MWDumper

From MediaWiki.org

(Redirected from Mwdumper)

MWDumper is a quick little tool for extracting sets of pages from a MediaWiki dump file.

An old JAR at download.wikimedia.org (http://download.wikimedia.org/tools/) doesn't work. To import current XML export dumps, you will probably have to build MWDumper from source (http://svn.wikimedia.org/svnroot/mediawiki/trunk/mwdumper/) or use third-party builds (http://csomalin.csoma.elte.hu/~tgergo/wiki/mwdumper.jar) (which starts in GUI mode by default so you won't need most of the parameters below, just run it with `java -jar mwdumper.jar`).

It can read MediaWiki XML export dumps (version 0.3, minus uploads), perform optional filtering, and output back to XML or to SQL statements to add things directly to a database in 1.4 or 1.5 schema.

It is still very much under construction.[1]

Note: While this can be used to import XML dumps into a MediaWiki database, it may not always be the best choice for this task. See Manual:Importing XML dumps for an overview.

Contents

Usage

Sample command line for a direct database import:

```
java -jar mwdumper.jar --format=sql:1.5 pages_full.xml.bz2 |
  mysql -u <username> -p <databasename>
```

or

```
javac src/org/mediawiki/dumper/Dumper.java
java -classpath ./src org.mediawiki.dumper.Dumper --format=sql:1.5 pages_full.xml.bz2 |
  mysql -u <username> -p <databasename>
```

Hint: The tables 'page', 'revision' and 'text' must be empty for a successful import.

Note: this command will keep going even if MySQL reports an error. This is probably not what you want - if you use the direct connection to MySQL, the import will stop when errors occur.

A note on character encoding

Make sure the database is expecting utf8-encoded text. If the database is expecting latin1 (which MySQL does by default), you'll get invalid characters in your tables if you use the output of mwdumper directly. One way to do this is to pass --default-character-set=utf8 to mysql in the above sample command.

Also make sure that your MediaWiki tables use CHARACTER SET=binary (http://dev.mysql.com/doc/refman/5.0/en/charset-table.html) . Otherwise, you may get error messages like `Duplicate entry in UNIQUE Key 'name_title'` because MySQL fails to distinguish certain characters.

You can also do complex filtering to produce multiple output files:

```
java -jar mwdumper.jar \
  --output=bzip2:pages_public.xml.bz2 \
    --format=xml \
    --filter=notalk \
    --filter=namespace:\!NS_USER \
    --filter=latest \
  --output=bzip2:pages_current.xml.bz2 \
    --format=xml \
    --filter=latest \
  --output=gzip:pages_full_1.5.sql.gz \
    --format=sql:1.5 \
  --output=gzip:pages_full_1.4.sql.gz \
    --format=sql:1.4 \
  pages_full.xml.gz
```

A bare parameter will be interpreted as a file to read XML input from; if "-" or none is given, input will be read from stdin. Input files with ".gz" or ".bz2" extensions will be decompressed as gzip and bzip2 streams, respectively.

Internal decompression of 7-zip .7z files is not yet supported; you can pipe such files through p7zip's 7za:

```
7za e -so pages_full.xml.7z |
  java -jar mwdumper.jar --format=sql:1.5 |
  mysql -u <username> -p <databasename>
```

Defaults if no parameters are given:

- read uncompressed XML from stdin
- write uncompressed XML to stdout
- no filtering

Output sinks

```
--output=stdout
    Send uncompressed XML or SQL output to stdout for piping.
    (May have charset issues.) This is the default if no output
    is specified.
--output=file:<filename.xml>
    Write uncompressed output to a file.
--output=gzip:<filename.xml.gz>
    Write compressed output to a file.
--output=bzip2:<filename.xml.bz2>
    Write compressed output to a file.
--output=mysql:<jdbc url>
    Valid only for SQL format output; opens a connection to the
    MySQL server and sends commands to it directly.
    This will look something like:
    mysql://localhost/databasename?user=<username>&password=<password>
```

Output formats

```
--format=xml
    Output back to MediaWiki's XML export format; use this for
    filtering dumps for limited import. Output should be idempotent.
--format=sql:1.4
    SQL statements formatted for bulk import in MediaWiki 1.4's schema.
--format=sql:1.5
    SQL statements formatted for bulk import in MediaWiki 1.5's schema.
    Both SQL schema versions currently require that the table structure
    be already set up in an empty database; use maintenance/tables.sql
    from the MediaWiki distribution.
```

Filter actions

```
--filter=latest
    Skips all but the last revision listed for each page.
    FIXME: currently this pays no attention to the timestamp or
    revision number, but simply the order of items in the dump.
    This may or may not be strictly correct.
--filter=list:<list-filename>
    Excludes all pages whose titles do not appear in the given file.
    Use one title per line; blanks and lines starting with # are
    ignored. Talk and subject pages of given titles are both matched.
--filter=exactlist:<list-filename>
    As above, but does not try to match associated talk/subject pages.
--filter=namespace:[!]<NS_KEY,NS_OTHERKEY,100,...>
```

```
    Includes only pages in (or not in, with "!") the given namespaces.
    You can use the NS_* constant names or the raw numeric keys.
--filter=notalk
    Excludes all talk pages from output (including custom namespaces)
--filter=titlematch:<regex>
    Excludes all pages whose titles do not match the regex.
```

