

OPUS 4.0.2

Manual

Version 1.4 (State: 21.02.2011)

by Doreen Thiede

translated by Sybille Mattrisch

Table of Contents

Part I Introduction - What is OPUS?	8
1 Standards.....	8
2 Bibliography Function.....	8
Part II Terms and Functions in OPUS	10
1 Start Page.....	10
2 Search.....	10
3 Browsing.....	10
Most recently published documents in the repository	10
Document Types	10
Collections	10
4 Publishing.....	11
5 FAQ Page.....	11
Part III Installation Requirements	14
Part IV Installation via Package Management	16
1 Installation.....	16
2 Deinstallation.....	17
Part V Installation without Package Management (with install script)	20
1 Download and unpack the tarball.....	20
2 Ubuntu.....	22
PHP	22
MySQL	22
Java	22
Apache	22
Execute Installation Script	23
3 openSUSE.....	24
PHP	24
MySQL	24
Java	25
Apache	25
Execute Installation Script	25
Part VI Manual Installation	28
1 Create config.ini.....	28
2 Installation of the Required Libraries.....	28
ZEND Framework	28

Solr PHP Client	29
JpGraph	29
jQuery JavaScript Framework	29
3 Database.....	29
4 Delivery of files with access protection.....	30
5 Configure Apache.....	33
6 Installation and Configuration of the Solr Server.....	34

Part VII Configuration 38

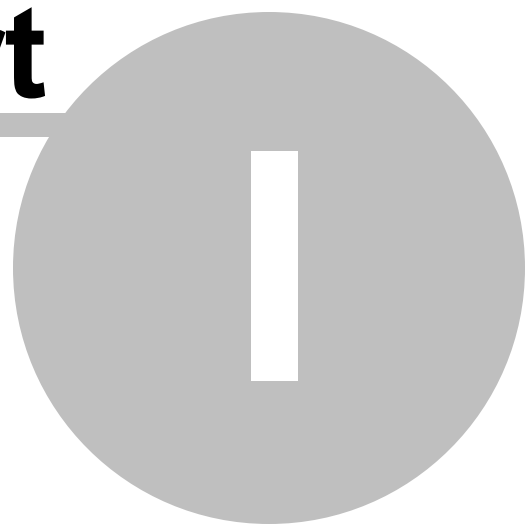
1 Mail Options.....	38
2 Referees.....	38
3 Layout/Theme.....	38
4 Start Module.....	38
5 Search Engine.....	38
6 URNs.....	39
7 Document Types.....	39
8 Formal Settings.....	40
9 Full Text Directory.....	40
10 OAI	40
11 Supported Languages.....	41
12 FAQ Page.....	41
13 How to create new document types.....	41
XML Document Type Definitions	41
Templates	43
14 How to create new fields.....	44
15 Configuration of the Start Page.....	47
16 Configuration of the Contact Page.....	47
17 Configuration of the Imprint Page.....	47
18 Configuration of the FAQ Page.....	47
Guidelines	48
19 Configuration of the Search Interface.....	48
20 Configuration of Mouse-Over Texts.....	49

Part VIII Administration 52

1 Account.....	52
2 Clearance.....	52
3 Access Control.....	52
User Roles	52
User Accounts	52
IP Address Ranges	52
4 Adminstrate Documents.....	53
5 Adminstrate Licenses.....	53

6	Administrate Collections.....	53
7	Administrate Languages.....	54
8	Show Publication Statistics.....	54
9	Show OAI Link.....	55
10	Institutions (Distributing Institutions).....	55
Part IX	Migration OPUS 3.2 to OPUS 4	58
1	Preparation.....	58
2	Migration.....	58
3	Troubleshooting.....	59
Part X	Attachment	62
1	Document types.....	62
	(Scientific) Article	62
	Bachelor's Thesis	62
	Book	62
	Part of a Book	62
	Conference Object	62
	Contribution to a (non-scientific) Periodical	62
	Course Material	62
	Doctoral Thesis	62
	Habilitation	63
	Image	63
	Lecture	63
	Master's Thesis	63
	Misc	63
	Moving Image	63
	Preprint	63
	Report	63
	Review	63
	Sound	63
	Study Thesis	64
	Working Paper	64
2	Meaning of the fields.....	64
3	Mapping of Document Types from Opus3 to Opus4.....	66

Part



1 Introduction - What is OPUS?

OPUS is an open-source software under the GNU General Public License for the operation of institutional document servers and/or repositories. OPUS is an acronym for Online Publikationsverbund Universität Stuttgart and was developed at the computing center of the Stuttgart university library at the end of the 1990s. Since then OPUS has been further developed in cooperation with national partners.

As freely available software OPUS is based on PHP and MySQL and can therefore be combined with other freely available components. In this way, OPUS 4 makes it possible to archive electronic documents, make them available to users, to search them, to browse and to simplify the publication process.

1.1 Standards

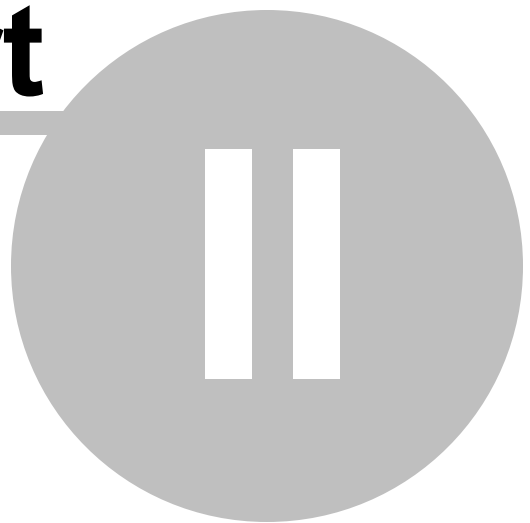
For the preparation of document types in OPUS4, the standard "Gemeinsames Vokabular für Publikations- und Dokumenttypen" (DINI AG Elektronisches Publizieren, DNB, BSZ, DINI Schriften 12-de, Version 1.0, Juni 2010) URL: <http://nbn-resolving.de/urn:nbn:de:kobv:11-100109998> was used as the basis. Document types delivered are defined in accordance with this standard (see appendix). This standard is being further developed and makes it possible for operators of repositories to define their own document types, if these are mapped to the corresponding DINI document types when being delivered via an OAI interface. The aim is a better standardization of the exchange of metadata in the area of document and publication types in institutional and specialized repositories.

Furthermore, for the delivery of OPUS documents via the OAI interface, the Metadata Core Set Definitions according to "Lieferung von Metadaten für Netzpublikationen an die Deutsche Nationalbibliothek" (Version 1.0, 30 November 2009) URL: http://www.d-nb.de/netzpub/ablieferung/pdf/metadaten_kernset_definitionen.pdf as well as "XMetaDissPlus - Format des Metadatensatzes der Deutschen Nationalbibliothek für Online-Hochschulschriften inklusive Angaben zum Autor (XMetaPers)" (DNB, Version 2.0, 4 June 2010) http://www.d-nb.de/standards/pdf/ref_xmetadissplus_v2-0.pdf were used in order to build a foundation for the standardized delivery to the DNB.

1.2 Bibliography Function

Bibliography function means the possibility to account for all publications of an institution within a certain period of time via OPUS, regardless of whether these are electronic publications and whether a full text is available or not. In principle, all documents are part of the repository. In addition, there may be documents that are also part of the bibliography. This is defined by putting a checkmark when being asked 'Add to bibliography?' in the publication process of a document.

Part



2 Terms and Functions in OPUS

The vocabulary used for functions and areas is sometimes specific for OPUS. At this point, all functions that may be accessed in the application are therefore introduced. Specific terms are defined and explained accordingly.

2.1 Start Page

On the start page you have the opportunity to create your own welcome text.

2.2 Search

The search in OPUS4 is realized via Apache Solr. This means that all common functions users are familiar with from other search engines are available, e.g. phrase search, truncation and a simple search in all relevant fields.

Simple Search

A simple search is conducted in titles, abstracts, authors, full texts (if available) and keywords (SWD or free) of the documents.

Advanced Search

The advanced search makes it possible to search specifically for authors, titles, reviewers, abstracts, full texts and years. Search fields can be combined with each other ('all words', 'at least one word', 'none of these words').

2.3 Browsing

The purpose of the 'Browse' function is to get an overview of documents according to specific selection criteria, e.g. document type or collection.

2.3.1 Most recently published documents in the repository

For a quick overview, this function shows the most recently published documents in the repository.

2.3.2 Document Types

This function makes it possible to browse for documents according to document type. In its standard version, OPUS4 is delivered with the document types described in the annex and formulated in the description language XML. These document types may be changed and new document types may be defined.

2.3.3 Collections

In OPUS4, hierarchal systems such as institutions (e.g. organizational charts of universities, working groups, research departments) and classifications are administrated via collections. With OPUS4, the following classifications are delivered as standard: DDC, CCS, PACS, JEL, MSC and BKL. These collections are accessible via the Browse function and can be created and edited in the administration section. Their configuration is described below under "[Administrate collections](#)".

2.4 Publishing

Apart from the search and browse functions, the publication of documents is the third central function in OPUS. The basis for the publication of documents are the document types. The standard version of OPUS4 contains document types that are always subordinated to other document types (e.g. “part of a book” and “book (monograph)”). However, this type of relationship is not functionally supported in the current version, i.e., there is, for instance, no possibility to functionally allocate documents that have been entered into the repository as “part of a book” to the corresponding parent document “book (monograph)”. If you want to make reference to a parent document within a document type, you can use the “TitleParent” field for this purpose. This makes the content of this field searchable, i.e., a search in all fields or a title search for the relevant parent document would also deliver documents as a result that only reference the parent document.

Documents are entered into OPUS via a 3-step form. The first step is to select the document type and to upload (optionally) the corresponding file(s). In addition, the document may be added to the bibliography by putting a checkmark in the relevant checkbox.

