs
```

Compilation :

```
cd /usr/local/src/tomcat-connectors-1.2.25-src/native/
./configure --with-apxs=/usr/sbin/apxs
make
su -c 'make install'
```

4) Apache configuration

Create the configuration file /etc/httpd/conf/workers.properties with the following content, adapting it with your installation path of Tomcat and Java.

```
# tomcat home
workers.tomcat_home=/data/tomcat

#localisation of the jvm
workers.java_home=/data/jdk1.6.0

# Define 1 real worker using ajp13
worker.list=worker1

# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=127.0.0.1
worker.worker1.port=8009
```

Add the JK module configuration to your httpd.conf file:

```

# ##### MOD JK #####
# Load mod_jk module
# Update this path to match your modules location
LoadModule      jk_module  modules/mod_jk.so

# Where to find workers.properties
# Update this path to match your conf directory location (put workers.properties
next to httpd.conf)
JkWorkersFile   "/etc/httpd/conf/workers.properties"

# Where to put jk shared memory
# Update this path to match your local state directory or logs directory
JkShmFile       "/etc/httpd/logs/mod_jk.shm"

# Where to put jk logs
# Update this path to match your logs directory location (put mod_jk.log next to
access_log)
JkLogFile       "/etc/httpd/logs/mod_jk.log"

# Set the jk log level [debug/error/info]
JkLogLevel      info

# Select the timestamp log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

# Send everything for context /examples to worker named worker1 (ajp13)
JkMount         /sdx/* worker1

```

Here, we can see that on the last line all the request starting by "sdx" are redirected to Tomcat.

Restart your web server:

```
/etc/init.d/httpd reload
```

Test: http://your_server/sdx/

You have to see the SDX administration page.

3.1.6.10. Use a MySQL database

By default, SDX works with a database of documents stored in a directory tree situated in its workspace. This system is acceptable for a small base of documents, but it quickly becomes a very weak basis for a large and growing charge.

It is therefore preferable, once the Cyberdocs installation is complete and tested, to change the storage system for a more efficient database engine. If you want to make this change on a production basis, note that you must re-index all your documents.

We see here an example of MySQL use, which is available in GPL and with most Linux distributions. Obviously, the use of another database system is possible.

Here are the configuration modifications to do to use MySQL:

1. Stop your Tomcat (cd tomcat/bin, ./shutdown.sh)
2. Download the java connector for MySQL on the following address:
<http://dev.mysql.com/downloads/connector/j/3.1.html>.
Copy the file ".jar" in the folder webapps/sdx/WEB-INF/lib. The file ".jar" is usually at the root of the downloaded archive. For example: mysql-connector-java-3.1.7-bin.jar.

3. Create a new MySQL database. For example, you can call it "pcd". We also advise you to create a MySQL user "pcd" with complete rights on this new database.
4. Modify your MySQL configuration to authorize important size requests. For that, add the following lines to the file /etc/my.cnf:

```
# Packets maximum size
max_allowed_packet=1024M
```

Restart your MySQL server (/etc/init.d/mysql restart).

5. Edit the file webapps\sdx\WEB-INF\web.xml, to add or modify the following lines:

```
<init-param>
  <param-name>load-class</param-name>
  <param-value>com.mysql.jdbc.Driver</param-value>
</init-param>
```

6. Edit the file webapps\sdx\WEB-INF\cocoon.xconf to add or modify the following lines - modifying the url, the login and mysql password as needed:

```
<datasources>
  <jdbc name="id_mysql" logger="sdx.rdbms.id_mysql">
    <pool-controller min="5" max="10"/>
    <dburl>jdbc:mysql://localhost:3306/pcd?autoReconnect=true</dburl>
    <user>myuser</user>
    <password>mypassword</password>
  </jdbc>
</datasources>
```

7. Modify the file src/web/conf/application.xconf :

- At the beginning of the node "<sdx:userDocumentBase>", add: "<sdx:database type='MYSQL' dsi='id_mysql'/">"

- Replace (or comment) in "<sdx:repositories>" of the node "<sdx:userDocumentBase>":

```
<sdx:repository id="users" type="FS" baseDirectory="users/xml" depth="0" extent="1000"/>
By :
<sdx:repository id="users" type="MYSQL" dsi="id_mysql"/>
```

- At the beginning of the node "<sdx:documentBase id='documents' type='lucene' default='true' maxFieldLength='100000'/" situated in "<sdx:documentBases>", add: "<sdx:database type='MYSQL' dsi='id_mysql'/">".

- Replace (or comment) in "<sdx:repositories>" of the node "<sdx:documentBases>":

```
<sdx:repository type="FS" id="rdocuments" baseDirectory="documents" default="true"/>
By :
<sdx:repository type="MYSQL" id="rdocuments" dsi="id_mysql" default="true"/>
```

To help you, you can download an application.xconf example file after modifications.

8. **Restart the script "installation-web" to copy the application.xconf in the SDX folder.**

9. Restart Tomcat (cd tomcat/bin, ./startup.sh)

You must see 4 tables in your MySQL database: documents, rdocuments, sdxuserdb, users. In addition, the names of the tables will be prefixed by the ID of the application specified in your "pcd.properties" file.

Please note:

- The database can be of a consistent size (several gigabytes): therefore be sure to have enough disk space.
- You can delete the directory webapps\sdx\pcd\conf\document.

3.1.6.11. Integrate the publishing module to your Web site

There is currently no preconceived solution to integrate the publishing module to an existing site. However, based on XML-RPC/SOAP projects are under consideration.

For now, under the [Cybertheses program](#), the university Lyon 2 has modify XSL Lyon 2 of the publishing module **to use it with the CMS Drupal**. However, it is entirely possible to use these "modified" sources with other dynamic sites, by doing some changes to PHP scripts.

We would like to clarify that this is a temporary solution pending further development.

The principle is as follows:

- To not use directly SDX interfaces for static pages and search into the documents.
- Integrating the search engine to an existing site.

These are the conditions to use these "re-modified" sources:

- You already have a functional installation of Cyberdocs.
- Your server configuration accepts PHP.
- You have knowledge in PHP.
- You have time to test and modify provided scripts.
- Your institution already has a dynamic site which you want to integrate the Cyberdocs publishing module's (notably for greater visibility and to keep a single access).

Installation and use:

Given the complexity of the installation and the risk of a security hole in the case of sources misuse, we chose not to directly distribute these changes: please contact us via the heading "contact" for more information.

3.1.7. Tools

You will find some tools for the Cyberdocs platform on <http://www.cyberdocs.org>

3.1.7.1. Re-indexing all your documents

In the event of a problem or following a change in configuration, it may be necessary to re-index all the documents in the database.

The proposed method is probably not the best, but it has the advantage of being relatively simple, for a development cost virtually nil.

Pre-requisite: the re-indexing script use « wget » and « awk ».

Re-indexing your documents, step by step:

- Download [this archive](#) and uncompress it: you will find a "reindex" folder containing the re-indexing tools.
- In the "reindex/web" folder, you will find three files allowing you to add to your SDX application a simplified re-indexing XSP page. Copy the three files in the Cyberdocs sources arborescence as following below:
 - reindex/web/rsimplechangement.xsp: place this file in your src/web/ folder.
 - reindex/web/xsl/rsimplechangement.xsl: place this file in your src/web/xsl/ folder.

- reindex/web/habillages/pcd/xsl/rsimplechangement.xsl : place this file in the xsl folder of your skin display.
- Execute again the "installation-web" script in order to install the new files in your SDX application. Restart Tomcat.
- If you use the publishing module, copy your institution "index.dat" file (for example production/lyon2/index.dat) in the "reindex" folder.
- If you don't use the management module, you have to create an "index.dat" text file containing the list of all the document spaces to re-index, with for each line the following format: "folder_name_doc:year_on_4_numbers:". For example: "grosjean_mf:2003:".
- Placed yourself in the "reindex" folder and execute the "reindex.sh" command as indicated below, in order to make the global re-indexing command file:

```
./reindex.sh URL_APPLICATION_PCD PRODUCTION_FOLDER_PATH LANGUAGE
ID_INSTUTION TOMCAT_FOLDER MANAGEMENT_FOLDER|all FILE_INDEX_DAT >
reindex_all.sh
```

For example, to re-index all Lyon 2 institution:

```
./reindex.sh http://mon_server/sdx/pcd /data/pcd-install/production fr
lyon2 /data/tomcat all index.dat > reindex_all.sh
```

And for example, to re-index only the Lyon 2 institution documents placed in the "archive" folder of the management module:

```
./reindex.sh http://mon_server/sdx/pcd /data/pcd-install/production fr
lyon2 /data/tomcat arch index.dat > reindex_all.sh
```

- Verify the "reindex_all.sh" file: it must contain calls to "wget".
- Attribute execution rights to all ".sh" files of the "reindex" folder and execute "reindexall.sh": ./reindexall.sh > resultat.txt &
- You can supervise the re-indexing by consulting the resultat.txt file (tail resultat.txt). If errors show, note the document locations concerned and treat those case manually.

Important:

- **Every 50 documents charge, the Tomcat server is automatically stopped and restart.**
- **Once all the documents reindexed, think to suppress the "rsimplechangement.xsl/xsp" files: they are no longer needed.**

3.1.7.2. Modify your metadata in mass

It may be helpful to modify certain metadata in a comprehensive manner. However, there is no integrated tool in Cyberdocs to perform these operations.

The easiest way is probably to use regular expressions and tools for automatic replacements. We will take the example of the suppression of the 8080 port in the documents reference URL, following an installation of JK module.

All of this is, of course, given as an example.

1) Creation of a replacement script

Create a "replace.sh" script :

```
cp $1 $1.orig  
sed 's/mon_serveur.fr:8080/mon_serveur.fr/g' $1.orig > $1
```

A backup copy of the file will be automatically created before the replacement.

2) Find the files to modify

One example to find the files to modify:

```
find pcd-install/production/ -name "*.xml" -exec grep "mon_serveur.fr:8080" -l  
'{}' \; > list_fichier.txt  
find pcd-install/production/ -name "*.dat" -exec grep "mon_serveur.fr:8080" -l  
'{}' \; >> list_fichier.txt
```

3) Do the replacement

```
awk '{ print "./replace.sh \"$0 }' list_fichier.txt > changeall.sh  
chmod +x changeall.sh  
./changeall.sh
```

Once the verification done, you can suppress the backup files.

