# Builder utilities

## ffanalyzer

ffanalyzer -i <inputfilename> -dir <inputdir> -odir <outputdir> -e <errorfile>

## ffindexer

ffindexer -o <outputdir> -i <inputfile> -k <extra\_file>

## ffplaylists

ffplaylists -i <inputfile> -l <levelsfile> -odir <outputdir>

## ffdictionary

ffdictionary -i <inputfile1> -i <inputfile2> ... -odir <outputdir>

## ffviews

ffviews -i <inputfile> -l <levelsfile> -odir <outputdir>

## ffcontents

ffcontents -i <inputfile> -odir <outputdir> -l <levelsfile>

## sqliteloader

sqliteloader –table <tablename> -cols <configstring> -i <inputfilename> -t <targetdbfilename>

Config string:

string := <col1>,<col2>,....,<colN>

col := <columnname>:<datatype>

datatype := integer | text | blob | blobfile

# Database structure

## Queries for creating tables

create table contents(title text, record integer, parent integer, level integer, simpletitle text, subtext text);

create index icontents on contents(parent);

create index icontents2 on contents(level);

create table words(word text, uid integer, indexbase integer, data blob, idx text);

create index iwords on words(word);

create table docinfo(name text, valuex text, idx integer);

create table objects(objectName text, objectType text, objectData blob);

create index iobject on objects(objectName);

create table groups(groupname text, recid integer);

create index igroup on groups(groupname);

create table texts(plain text, recid integer, showid integer, levelname text, stylename text);

create index itexts on texts(recid);

create index itxstyle on texts(stylename);

create table levels(level text, id integer, original text);

create table popup(title text, class text, plain text);

create index ipopup on popup(title);

create table jumplinks(title text, recid integer);

create table styles(name text, id integer);

create table styles\_detail(styleid integer, name text, valuex text);

create table playlists(id integer, parent integer, title text);

create table playlists\_detail(parent integer, ordernum integer, objectName text);

create table textviews(id integer, parent integer, title text);

create table textviews\_texts(parent integer, textid integer);

create table dictionary(id integer, name text);

create table dict\_words(id integer, word text, simple text);

create table dict\_means(wordid integer, dictid integer, recid integer PRIMARY KEY ASC ON CONFLICT REPLACE AUTOINCREMENT, meaning text);

create index idict\_means on dict\_means(dictid,wordid);

# Files structure

## contents.txt

structure:

<levelid>\t<recordid>\t<parentrecordid>\t<text>\t<asciititle>\t<subtext>

## dictionary.txt

structure:

<dictionaryid>\t<title>

## dict\_means.txt

structure:

<wordid>\t<dictionaryid>\t<textmeaning>

## dict\_words.txt

structure:

<wordid>\t<unicodetext>\t<asciitext>

## docinfo.txt

structure:

<propertyname>\t<propertyvalue>\t<index>

## groups.txt

structure:

<groupname>\t<recordid>

## jumplinks.txt

structure:

<jumplinkname>\t<recordid>

## levels.txt

structure:

<id>\t<originalname>\t<safename>

## objects.txt

strcuture:

<objectname>\t<fileurl\_of\_blob>\t<objectmimetype>

## playlists.txt

structure:

<playlistid>\t<parentid>\t<title>

## playlists\_detail.txt

structure:

<playlistid>\t<objectid>\t<objectname>

## popup.txt

structure:

<title>\t<classname?>\t<plaintext>

## styles.txt

structure:

<styleid>\t<stylename>

## styles\_detail.txt

structure:

<styleid>\t<propertyname>\t<propertyvalue>

## texts.txt

This file contains still some tags aimed for content building

structure:

<recordid>\t<plaintext>\t<levelname>\t<stylename>

## texts\_b.txt

This file is cleaned from content building tags.

structure:

<recordid>\t<plaintext>\t<levelname>\t<stylename>

## views.txt

structure:

<parentid>\t<viewid>\t<viewtitle>

## view\_details.txt

structure:

<viewid>\t<recordid>

## words\_a.txt

structure:

<groupname>\t<wordid>\t<word>\t<indexbaseoffset>\t<hexadata\_of\_blob>

## words\_b.txt

structure:

<groupname>\t<wordid>\t<word>\t<indexbaseoffset>\t<fileurl\_of\_blob>

## words\_raw.txt

structure:

<wordid>\t<recordid>\t<proximity>