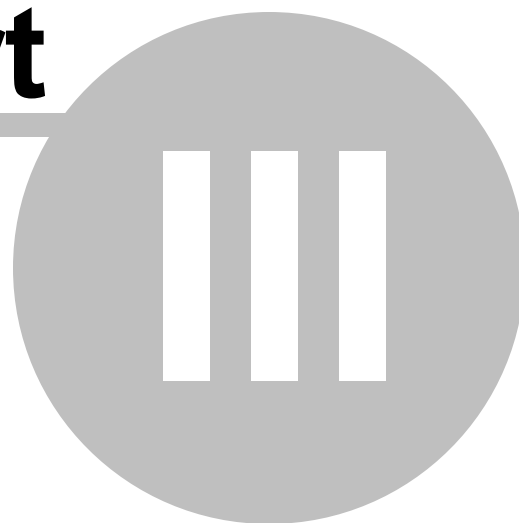
The second step is to enter the metadata on the basis of the selected document type. There are mandatory and optional fields. Mandatory fields are marked with an asterisk. If all mandatory fields have been filled, the form data are validated by clicking the send button. If an error occurred, you will be redirected to the form in order to correct possible errors.

If all entries are correct, step 3 gives you the opportunity to check your entries once again and to correct them by using the back button, if necessary. Furthermore, you have the possibility (optionally) to allocate the document to a collection. By clicking on the save button the document is saved on the document server as “unpublished”. Following examination by the administrator, it then has to be activated by (an) authorized person(s). After this, the document is available as “published”.

2.5 FAQ Page

The FAQ page makes it possible to offer help texts for users about frequently asked questions. OPUS4 already contains a couple of texts that can be adjusted and/or edited. Chapter [Configuring the FAQ Page](#) explains how to conduct such changes.

Part



3 Installation Requirements

The following section describes the installation of OPUS4. The instructions have been formulated for Linux distributions Ubuntu 10.04, Ubuntu 10.10 and OpenSuSE 11.3. In principle, the software should also work with other distributions / operating systems. However, that might require additional installation steps not described here.

Applications, frameworks, libraries:

System packages

- MySQL >= 5.1
- Apache 2
- PHP 5.3
- PEAR
- JRE min. 1.5; better 1.6

Additional components

- current, standards-compliant Servlet/JSP container: Solr includes Jetty 6 in its delivery, for other servlet containers (Tomcat 6, Resin, ...) we recommend a look inside the Solr Wiki: <http://wiki.apache.org/solr/SolrInstall>
- Solr 1.4.1

Part

IV

4 Installation via Package Management

If installation is done via package management, the installation directory is **/var/local/opus4**.

4.1 Installation

1. Register OPUS 4 Package Repository:

for Ubuntu 10.04:

```
sudo sh -c "echo 'deb http://opus4.kobv.de/repository/ lucid main' >/etc/apt/
sources.list.d/opus4-lucid.list"
```

for Ubuntu 10.10:

```
sudo sh -c "echo 'deb http://opus4.kobv.de/repository/ maverick main' > /etc/
apt/sources.list.d/opus4-maverick.list"
```

2. Update package list

```
sudo apt-get update
```

3. Install OPUS 4

```
sudo apt-get install opus
```

4. The OPUS installer now prompts for some parameters (default values are always in square brackets; if no input is made, these values are accepted)

OPUS requires a dedicated system account under which Solr will be running. In order to create this account, you will be prompted for some information. System Account Name [opus4]:

New OPUS Database Name [opus400]:

New OPUS Database Admin Name [opus4admin]:

New OPUS Database Admin Password:

New OPUS Database User Name [opus4]:

New OPUS Database User Password:

MySQL DBMS Host [leave blank for using Unix domain sockets]:	#If host and port are left empty, connection to the database server is established via Unix domain sockets (not via TCP).
MySQL DBMS Port [leave blank for using Unix domain sockets]:	

MySQL Root User [root]:

Next you'll be now prompted to enter the root password of your MySQL server
Enter password:

Install and configure Solr server? [Y]:

Solr server port number [8983]:

Install init.d script to start and stop Solr server automatically? [Y]:

Import test data? [Y]:

#6 test documents are imported.

Delete downloads? [N]:

#Deletes downloads of the required
libraries in directory /var/local/
opus4/downloads.

5. End message:

"OPUS 4 is running now! Point your browser to <http://localhost/opus4/>".

6. A search <http://localhost/opus4/solrsearch/index/search/searchtype/all> should deliver 6 matches.

4.2 Deinstallation

1. Start deinstallation:

```
sudo apt-get purge opus
```

2. As during installation, the various parameters (defined during installation) are prompted:

Do you want to continue [Y/n]?

Delete OPUS4 Apache HTTPD config files in /etc/apache2/sites-available/
opus4 [Y]:

Delete OPUS4 Database <opusdb> [Y]:

Delete OPUS4 Database User <opususer> [Y]:

Delete OPUS4 Database Admin User <adminuser> [Y]:

MySQL Root User [root]:

MySQL DBMS Host [leave blank for using Unix domain sockets]:

MySQL DBMS Port [leave blank for using Unix domain sockets]:

Next you'll be now prompted to enter the root password of your MySQL server
Enter password:

Remove OPUS4 instance directory? [N]:

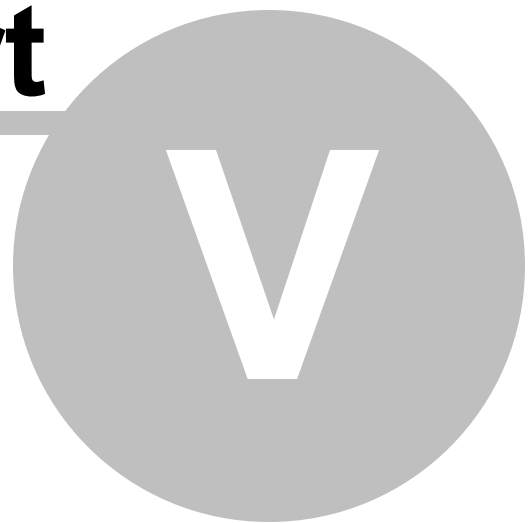
Note: current directory is /var/local/opus4

Remove OPUS4 system account <systemaccount> [Y]:

3. Upon successful deinstallation the message reads:

Deinstallation of OPUS4 completed.

Part



5 Installation without Package Management (with install script)

5.1 Download and unpack the tarball

First, create a directory:

```
mkdir -p $BASEDIR
```

In the following we will assume that **\$BASEDIR** refers to directory **/var/local/opus4**. Therefore, all further text will only contain **\$BASEDIR**. If another directory is to be used, the path must be changed in scripts

- install/install.sh
- install/uninstall.sh
- install/opus4-solr-jetty.conf
- apacheconf/opus4

in order to ensure the correct functioning of the installation routine.

After creating the directory **\$BASEDIR**, the tarball must be downloaded and unpacked:

```
cd $BASEDIR
wget http://opus4.kobv.de/opus-4.0.0.tgz
tar xfvz opus-4.0.0.tgz
```

As a result, you will see the following **directory structure** below **\$BASEDIR**:

```
|-- apacheconf
|-- install
|-- libs
|-- opus4
|   |-- application
|   |   |-- configs
|   |-- db
|   |   |-- createdb.sh.template
|   |   |-- masterdata
|   |   |-- schema
|   |-- import
|   |   |-- importer
|   |   |-- stylesheets
|   |-- library
|   |   |-- Apache -> ../../libs/SolrPhpClient/Apache
|   |   |-- Application
|   |   |-- Controller
|   |   |-- Form
|   |   |-- jpgraph -> ../../libs/jpgraph/src
|   |   |-- Mail
|   |   |-- Opus
|   |   |-- Rewritemap
|   |   |-- Util
|   |   |-- View
|   |   |-- Zend -> ../../libs/ZendFramework/library/Zend
|   |-- modules
|   |   |-- account
|   |   |-- admin
|   |   |-- citationExport
|   |   |-- collections
|   |   |-- default
|   |   |-- frontdoor
|   |   |-- home
|   |   |-- import
|   |   |-- oai
|   |   |-- pkm
|   |   |-- publicationList
|   |   |-- publish
|   |   |-- review
|   |   |-- solrsearch
|   |   |-- statistic
|   |-- public
|   |   |-- htaccess-template
|   |   |-- index.php
|   |   |-- js
|   |   |-- layouts
|   |-- scripts
|   |   |-- common
|   |   |-- cron
|   |   |-- indexing
|   |   |-- install
|   |   |-- SolrIndexBuilder.php
|   |-- workspace -> ../workspace
|-- solrconfig
|-- testdata
|-- workspace
|   |-- cache
|   |-- files
|   |-- log
|   |-- opus.log
|   |-- tmp
```

Now the following access rights for subdirectories must be changed in **workspace**:

```
chmod -R 777 workspace
```

The further steps are now described separately for Ubuntu and openSUSE.

5.2 Ubuntu

The following package installations may be executed via apt-get:

```
sudo apt-get install <packagename> # alternatively: aptitude
```

5.2.1 PHP

It is highly recommended to install at least Version 5.3 of PHP. PHP versions < 5.3 contain bugs that lead to malfunctions in OPUS 4.

Use command `sudo apt-get install` for the following packages:

```
php5
php5-cgi
php5-cli
php5-common
php5-curl
php5-dev
php5-gd
php5-mcrypt
php5-mysql
php5-uuid
php5-xdebug
php5-xsl
php-crypt-gpg
```

Following installation of the PHP5 packages, Apache must be restarted via

```
sudo /etc/init.d/apache2 restart
```

5.2.2 MySQL

It is recommended to install MySQL in a 5.x version; the version currently in use is 5.1.

```
sudo apt-get install mysql-server-5.1
```

5.2.3 Java

To be able to excute Solr / Jetty, a current Java Runtime Environment (JRE) must be installed:

```
sudo apt-get install openjdk-6-jdk
```

5.2.4 Apache

The version currently in use and therefore recommended is Apache 2.2.

Installation:

```
sudo apt-get install apache2
sudo vi /etc/apache2/apache2.conf
# adjust ServerName here
```

Activation of the required Apache modules

In order to activate Apache modules **rewrite**, **proxy**, **proxy_http** and **php5**, the following command must be executed:

```
sudo a2enmod rewrite proxy proxy_http php5
```

Configuration of Apache

The tarball already contains an Apache configuration file under **\$BASEDIR/apacheconf/opus4**. This can be activated as follows:

```
cd /etc/apache2/sites-available
sudo ln -s $BASEDIR/apacheconf/opus4 opus4
sudo a2ensite opus4
sudo /etc/init.d/apache2 restart
```

5.2.5 Execute Installation Script

As a final step, the installation script under **\$BASEDIR/install** must now be executed:

```
cd install
sudo ./install.sh
```

The OPUS installer now prompts some parameters (default values are always in square brackets; if no entry is made, these values are used)

OPUS requires a dedicated system account under which Solr will be running. In order to create this account, you will be prompted for some information. System Account Name [opus4]:

New OPUS Database Name [opus400]:

New OPUS Database Admin Name [opus4admin]:

New OPUS Database Admin Password:

New OPUS Database User Name [opus4]:

New OPUS Database User Password:

MySQL DBMS Host [leave blank for using Unix domain sockets]: # If host and port are left empty, connection to the database server is established via Unix domain sockets (not via TCP).