```
find pcd-install/production/ -name "*.xml.orig" -exec rm -f '{}' \;  
find pcd-install/production/ -name "*.dat.orig" -exec rm -f '{}' \;
```

3.2. Help to users

3.2.1. Technological approach

The Cyberdocs platform's conversion module aims to automate a process of conversion from a word processing format to a structured document in XML format, according to the DTD TEILite. Such an operation is therefore to identify the most possible *structure* in the original document to make the XML document as rich as possible. The structure elements present in a word-processing document can be:

- Typographical enrichment : bold or italics, colors, particular fonts, etc.
- Structures already managed by word processor: tables, illustrations, hyperlinks, footnotes, etc.
- Styles (paragraphs or characters) with a common semantic to all documents: Title 1, Title 2, etc.
- Styles with a particular semantic for Cyberdocs platform: author, document title, publishing year, etc.

The Cyberdocs platform's conversion module does not only produce this XML reference document. Indeed, once it is obtained, the module can produce versions of the document ready for static publication, in HTML, PDF or XHTML format. In addition, the module will prepare a bunch of documents that will fuel the publishing module to make the consultation interface even richer.

A series of treatments will be carried out, in sequence, in order to take maximum advantage of these information. These treatments involve five major technologies:

- [XSLT](#) (*eXtensible Stylesheet Language Transformations*): This programming language was developed specifically for process information in XML format. It is perfectly appropriate for treatments made by the conversion module, and except from the initial stage which uses OpenOffice.org, all other developments have been carried out using this standard language. The conversion module includes an XSLT processor: [Saxon](#) 6.5.2 version, developed by Michael Kay and freely available.
- [XSL-FO](#) (*eXtensible Stylesheet Language Formatting Objects*): This is the other aspect of the XSL standard, which consists of a layout format (XML), thus preparing documents suitable for printing. The production of PDF documents in the conversion module passes in advance by generating a XSL-FO format, which will then be processed through an XSL-FO processor, included in the Cyberdocs distribution. This processor is [FOP](#) from the Apache XML Project.
- [SVG](#) (*Scalable Vector Graphics*): this graphical vector format is used when converting to the XSL-FO format, in order to allow finer and more precise images included in the documents processed. The XSL-FO processor FOP, when dealing with documents, gives a better display when images are included in the SVG documents.
- [ANT](#): this Apache project provides a flexible and powerful environment for performing different operations. It already includes a series of tasks, including driving XSLT or XSL-FO transformations, copying files, and so on. That is why ANT is used to control the various operations of conversion proposed by the Cyberdocs's conversion module.
- [Java](#): ANT environment is a Java application, a Java virtual machine is used to support all the operations carried out by the conversion module, with the exception of the initial treatment done by OpenOffice.org.

An operator converting documents using Cyberdocs does not necessarily need to know all the technical details. However, when things go wrong or when installation must be carried out properly, it may be important to understand the role of different technologies and their implementation.

The use of management module provides to know precisely the names of the stages of conversion and particularly to use the conversion module from the command line. We therefore separate documentation into two distinct parts: a visual documentation for users of the management module and a very detailed documentation for users from the command line.

3.2.2. Convert a document via the management module

The **Cyberdocs** tool allows users to convert text documents into an XML format guaranteeing a permanent archiving of documents (which thus no longer depend to a certain program), and from which they are generated consultation formats (HTML, XHTML) and printing format (PDF).

3.2.2.1. Identification under Cyberdocs

To use Cyberdocs, the administrator has to give you a user name and password that allow you to access your institution. For example here, we'll take the University Lumière Lyon 2.

Cyberdocs - Module de gestion

Utilisateur :

Mot de passe :

Institution :

You can then select the institution from which you will treat the documents:

Cyberdocs - Module de gestion

Accueil

Utilisateurs

Conversions

Bienvenue dans l'outil d'administration de Cyberdocs.

- Listes des institutions disponibles:
 - ◊ Université Lumière Lyon 2

The next screen allows you to access to already treated documents on the server, or to create new documents.

3.2.2.2. Create a space for a thesis

If you want to convert a document that you just prepared, you first have to create a space for this new document:

Cyberdocs - Module de gestion

Accueil

Utilisateurs

Université Lumière Lyon 2:

- ◊ Création d'un espace pour un nouveau document

Espace des documents:

Cyberdocs - Module de gestion

Accueil

Utilisateurs

Conversions

Se déconnecter

Université Lumière Lyon 2:

Création d'un nouvel espace

- Identifiant du document:
- Année de soutenance:
- Langue du document:

The **identifiant** is the author's name followed by the first letter of the first name from which it is separated by an underscore, all of it in lowercases; for example, for Edith Jaillardon's thesis, we will have the following identifier: **jaillardon_e**. This identifier can be modified, but usually it's not necessary.

The new space you have just created in Cyberdocs will look like this:

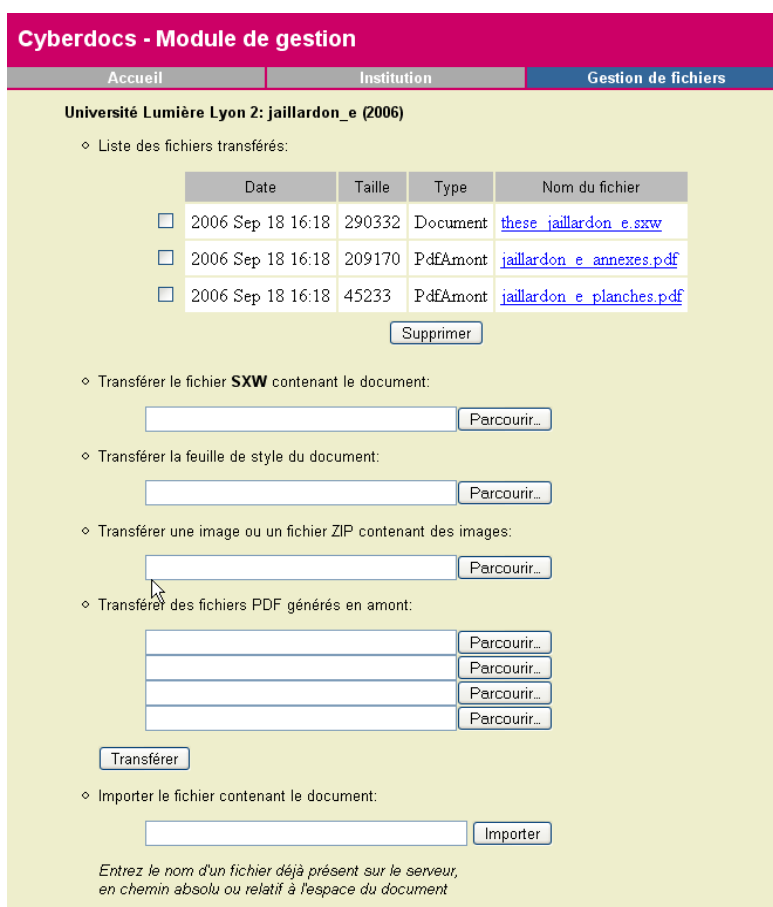


3.2.2.3. Conversion procedure

The navigation inside this space will be done through the following tabs: **Files Management**, **Metadata** and **Treatments**. Proceed in this order:

3.2.2.3.1. *Styled files and associated files transfer (Files Management tab)*

Load files that must be addressed by the chain in the appropriate fields. The main file must be a text document with the **.odt** or **.sxw** extension, created with OpenOffice. The files associated with the main document can be image files (with a **.jpg**, **.gif** or **.png** extension, which can be grouped into an archive. Zip) or **.pdf** files (it is not possible to regroup them in an archive **.zip**).



3.2.2.3.2.

Metadata input (Metadata tab)

- It is vital to type and validate the metadata before the conversion for them to be associated to the document.

Cyberdocs - Module de gestion

Accueil Institution Gestion de fichiers Métadonnées

Université Lumière Lyon 2: jaillardon_e (2006)

◊ Métadonnées du document:

- Editer
- Visualiser

Cyberdocs - Module de gestion

Accueil Institution Gestion de fichiers Métadonnées

Métadonnées du document - Université Lumière Lyon 2: jaillardon_e (2006)

Les champs avec une * doivent être renseignés obligatoirement

Auteur* (NOM Prénom) JAILLARDON Edith

Titre* Français

La suppléance parlementaire sous la Cinquième République

Français

Français

- When a thesis has two authors, names and firstnames of the two authors must be indicated in the same "author" field:

Cyberdocs - Module de gestion

Accueil Institution Gestion de fichiers Métadonnées Traitements

Métadonnées du document - Université Lumière Lyon 2: gerfaud-tourrel (2000)

Les champs avec une * doivent être renseignés obligatoirement

Auteur* (NOM Prénom) TOURREL Jean-Paul & GERFAUD Jean-Pierre

- When a thesis has been written in an other language than French (English ??) (case of some under joint supervision thesis written in English, Italian, Portuguese, ...) you indicate, in the same field "French Title", the title in French and in the other language:

Auteur* (NOM Prénom)	MARIÉTHOZ Johnny
Titre*	Français <input type="button" value="v"/> Discriminant Models for Text-independent Speaker Verification / Algorithmes d'apprentissage discriminants en vérification du locuteur

- For PDF format thesis (but not those containing only certain parts in PDF), you indicate, in the field "Publishing comments", the following mention: "PDF format published thesis".



In order to validate these metadata, you **imperatively** have to click a first time, at the bottom of the first screen, on **Modify**, then at the bottom of the second screen on **Continue**.

Métadonnées du document - Université Lumière Lyon 2: jaillardon_e (2006)	
Auteur (NOM Prénom)	JAILLARDON Edith
Titre	Français : La suppléance parlementaire sous la Cinquième République
Sous-titre	
Ecole doctorale	Sciences des sociétés et du droit
Faculté de soutenance	
Département	
Discipline	Science politique
NOM Prénom du directeur de thèse	DEMICHEL Francine
Résumé	
Mots-clés (séparés par des ;)	
Mots-clés temporels (séparés par des ;)	
Mots-clés géographiques (séparés par des ;)	
NOM prénom d'un membre du jury	
Date de soutenance (aaaa-mm-jj)	1976-01-01
Nature du contenu	
Langue principale de la thèse	Français
Copyright	
Type de diffusion	Intranet
Commentaires de diffusion	
Références à d'autres ressources	
URL de référence	http://demeter.univ-lyon2.fr:8080/sdx/theses/lyon2/2006/jaillardon_e

Continuer

Now the metadata are registered and you can launch the conversion of the documents.

3.2.2.3.3.

Conversion (Treatments tab)

When working on a new document, you launch a **Complete treatment** which will generate an XML document and derived products (HTML, PDF).

The screenshot shows the 'Cyberdocs - Module de gestion' interface. At the top, there is a navigation bar with five tabs: 'Accueil', 'Institution', 'Gestion de fichiers', 'Métadonnées', and 'Traitements'. The 'Traitements' tab is active. Below the navigation bar, the document title is 'Université Lumière Lyon 2: jaillardon_e (2006)'. The main content area is divided into several sections:

- ◇ Répertoire du document: **Nouveaux documents**
 - Modifier le répertoire du document: Nouveaux documents (dropdown menu)
- ◇ Commentaire du document:
 - Modifier le commentaire du document: [input field] [Modifier button]
- ◇ Commandes disponibles:
 - **Traitement complet** (highlighted with a mouse cursor)
 - [Production du document XML de référence](#)
 - [Validation du document XML de référence](#)
 - [Génération des produits dérivés \(html, pdf...\)](#)
 - [Autres traitements - document non XML](#)
- ◇ Résultat de la conversion: *Aucun fichier*
- ◇ Liste des fichiers produits par l'outil de conversion:
 - [PDF en Amont](#)
- ◇ [Effacer](#) tous les fichiers produits par les conversions.
- ◇ [Supprimer](#) ce document

The time needed for the document conversion depends on its size and its contents complexity (contains illustrations for example); for a thesis, you have generally to count few minutes (less than 15 minutes) but some documents may take up to one hour or more for their treatment. The heading **Conversion result** : See allows to follow conversion steps, to be informed of the problems and to verify the results:

Cyberdocs - Module de gestion

Accueil Institution Gestion de t

Université Lumière Lyon 2: jaillardon_e (2006)

- ◊ Répertoire du document: **Nouveaux documents**
 - Modifier le répertoire du document: Nouveaux documents ▾
- ◊ Commentaire du document:
 - Modifier le commentaire du document:
- ◊ Conversion : **en cours**
 - Utilisateur : **jerome**
 - Date de lancement : **19/09/2006 08:32:33**
 - Date de fin :
 - Statut : **En cours**
 - Action : [Annuler](#)
- ◊ Résultat de la conversion: [Voir](#)
- ◊ Liste des fichiers produits par l'outil de conversion:
 - [OpenOffice](#)
 - [PDF en Amont](#)
- ◊ [Effacer](#) tous les fichiers produits par les conversions.
- ◊ [Supprimer](#) ce document

Cyberdocs - Résultat de la conversion

Note : si vous laissez cette page ouverte, elle se rechargera automatiquement tous les 30 secondes.

lyon2 - 2006 - jaillardon_e

Buildfile: oo-vers-tei.xml

initialisation

Conversion du fichier de traitement de texte vers OpenOffice.org

Décompresse le fichier produit par OpenOffice

