

DWDS-DDC-Mini-Howto
Version 1.74
Kai Zimmer zimmer@bbaw.de

Content

- 1) necessary files
- 2) indexing
- 3) connecting the corpus
- 4) restarting ConcordDaemon
- 5) check the logfile

Shell commands are written in *italics* .

1) Necessary files

Before you start indexing a corpus, you'll need

- 1) a list of files you want to index (e.g. Corpus.con) – this can easily be done by entering:

```
find . -name "*.txt" >corpus.con
```

The dot tells the find command where to search for files – in this case it's the current directory

- 2) an options-file (e.g. Corpus.opt), which tells ddc about the type of data you're going to index (text, HTML, XML) and some parameters.

Some examples:

- a. "classical" DWDS-XML-TEI documents:

German

IndexType DWDS_Index

UseParagraphTagToDivide

EmptyLineIsNotSentenceDelim

DontUseIndention

UserMaxTokenCountInOnePeriod 10000000

IndexMorphPatterns

OutputBibliographyOfHits

IndexPunctuation

Bibl 1 textClass TEI.2/teiHeader/profileDesc/textClass/keywords/term

Bibl 0 author

```
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/titleStmt/author
```

Bibl 0 date

```
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/publicationStmt/  
date[@id="first"]
```

```
Bibl 0 orig /TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/bibl
```

Bibl 0 scan /TEI.2/teiHeader/fileDesc/sourceDesc[@id="scan"]/bibl
Bibl 0 page
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/seriesStmt/idno[@type="page"]
Bibl 0 body /TEI.2/text/body

b. “free” indexed XML documents:

German
IndexType Free_Index
Indices [Token w normal]; [Lemma l normal]; [Pos p normal]; [Thes t normal]
HitBorders [s:sentence:default]; [c:clause]
OutputBibliographyOfHits
ResumeOnIndexErrors
Bibl string 1 textClass
/TEI.2/teiHeader/profileDesc/textClass/keywords/term
Bibl string 1 title
/TEI.2/teiHeader/fileDesc/titleStmt/title[@type="main"]
Bibl integer 0 date
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/publicationStmt/date[@id="first"]
Bibl string 1 avail
/TEI.2/teiHeader/fileDesc/publicationStmt/availability/@n
Bibl string 0 orig
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/bibl
Bibl integer 1 page /TEI.2/text/body/pb/@n
Bibl string 1 idno /TEI.2/teiHeader/fileDesc/publicationStmt/idno
Bibl string 0 author
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/titleStmt/author
Textarea body /TEI.2/text/body

c. morphologically annotated XML documents:

German
IndexType MorphXML_Index
UserMaxTokenCountInOnePeriod 10000000
OutputBibliographyOfHits
Bibl string 0 body /ddc_document/text
Bibl string 0 author /ddc_document/header/author
Bibl integer 0 date /ddc_document/header/date
Bibl string 0 title /ddc_document/header/title
Bibl string 0 scan /ddc_document/header/bibl

Bibl string 1 textClass /ddc_document/header/textClass
Bibl integer 0 volume /ddc_document/header/idno[@type="volume"]

d. TEI-XML documents with annotation and relevance ranking

German
IndexType Free_Index
Indices [Token w normal]; [Lemma l normal]; [Pos p normal]; [Thes t normal]
HitBorders [s:sentence:default]; [c:clause]
OutputBibliographyOfHits
ResumeOnIndexErrors
CaseInsensitive
TfIdfRank 1.0
NearRank 1.0
PositionRank 1.0
Bibl string 1 textClass
/TEI.2/teiHeader/profileDesc/textClass/keywords/term
Bibl string 1 title
/TEI.2/teiHeader/fileDesc/titleStmt/title[@type="main"]
Bibl string 1 biblfilename /TEI.2/teiHeader/fileDesc/filename
Bibl integer 1 date
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/publicationStmt/date[@id="first"]
Bibl string 1 author
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/titleStmt/author
Bibl integer 1 PageRank
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/relevance
TextArea filename
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/filename
TextArea txttitle
/TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/titleStmt/title
Textarea body /TEI.2/text/body

e. TEI-XML documents showing annotations and relevance

German
IndexType Free_Index
Indices [Token w normal]; [Lemma l normal storage]; [Pos p normal storage]; [Token2 v normal storage]; [Spk s normal storage]; [Mysql m normal storage]
HitBorders [s:sentence]; [c:clause]; [u:utterance:default]
OutputBibliographyOfHits
ResumeOnIndexErrors

TfIdfRank 1.0
 NearRank 1.0
 PositionRank 1.0
 Bibl string 1 textClass
 /TEI.2/teiHeader/profileDesc/textClass/keywords/term
 Bibl string 1 title
 /TEI.2/teiHeader/fileDesc/titleStmt/title[@id="main"]
 Bibl integer 1 date
 /TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/publicationStmt/date[@id="first"]
 Bibl integer 1 page /TEI.2/text/body/pb/@n
 Bibl string 0 author
 /TEI.2/teiHeader/fileDesc/sourceDesc[@id="orig"]/biblFull/titleStmt/author
 TextArea body /TEI.2/text/body
 IndicesToShow 1 2 3 5 6
 InterpDelimiter ^

f. plain HTML documents (e.g. a homepage) :