MySQL DBMS Port [leave blank for using Unix domain sockets]:

MySQL Root User [root]:

Next you'll be now prompted to enter the root password of your MySQL server
Enter password:

Install and configure Solr server? [Y]:

Solr server port number [8983]:

Install init.d script to start and stop Solr server automatically? [Y]:

```

Import test data? [Y]:                #6 test documents are imported.
Delete downloads? [N]:                #Deletes downloads of the required
                                      libraries in directory /var/local/
                                      opus4/downloads.

```

End message: "OPUS 4 is running now! Point your browser to <http://localhost/opus4/>".

A search <http://localhost/opus4/solrsearch/index/search/searchtype/all> should produce 6 matches.

5.3 openSUSE

The following package installations can be executed via zypper:

```
sudo zypper install <packagename>      # alternatively: yast
```

5.3.1 PHP

It is highly recommended to install at least Version 5.3 of PHP. PHP versions < 5.3 contain bugs that lead to malfunctions in OPUS 4.

Use command `sudo zypper install` for the following packages:

```

gcc
make
libuuid-devel
php5
php5-mcrypt
php5-devel          # corresponds to php5-dev
php5-curl
php5-gd
php5-mysql
php5-pear
php5-xsl
apache2-mod_php5    # corresponds to php5-cgi

pecl install xdebug    # installed version 2.1.0
pear install Crypt_GPG # installierte Version 1.1.1
pecl install uuid      # installed version 1.0.2

```

Package **php5-cgi** does not have to be installed separately.

Then save file **xdebug.ini** with the content `extension=xdebug.so` in directory `/etc/php5/conf.d`

Then save file **uuid.ini** with the content `extension=uuid.so` in directory `/etc/php5/conf.d`

Following installation of the PHP5 packages, Apache must be restarted via

```
sudo /etc/init.d/apache2 restart
```

5.3.2 MySQL

It is recommended to install Version 5.x of MySQL, the currently used version is 5.1.

```

install: sudo zypper install mysql-community-server
start server: rcmysql start
set root password: /usr/bin/mysql_secure_installation

```


5.3.3 Java

To be able to execute Solr / Jetty, a current Java Runtime Environment (JRE) must be installed:

```
sudo zypper install java-1_6_0-openjdk
```

5.3.4 Apache

The version currently in use and therefore recommended is 2.2.

```
sudo zypper install apache2
```

Activation of the required Apache modules

In order to activate Apache modules **rewrite**, **proxy**, **proxy_http** and **php5**, the following command must be executed:

```
sudo a2enmod -l          ## lists all modules loaded
sudo a2enmod rewrite
sudo a2enmod proxy
sudo a2enmod proxy_http
sudo a2enmod php5
rcapache2 restart        ## restart Apache
```

Apache Configuration

The tarball already contains an Apache configuration file under **\$BASEDIR/apacheconf/opus4**. This can be activated as follows:

```
cd etc/apache/vhost.d
sudo ln -s $BASEDIR/apacheconf/opus4 opus4.conf
sudo /etc/init.d/apache2 restart
```

5.3.5 Execute Installation Script

As a final step, the installation script under **\$BASEDIR/install** must now be executed:

```
cd install
sudo ./install.sh
```

The OPUS installer now prompts some parameters (default values are always in square brackets; if no entry is made, these values are used)

OPUS requires a dedicated system account under which Solr will be running.
In order to create this account, you will be prompted for some information.
System Account Name [opus4]:

New OPUS Database Name [opus400]:

New OPUS Database Admin Name [opus4admin]:

New OPUS Database Admin Password:

New OPUS Database User Name [opus4]:

New OPUS Database User Password:

MySQL DBMS Host [leave blank for # If host and port are left empty,

```
using Unix domain sockets]:          connection to the database server is
MySQL DBMS Port [leave blank for     established via Unix domain sockets
using Unix domain sockets]:          (not via TCP)
```

```
MySQL Root User [root]:
```

```
Next you'll be now prompted to enter the root password of your MySQL server
Enter password:
```

```
Install and configure Solr server? [Y]:
```

```
Solr server port number [8983]:
```

```
Install init.d script to start and stop Solr server automatically? [Y]:
```

```
Import test data? [Y]:
```

```
#6 test documents are imported.
```

```
Delete downloads? [N]:
```

```
# Deletes downloads of the required
libraries in directory /var/local/
opus4/downloads.
```

```
End message: "OPUS 4 is running now! Point your browser to http://localhost/
opus4/".
```

```
A search http://localhost/opus4/solrsearch/index/search/searchtype/all should produce 6 matches.
```

Part

VI

6 Manual Installation

In the event that the script described in chapter “Execute Installation Script“ cannot or must not be used, the following section describes which steps have to be executed manually.

6.1 Create config.ini

OPUS 4 is delivered with a **config.ini.template**. This file can be found under **\$BASEDIR/application/configs** and should also be saved there (with the extension 'template' being removed from the file name). In **config.ini** the values for local conditions must be defined, without which OPUS 4 cannot be run. All values defined in **config.ini** overwrite the corresponding values in **application.ini**, therefore not only additional, but **all** desired values should always be written into **config.ini**. By default, some values are commented out in the file config.ini.template (by a semicolon at the beginning of the line). If this value is to be used, these semicolons must be deleted in the appropriate places. A semicolon may also be used to insert comments.

6.2 Installation of the Required Libraries

For the installation of Opus4 the following 3rd Party Libraries are required. For licensing reasons, these are not included in the standard delivery but must be downloaded and installed by the user.

Library Name	Website	License
Apache Solr	http://lucene.apache.org/solr	Apache License Version 2.0, January 2004 http://www.apache.org/licenses/
jQuery JavaScript Framework	http://docs.jquery.com/Downloading_jQuery	http://jquery.org/license
jpggraph	http://jpggraph.net/	The Q Public License Version 1.0 http://www.opensource.org/licenses/qtpl.php
solr-php-client	http://code.google.com/p/solr-php-client/	New BSD Licence http://www.opensource.org/licenses/bsd-license.php
ZEND Framework	http://framework.zend.com/download/latest	http://framework.zend.com/license

6.2.1 ZEND Framework

Step 1: Download current version of the Zend Framework: <http://framework.zend.com/download/latest>
Currently used Zend version (minimum is sufficient, full contain documentation etc.)
1.10.6 minimum or 1.10.6 full

Step 2: After download unpack to **\$BASEDIR/libs**

Step 3: Link the content of **\$BASEDIR/libs/ZendFramework-1.xx.xx** to **\$BASEDIR/libs/ZendFramework**:

```
ln -sv ZendFramework-1.xx.xx ZendFramework
```

6.2.2 Solr PHP Client

Step 1: Download library at GoogleCode from the Subversion (SVN). We currently use Revision 36:

```
cd $BASEDIR/libs
svn export -r 36 http://solr-php-client.googlecode.com/svn/trunk/ SolrPhpClient_r36
```

Step 2: Link the content of **SolrPhpClient_r36** to **SolrPhpClient**:

```
ln -sv SolrPhpClient_r36 SolrPhpClient
```

If you experience any problems with the download, the following page will help: <http://code.google.com/p/solr-php-client/source/checkout>

6.2.3 JpGraph

Step 1: Download library from the project page (we currently use Version 3.0.7 (2010-01-11) for PHP5): <http://jpgraph.net/download/>

Step 2: Unpack the downloaded archive to **\$BASEDIR/libs/jpgraph-3.0.7**

Step 3: Link the content of **\$BASEDIR/libs/jpgraph-3.0.7** to **\$BASEDIR/libs/jpgraph**:

```
ln -sv jpgraph-3.0.7 jpgraph
```

6.2.4 jQuery JavaScript Framework

Note: This installation should be made **after** installing OPUS4 because the OPUS4 directory structure must be factored in.

Step 1: Download jQuery library (in its minified variant) under http://docs.jquery.com/Downloading_jQuery in Version 1.4.2 (minimum).

Step 2: Save JavaScript file **jquery-x.x.x.min.js** under **\$BASEDIR/opus4/public/js**

Step 3: Create symlink jquery:

```
ln -sv jquery-x.x.x.min.js jquery.js
```

If you want to work with another jQuery version for a particular theme, the following entry must be made in **config.ini**:

```
; PATH TO JQUERY JAVASCRIPT LIBRARY (relative to public)
javascript.jquery.path = layouts/<theme>/js/jquery-x.x.x.min.js
```

The relevant file **jquery-x.x.x.min.js** must be saved under the directory mentioned.

6.3 Database

1. Create database opus400:

```
create database opus400 default character set = utf8 default collate = utf8_general_ci
```

2. Create user (default is 'opus4admin') for database initialization:

```
create user 'opus4admin'@'localhost' identified by '<passwd>'
grant all privileges on opus400.* to 'opus4admin'@'localhost'
```

3. Create user (default is 'opus4') for Web application:

```
create user 'opus4'@'localhost' identified by '<passwd>'
grant select,insert,update,delete on opus400.* to 'opus4'@'localhost'
```

4. In order to customize database relevant data (user name and password) in the various admin scripts / config files, the following settings are required. The values entered into config files should always be written in single quotes (e.g. `somevar = 'value1'`). Config files in the tarball always have the additional extension `.template` – no changes should be made to these files. Make a copy of the file, remove the extension `.template` in the file name and make any changes in this file (as already described for `config.ini`), e.h.:

```
cp opus4/db/createdb.sh.template opus4/db/createdb.sh
```

5. In order to create an empty database (opus4admin), **\$BASEDIR/db/createdb.sh** must be customized and executed:

```
user=
password=
host=
port=
dbname=
```

6. In **config.ini** the following values (defined above) **must** be entered:

DB SETTINGS

```
db.params.host =
db.params.port =
db.params.username =
db.params.password =
db.params.dbname =
```

#If host and port are left empty, the connection to the database server is established via Unix domain sockets (not via TCP).