```
[unzip]Expanding: /data/oo2xml/production/lyon2/2006/jaillardon_e/oo/jaillardon_e.sxw into /data/oo
[touch]Creating /data/oo2xml/production/lyon2/2006/jaillardon_e/oo/xml/office.dtd
[touch]Creating /data/oo2xml/production/lyon2/2006/jaillardon_e/oo/xml/META-INF/Manifest.dtd
```

- If the conversion is reached, the message "BUILD SUCCESSFUL" is displayed at the bottom of the page; before going to the verification step, you have to verify that, at the top of the page, under the heading **XSLT conversion first step, from OpenOffice.org to an intermediate XML format** there is no message saying that the treated document contains one or more *unregistered styles*:

```
Première étape XSLT de conversion, depuis OpenOffice.org vers un format XML intermédiaire

[style]Processing /data/oo2xml/production/lyon2/2004/teil_g/oo/xml/content.xml to /data/oo2xml/production/lyon2,
[style>Loading stylesheet /data/oo2xml/outils/xslt/01-validation/pilote.xsl
[style]INFO:100:Le traitement de l'étape débute à 2006-09-25T08:45:19
[style]INFO:200:Attention, on ne traite pas office:script
[style]AVERTISSEMENT:400:Il manque le style l[EcoleDoct, l|Auteur,
[style]AVERTISSEMENT:300:Attention, le style Contents 5 n'est pas déclaré dans le fichier utiles/styles.xml
[style]AVERTISSEMENT:300:Attention, le style Contents 2 n'est pas déclaré dans le fichier utiles/styles.xml
[style]AVERTISSEMENT:300:Attention, le style Contents 2 n'est pas déclaré dans le fichier utiles/styles.xml
[style]AVERTISSEMENT:300:Attention, le style Contents 1 n'est pas déclaré dans le fichier utiles/styles.xml
[style]AVERTISSEMENT:300:Attention, le style Contents 1 n'est pas déclaré dans le fichier utiles/styles.xml
[style]AVERTISSEMENT:300:Attention, le style Contents 1 n'est pas déclaré dans le fichier utiles/styles.xml
[style]AVERTISSEMENT:300:Attention, le style Contents 1 n'est pas déclaré dans le fichier utiles/styles.xml
[style]AVERTISSEMENT:300:Attention, le style Contents 2 n'est pas déclaré dans le fichier utiles/styles.xml
[style]AVERTISSEMENT:300:Attention, le style Contents 3 n'est pas déclaré dans le fichier utiles/styles.xml
```

The presence of unregistered styles in the document does not block the conversion, but any paragraph that has an unregistered style is **deleted** when converting. If you see your messages indicating the presence of unregistered styles, we must consider that the conversion is not correct. You have to go back to the OpenOffice document, replace these unregistered styles by styles belonging to the model Thesis Lyon2, and repeat the conversion in accordance with the procedure described below for a conversion that failed.

It is sometimes impossible to replace a style in Word. In that case, several possibilities are available to you:

- **applied the style "Font by default"** to the concerned paragraph;
- to **cut** it and **pas tit as text without formatting** (menu Edition / special paste);
- **suppress totally the style**.



Without forgetting that in all case to resume manually the texts enrichment which have been suppressed.

- If no unregistered style appears, you can go to the next step: verification.
- If the conversion is not successful, the message "BUILD FAILED" is displayed at the bottom of the page; the information figuring on the page will allow you to identify the problem that caused the conversion to fail.
- If the error came earlier in the conversion, that means it exists a problem in the documents under conversion, which prevent the smooth running of it (problems list to establish).

In this second case, the conversion has to be redone. Proceed like this:

- Go back in your source documents and correct the problems that blocked the conversion.

- In Cyberdocs, erase all the files products by the conversion from the page **Treatments:**

Université Lumière Lyon 2: jaillardon_e (2006)

- ◊ Répertoire du document: **Nouveaux documents**
 - Modifier le répertoire du document: Nouveaux documents
- ◊ Commentaire du document:
 - Modifier le commentaire du document:
- ◊ Commandes disponibles:
 - [Traitement complet](#)
 - [Production du document XML de référence](#)
 - [Validation du document XML de référence](#)
 - [Génération des produits dérivés \(html, pdf...\)](#)
 - [Autres traitements - document non XML](#)
- ◊ Conversion : **terminée**
 - Utilisateur : **jerome**
 - Date de lancement : **19/09/2006 08:32:33**
 - Date de fin : **19/09/2006 08:36:38**
 - Statut : **Terminé**
 - Action : [Fermer](#)
- ◊ Résultat de la conversion: [Voir](#)
- ◊ Liste des fichiers produits par l'outil de conversion:
 - [OpenOffice](#)
 - [XML TEI Lite](#)
 - [HTML](#)
 - [XHTML](#)
 - [PDF](#)
 - [PDF en Amont](#)
- ◊ **Effacer** tous les fichiers produits par les conversions.
- ◊ [Supprimer](#) ce document

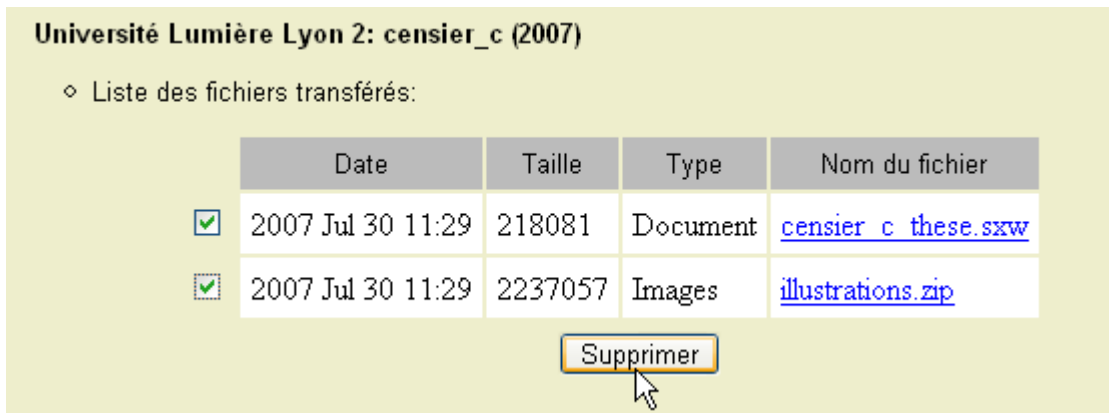
Suppress the sources documents that you've loaded, from the **Files management** tab:

Université Lumière Lyon 2: censier_c (2007)

◊ Liste des fichiers transférés:

	Date	Taille	Type	Nom du fichier
<input checked="" type="checkbox"/>	2007 Jul 30 11:29	218081	Document	censier_c these.sxw
<input checked="" type="checkbox"/>	2007 Jul 30 11:29	2237057	Images	illustrations.zip

Supprimer



Load the corrected files from the **Files management** tab.

Launch a new complete treatment from the **Treatment** tab.

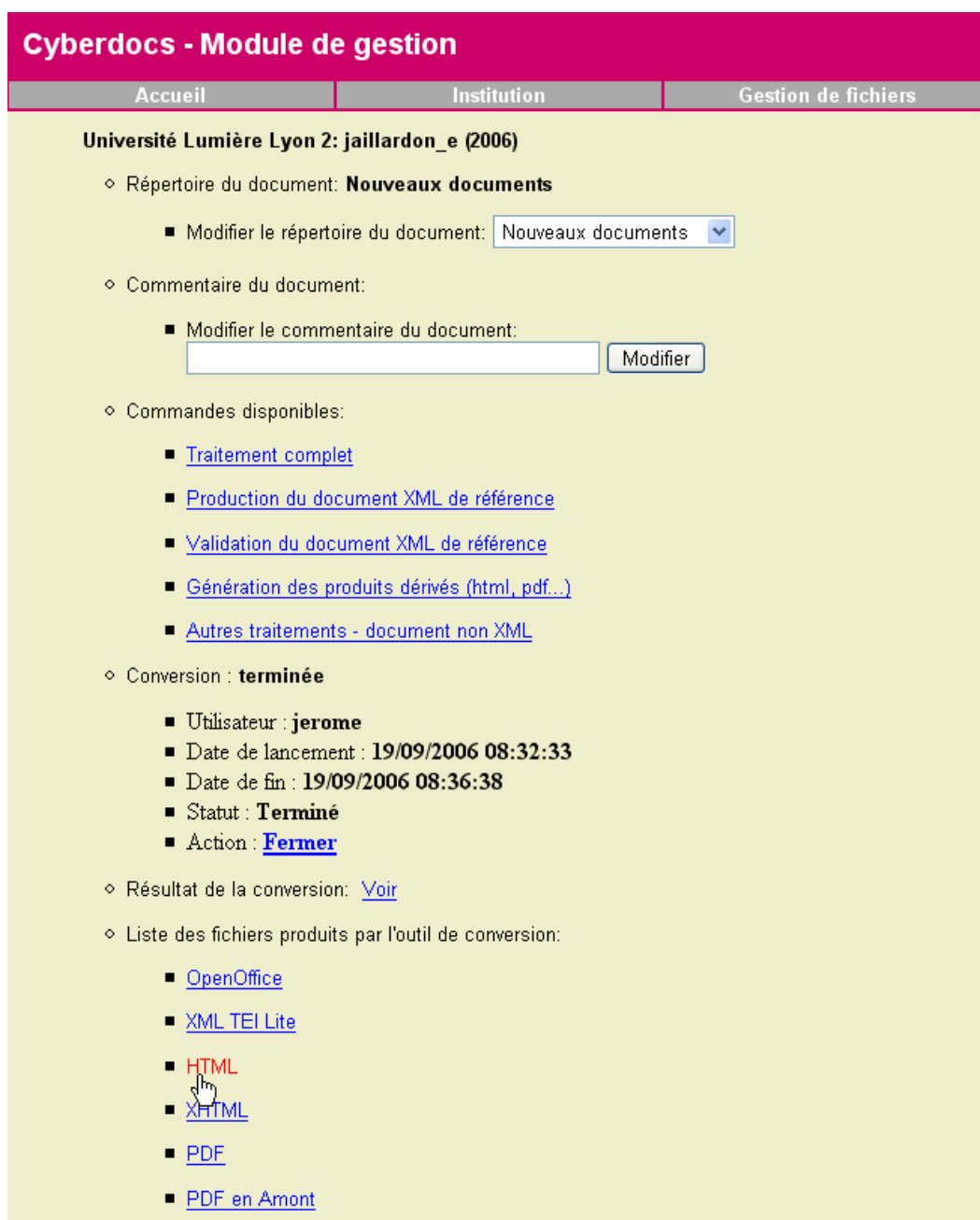
In case of a new conversion failure, this procedure has to be repeat as many times as necessary, until you got a "BUILD SUCCESSFUL" message.

N.B. The Cyberdocs chain can treat several documents at the same time.

3.2.2.4. Verification of the conversion result

When the conversion was completed, it is still necessary to verify that the materials do not contain residual problems, before publishing them on the online thesis website. For this, you conduct a verification from the HTML files produced by the conversion, which give an overview of the documents when they are online and accessible by Internet users.

For that, in the **Treatments** tab, heading **File list produced by the conversion tool**: activate the **HTML** link and choose the **index-frames.html** document:



The screenshot shows the 'Cyberdocs - Module de gestion' interface. At the top, there is a navigation bar with three tabs: 'Accueil', 'Institution', and 'Gestion de fichiers'. Below the navigation bar, the main content area is titled 'Université Lumière Lyon 2: jaillardon_e (2006)'. The interface is organized into several sections:

- Répertoire du document: Nouveaux documents**
 - Modifier le répertoire du document: Nouveaux documents (dropdown menu)
- Commentaire du document:**
 - Modifier le commentaire du document: [input field] [Modifier button]
- Commandes disponibles:**
 - [Traitement complet](#)
 - [Production du document XML de référence](#)
 - [Validation du document XML de référence](#)
 - [Génération des produits dérivés \(html, pdf...\)](#)
 - [Autres traitements - document non XML](#)
- Conversion : terminée**
 - Utilisateur : **jerome**
 - Date de lancement : **19/09/2006 08:32:33**
 - Date de fin : **19/09/2006 08:36:38**
 - Statut : **Terminé**
 - Action : [Fermer](#)
- Résultat de la conversion: [Voir](#)**
- Liste des fichiers produits par l'outil de conversion:**
 - [OpenOffice](#)
 - [XML TEI Lite](#)
 - [HTML](#) (highlighted with a mouse cursor)
 - [XHTML](#)
 - [PDF](#)
 - [PDF en Amont](#)

Cyberdocs - Module de gestion

Accueil

Institution

Université Lumière Lyon 2: jaillardon_e (2006)

◇ Fichiers HTML (html) :

- [TH.1.html](#) - 22.8 Ko
- [index-frames.html](#) - 1.4 Ko
- [TH.2.html](#) - 92.5 Ko
- [TH.3.html](#) - 502.6 Ko
- [TH.4.html](#) - 50.3 Ko
- [TH.5.html](#) - 23.3 Ko
- [TH.6.html](#) - 10.1 Ko

You will access the document as it will appear on the online theses site (except some formatting differences), the active table of contents, in the left frame, allows you to navigate within the document; Tabs lead directly to the list of tables and figures contained in the document:

The screenshot displays a web interface with two main sections. On the left, there is a table of contents with a 'Contenu' tab selected. The table of contents lists the following items:

- ☐ Page de titre
- ☐ introduction
- ☐ 1e partie. cadres d'intelligibilité;
- ☐ 2e partie. différenciation des modes de socialisation;
- ☐ 3e partie. équilibres de vie spécifiés par la sexualité féminine;
- ☐ conclusion générale
- ☐ bibliographie
- ☐ ANNEXES

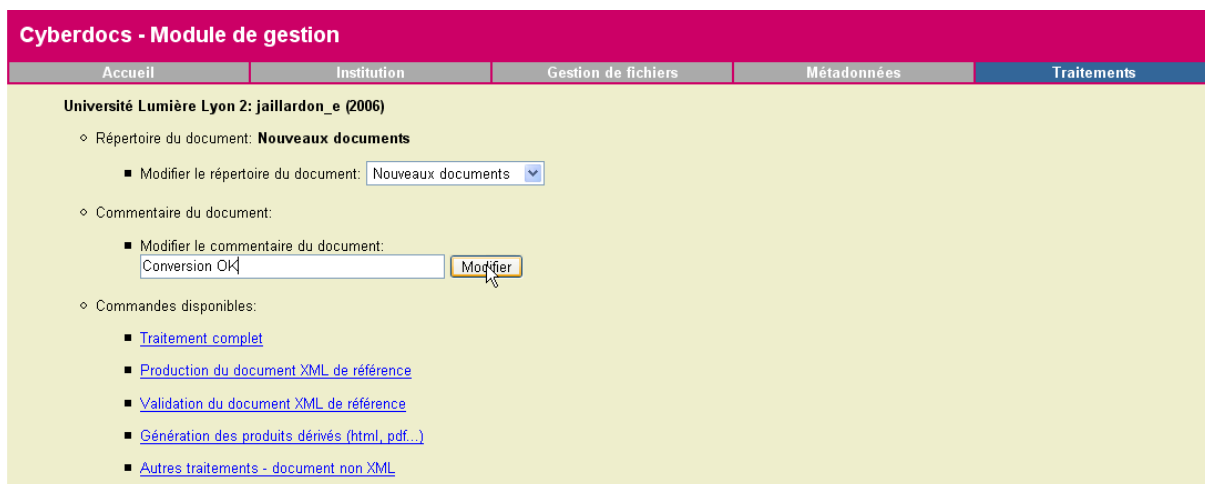
Below the table of contents, there is a small text: 'Toc non lisible avec displayToc'.

On the right, there is a document preview. At the top, it reads 'UNIVERSITÉ LUMIÈRE LYON 2' and 'faculté de sociologie et d'anthropologie'. Below this, it says 'sociologie et sciences sociales'. The main title of the document is 'LOGIQUES DE DIFFÉRENCIATIONS : ANCIENNES LYCÉENNES D'ORIGINE ALGÉRIENNE'. Below the title, it states 'Présentée et soutenue publiquement par : Monique DESCHAMPS'. The document is a 'THÈSE pour obtenir le grade de : DOCTEUR DE L'Université LUMIÈRE LYON 2'. It is 'Dirigée par : Yves GRAFMEYER, Professeur de sociologie, université Lumière Lyon 2'. The document was 'Présentée et soutenue publiquement le 7 juillet 2006'. The jury members are listed as: 'Devant un jury composé de : Isabelle BERTAUX-WIAME, chargée de recherche au CNRS; Yves GRAFMEYER, professeur à l'université Lumière Lyon 2; Yves LEQUIN, professeur émérite à l'université Lumière Lyon 2; André PETITJEAN, professeur à l'université de Metz; Olivier SCHWARTZ, professeur à l'université René Descartes Paris 5'. At the bottom, it says 'Copyright DESCHAMPS Monique et Université Lumière - Lyon 2 - 2006.' and 'Ce document est protégé par un droit de propriété intellectuelle.'