IndexType DWDS_Index
 German
 LocalPathPrefix pages/
 InternetPathPrefix www.dwds.de/pages/
 QueryOnlyFiles
 CaseInsensitive

2) indexing

Once you have created the necessary files (see 1)) indexing is simple:

- export the environment variable RML to the RML-directory, e.g.
export RML=/opt/ddc
- index the corpus
\$RML/Bin/ConcordIndex Corpus.con

3)connecting the corpus

the following steps are necessary:

- a) customize the file \$RML/Bin/ddc_local_corpora.cfg:

```
// CorporaName IP PORT LocalName
```

```
dwds0 192.168.1.83 50005 /home/sokirko/M5/01/01.con
dwds1 192.168.1.83 50006 /home/sokirko/M5/02/02.con
dwds2 192.168.1.83 50007 /home/sokirko/M5/03/03.con
dwds3 192.168.1.83 50008 /home/sokirko/M5/04/04.con
dwds4 192.168.1.83 50009 /home/sokirko/M5/05/05.con
dwds5 192.168.1.83 50010 /home/sokirko/M5/06/06.con
dwds_hp 192.168.1.83 50100 /home/dwds_hp/pages/dwds_hp.con
```

Here I added the corpus “dwds_hp” to the end of the file. The IP is the address of the ddc server, the portnumber needs to be setup (traditionally a TCP-port > 50001) and needs to be unique (every new corpus needs an IP of its own), the last entry is the absolute path to the corpus.con file.

b) customize the file \$RML/Bin/ddc_server.cfg:

```
// CorporaName IP PORT
server 192.168.1.83 50011
dwds0 192.168.1.83 50005
dwds1 192.168.1.83 50006
dwds2 192.168.1.83 50007
dwds3 192.168.1.83 50008
dwds4 192.168.1.83 50009
dwds5 192.168.1.83 50010
dwds_hp 192.168.1.83 50100
```

these values are similar to the ddc_local_corpora.cfg above – but without path of the Corpus.con file. The sense of this file is an abstraction of multiple DDC-servers (e.g. a cluster) into a frontend-server, but it’s necessary to setup also without making use of clustering. The keyword “server” must be in the first line.

c) customize the file \$RML/Bin/ddc_xml_server.cfg:

```
server 192.168.1.83 50011
```

This file is used especially for querying with the command line tool ‘ddc_xml’. There’s only one server address to configure here, which ddc_xml will query.

4) restarting ConcordDaemon

Finally, we need to restart the ConcordDaemon:

ConcordDaemon stop
ConcordDaemon start

Attention – if you compiled ddc-concordance yourself from the sources, the daemon is named ‘ConcordDaemontst’.

The whole process of restarting can take several minutes (depending on the corpus size). You should watch the logfile growing now (see below).

5) check the logfile

You can check in the logfile whether the Daemon started properly. It's located in `$RML/Logs/concord/date.log`. You have to replace *date* with the current date:

```
tail -f 14May2007.log
```

for the configuration shown above, the entries look like this:

```
16:11:30 > Entering ConcordDaemon
16:11:30 > SocketInitialize
16:11:30 > InitLemmatizers
16:11:34 > LoadCorpora
16:11:34 > Start loading corpora from
/home/sokirko/RML/Bin/ddc_local_corpora.cfg
16:11:34 > Found 7 Hosts
16:11:34 > Loading corpora index for /home/sokirko/M5/01/01.con
16:11:58 > init graphan
16:11:58 > ok
16:11:58 > Loading corpora index for /home/sokirko/M5/02/02.con
16:12:24 > init graphan
16:12:24 > ok
16:12:24 > Loading corpora index for /home/sokirko/M5/03/03.con
16:12:53 > init graphan
16:12:53 > ok
16:12:53 > Loading corpora index for /home/sokirko/M5/04/04.con
16:13:18 > init graphan
16:13:18 > ok
16:13:18 > Loading corpora index for /home/sokirko/M5/05/05.con
16:13:38 > init graphan
16:13:38 > ok
16:13:38 > Loading corpora index for /home/sokirko/M5/06/06.con
16:14:00 > init graphan
16:14:00 > ok
```

```
16:14:00 > Loading corpora index for /home/dwds_hp/pages/dwds_hp.con
16:14:01 > init graphan
16:14:01 > ok
16:14:01 > Create Listener dwds0 (192.168.1.83:50005)
16:14:01 > Create Listener dwds1 (192.168.1.83:50006)
16:14:01 > waiting for accept
16:14:01 > Create Listener dwds2 (192.168.1.83:50007)
16:14:01 > waiting for accept
16:14:01 > Create Listener dwds3 (192.168.1.83:50008)
16:14:01 > waiting for accept
16:14:01 > Create Listener dwds4 (192.168.1.83:50009)
16:14:01 > waiting for accept
16:14:01 > waiting for accept
16:14:01 > Create Listener dwds5 (192.168.1.83:50010)
16:14:01 > Create Listener dwds_hp (192.168.1.83:50100)
16:14:01 > waiting for accept
16:14:01 > LoadServer
16:14:01 > waiting for accept
16:14:01 > Create Listener server (192.168.1.83:50011)
16:14:01 > waiting for accept
```

Most important are the “waiting for accept”-entries. If they’re not yet completely started (e.g. if you use huge corpora) you can’t access the corpus. Or maybe you still have a configuration error.