SEARCH ENGINE SETTINGS

```
; Indexer
searchengine.index.host = solr.example.org
searchengine.index.port = 8983
searchengine.index.app = solr

; Full text extractor
searchengine.extract.host = solr.example.org
searchengine.extract.port = 8984
searchengine.extract.app = solr
```

7. If MySQL cannot load the INNODB plugin, this is written down in the error log (if error logging is activated in the configuration). After this, MYISAM tables (instead of INNODB tables) are generated automatically. OPUS, however, does not work with MYISAM tables. For instance, there will be problems when deleting documents. It is therefore recommended to check which engine MySQL uses.

6.4 Delivery of files with access protection

OPUS 4 has the ability to access protected files. Via `OPUS_Security` you can check whether the current user (with his/her IP and possibly authenticated via user name/password) is allowed to read a certain file or not. The delivery of the files is executed by the Web server, authorisation, however, is to be assumed by our application. Access protection works only if the configuration does not have the setting `security = 0!`

For delivery in Opus the following mechanism has been realized:

The Web server has an area that may only be read by 127.0.0.1 (currently /files in the standard configuration). Here, for each document there is a directory with the docId, in which all files for this document are saved, e.g. /files/123/document.pdf.

In Apache it is checked by means of Module Rewrite and a RewriteMap from where the requested file is to be loaded. If it is allowed to deliver the file, the RewriteMap prompts Apache to load the document from the protected area (via module_proxy) and to deliver it to the user. If it is not allowed to deliver the file, a file containing a HTTP status code and an error message is loaded from the protected area.

Apache loads the RewriteMap upon start. Zend_Session is required to identify a logged-in user, but can only be initialised once. Therefore, Apache starts a shell script (server/scripts/opus-apache-rewritemap-caller-secure.sh), which starts the Opus_Apache_RewriteMap at every request, transmits the Apache parameters and delivers the output of the RewriteMap back to Apache.

Then use command server/scripts/opus-apache-rewritemap.php, which executes the Zend_Bootstrapping, loads the configuration, parses the Apache parameters and transmits them to the Rewritemap_Apache (server/library/Apache/Rewritemap.php). The Rewritemap now identifies the file that is to be loaded, identifies existing user sessions and IPs and queries Opus_Security_Realm as to the authorization of the current user. After this, the output for the Apache is generated, which will then forward accordingly. Keys in config.ini/application.ini

There are two keys in the configuration that control the Opus delivery mechanism:

- * deliver.target.prefix = /files – The directory for full text documents is then workspace/files/document-id
- * deliver.url.prefix = /documents – The URL for full text documents is then http://.../documents/document-id/dateiname

Both keys are assigned default values in application.ini, however, they can be overwritten in **config.ini**:

```
; deliver.target.prefix = /files  
; deliver.url.prefix = /documents
```

If you make any changes, please also keep them in mind in the Apache config described below.

Required Apache configuration

Modules **mod_rewrite** and **mod_proxy** are required. It is important that **mod_proxy** is configured safely: The global configuration area (in Debian derivatives realized via Include and the file **/etc/apache2/mods-enabled/proxy.conf**) should contain the following:

```
<IfModule mod_proxy.c>  
    #turning ProxyRequests on and allowing proxying from all may allow  
    #spammers to use your proxy to send email.  
    ProxyRequests Off  
  
    #disable proxy for all sites except explicit allowed below.  
    <Proxy *>  
        AddDefaultCharset off  
        Order deny,allow  
        Deny from all  
    </Proxy>  
</IfModule>
```

Note: This type of delivery works safely only when ProxyRequests are deactivated (above realized by the line ProxyRequests Off)!

In order to use a program, module mod_rewrite requires a lock file as RewriteMap that must not be located in the nfs file system. For this purpose, the following code must also be included in the general part of Apache (under Ubuntu: `/etc/apache2/httpd.conf`):

```
<IfModule mod_rewrite.c>
    RewriteLock "/var/run/apache2/opusdeliver-rewrite.lock"
</IfModule>
```

The virtual host, which is responsible for the Opus installation, should contain the following, so that the files directory is protected, access via proxy is possible and the RewriteMap is used. Paths must be adapted as required, 8 blanks must be replaced by a tabulator, so that the fields transmitted to the RewriteMap contain `<Backslash><Tabulator>`! The RewriteMap receives several fields within one line by the Apache. To separate these fields, a character is needed that is definitely not contained in these fields. The tabulator seems to be the best solution here. Unfortunately, the Apache does translate `\t` appropriately; therefore the tabulator must be entered and then escaped, so that the Apache does not interpret it.

```
<VirtualHost *:80>
```

This part has already been included in the OPUS4 installation instructions

```
Alias /opus4-devel "$BASEDIR/opus4/public"
<Directory "$BASEDIR/opus4/public">
    ##
</Directory>

##
## new text to be added
##
Alias /files "$BASEDIR/opus4/workspace/files"
<Directory "$BASEDIR/opus4/workspace/files">
    Options -Indexes
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Directory>

<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteLog $BASEDIR/opus4/workspace/log/rewrite.log
    RewriteLogLevel 0

    RewriteLock $BASEDIR/opus4/workspace/opus-apache-rewritemap-caller.lock

    RewriteMap opus4deliver-devel 'prg:$BASEDIR/opus4/scripts/opus-apache-rewritemap-caller-secure.sh'
    RewriteRule ^/documents/(.*)$ http://127.0.0.1/${opus4deliver-devel:$1\
%{REMOTE_ADDR}\ COOKIES=%{HTTP_COOKIE}} [P]

    <IfModule mod_proxy.c>
        <Proxy http://127.0.0.1/*>
            Order deny,allow
            Allow from all
        </Proxy>
```



```

        </IfModule>
    </IfModule>
</VirtualHost>

```

As explained above, this configuration cannot simply be transferred via copy&paste. In the line starting with RewriteRule, there are two points at which 8 blanks must be replaced by a tabulator each!

In the above configuration, the paths have to be adapted, of course. In addition, the file **\$BASEDIR/opus4/scripts/opus-apache-rewrite-map-caller-secure.sh.template** in **opus-apache-rewrite-map-caller-secure.sh** must be renamed and adapted, if required, e.g. in order to adjust the value of the USER variable – user under which the Apache is run (Ubuntu: www-data; SuSE: wwwrun).

In order to bypass the RewriteMap and, consequently, the access protection, you could try to use the Web server as a proxy and access the path containing the files directly (/files/123/document.pdf). However, ProxyRequests are deactivated and ProxyPass is only set by the RewriteMap. Direct access to the files and bypassing the RewriteMap should therefore not be possible.

6.5 Configure Apache

The tarball already contains an Apache configuration file under **\$BASEDIR/apacheconf/opus4**. The following section describes what it contains and how to create it if access to the existing file is not possible or wanted.

Create file **opus4** for Ubuntu in subdirectory **/etc/apache2/sites-available/** with the following content:

```

<VirtualHost *:80>
    AllowEncodedSlashes On

### if required, separate log files can be created with the following three lines ###
# LogFormat "%h %l %u %t \"%r\" %>s %b" common
# CustomLog /somedir/access_log common
# ErrorLog /somedir/error_log

    Alias /opus4 "$BASEDIR/opus4/public"
    <Directory "$BASEDIR/opus4/public">
        Options FollowSymLinks

        AllowOverride All
        Order deny,allow
        Deny from all
        Allow from 0.0.0.0/0.0.0.0 # Allow global access
### if required, access can be restricted with the following two lines ###
        # Allow from 127.0.0.0/255.0.0.0 ::1/128 # only current computer
        # Allow from 130.73.63.0/255.255.0.0 # institute-wide access (e.g.
ZIB, adaptation to corresponding IP range necessary)
    </Directory>
</VirtualHost>

```

Rename file **opus4/public/htaccess-template** to **opus4/public/.htaccess**. Edit this file and replace

```

RewriteBase <template>
by
RewriteBase /opus4

```

In **.htaccess** the value for APPLICATION_ENV may also be changed, e.g. in order to switch between test and production. If an invalid value is used, the Web pages remain empty and an error message occurs in the Apache2 error.log.

Then activate the Apache module `rewrite_module` ([see above](#)). Now activate the site and restart the Apache to activate the changes:

```
sudo a2ensite opus4
sudo /etc/init.d/apache2 restart
```

Under openSUSE the Apache is configured as follows:

Create file `opus4.conf` in directory `$BASEDIR` with the content described above. Create a symlink under `/etc/apache2/vhosts.d` (Please note: the file must have the suffix.conf)

```
sudo ln -s /home/user/opus4.conf opus4.conf
```

Then restart the Apache:

```
sudo /etc/init.d/apache2 restart
```

6.6 Installation and Configuration of the Solr Server

General

We use Version 1.4.1 of Solr. The jetty supplied serves a servlet container– please find further information under <http://wiki.apache.org/solr/SolrJetty>. Access to the Solr server is done via [Solr PHP Client](#) (currently Revision 36).

Installation Instructions

1. Create solruser on target system `solr.example.org`
2. Log in on the target system as solruser
3. Download Apache Solr 1.4.1: A Mirror Site List is available under <http://www.apache.org/dyn/closer.cgi/lucene/solr/> e.g.

```
cd /home/solruser
mkdir solr_install
cd solr_install
wget ftp://ftp.fu-berlin.de/unix/www/apache//lucene/solr/1.4.1/apache-solr-1.4.1.tgz
```

4. Unpack file `apache-solr-1.4.1.tgz`

```
tar -xzf apache-solr-1.4.1.tgz
```

5. Create a symbolic link

```
ln -sv apache-solr-1.4.1 solr
```

6. Create Solr instance `opus4`

```
cd solr
cp -r example opus4
```

7. Configure logging

1. Copy file `/solrconfig/logging.properties` from the OPUS4 tarball into the instance directory:

```
cd opus4
```

```
cp <path_to_opus4_archive>/solrconfig/logging.properties .
```

2. To adapt the jetty configuration file in directory /home/solruser/solr_install/solr/opus4/etc/jetty.xml: Enter the following definition as child element of Configure:

```
<!-- configure logging -->
<Call class="java.lang.System" name="setProperty">
  <Arg>java.util.logging.config.file</Arg>
  <Arg>logging.properties</Arg>
</Call>
```

8. Copy Solr index schema definition and configuration file from the OPUS4 installation archive into the instance directory

```
cp <path_to_opus4_archive>/solrconfig/solrconfig.xml solr/conf/
cp <path_to_opus4_archive>/solrconfig/schema.xml solr/conf/
```

9. Start Jetty/Solr server

```
java -server -jar start.jar
```

10. Access Admin-Frontend <http://solr.example.org:8983/solr/admin/>

It is possible to use a separate Solr server for full text extraction. For this purpose, execute the steps described above again (starting with item 6). The instance directory to be created in item 6 is **opus4-extract**.