The points to verify in particular are:

- o Title page has all required information; these information are conform to those on the metadata note;
- o Parts and sub-parts are correctly displayed and do not have organisation mistakes; titles are placed in accentuated lower cases, with a capital letter at the begining;
- o Tables and illustrations;
- o Test all possible links from the tabs and be sure that tables and illustrations are correctly displayed;
- o When there's numbered lists (ex. Figure 1, Tableau 1), be sure that there is no missing number;
- o If several figures or tables have the same title, you have to distinguish them by precisng the legend (for ex. "Tableau 1" in the text body and "Annex - Table 1").
- o Spcecial caracters are correctly displayed;
- o Illustrations are visibles and do not exceed the border (excep in big sized illustration with many details, that you cannot reduce without ruining the readability);
- o Linked files (PDF, audio, videos) opens properly.

If the document is correct, you report that the verification has been done by adding a comment in the **Treatments** tab, heading **Document comments**:



The screenshot shows the 'Cyberdocs - Module de gestion' interface. At the top, there is a navigation bar with tabs: 'Accueil', 'Institution', 'Gestion de fichiers', 'Métadonnées', and 'Traitements'. The 'Traitements' tab is active. Below the navigation bar, the document title 'Université Lumière Lyon 2: jaillardon_e (2006)' is displayed. Underneath, there are several sections:

- ◊ Répertoire du document: **Nouveaux documents**
 - Modifier le répertoire du document: Nouveaux documents (dropdown menu)
- ◊ Commentaire du document:
 - Modifier le commentaire du document: Conversion OK (text input) [Modifier] (button)
- ◊ Commandes disponibles:
 - [Traitement complet](#)
 - [Production du document XML de référence](#)
 - [Validation du document XML de référence](#)
 - [Génération des produits dérivés \(html, pdf...\)](#)
 - [Autres traitements - document non XML](#)

You can then proceed to the document publishing. This publication can be done with the help of the SDX platform (**to install the publishing module, please refer to the site www.cyberdocs.org/**)

3.2.3. Convert a document without the management module: in command lines

The Cyberdocs conversion module performs an overall conversion task quite complex because of input documents variability requires great flexibility in treatments to be made, which is finally complex. That is why the task of overall conversion has been divided into several stages or operations which are explained here.

Well understanding these operations is crucial for the conversion operator, because even though we try to avoid it, it may be necessary to intervene manually on a document produced at either of the stages.

3.2.3.1. Operations and conversion steps

Hat : All operations done by the conversion module, explained step by step.

A more detailed way to present the conversion is to elaborate the processed steps. In the list below, the code name which figures here is linked to the ANT task that allows the execution of this step only. Folders mentioned are always created inside the working folder of a document.

step_oo

This step will allow the conversion of the source file in an XML file thanks to the OpenOffice.org tool. The folder oo is created and contain the result of this process.

step_01

This step will recover the output of the precedent step and make a first processing which allows to well identify what will be processed or excluded. That way, we'll give the right styles names to the elements, verify that the mandatory styles are indeed present, and exclude OpenOffice.org non processed elements. The result is sent in the folder prod/01.

step_02

This step will recover the output of the prior step and transform every element in TEILITE element, according to its associated name or style. The product of this operation, prod/02, is then treated by the **step_02-bis** which simply numbered each element in order to regroup them afterwards. The result folder is placed in the folder prod/02-bis.

step_03

This step will recover the output of the prior step and organise into a hierarchy the element according to the title levels. The folder prod/03 contains the result of this step.

step_04

At least, this step will transform the output of the prior step into a TEI document, and generate de final XML document in the XML folder.

Once the **step_04** done, we have a TEILITE XML document which will be used as a reference; It's in a way the true product of this conversion component. However, processing can continue, in order to validate this document or obtain derived products. The below steps described make those processing.

Metadata

This task allows creating a metadata file in text format which can be recovered by the management module. The file created is in the folder xml.

HTML

This task enables to generate two statics HTML versions of the document. The first one, in the HTML folder, is a version that can be used in any kind of context, for a static publication. The second one, same as the first one but without a dynamic table of contents, is a print use version, and it will be used by the publishing module. It is in the folder html/sdx.

All this operation is done a second time, to produce two XHTML format versions. These versions are utterly in phase with the W3C standards, with, notably the mathematics equations included in MathML format, and the Mozilla browser allows browsing them easily.

pdf

This task enables to generate a PDF version of the document, which will be placed in the pdf folder.

web

This task enables to create the set of the documents which will be used by the publishing component (XML TELITE version, HTML version for printing, PDF version).

Main operations to know

The previous section describes all the conversion steps, but the tools also provide *operations*, which are sequences of tasks allowing to make a treatment, from a point of departure (for example the source document) to a point of arrival (the reference document for example). The file bin/taches.xml in the conversion module installation folder (or file src/oo-to-tei/bin/taches.xml in Cyberdocs sources) is the ultimate reference for conversion operations (and also steps), but we resume the key information here. At the beginning of the item, in bold type, is the name of the ANT task that performs the operation.

Tout : Complete treatment, from the source to the reference document, as well as derivative products.

Reference document production:

- **document-reference** : from the treatment document to the text document
- **depuis-oo** : from the OpenOffice.org document
- **depuis-xslt-01** : from the end of the first step of XSLT transformation
- **depuis-xslt-02** : from the end of the second step of XSLT transformation
- **depuis-xslt-03** : from the end of the third step of XSLT transformation
- **depuis-xslt-04** : from the end of the fourth step of XSLT transformation
- **validation** : validation of the reference document

From the reference document:

- **produits-derives** : generating all derivative products
- **metadonnees** : generating a minimal metadata file in text format
- **html** : generating static HTML and XHTML versions
- **pdf** : generating PDF version
- **web** : production of all files necessary for the publishing module

3.2.3.2. Preparing a working space

Hat: Folders and documents to prepare before applying the conversion module operations on a document.

In order to simplify the number of parameters in a conversion operation, the conversion module has been programmed according to a specific directory structure. In addition, the documents it will produce will always be located in the same directories compared to the current working directory of a document. We will explore here the general organization of the production folder then the specific organization of a document folder.

1. Organization of the production folder

The production folder is a folder named "production" in the conversion module installation folder. The folder will contain all the documents to be converted, as well as all the documents from the conversion process.

The organization of this conversion folder follows an immutable logic: the documents are organized on three levels, the first being an institution, the second being the publishing year, and the third a document to be processed. This relatively rigid organization could evolve in the future.

Example 1. Simplify content of a production folder