Misc options

```
--progress=<n>
    Change progress reporting interval from the default 1000 revisions.
--quiet
    Don't send any progress output to stderr.
```

Direct connection to MySQL

Example of using mwdumper with a direct connection to MySQL

```
java -server -classpath mysql-connector-java-3.1.11/mysql-connector-java-3.1.11-bin.jar:mwdumper.jar \
  org.mediawiki.dumper.Dumper --output=mysql://127.0.0.1/testwiki?user=wiki\&password=wiki \
  --format=sql:1.5 20051020_pages_articles.xml.bz2
```

Notes:

- You will need the mysql-connector JDBC driver (http://www.mysql.com/products/connector/j/) .
- The JRE does not allow you to mix the -jar and -classpath arguments (hence the different command structure).
- The --output argument must before the --format argument.
- The ampersand in the MySQL URI must be escaped on Unix-like systems.

Example of using mwdumper with a direct connection to MySQL on WindowsXP

Had problems with the example above... this following example works better on XP....

1.Create a batch file with the following text.

```
set class=mwdumper.jar;mysql-connector-java-3.1.12/mysql-connector-java-3.1.12-bin.jar
set data="C:\Documents and Settings\All Users.WINDOWS\Documents\en.wiki\enwiki-20060207-pages-articles.xm
java -client -classpath %class% org.mediawiki.dumper.Dumper "--output=mysql://127.0.0.1/wikidb?user=<user

pause
```

2.Download the mysql-connector-java-3.1.12-bin.jar and mwdumper.jar

3.Run the batch file.

Note

1. It still reports a problem with the import files, "duplicate key"...
2. The class path separator is a ; (semi-colon) in this example; different from the example above.

The "duplicate key" error may result from the page, revision and text tables in the database not being empty, or from character encoding problems. See A note on character encoding.

Trouble shooting

If strange XML errors are encountered under Java 1.4, try 1.5:

- http://java.sun.com/j2se/1.5.0/download.jsp
- http://www.apple.com/downloads/macosx/apple/java2se50release1.html

If mwdumper gives **java.lang.IllegalArgumentException: Invalid contributor** exception, see https://bugzilla.wikimedia.org/show_bug.cgi?id=18328

If it gives **java.lang.OutOfMemoryError: Java heap space** exception, run it with larger heap size, for example `java -Xms128m -Xmx1000m -jar mwdumper.jar ...` (first is starting, second maximum size) (bug 21937 (https://bugzilla.wikimedia.org/show_bug.cgi?id=21937))

Progamming

Contains code from the Apache Commons Compress project for cross-platform bzip2 input/output support (Apache License 2.0).

Performance Tips

To speed up importing into a database, you might try:

- Temporarily remove all indexes and auto_increment fields from the following tables: page, revision and text. This gives a tremendous speed bump, because MySQL will otherwise be updating these indexes after each insert. Don't forget to recreate the indexes afterwards.
- Java's -server option may significantly increase performance on some versions of Sun's JVM for large files. (Not all installations will have this available.)
- Increase MySQL's innodb_log_file_size. The default is as little as 5mb, but you can improve performance dramatically by increasing this to reduce the number of disk writes. (See the my-huge.cnf sample config.)
- If you don't need it, disable the binary log (log-bin option) during the import. On a standalone machine this is just wasteful, writing a second copy of every query that you'll never use.
- Various other wacky tips in the MySQL reference manual (http://dev.mysql.com/mysql/en/innodb-tuning.html) .

Reporting bugs

Bugs can be reported to the mwdumper product in the MediaWiki Bugzilla (https://bugzilla.wikimedia.org/enter_bug.cgi?product=mwdumper) .

Todo

- Add some more junit tests
- Include table initialization in SQL output
- Allow use of table prefixes in SQL output
- Ensure that titles and other bits are validated correctly.
- Test XML input for robustness
- Provide filter to strip ID numbers
- <siteinfo> is technically optional; live without it and use default namespaces
- GUI frontend(s)
- Port to Python? ;)

Change history (abbreviated)

- 2005-10-25: Switched SqlWriter.sqlEscape back to less memory-hungry StringBuffer
- 2005-10-24: Fixed SQL output in non-UTF-8 locales
- 2005-10-21: Applied more speedup patches from Folke
- 2005-10-11: SQL direct connection, GUI work begins
- 2005-10-10: Applied speedup patches from Folke Behrens
- 2005-10-05: Use bulk inserts in SQL mode
- 2005-09-29: Converted from C# to Java
- 2005-08-27: Initial extraction code

Notes

1. ↑ MIT-style license like our other Java/C# tools; boilerplate to be added.

See also

- Manual:Importing XML dumps

Language:	**English** • □□□