To start a second Jetty/Solr server, the standard port (8993) in the jetty configuration file jetty.xml must be changed. For this purpose, the port entry default="8983" in section

```
<Call name="addConnector">
  <Arg>
    <New class="org.mortbay.jetty.bio.SocketConnector">
      <Set name="port"><SystemProperty name="jetty.port" default="8983"/></Set>
      <Set name="maxIdleTime">50000</Set>
      <Set name="lowResourceMaxIdleTime">1500</Set>
    </New>
  </Arg>
</Call>
```

must be adapted accordingly.

Configuration Opus4

Specify search server in file **opus4/application/configs/config.ini**:

```
; Indexer
searchengine.index.host = solr.example.org
searchengine.index.port = 8983
searchengine.index.app = solr
```

Specify extraction server in file **opus4/application/configs/config.ini**: If no separate Solr server is used for full text extraction, the following entries must be used. Alternatively, the port number should be replaced:

```
; Full text extractor  
searchengine.extract.host = solr.example.org  
searchengine.extract.port = 8984  
searchengine.extract.app = solr
```

Create Search Index Manually

The following command starts the indexing of all documents with status published:

```
php5/opus4/scripts/SolrIndexBuilder.php
```

Part

VII

7 Configuration

After successful installation, further configurations can be made in OPUS4. All mandatory configurations have already been explained in the relevant chapters. The settings in **config.ini** described here are optional, e.g. individual layouts or definitions as to which document types are to be made available to the user. In addition, several language files can be edited in OPUS4 in order to customize their content for the user. All changes made in language files are administrated in the translation resources of OPUS4, i. e. you only edit the content of a specific key in the corresponding tmx file, and the application automatically inserts the new text at the correct position.

7.1 Mail Options

The following settings define the e-mail address to be used when mails are sent by OPUS4 Mails:

```
;MAIL SETTINGS
; mail.opus.smtp = localhost; SMTP server for sending email
; mail.opus.port = 25 ; SMTP server port for sending email
mail.opus.address = ; <E-Mailadresse, z.B. noreply@bibliothekxyz.de>
mail.opus.name = ; <Name, z.B. BibliothekXYZ>
```

7.2 Referees

If certain persons (e.g. admin, referee) are to be notified when new documents are published, their e-mail addresses can be specified at the following position in config.ini:

```
;REFEREES
; Who should be informed, if a new document gets published?
; referees['John Doe'] = john.doe@example.org
```

Multiple e-mail addresses are separated by a comma.

7.3 Layout/Theme

Here you can define which layout ('theme') is to be used if you do not want to use the standard OPUS4 layout.

```
; The 'theme' setting can be used to select a different theme.
; theme = opus4
```

7.4 Start Module

The default start module in OPUS4 is the HOME module. At this position in config.ini you can define which module is started when the URL does not contain any URL.

```
; The 'startmodule' sets the module, which will be loaded when
; no module is given in the url. Defaults to 'home'.
; startmodule = home
```

7.5 Search Engine

For search functions OPUS4 uses the Browsing Apache Solr, which has to be installed locally. Therefore, the local solr parameters for the index and for text extraction are entered at this position in the config.ini. Default values are explained in chapter '[Installation and Configuration of the Solr Server](#)'. If the

same solr server is used for both, the same parameters must be entered for both areas. In addition, it is possible to show or hide facets. By default all facets are shown. Note: Blanks must not be used in and between facet names.

```
; OPUS4 uses a solr server for search and browsing.  
; the solr server can be run on the same system or on a different host.  
; Please enter the credentials to connect to your solr server.
```

```
; SEARCH ENGINE SETTINGS  
searchengine.index.host =  
searchengine.index.port =  
searchengine.index.app =
```

```
; The text extraction can be run on a different solr server than the metadata indexer.  
; You can also use the same solr server, in this case you have to enter  
; the same credentials as above.  
searchengine.extract.host =  
searchengine.extract.port =  
searchengine.extract.app =
```

```
; Facets fields to be considered.  
; searchengine.solr.facets = author_facet,year,doctype,language,has_fulltext,belongs_to_bibliography,  
subject,institute
```

7.6 URNs

In OPUS4 there is the possibility to automatically assign URNs. If this option is not required, it is sufficient to leave these values commented out.

```
; URN SETTINGS  
; If you do not want to set URNs automatically, set these values blank or  
; comment them out  
; urn.nid = nbn  
; urn.nss = de:kobv:test-opus
```

7.7 Document Types

At this position in the **config.ini** you have the possibility to define settings for document types. Here you can define document types that are to be available to the user (include). If this parameter is commented out, all document types are shown. Document types you do not want to be shown can be excluded by exclude. In addition, you can define via templates for selected document types which template should be used for form generation.

Parameter `maxfilesize` defines the maximum file size. Initially, this value is set to 10 MB. This is indicated in bytes (10 MB = 10240000 bytes). A user uploading a file that is larger than the defined value will receive an error message. Note: The defined value must be smaller than parameter "upload_max_filesize" in the **php.ini**.

Via the value `filetypes.allowed` you can define which file extensions are allowed for uploaded files. Note: For your own safety you should take care not to allow any executable file types. For a trouble-free upload of documents you should ensure that parameter "file_uploads" is set to "On" in the **php.ini**.

```
; DOCUMENTTYPE SETTINGS  
; You can define which document types should be shown in the publish module or  
; which ones should be excluded (comma separated names of XML files without
```

```
; extension). If you don't set documentTypes.include and documentTypes.exclude
; all document types will be shown.
; documentTypes.include = preprint, doctype1
; documentTypes.exclude = doctype2

; Use to configure templates names that do not match the document type name
; documentTypes.templates.preprint = defaulttmpl
; documentTypes.templates.doctype1 = doctype2

; publish.maxfilesize defines the allowed maximum size of a file.
; This does not changes any values of your Apache or php.ini. Please assure
; the values in your Apache or php settings are big enough.
; publish.maxfilesize = 10240000
; publish.filetypes.allowed defines which filetypes can be uploaded.
; publish.filetypes.allowed = pdf,txt,html,htm ; filetypes that are accepted in publication form
```

7.8 Formal Settings

The value 'numberoffiles' defines the number of documents/files that can be uploaded. Initially, this value is set to 2, which can, however, be customized. Note: The defined value must be smaller than parameter "upload_max_filesize" in the **php.ini**.

The value 'bibliographie' defines whether a checkbox for the allocation to the bibliography is to be shown. Possible values: 0 – do not show / 1 - show.

The value "requireupload" defines whether the file upload is required. Possible values: 0 - optional / 1 - required.

```
;FORM SETTINGS
; This configures the count of files a user can upload while publishing a document.
; form.first.numberoffiles = 2
; If you would like to use opus to save bibliographical items you should set
; this to 1. It will ask on the first site of the publish module if a new
; document should be added to the bibliography.
; form.first.bibliographie = 0
; States if the upload-fields are required to enter the second form.
; form.first.requireupload = 0
```

7.9 Full Text Directory

Information about these settings can be found in chapter [Delivery of Files with Access Protection](#).

```
;DOCUMENT-DOWNLOAD-DIRECTORIES
; Please read the documentation about the delivery process of OPUS.
; Here you can configure the internal address of the fulltext directory.
; deliver.target.prefix = /files
; Here you can configure the external address of the fulltext directory.
; deliver.url.prefix = /documents
```

7.10 OAI

At this point, the OAI parameters are defined.

```
; OAI SETTINGS
; oai.baseurl =
```



```
;oai.repository.name = Opus4 Demo Instance  
;oai.repository.identifier = opus4.demo string  
;oai.sample.identifier = oai:opus4.demo:90 Bsp.dokument  
;oai.dnb.contactid =
```

7.11 Supported Languages

At present, the languages supported for the translation resources in OPUS are German and English. However, it is possible to offer OPUS in further languages. In this case, the following parameters can be defined in the config.ini and added accordingly:

```
; SUPPORTED LANGUAGES  
supportedLanguages = de,en
```

Note: In order to switch languages in the Web interface, cookies must be activated.

7.12 FAQ Page

There are two possibilities to show the content of the FAQ page: All help texts on one page (default) or a new page for each help text.

For the second option, the following value must be defined in the **config.ini**:

```
help.separate = true
```

7.13 How to create new document types

By default, Opus4 comes with 20 pre-defined document types ([see attachment](#)). Each document type consists of an XML document type definition and a template. In the following section, the general structure and meaning of the entries in the XML files are explained in more detail. We recommend an XML editor for the creation and validation of document types.

The following changes are possible:

1. Add/remove fields to/from a document type --> The relevant fields must be deleted (commented out) or added in the XML document type file first and after this in the corresponding template
2. Rename document types --> The names of the relevant XML document type file and the corresponding template must be changed
3. New document type --> In this case it is recommended to use an existing document type that already contains many of the fields required and proceed according to steps 1 and 2
4. Change the sequence of fields shown for a document type --> In this case it is sufficient to bring the fields in the template in the desired order

7.13.1 XML Document Type Definitions

The XML document type definitions are in directory **opus4/application/configs/doctype**. They define which fields a document type should contain. In addition, you can influence whether a field is required or optional.

The root element in a document type is called documenttype and has an attribute name as well as further name spaces.

Example:

```
<documenttype name="article"> ... </documenttype>
```

name: Name of the document type, freely selectable. Format: lower-case letters and underscore

It is **recommended** to name document types solely with lower-case letters and without special characters and to name the corresponding templates identically (e.g. doctoralthesis). Currently there is also the following alternative:

```
XML-Dokumenttyp:    doctoral_thesis
Template:           doctoral-thesis
```

However, there is no guarantee that this alternative will be supported by future versions of the ZEND framework.

Within the element documenttype the fields required for this document type are defined by the element field. It has the attributes name, formelement, required, datatype and multiplicity.