```
production up.bat up.sh lyon2 lyon2.css 2000 dieng_sa.bat
           dieng_sa.sh dieng_sa sources dieng_sa-md.xml dieng_sa.css dieng_sa.doc images
           images-dieng_sa.zip
```

The above example shows us a part of the production folder as it is when installing the production module. We show in this case files and folders relating to a single document. Directly in the production folder is a script `up.{bat|sh}` which will call the startup script conversion module.

Foremost, we notice a first level, in the case "lyon2", which corresponds to the **institution code** of the document editor. This folder contains a CSS style sheet that is specific; this stylesheet must be present, even if you have to copy the one provided with the platform. The filename should be the code of the institution with the extension `.css`.

At the second level, there are a folder 2000, which matches this **production year**. There is also a script `dieng_sa.{bat|sh}` which allow starting the document conversion whose code is the same as the name of the script.

At the third level, there is a folder `dieng_sa` which is the **code of the document** to be processed. In this folder is a sources folder where has to be the source word processor to convert. If this document has illustrations, this should be in the subfolder images, or united in a ZIP file or directly in this folder.

When one or the other of conversion step had been executed, other folders and files would be added, but they will all be situated inside the document folder, as to say the folder `dieng_sa` in the above example.

2. What to do before treating a new document

When you want to treat a new document, the following operations has to be done:

- If it's a document coming from a new institution, create a folder for that institution in the "production" folder, and inside this one, add a CSS style sheet with the same name that the folder but with a `.css` extension.
- If it's a new publishing year, create a folder for that year within the folder of the institution publishing the document.

In the year of publication folder, create a folder whose name will be the identifier of the document throughout his transition in the Cyberdocs platform. This identifier is very important, and it should not contain special characters (accents, punctuation, and so on.).

In the document folder, create a folder "sources" and deposit there the word processor document that will be converted.

- If you have metadata for the publishing, place a file [document identifier]-md.xml in the sources folder.
- If there are external illustration in the document, create an images folder inside the sources folder and place the illustrations there.

In the publishing year folder, copy execution scripts given with the example (for example test.{bat|sh}) and modify them by adding the right parameters, in particular the document identifier which had to be the second parameter. For that matter, a later section gives more explanations.

We remind you that if you have installed the management module, these manipulations are greatly simplified, as you only have to ask the creation of a working space for a document and then upload there the sources files.

3.2.3.3. Prepare a document

hat: Description and role of the different styles used in the documents.

The Cyberdocs platform is able to handle any word processor document. However, in order to make the tools more interesting, a documents preparation phase can be used. This phase consists essentially to apply styles to the various paragraphs of the document, and somehow clean up their content.

For Cyberdocs, style is identified by a *code* and not by a name. It is possible to use different style sheets in the word processor, while obtaining adequate treatment with the conversion module. To use your own stylesheet, you need to simply modify the file `outils/xslt/utiles/styles.xml` to include the names of your styles for each supported style by Cyberdocs (see list below).

Styles list supported by Cyberdocs

This list gives the code (defined in the file `outils/xslt/utiles/styles.xml`), the XML structure produced by the presence of this style, as well as the name of the corresponding style in the *lyon2* style sheet.

Table 1. Styles list

Stylesheet Code	Corresponding XML element	Corresponding name for the Lyon2 stylesheet
auteur	<docAuthor>	1 Auteur
copyright	<titlePart type="copyright">	1 Copyright
dedicace	<div type="dedicace">	1 Dedicace
depot	<docDate>	1 Depot
dept	<titlePart type="dept">	1 Dept
directeur	<titlePart type="directeur">	1 Directeur
discipline	<titlePart type="discipline">	1 Discipline
ecole-doct	<titlePart type="ED">	1 EcoleDoct
epigraphe	<epigraph>	1 Epigraphe
faculte	<titlePart type="faculte">	1 Faculte
grade	<titlePart type="grade">	1 Grade
jury	<titlePart type="jury">	1 Jury
no-officiel	<titlePart type="Reference">	1 NoOfficiel
sous-titre	<docTitle>/<titlePart type="sub" lang="fr">	1 Sous-titre
titre-front	<div type="***">	1 TitreFront
titre-these	<docTitle>/<titlePart type="main" lang="fr">	1 TitreThese
universite	<titlePart type="univ">	1 Universite
ann-titre	----- -----	3 Ann_titre
ann-titre1**9	<div type="appendix">	3 Ann_titre1**9
bibli-item	<bibl>	3 Bibli_item
bibli-tit	----- -----	3 Bibli_tit
bibli-tit1**4	<div type="bibl">	3 Bibli_tit1**4
citation	<q rend="block">	Citation
citation-bloc1**2	<q rend="block">	CitatioBloc1**2
closer	<closer>	closer
conclu	<div type="conclusion">	Conclu
entree	<list type="gloss">/<item>	Entree
figure	<figure>	Figure
intro	<div type="***">	Intro
legende-fig	<* id="fig">/<head>	LegendeFig
legende-tab	<* id="tab">/<head>	LegendeTab
liste-num	----- -----	ListeNum
liste-num1	<list type="ordered">/<item>	ListeNum1
liste-num2	<list type="ordered">/<item>	ListeNum2
liste-puce	----- -----	ListePuce
liste-puce1**8	<list type="bulleted">/<item>	ListePuce1**8
liste-simple	<list type="simple">/<item>	ListeSimple
liste-titre	<list>/<head>	ListeTitre
partie	<div type="part">	Partie
source	<note place="interlinear">	Source

Stylesheet Code	Corresponding XML element	Corresponding name for the Lyon2 stylesheet
caption	<head>	Caption
heading1**9	<div>/<head>	heading 1**9
image-ligne	<figure>	ImageLigne
image-tab	<figure>	ImageTab

3.2.3.4. Doing a conversion

Hat: Scripts to execute to convert a document, as well as their potential parameters.

The conversion module can be used with a cascade of scripts calling each other. The simplest is to copy an existing example, but we will present here the details in reference.

The cascade can be resumed like this:

- A **specific script to a document** calls a general conversion script passing parameters that determined where are the documents to convert.
- A **general conversion script** calls the ANT starting script specifying the task to execute as well as the environmental variable which correspond to the predecesing script parameters in context.
- The **ANT starting script** will prepare the Java environment by delimiting an appropriate CLASSPATH, then will start a virtual Java machine asking it to run ANT.

As can be seen, only the first script can be changed when dealing with a new document, the other two are generic and do not have to be changed from one document to another. Therefore, the preparation of a document, it is only necessary to copy a script to a specific document and modify it to set up the right parameters.

1. ANT starting script

This script corresponds to the file `outils/bin/oo-vers-tei.{bat|sh}` in the conversion module installation folder. That script will itself call the `ant.{bat|sh}` script which is given with ANT.

If we take a look at the script's main command, we note that it is very simple:

```
call ant -logger org.cyberdocs.ant.HTMLLogger -buildfile
oo-vers-tei.xml %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Essentially, the script defines two important parameters and set up the other parameters it received. These two parameters are:

-logger

This is the type of log (*logging*) that is desired. Indeed, in order to make the management module very intuitive and self-documented, a specific type of logging to the Cyberdocs platform, in HTML format, has been defined. That is the one that is used by default, as we saw in the example above. If one wants to use the type of logging by default (which is to leave messages in standard text format), you can remove the `part-logger org.cyberdocs.ant.HTMLLogger` in this command.

-buildfile

ANT operates from an XML document where all the possible tasks and their dependencies are. This setting allows you to specify it, and in the case of Cyberdocs there is no reason to change it.

Except for changing the type of logging, there is therefore no reason to modify this script.

2. The generic conversion script

This script is the `production/up.{bat|sh}` file situated in the conversion module installation folder. That script will call the ANT starting script with a command like this one:

```
call oo-vers-tei.bat
    "-Ddossier.production=..\production" "-Dnom.fichier=%2"
    "-Dcode.institution=%3" "-Dcode.doc=%4" "-Dcode.styles=%5" "-Dlangue.doc=%6"
    "-Dannee.soutenance=%7" %1
```

In that example, the command line has been cut in several lines to facilitate the reading.

As you can see, this script will send essentially parameters of the form `-D[name of the parameter] = [value of the parameter]`. This syntax is important because it is operated at the start of Java and allows specification of environment variables. These variables can then be used in a ANT script, which is actually the case in `outils/bin/oo-to-tei.xml`. So, if you want to send other parameters to ANT script, or even if you want to redefine the properties used in this script, you can do it in the command line as in the example above, without changing the file `tools/bin/oo-to-tei.xml`. For more information, consult the [documentation of ANT](#).

The meaning of this environment variable will be indirectly explained in the next section.

3. The document specific script

This script is for example the `production/lyon2/2002/essai.{bat|sh}` file. The main command line is similar to this one:

```
call up.bat tout essai.doc lyon2 essai lyon2 fr 2002
```

You can note that the script only call the generic script, setting up seven parameters which are:

- *tout* is one of the conversion module task, as explained in another section or detailed in the `outils/bin/taches.xml` file in the Cyberdocs sources.
- *essai.doc* is the file name of the document to convert, file which has to be in the sources folder.
- *lyon2* is the institution code that published the document, and has to match the name of a parent folder.
- *essai* is the document identifier, which has to match the proper folder name of that document.
- *lyon2* (the second) represents the style sheet code used, and this code has to be defined in the `outils/xslt/utiles/styles.xml` file.
- *fr* is the document main language ISO code.
- *2002* is the document publishing year, which has to match the folder name where the document is.

For the script to properly work, you have to be careful that the arguments order is not modified.

3.2.3.5. Conversion result

Hat: Description of the different documents and files produced during a document conversion.

As a result of a conversion operation, a number of files and folders will be added; these files and folders are found within the document folder, and still outside of the folder containing the sources (sources folder) Here is some information about these files and their content.

- The oo folder contains the conversion result of the word processor document by OpenOffice.org.
- The prod folder itself is separated in 4 sub-folders: 01, 02, 02-bis, 03, containing respectively the conversion result after the first (01), second (02 et 02-bis) and third (03) conversion steps.
- An xml folder containing:
 - o The final document, result of the conversion, in XML format (in our example dieng_sa.xml)
 - o The DTD TEILite associated
 - o A text file with few metadata (in our example dieng_sa-meta-auto.txt)
 - o A notes.xml file containing the document's footnotes, this document is used by the publishing module to optimize the display.
 - o A config.xml file indicating the configuration information for that document, essentially by now, the number of levels in the table of contents.
 - o The XML metadata file located in the sources folder (in our example dieng_sa-md.xml)
 - o A tocTab.js javascript file containing the document's table of contents.
 - o A ressources folder in which are stocked the document's images.
 - o A formules folder containing the formules present in the document in MathML format.
- The html folder contains a static HTML version of the document, starting page being index-frames.html, and printable HTML versions - of the document and its parts - in the sub-folder sdx.
- The xhtml folder contains a static XHTML version of the document, with the same notes as the HTML version.
- The pdf version contains a PDF version of the document and its parts, as well as the XSL-FO files which served to its production.

In all those files, some can be deleted after the conversion, being:

- The prod folder because it contains work files for intermediary conversion steps.
- The .fo files in the pdf folder because they only served for the pdf file production.

3.2.4. Publishing module

The publishing component of the Cyberdocs production line is a dynamic Web application using the SDX platform. The main functionalities are search and consultation of published documents.

In this documentation, we will briefly describe the present functionalities and their main characteristics.

3.2.4.1. Management and users identification

Hat: Management and users identification in SDX and in Cyberdocs publishing module in particular.

The publishing module requires users authentication to allow to add (or *index*) a document. In this section, we present how to create and how to identify users.

In the initial application configuration, there is a user who is part of the admins group: its code is admin, and there is no password. Like all users of this group, he has the right to add or remove documents in the publishing module.