Example:

```
<field name="ThesisDateAccepted" formelement="text" datatype="Date" required="yes"
multiplicity="1" />
```

name: Name of the field. The names of the fields cannot be assigned arbitrarily. They are defined in the corresponding XML Schema and are formulated in CamelCase writing.

Possible: You can choose between 70 different fields that can be taken from XML Schema under type definition "validfieldname".

required: Required field? If a field is a required field, it is marked with an asterisk and must be completed.

Possible: yes, no

formelement: HTML form element. Defines which HTML form is to be used for representation.

Possible: text, textarea, select

datatype: Type of field content. Defines what the content of a field may look like. Builds the basis for validation.

Possible: Date, ddc, Email, Institute, Integer, Language, Licence, msc, Person, Project, Text, Title, ThesisGrantor, ThesisPublisher, Year

multiplicity: Number of repetitions. Defines how often a field may occur in total. If the value is larger than 2 or *, an add button and later a delete button is shown for the field, with which the user can define the number according to document type. Possible: 1, 2, 3, 4, ... , *

Certain fields may only be saved once per document, that is why a multiplicity of 1 is **absolutely mandatory** for the following fields:

- * CompletedDate
- * CompletedYear
- * ContributingCorporation
- * CreatingCorporation
- * ThesisDateAccepted
- * Edition
- * Issue
- * Language (of publication)
- * PageFirst
- * PageLast
- * PageNumber
- * PublishedDate
- * PublishedYear
- * PublisherName
- * PublisherPlace

* Volume

If a field is not to be shown at all, it should be deleted (recommended) or commented out from the document type (do **not** set the multiplicity to "0" setzen). To comment out use:

```
<!-- <field> -->
```

Every `field` in the document type must always have **all** attributes mentioned and these must be declared.

Form `field` `subfield`

With `subfield` you have the possibility to define subfields for `field`. In analogy to `field` it has the attributes `name`, `required`, `datatype` and `formelement.multiplicity` is not required here.

Example:

```
<subfield name="AcademicTitle" formelement="text" datatype="Text" required="no" />
```

* `name`: Name of the subfield. The names of the fields are assigned in the same way as for `field`.

* `formelement`: HTML form element.

* `datatype`: Type of field content.

* `required`: Required field?

At present, subfields cannot yet be defined for various fields without restrictions, but only for persons. For a person field (PersonAuthor, PersonSubmitter, etc.), the subfields Email, BirthPlace, BirthDate and AcademicTitle can be created.

7.13.2 Templates

The templates for the document types are in directory **opus4/modules/publish/views/scripts/form**. In addition to define the sequence of elements, you also have the possibility to change the appearance by making amendments to the embedded css classes. Furthermore, you have the possibility to assign option values to elements. The signature of the methods is as follows:

```
element($value, $options = null, $type = null, $name = null)
group($value, $options = null, $name = null)
```

"value" is always assigned together with the name, "options" are also optional and represent a string that can contain possible html attributes of the element. "type" refers to the type of the element and is optional, just as "name".

Option values can be used as follows:

```
<?= $this->group($this->groupTitleAbstract, "cols='60' rows='9'"); ?>
```

In this example the string "cols='60' rows='9'" corresponds to the value `$options`. To avoid errors you should use exactly this syntax. In this way you define that the text field for an abstract is 60 columns wide and 9 lines high. This element already uses the css class `form-textarea`, in which you can adapt the appearance.

```
<?= $this->group($this->groupTitleMain, "size='60'"); ?>
```

With this option value the length of the title field is set to, e.g., 60 characters. By default, the element uses css class form-textfield, which can be adjusted accordingly.

If an element does not use a class, there is the following alternative:

```
<?= $this->element('button_label_abort', "class='form-button submit-button'",  
"Submit", "send"); ?>
```

In this case, the option value for the button is in second place. In this way, css class form-button submit-button is used for the button.

As a rule, all possible attributes of html elements can be used. With "formelement" you define in the document type which html element this is and which attributes are at all possible.

7.14 How to create new fields

You have the possibility to define new fields, e.g., in order to save local or very specific information.

The new field is created in analogy to the existing fields in the desired XML document type definition. It is important to enter `Enrichment` as `datatype` and not to choose `select` as `formelement` (because at this point knowledge about the database is missing).

Syntax

The syntax for a simple field looks as follows:

XML document type definition:

```
<field name="<NewField>" formelement="text" datatype="Enrichment" required="no"  
multiplicity="1" />
```

In the associated template, enter correspondingly:

```
<?= $this->element($this->NewField); ?>
```

The syntax for a group field looks as follows:

XML document type definition:

```
<field name="<NewField>" formelement="text" datatype="Enrichment" required="no"  
multiplicity="*" />
```

In the associated template, enter correspondingly:

```
<$this->group($this->groupNewField); ?>
```

In both examples the attributes `name`, `required` and `multiplicity` can be changed at one's own option.

Translations

In analogy to the example above the following language resource files must be edited:

Write the following entry into **\$BASEDIR/modules/publish/languages/field_header.tmx**:

```
<tu tuid="header_EnrichmentNeuesFeld">
  <tuv xml:lang="en">
    <seg>New field</seg>
  </tuv>
  <tuv xml:lang="de">
    <seg>Neues Feld</seg>
  </tuv>
</tu>
```

Write the following entry into **\$BASEDIR/modules/publish/languages/field_hints.tmx**:

```
<tu tuid="hint_EnrichmentNeuesFeld">
  <tuv xml:lang="en">
    <seg>You can explain the field for your users here.</seg>
  </tuv>
  <tuv xml:lang="de">
    <seg>Hier können Sie Erläuterungen oder einen Hilfetext für Ihre Nutzer eingeben.</seg>
  </tuv>
</tu>
```

Write the following entry into **\$BASEDIR/modules/publish/languages/field_fieldnames.tmx**:

```
<tu tuid="EnrichmentNeuesFeld">
  <tuv xml:lang="en">
    <seg>A new defined field</seg>
  </tuv>
  <tuv xml:lang="de">
    <seg>Ein neudefiniertes Feld</seg>
  </tuv>
</tu>
```

In case of a group field the following files must be edited, too:

Write the following entry into **\$BASEDIR/modules/publish/languages/buttons.tmx**:

```
<tu tuid="button_label_add_one_moreEnrichmentNewField">
  <tuv xml:lang="en">
    <seg>Add one more new field</seg>
  </tuv>
  <tuv xml:lang="de">
    <seg>Eine weiteres neues Feld hinzufügen</seg>
  </tuv>
</tu>

<tu tuid="button_label_deleteEnrichmentNewField">
  <tuv xml:lang="en">
    <seg>Delete the last new field</seg>
  </tuv>
  <tuv xml:lang="de">
    <seg>Das letzte neue Feld löschen</seg>
  </tuv>
</tu>
```

```

    </tuv>
  </tu>

```

Write the following entry into **\$BASEDIR/modules/publish/languages/field_header.tmx**:

```

<tu tuid="groupNewField">
  <tuv xml:lang="en">
    <seg>New field</seg>
  </tuv>
  <tuv xml:lang="de">
    <seg>Neues Feld</seg>
  </tuv>
</tu>

```

Write the following entry into **\$BASEDIR/modules/publish/languages/field_hints.tmx**:

```

<tu tuid="hint_groupNewField">
  <tuv xml:lang="en">
    <seg>You can explain the field for your users here.</seg>
  </tuv>
  <tuv xml:lang="de">
    <seg>Hier können Sie Erläuterungen oder einen Hilfetext für Ihre Nutzer eingeben.</seg>
  </tuv>
</tu>

```

The mentioned translation resource files can be edited in the same way as explained above for existing fields, i.e. to customise help texts (field_hints.tmx).

Enrichments in document overview

New fields are not automatically displayed on the detail view (Frontdoor) of a document. To display them the file

\$BASEDIR/modules/frontdoor/views/scripts/index/index.xslt must be edited in two steps.

Step 1:

In section

```

<!-- Enrichment Section: add the enrichment keys that have to be displayed in frontdoor --> #Zeile
...
<!-- End Enrichtments -->

```

is defined which Enrichment fields shall be displayed . Put the following line between start and end of the comment, if `NewField` shall be displayed:

```

<xsl:apply-templates select="Enrichment[@KeyName='NeuesFeld']" />

```

Step 2:

In section

```

<!-- Templates for "enrichments". --> #Zeile 456

```

is defined how the new field shall be displayed.

In analogy to the example above the additional lines after the comment must look like this:

```
<xsl:template match="Enrichment[@KeyName='NeuesFeld']">
  <tr>
    <th class="name">
      <xsl:call-template name="translateString">
        <xsl:with-param name="string">EnrichmentNeuesFeld</xsl:with-param>
      </xsl:call-template>
      <xsl:text>:</xsl:text>
    </th>
    <td>
      <xsl:value-of select="@Value" />
    </td>
  </tr>
</xsl:template>
```

Newly created fields are shown in the administration section -> Administrate documents under "Extension" and their content can be edited there.

7.15 Configuration of the Start Page

To edit the start page, you have to make changes in the file **opus4/modules/home/language/index.tmx**.

- to edit the page title, the text in the key 'home_index_index_title' must be changed
- to edit the page content on the left, the text in the key 'home_index_index_welcome' must be changed
- to edit the page content on the right, the text in the key 'home_index_index_instructions' must be changed

7.16 Configuration of the Contact Page

To edit the contact information, you have to make changes in the file **opus4/modules/home/language/contact.tmx**:

- to edit the page title, the text in the key 'home_index_contact_title' must be changed
- to edit the page content, the text in the key 'home_index_contact_content' must be changed.

7.17 Configuration of the Imprint Page

To edit the imprint page, you have to make changes in the file **opus4/modules/home/language/imprint.tmx**:

- to edit the page title, the text in the key 'home_index_imprint_title' must be changed
- to edit the page content, the text in the key 'home_index_imprint_content' must be changed

7.18 Configuration of the FAQ Page

To edit the FAQ page, you have to make changes in the file **opus4/modules/home/views/scripts/index/help.phtml** and in the file **opus4/modules/home/language/help.tmx**:

The contents of the FAQ page are divided into so-called sections that may contain an unlimited number of contents. Sections are administered in the file help.phtml. Each section is represented by a key in this file help.phtml.

The keys' function is to administrate the designation of the sections on the one hand and the title and the contents of the entries within a section on the other hand in their associated language file help.tmx. Title and content can be edited directly in this file. For longer texts you have the possibility to transfer them to the directory **opus4/modules/home/views/scripts** and to use the key for these text files in the content element of the **elp.tmx** file as a reference.

Example:

```
<tu tuid="help_index_authorhelp">                                #Key for the section
  <tuv xml:lang="en">
    <seg>Help for authors</seg>                                #Title of the section (eng)
  </tuv>
  <tuv xml:lang="de">
    <seg>Hilfe für Autoren</seg>                                # Title of the section (ger)
  </tuv>
</tu>

<tu tuid="help_title_format">                                    #Key for the title within a section
  <tuv xml:lang="en">
    <seg>What's a presentation format?</seg>                    #Title within the section (eng)
  </tuv>
  <tuv xml:lang="de">
    <seg>Was ist das Präsentationsformat?</seg>                # Title within the section (ger)
  </tuv>
</tu>

<tu tuid="help_content_format">                                  #Key for the content of the title
  <tuv xml:lang="en">
    <seg>format.en.txt</seg>                                    #Designation of the (transferred) Text file (eng)
  </tuv>
  <tuv xml:lang="de">
    <seg>format.de.txt</seg>                                    # Designation of the (transferred) Text file (ger)
  </tuv>
</tu>
```

7.18.1 Guidelines

It is of utmost importance to mention the operator's guidelines, since these govern the transfer of rights upon publication. The relevant texts must be contained in the following files:

opus4/modules/home/views/scripts/policies.en.txt
opus4/modules/home/views/scripts/policies.de.txt

7.19 Configuration of the Search Interface

The texts and the edition statements for the search interface, the results page and facets are administrated in the file **opus4/modules/solrsearch/language/solrsearch.tmx**. To edit or make adjustments to this file, open the file, execute the requested changes and save the file in the same directory. Facets can also be modified by means of the file **solrsearch.tmx**:

Example:

```
<tu tuid="author_facet_facet_heading">
  <tuv xml:lang="en">
    <seg>Author</seg>
  </tuv>
```



```
<tuv xml:lang="de">
    <seg>Autor</seg>
</tuv>
```

7.20 Configuration of Mouse-Over Texts

Help texts shown for fields by means of mouse-over can be changed in the file **opus4/modules/publish/language/fieldheader**. The same applies to the designation of fields (**opus4/modules/publish/language/fieldnames**).

Part



8 Administration

The administration area is only accessible for registered users. On the following pages, the administrative functions are described in detail.

8.1 Account

The administrator's user name is "admin" and cannot be changed. The default password is "adminadmin" and should be changed upon the first login. To do so, click on 'Account' next to the login/logout button and define a new password. In addition, details about the administrator (name, e-mail address) can be entered here.

8.2 Clearance

In the Clearance section documents can be cleared by authorized persons after they have been published in OPUS. As a first step, the relevant documents are selected. A click on 'Clear selected documents' takes you to step 2. Before the documents are finally activated, a 'Referee' has to be entered (this can be any authorized person). In this way, it is possible to reconstruct who cleared which documents, even if several people have been involved in the process.

8.3 Access Control

In the Access Control section it is possible to administrate users and define IP address ranges and user roles.

8.3.1 User Roles

In this section you can define user roles that determine the rights a particular user has. By default, OPUS 4 has two user roles: Administrator (has all rights) and Guest (may publish and see published documents). If a new user role is created, it has to be given a name first (e.g. 'Library'). Then the individual rights to be assigned to this user role can be selected:

Simple rights:

- administrate – Access to administration
- clearance – Clearance of newly posted documents
- publish – Publish documents
- publishUnvalidated – Publish documents despite validation failure (should be restricted to administrators)

Read metadata:

- published – Read published documents that have been cleared
- unpublished – Read published documents that have *not yet* been cleared
- deleted – Read deleted documents

8.3.2 User Accounts

In the User Accounts section existing users can be edited and new users created ('Add account'). If a new user is created, an account name ('Login') and a password must be assigned (minimum 6 characters). After this the new user account is assigned to a role.

8.3.3 IP Address Ranges

In this section you can define IP address ranges in which certain user roles are applicable. This means that all users accessing the repository from a certain IP range have automatically the respective rights.

If a new IP range is added, a name must be entered first. This may only consist of letters and numbers (no special characters, no blanks). Below that you enter the relevant IP range from the start IP to the end IP. If the range is only a single IP address, this is entered in the field 'Start IP'. Lastly, the user role is selected.

8.4 Adminstrate Documents

This section is the core of the administration. This is where metadata can be edited or deleted, full texts can be added subsequently and the publication status of a document can be changed. Documents can be sorted by ID (first column), title, author, date or document type. In addition, documents can be assigned to collections and existing assignments can be deleted.

8.5 Administrate Licenses

In this section licenses available to the user in the course of the publication process can be administrated. Only when licenses are created here, is it possible for users to select them in the document types.

8.6 Administrate Collections

In OPUS4 it is possible to create collections in order to offer documents via Browsing, e.g. structured according to subject, geography or institution. Each collection can have an unlimited number of collection entries. On the collection level, these can be moved upwards or downwards in their position. On the collection entry level this is not that easy. Here, collection entries must be deleted and re-created, and the documents assigned to them must be re-linked.

If a new collection is to be created, the following settings must be made upon clicking on 'Create new collection':

- Name --> to be assigned at random
- Name of the collection root in the OAI set entry -->
- Sort direction --> defines which position the new collection will have in the display
- Collection is used --> determines the visibility of the collection
- Metadata fields to be shown when Browsing, on the front door and in the OAI set --> For the display of the collection entries there is the choice between Name | Number | Name, Number | Number, Name

Translation

For the correct display of newly created collection when Browsing, the translation file **modules/solrsearch/language/browsing.tmx** must be included:

Example:

```
<tu tuid="search_index_custom_browsing_<name of collection defined above>">
  <tuv xml:lang="en">
    <seg><English title></seg>
  </tuv>
  <tuv xml:lang="de">
    <seg><German title></seg>
  </tuv>
</tu>
```

Visibility

Collections and collection entries can be set to visible or invisible. Since the visibility setting for a collection is not 'inherited' by the collection entries, it is, in principle, possible to guess the ID of a collection entry that does not have the visibility setting 'invisible'. This collection entry is then shown in the current implementation, as are the attached documents and any existing deeper-lying collection entries (unless they have the visibility setting invisible).

If you want or have to be absolutely sure that no collection entries that are supposed to be invisible are shown in the front end, you have to manually set the visibility of **all** collection entries of a specific collection to 'invisible'.

Add Collections to Documents

In the 'Administrate Documents' section, collections can be linked to documents.

During Collection Browsing only documents that are directly assigned to the collection can be shown. If you want to 'Show all documents in sub-tree', all documents assigned to a sub-collection must **additionally** be assigned to a superordinate collection.

CSV Export

In order to export a collection or to export linked documents als CSV file, you have to enter the name of a collection (role) and the name or number of an associated collection entry, e.g. <http://localhost/opus4-devel/remotecontrol/collection/list?role=institutes&name=Bauwesen> or <http://localhost/opus4-devel/remotecontrol/collection/list?role=ddc&number=521>

If this has been successful, the function returns a CSV file containing all documents assigned to the referenced collection: Each line contains a document ID, title and year.

If no documents are assigned to the referenced collection, the function returns an empty file.

The function may return the following errors:

- HTTP Response Code 400 (Bad Request), if an internal error has occurred (e.g. missing URL parameter role and/or number; name of CollectionRole unknown; Collection Number unknown; Solr server not available)
- HTTP Response Code 501 (Not Implemented), if the number of a collection is ambiguous (with reference to all collection entries assigned to the collection in question) <http://localhost/opus4-devel/collections/list/csv?role=ddc&number=520>

8.7 Administrate Languages

Under this menu item you can administrate the languages that are to be available for the publication process. In order to create new languages the relevant ISO values of ISO standards 639-2 and 639-1 must be entered:

Part2_B: Identifier of the "bibliographic applications code set" (ISO 639-2), 3 characters, e.g. ger

Part2_T: Identifier of the "terminology applications code set" (ISO 639-2), 3 characters, e.g. deu

Part1: Identifier (ISO 639-1), z.B. de

Scope: 'I' (Individual), 'M' (Macrolanguage) oder 'S' (Special),

Type: 'A' (Ancient), 'C' (Constructed), 'E' (Extinct), 'H' (Historical), 'L' (Living), 'S' (Special),

RefName: Reference name of the language, e.g. German

8.8 Show Publication Statistics

Under this menu item you can retrieve the current publication statistics of the repository for a certain year: total number of documents, newly published documents per month, documents per document type and documents per institute.

8.9 Show OAI Link

In this section the OAI link is shown with examples if the metadata of the repository are to be embedded into other services (e.g. BASE).

8.10 Institutions (Distributing Institutions)

Here you can enter DNB institutes that build the basis for fields ThesisGrantor and ThesisPublisher in document types bachelorthesis, masterthesis, habilitation and doctoralthesis. ThesisPublisher (Publishing Institution) is created with entries Name (required), Address (optional), City (required), Phone (optional), DNB Contact ID (optional). Then you can select whether the institution that has been entered is also the granting institution (isGrantor).

Part

IX

9 Migration OPUS 3.2 to OPUS 4

This Chapter explains how to migrate from existing OPUS 3 applications to OPUS 4.

9.1 Preparation

Scenario for migration

OPUS 3.x application on server `opus3.example.org` shall be migrated to OPUS 4 application on server `opus4.example.org` (Note: servers can be identical, in this case the following `scp` commands can be replaced by simple `cp` commands; the packing of the full texts via `tar` can be omitted)

1. Generation of an Opus3-XML-DB-Dump

First a XML-Dump of the OPUS 3 documents (metadata) must be generated and copied on the same server, on which OPUS 4 was installed (in this example: `opus4.example.org`):

```
// Log on to opus3.example.org
mysqldump --xml -u<user> -p<password> <database> > opus3_dump.xml
scp opus3_dump.xml <opus4user>@opus4.example.org:$BASEDIR/migration
```

2. Pack full texts

After that the OPUS 3 full texts have to be packed and copied on the same server, on which OPUS 4 was installed:

```
// change into OPUS3 directory
tar cvfz fulltexts.tar.gz ./fulltexts/
scp volltexte.tar.gz <opus4user>@opus4.example.org:$BASEDIR/migration
```

3. Extract full texts

Now the full texts are extracted on the same server, on which OPUS 4 was installed:

```
//Log on to opus4.example.org
cd $BASEDIR/migration
tar xvfz volltexte.tar.gz
```

9.2 Migration

Execute migration script

After these preparations the migration itself can start by executing the migration script:

```
cd $BASEDIR/opus4/scripts
sh ./opus3-migration.sh [-f xmlfile] [-p fulltextpath] [-z stepsize for
looping] [-i ]
```

with a semantic as follows:

-f stands for "OPUS3-XML-database export file" (in the example above: `$BASEDIR/migration/opus3_dump.xml`)

-p stands for "OPUS3 fulltext files" (in the example above: `$BASEDIR/migration/fulltexts`)

-z stands for "Stepsize for looping", this is default 50

-i stands for "Build Index after each loop" and is default "off", which means the Solr Index will be built

after importing all data

After the import is finished you will see the following message:

Operation completed successfully in xx seconds.