1) Users management

The publishing module presents no features proper to users management. To this end, you'd rather use the SDX administrative interface, which has such features. To get there, go to the URL **sdx/admin/index.xsp** your SDX installation (warning, it can give a URL with twice *sdx*, it is normal!), Then click on the button *Administer* next to Cyberdocs application. You can at this time set up the code and administrator password (admins group), then click on the *Identification* button to manage users accounts.

2) Identification

To identify yourself in Cyberdocs application and get administrator privileges, you have to go to the **identification.xsp** URL of the application.

3.2.4.2. Documents management

Hat: Procedure to index and delete documents in the publishing module.

The Cyberdocs's publishing module is an SDX application. For this search tool, the documents have a meaning only if they are *indexed*, so we are going to talk about documents indexing and deleting.

These functions are of course reserved for application administrators. To implement this restriction, the publishing module uses the users management features available with the SDX platform. To index or delete documents, you must first login as an administrator of the application.

1) Indexing documents

The application administrator, once identified, can upload (indexing) documents so that they are searchable and readable via the SDX application.

For this, both forms are at its disposal, both forms are located in the same page accessible from the **chargement.xsp** URL:

To index documents which are stored on a disk directly accessible by the server

Formulaire d'indexation par institution, année, code de document :

Répertoire de base contenant les documents :

Code de l'institution :

Année de publication :

Identifiant du document :

Document non XML :

In this case, a pre-filled text field shows the conversion module's production folder location. You can keep the value or change it if your documents are elsewhere. You will only need to fill in the code of the institution, the year of publication and the identifier of the document. These information should match subfolders in the folder production.

If the document is not available in XML format (for example if it's in PDF format), you have to tick the **non-xml document** button.

To index documents located on a distant server:

Formulaire d'indexation par URL complète :

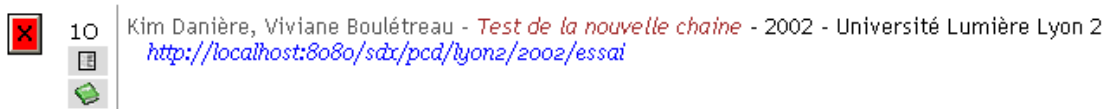
URL du document à indexer :

In that case, you have to type the complete XML document's URL address, for example **<http://myserver.org/my-documents/test/xml/test.xml>**. If it's a document non-available in XML format, you have to give its metadata file's URL, for example: **<http://myserver.org/my-documents/test/xml/test-md.xml>**.

Please note that it is necessary for the document to be located in a folder that follows the structure procedure of the xml folder by the conversion module.

2) Deleting documents

An administrator can delete an indexed document, consequently it cannot be anymore find by a search or displayed. To delete, simply click on the little X that appears next to a document in a search result; meaning you first need to carry out a search by the method that seems most appropriate, to find the document that should be deleted.



Note that the deleting button is only available with the complete documents, not for parts of documents (chapters, for example). However, if one removes a document, all parts are also removed to maintain a certain consistency.

3.2.4.3. Document's metadata use

Hat: Metadata format wanted by the publishing module and treatments done with those information.

The metadata associated with the document are operated by the publishing module, and they are very essential. Regardless of how they are produced (the management module includes a Web interface to capture them), they must be in the form of an XML document and the document must be in a file located in the same folder as the main XML document, and this file must have the same name as the main document, but beginning with -md before the .xml.

Metadata elements are parts of the [Dublin Core Metadata Element Set](#) or of the [Metadata Standard for Electronic Thesis and Dissertations](#) (ETDMS). Other format could eventually be supported.

1) Metadata use

Metadata are operated following different ways by the publishing module:

The detailed summary of the document, which can be displayed in its own window, is created from the metadata.

- The document's summary form, which can be displayed in its own window, is created from metadata.
- Search results are constituted from metadata.
- Most of the advanced search form fields use one or the other metadata.
- Information given by an OAI warehouse are done from metadata.
-

2) Metadata format

The example below is a reference for Cyberdocs's metadata treatments:

Example 1. Métadonnées supported by Cyberdocs

```
<doc
  <!-- some namespaces -->
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:md-cyb="http://cyberdocs.org/md/1.0"
  xmlns:etdms="http://www.ndltd.org/standards/metadata/etdms/1.0/"
>

  <!-- Unique identifier (institution.annee.code_du_document) -->
  <md-cyb:identifiant>lyon2.2002.essai</md-cyb:identifiant>

  <!-- Three parts of the unique identifier -->
  <md-cyb:identifiant-doc>essai</md-cyb:identifiant-doc>
  <md-cyb:annee>2002</md-cyb:annee>
  <md-cyb:code-institution>lyon2</md-cyb:code-institution>

  <!-- Identification of the tool used to produced / publishing the document.
-->
  <md-cyb:plate-forme>Plate-forme
  [http://sourcesup.cru.fr/cybertheses/]</md-cyb:plate-forme>
  CyberThèses

  <!-- Tool Version -->
  <md-cyb:version-plate-forme>1.0</md-cyb:version-plate-forme>

  <!-- Author -->
  <dc:creator>Kim Danière, Viviane Boulétreau</dc:creator>
```

```

<!-- Title in its original language, with its sub-title -->
<dc:title xml:lang="fr">Test de la nouvelle chaine</dc:title>

<!-- Title in an other language -->
<dcterms:alternative xml:lang="en">Test</dcterms:alternative>

<!-- Thesis supervisor -->
<dc:contributor etdms:role="directeur">Sous notre propre
direction</dc:contributor>

<!-- Jury members -->
<dc:contributor etdms:role="jury">Martin Sévigny</dc:contributor>
<dc:contributor etdms:role="jury">Jean-Paul Ducasse</dc:contributor>

<!-- Diploma, academic defense place, etc. -->
<etdms:degree>

  <!-- Diploma -->
  <etdms:level>Thèse pour obtenir le grade de CyberThéséur</etdms:level>

  <!-- Discipline -->
  <etdms:discipline>Discipline : Edition Electronique</etdms:discipline>
  <!-- Academic defense place, in several levels -->
  <etdms:grantor md-cyb:role="universite">Université Lumière Lyon
2</etdms:grantor>
  <etdms:grantor md-cyb:role="ecole-doctorale">Ecole Doctorale
tagada</etdms:grantor>
  <etdms:grantor md-cyb:role="faculte">Division ERAD</etdms:grantor>
  <etdms:grantor md-cyb:role="departement">Cellule Edition
Electronique</etdms:grantor>
</etdms:degree>

<!-- La date de soutenance -->
<dcterms:dateAccepted>2002-11-27</dcterms:dateAccepted>

<!-- Summaries -->
<dcterms:abstract xml:lang="fr">Un résumé en français</dcterms:abstract>
<dcterms:abstract xml:lang="en">An English abstract</dcterms:abstract>

<!-- Key words -->
<dc:subject xml:lang="fr">[Test ; conversion ; cyberthèses]</dc:subject>

<!-- Editor -->
<dc:publisher>CyberThèses</dc:publisher>

<!-- Format -->
<dc:format>text/xml</dc:format>

<!-- Language -->
<dc:language>fr</dc:language>

<!-- Differents rights -->
<dc:right md-cyb:role="copyright">[Copyright 2002 CyberThèses - Université
Lumière Lyon 2</dc:right>

<!-- Available values for the publishing : nothing or [intranet] -->
<dc:right md-cyb:role="diffusion"></dc:right>
<dc:right md-cyb:role="commentaires-diffusion"> [Cette thèse est en accès
restreint pour
protéger la vie privée des personnes qui sont mentionnées]</dc:right>

```

```

<!-- Reference URL -->
<dc:identifier>http://localhost:8080/sdx/pcd/lyon2/2002/essai</dc:identifier>

<!-- Temporal's key words -->
<dcterms:temporal xml:lang="fr"/>

<!-- Geographical key words -->
<dcterms:spatial xml:lang="fr"/>

<!-- File size (in octets) -->
<dcterms:extent>[13416]</dcterms:extent>
</doc>

```

3.2.4.4. Consulting a document

Hat: Document's consultation window informations Informations and their different options.

The Cyberdocs' publishing module includes an advisory function relatively sophisticated, particularly suited to the consultation document lengthy and structured, such as doctoral dissertations. We will explain the principles behind this method of consultation, as well as the various functions associated with the consultation.

1) Documents representation types

In the Cyberdocs' publishing module, these different documents consultation types are available:

- A brief format, used in the search results, contains mains bibliographical coordinates of a document (title, author, publishing date, etc.).
- Document's metadata, displayed in a specific window under a table format.
- Printable versions of the documents.
- Structured consultation window of the document and its part.

In this part of the documentation, it is only the latter method that interests us and for other aspects you are invited to follow the respective links to view other parts of the documentation.

2) Structured consultation window

The formal consultation window is divided into three frames. The upper frame presents major bibliographics information of material, information on the research done to get to this document, as well as a simple search form.

Please note that this is a window specific to this feature in the consultation module. So, when a search is performed and we want to consult a document, it appears in a window different from the results. However, all documents are viewed in the same window, in order to avoid a multiplicity of open windows.

2.1) Bibliographics information frame

The information given in this frame comes from the document's associated metadata.

The frame also offers search and browse fonctionnalités in the shown document. First, words searched are highlighted in the shown document. If you want to go through one searched word's occurrence to an other, you simply have to use the two provided buttons:



Then you can also do a search in the document or one of its part using the search form you can find in this frame:

A search form with the label "Mot(s) recherché(s)" on the left. To its right is a text input field containing the text "sdxall:1". To the right of the input field is a button labeled "Envoyer".

This will display a list result in the frame of the document, results that will only contains parts of the document you are consulting.

2.2) Table of contents' frame

The table of contents' frame is located in the left part of the consultation window. It offer a dynamic table of contents which allow to select quickly, by folding or unfolding parts, document's section we would like to consult.

Usually, three different tables of contents are shown:

1. A table of contents constructed from documents' titles.
2. An Illustration table, construted from illustrations' titles.
3. A tables list, constructed from table's titles.