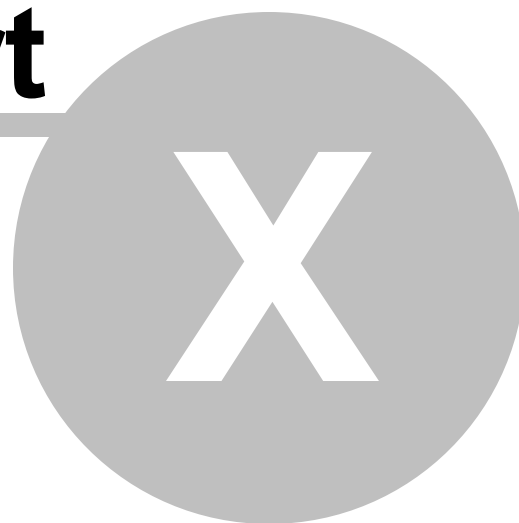
9.3 Troubleshooting

During the migration the import of the documents will be shown as follows:

```
Successfully imported old ID 1 with new ID 1 -- memory 5414 (KB), peak
memory 7872 (KB)...
1 file(s) have been imported successfully for document ID 1 -- memory 41142
(KB), peak memory 44634 (KB)
...
```

When errors occur, these messages are shown directly below the corresponding ID numbers.

Part



10 Attachment

10.1 Document types

Document types in OPUS4 have been named on the basis of the "Gemeinsames Vokabular für Publikations- und Dokumenttypen" and are also defined along these lines. OPUS users are free to rename document types, if required, or to define them according to their own criteria. However, it should be noted that the "Gemeinsames Vokabular" is one of the foundations for the XMetaDissPlus2.0 format, so that during the creation of document types in OPUS 4 the potential automatic delivery of all documents in the repository via this standard to the DNB has already been factored in.

10.1.1 (Scientific) Article

Document type **(scientific) Article** includes documents that have been published as article, editorial, register, table of contents or editorial section of a scientific journal or scientific periodical (postprint).

10.1.2 Bachelor's Thesis

Document type **Bachelor's Thesis** refers to the lowest level of a written thesis (usually after 3 years of study).

10.1.3 Book

Document type **Book (Monograph)** is intended for classic monographic publications.

10.1.4 Part of a Book

Document type **Part of a Book (Chapter)** represents documents that have been prepared within the framework of a monographic publication, such as chapters or contributions to compilations.

10.1.5 Conference Object

Document type **Conference Object** includes all kinds of documents connected to a conference (conference papers, conference reports, conference lectures, contributions to conference proceedings, conference contributions, abstracts, volumes of conference contributions, conference posters).

10.1.6 Contribution to a (non-scientific) Periodical

Document type **Contribution to (non-scientific) Periodical** refers to contributions in newspapers, weekly magazines or other non-scientific periodicals.

10.1.7 Course Material

Document type **Course Material** refers to teaching material in the broadest sense, e.g. lecture recordings as video or audio files, exercise material, preparation or exam material. Lecture texts as such, however, are represented by document type **Lecture**.

10.1.8 Doctoral Thesis

Document type **Doctoral Thesis** refers to the highest level of a written thesis, which is equivalent or higher than a doctoral thesis, but does not follow the Bologna Convention. This also includes document type habilitation thesis.

10.1.9 Habilitation

For a definition of document type **Habilitation** refer to Doctoral Thesis.

10.1.10 Image

Document type **Image** refers to a non-textual visual representation. Examples are pictures of photographs of objects, paintings, prints, drawings, other images and graphics, animations and moving images, films, diagrams, maps or sheet music. This document type can be used for digital and physical objects.

10.1.11 Lecture

Document type **Lecture** includes university speeches, lectures and inaugural lectures.

10.1.12 Master's Thesis

Document type **Master's Thesis** refers to the medium level of a written thesis and also includes written theses completed before the Bologna process for academic degrees equivalent to the current master degree ('Magister', 'Uni-Diplom', 'Staatsexamen').

10.1.13 Misc

Document type **Misc** is intended for everything that does not fit in any of the existing document types.

10.1.14 Moving Image

Document type **Moving Image** refers to a series of visual representations that convey the impression of movement when shown sequentially. Examples are animations, films, TV shows, videos, zoetropes or the visual representation of a simulation.

Note: Objects of the document type **Moving Image** also correspond to the type **Image**. Therefore, during delivery via XMetaDissPlus2.0, in element `dini:PublType` both `MovingImage` and `Image` are requested.

10.1.15 Preprint

Document type **Preprint** includes preliminary scientific or technical papers that are not published in a series of an institution, but are to appear in a scientific journal or as part of a book.

10.1.16 Report

Document type **Report** includes textual material that cannot be categorized as any of the other types, e.g. reports, external research reports, internal reports, memos, statistical reports, project completion reports, technical documentations and instructions.

10.1.17 Review

Document type **Review** refers to reviews of books or article and/or summaries of a publication that have not been written by the author.

10.1.18 Sound

Document type **Sound** refers to a resource whose primary aim is to be heard, e.g. music files, audio CDs, speech and sound recordings. No differentiation is made between sounds, noise and music.

10.1.19 Study Thesis

Document type **Study Thesis** refers to textual elaborations that are prepared as part of a course of study (term papers, seminar reports, investigation and project reports) and are not categorized as thesis.

10.1.20 Working Paper

Document type **Working Paper** refers to a preliminary scientific or technical paper that is published in a series of an institution (also: Research Paper, Research Memorandum, Discussion Paper).

10.2 Meaning of the fields

The following table shows all field names available in OPUS. In contrast to labels, field names must not be changed.

Field Name	Meaning	Description
PersonSubmitter	Submitter	Submitter's contact data
Licence	Licence	Describes the rights / access to the document (with / without print on demand, which CC licence)
Language	Language	Language in which the electronic resource has been prepared
ServerDateUnlocking	Unlocked on	Serves to select the point in time at which the document is be available in the repository
IdentifierOpus3		The old OPUS 3-ID
TitleMain	Main title	Original title of the document or object
TitleAbstract	Abstract / Short version	A summary of the document or object
TitleSub	Subtitle	Subtitle (addition to the title of) of the document or object
TitleParent	Title of the parent document	e.g. Title of the journal, compilation etc.
PersonAuthor	Author	Information about the author
ServerDateValid	Date of expiry of validity	Serves to select the point in time up to which the document is be available in the repository
PersonReferee	Referee	An external referee who has decided whether a document is published or not
PersonEditor	Editor	The editor of the document (bibliographic information)
PersonTranslator	Translator	The translator of the document
PersonContributor	Other person involved	Name of person who has made a significant intellectual contribution to the document

CreatingCorporation	Creating corporation	Name of the organization responsible for the intellectual content of the document (bibliographic information)
ContributingCorporation	Contributing corporation	Name of the organization that has made a significant intellectual contribution to the document
Journal	Journal	Journal in which, e.g., the article has been published, bibliographic field
Volume	Volume	Volume in which, e.g., the article has been published, bibliographic field
Issue	Issue	Issue in which, e.g., the article has been published, bibliographic field
PageNumber	Page Number	Number of pages in the document
PageFirst	Number of the first text page	
PageLast	Number of the last text page	
CompletedYear	Year of publication(online)	
CompletedDate	Date of publication (online)	Default date in each document; by default the current date is assigned when the document is published
PublishedYear	Year of initial publication	Is used when the document has already been published (e.g. by a publishing house), bibliographic field
PublishedDate	Date of initial publication	
SubjectSwd	SWD keyword	Controlled vocabulary(SWD)
SubjectUncontrolled	Free keyword / Tag	Freely selectable description (no controlled vocabulary)
IdentifierUrn	URN	Unique identifier of the document
IdentifierOpac	OPAC-ID	
Note	Note / Comment	Free field for comments
PersonAdvisor	Advisor	Responsible advisor of a written thesis or study thesis
Institute	Allocation to institution	Institutions that have been created as collections
PublisherName	Verlagsname	Name of the publishing house, bibliographic field
PublisherPlace	Verlagsort	Place of the publishing house, bibliographic field
IdentifierIsbn	ISBN	ISBN Number

IdentifierIssn	ISSN	ISSN Number
TitleAdditional	Translated title	This field is used if a translated title is required apart from the original title
ThesisDateAccepted	Date of exam	Date on which the written thesis was accepted according to examination regulations, usually date of the oral exam
ThesisPublisher	Publishing Institution	The content of these two fields is administrated by the institutions (publishing institution).
ThesisGrantor	Title-granting institution	
Type	Document type	

10.3 Mapping of Document Types from Opus3 to Opus4

In Opus4 a couple of new document types have been introduced, whereas others (already existing) have been standardised according to the "Gemeinsames Vokabular für Publikations- und Dokumenttypen" (DINI Schriften 12-de).

Therefore, in the following please find a mapping of (standard) document types from Opus3 to those used in Opus4.

Document Types Opus3	Document Types Opus4
Manual	report
Article	article
Book	book
InBook	bookpart
Thesis.Master	masterthesis
Thesis.Doctoral	doctoralthesis
Festschrift	book
Journal	misc
Proceedings	conferenceobject
InProceedings	article
ResearchPaper	workingpaper
Paper	workingpaper
TechReport	report
Preprint	preprint
Postprint	article
Misc	misc
Thesis.Habilitation	habilitation
Thesis.Bachelor	bachelorthesis

Lecture	lecture
Image	image
Video	movingimage
Music	sound
CourseMaterial	coursematerial