2.3) Document's frame

The main frame, located in the right side of the window, allows reading the document. Note that if you click on the icon situated at the beginning of a title, the table of contents will be sinchronized, meaning that this title will be displayed in the table of contents' frame.

3.2.4.5. Search

Hat: Different search mode, way of using them, kind of outcomes.

Research is a central feature in the publishing module. It exists basically four ways to carry out researches, which we present below. Regardless of the ways of making them, the search results are always in the same form.

1) Simple search

The simple search function is the one which consist to use the present text zone at the top of each page to express a search request. Most of the time, this request will be set up with one or more words, but a more [developped query language](#) is, in fact, available. For example, if you process the query "*Africa farming*", both words won't be mandatory (OR implicit Boolean) but if you process the query "*Africa+farming*", the word *farming* will be compulsory. When we do such research, the search zone is the documents integral text.

When doing such searches, the search zone is the integral text of the documents.

2) Advance search

The advanced search function is reachable by clicking on the link "*Advanced search*" in the upper headpiece of each page. We yet obtain a form which allows to type several search criterions and to link them by Boolean operator "AND" or "OR".

The first zones of the form allow to type words and find them in the main documents metadata. Then, the following zones allow to search in precise index.

3) Search by lists

The search by lists consists in searching document while obtaining a list according to certain kind of criterions. By now, the only criterion available is *Universities and Institutions*, especially adapted for thesis. To activate this list, you have to choose this entry in the drop-down menu called *Choose an index*, present in the upper part of all pages.

The result will be a universities and institutions list, and for each documents that are published.

4) Search in a document

When consulting a document, you have the possibility to do a search within the document only. To do that, you only have to use the text zone on the upper frame of the document consulting window. You can do searches the same way as for the simple search presented above.

Results will be displayed in the main consultation frame of the document.

3.2.4.6. Search results

Hat: Information about the search results presentation page, browsing in those results, etc.

The search results are always represented the same way, whether you came from the simple search, the advanced search, lists or the search in one document.

For each search result, a brief summary of the document is listed, including the following information:

- Document author.
- The title of the document and, eventually, subheading.
- Publishing year.
- The publisher (for example the University for the thesis)
- Document address.

This last information, the address, is clickable to access to the consultation window.

Then, on the left of this brief presentation, we find the number of this result in the list, with a link to consult the card (metadata) and an other to consult the document.

3.2.4.7. Printable versions

Hat: How to identify document's printable versions, and how to get them on the screen.

You can have, from the publishing module, printable version of the documents you are consulting. To do that, you have to go on the document's consultation window, and then click on the print shape icon on the right corner of the window. You then get a window that allows you to choose the complete document or the part of the document to send, as well as the format (HTML or PDF)

There's only to use the print possibilities of your browser and your working station.

3.3. Help for programmers

3.3.1. Bugs reporting

Any bug report is of course welcome!

For that, we would like that the talkings go through disussion boards. You don't have to be a confirmed programmer to report bugs: problems in the documentation, installation problems or document conversion troubles, dead links on our website... improving Cyberdocs cannot be done without its users.

3.3.2. Contributions to Cyberdocs

To contribute to the platform developpment, you should present yourself to the programmer's community by adding your name to the discussion boards [cybertheses-dev](#). An [online form](#) allows you to simply do it.

3.4. Downloading Cyberdocs

For downloading instructions, report to the "download" page.

4. Downloading

Cyberdocs is a free software that can be downloaded on Internet. For those who cannot wait, there are the two essential link to download the platform, a distributed version or the latest sources:

- Downloading [Cyberdocs' official distribution](#)
- Get back the [latest sources with Subversion](#)

4.1. Cyberdocs' distributions

The collaborative development site [Sourcesup](#) hosts [Cyberdocs project](#) and propose at the same time a page where you can download the different distributions of the platform. This page is available from the *File* tab on the site Sourcesup.

This download page offers different versions that are still available for download. For each version, different formats will be available, or different packages.

In all cases, it is recommended that you install the latest version: support and documentation of the older versions are no longer insured.

4.2. Getting back the sources with Subversion

Subversion is a versioning sources management software. The developers of the Cyberdocs' project use this tool to facilitate collaboration and healthy evolution of various versions of the platform. Beyond that developing use, any user interested in the latest sources of Cyberdocs can do this using *client* software of Subversion.

In this page, we provide some information about Subversion, as well as the parameters to be used to check the Cyberdocs sources, and finally an example with TortoiseSVN software under Windows.

About Subversion

To learn more about this tool, we suggest that you read the [matching article in Wikipedia](#). You will find an explanation of the tool's origins, on the main commands available, as well as links to various software allowing to work with Subversion.

We also suggest that you read a [help page on Subversion](#) maintained by the Universities Network Committee (CRU, France), where the Cyberdocs sources are hosted.

If you only want to download Cyberdocs sources, you can directly go to the following sections that provide all necessary information.

Cyberdocs' Subversion parameters

On the Sourcesup site where Cyberdocs sources are hosted, the tab Subversion provides the necessary information to access Cyberdocs sources via Subversion. For a limited access mode, which allows only grab the source, and not to modify them, you must use the following address with your client software of Subversion:

checkout <http://subversion.cru.fr/cyberdocs>

Depending on the tool you are using, the exact procedure will be different. For example if you are working in command line, the command will be:

```
svn checkout http://subversion.cru.fr/cyberdocs [local_folder]
```

Where [local_folder] is the folder's path where you want to copy the sources on your computer.

Using TortoiseSVN under Windows

The [TortoiseSVN](#) software is a tool that integrates with Windows Explorer, which allows you to easily use all the Subversion's features in graphical mode. Here is the procedure to install TortoiseSVN then to download the Cyberdocs sources.

Downloading and installing TortoiseSVN

A [downloading page](#) allows you to download the latest version. You have to download the installer.

The installation is simple from the installer. Please note that you will have to restart your computer to benefit from the specific icons installed by TortoiseSVN.

Downloading Cyberdocs with TortoiseSVN

We will consider that you get the source for the first time. First, you have to create a new folder on your computer, which will contain the Cyberdocs sources. Once done, with your Windows Explorer, click on the folder with the right mouse button and get a menu like this:

(Image)

On the menu that appears, click on the entry SVN Checkout ... And get a window like this:

In the *referential URL* zone, you have to choose the following value:

<http://subversion.cru.fr/cyberdocs/trunk>

In the *Extraction directory* area, the path to the folder where you clicked will be recorded and you can keep it. For other parameters, default values are correct, and are illustrated in the previous image.

Once you gave the proper information, simply click the *OK* button and the sources recovery will start. A window permanently tells you where you stand.

Update sources with TortoiseSVN

If you already have recovered Cyberdocs sources and want to update them (for example, if developers have made changes), you can use Windows Explorer to start TortoiseSVN, a bit like when retrieving original sources.

With Explorer, open the folder where Cyberdocs sources are. Then click with the right mouse button, and get a context menu like this:

(Images)

This time, in the context menu, select the entry *SVN Update* and TortoiseSVN will be checking on the server if new sources are available (additions or modifications) and it will download them.

5. Lists

The collaboration between the various participants in Cyberdocs or Cyberthèses is done through several discussion lists. All these lists are hosted by the CRU, they work with the multilingual software [Sympa](#), archiving is done by the software [MhOnArc](#). **The mailing lists are the best way to get in touch with the Cyberthèses or Cyberdocs communities**, and all questions or comments should first go through one or the other of these lists.

All lists related to Cyberdocs and Cyberthèses are accessible from a [single page](#) run by Sympa. By accessing the home page of a list, you can perform different actions, particularly subscribe, unsubscribe and consult the archives of the list. **Please note that it is strongly advised to consult the archives of a list before asking questions.**

The rest of this page will explain the role and procedures specific to each discussions list associated with the project. We invite you to read the specific goals of each list before sending a message, to make sure that you choose the right list, the more relevant to the question or comment to make.

1) cybertheses-users

This list is reserved to the platform's users. It addressed all the issues relating to the processing of documents by the platform Cyberdocs: stylage, treatment, understanding error messages, and so on.

[List homepage](#)

[List subscription](#)

[Unsubscribe the list](#)

[Consulting list's archives](#)

Send a message on the list: cybertheses-users@cru.fr

2) cybertheses-tech

This mailing list is destined to people in charge of the installation, management and maintain a Cyberdocs platform, and therefore are likely be interested by its technical surrounds: how is works, how customize or adapt it, how to solve installation problem or parameter problems, etc.

[List homepage](#)

[Subscription](#)

[Unsubscribe](#)

[Consulting the list's archives](#)

Send a message on the list: cybertheses-tech@cru.fr

3) cybertheses-dev

This list is available in consultation to anyone; the content is essentially technical, since it is here that people actively participating to Cyberdocs developments, help each other, exchange information, tips and tricks, tips programs, and so on. Subscription is possible, but it must first be authorized by one of the project managers.

[List homepage](#)

[Subscription](#)

[Unsubscribe](#)

[Consulting the list's archives](#)

Send a message on the list: cybertheses-dev@cru.fr

4) cybertheses-latex

This group aims the integration of the [LXir](#) tool which allows a LaTeX entry to the conversion chain of the Cyberdocs platform, in particular for the Cyberthèses project.

[List homepage](#)

[Subscription](#)

[Unsubscribe](#)

[Consulting the list's archives](#)

Send a message on the list: cybertheses-latex@cru.fr

5) cybertheses-multilingue

This working group animated by Gabriela Ortúzar of the Universidad de Chile, will address the issue of improving access to CyberThèses resources through integration with multilingual databases authorities.

[List homepage](#)

[Subscription](#)

[Unsubscribe](#)

[Consulting the list's archives](#)

Send a message on the list: cybertheses-multilingue@cru.fr

6. FAQ

See: <http://www.cyberdocs.org/en/faq>

7. Contacts

Service Général de la Recherche et des Ecoles doctorales - ERAD
86 rue Pasteur
69007 Lyon
FRANCE

For any information, please write to: cybertheses@univ-lyon2.fr