

Zone files reference manual

Whois API LLC
<http://www.whoisxmlapi.com>

Copyright ©2010-2017

This data feed subscription is licensed to you or your organization only, you may not resell or relicense the data without explicit written permission from Whois API LLC. Any violation will be prosecuted to the fullest extent of the law.

About this document

File URLs:

1. http://bestwhois.org/zone_file

File version 1.2.

Approved on 2019-05-31.

Contents

1	About zone files	1
2	Directory structure and file formats	1
2.1	Term definitions	1
2.2	Directories with data files	2
2.3	Auxiliary files	2
2.4	Data file formats	3
3	Script for automated downloading	3
4	Decompressing multipart archives	3
4.1	Linux/Mac OS X	4
4.2	Windows	4
5	Handling large csv files	6
5.1	Line terminators in CSV files	6
5.2	Opening a large CSV file on Windows 8 Pro, Windows 7, Vista & XP	7
5.3	How can I open large CSV file on Mac OS X?	8
5.4	Tips for dealing with CSV files from a shell (any OS)	8

1 About zone files

Zone files describe DNS zones. They are text files that are typically used by name servers but they are human readable. For their detailed description, see e.g. https://en.wikipedia.org/wiki/Zone_file.

In this subscription we provide zone files for various top level domains on a given day as well as a list of active domain registrations on that day.

2 Directory structure and file formats

2.1 Term definitions

The gTLDs are subdivided into two categories:

major gtlds: Till 23 October 2017:

`.com, .net, .org, .info, .mobi, .us, .biz, .asia, .name, .tel. aero`

From 23 October 2017 on, for certain organizational changes, the domains `.net` and `.mobi` will appear amongst new gtlds. Hence, the major gtlds' list from this date reads

`.com, .org, .info, .us, .biz, .asia, .name, .tel. aero`

new gtlds: The *new* gTLDs released later by ICANN starting in 2014 in the framework of the “New Generic Top Level Domain Program”, please see this dynamic list:

https://www.whoisxmlapi.com/support/supported_ngtlds.php

2.2 Directories with data files

The files in the subdirectory

`yyyy_MM_dd/$tld`

subdirectory contains files named

`$tld.tar.gz.nnnn`

forming a multipartite tar archive containing the data for the given domain on the given day.

E.g. http://bestwhois.org/zone_file/2017-03-15/com contains the data for “.com” on 2017-03-15, the files are:

`com.tar.gz.0000`
`com.tar.gz.0001`
`com.tar.gz.0002`
`com.tar.gz.0003`
`com.tar.gz.0004`
`com.tar.gz.0005`
`com.tar.gz.0006`
`com.tar.gz.0007`
`com.tar.gz.0008`
`com.tar.gz.0009`
`com.tar.gz.0010`
`com.tar.gz.0011`
`com.tar.gz.0012`
`com.tar.gz.0013`
`com.tar.gz.0014`
`com.tar.gz.0015`

All the files are of size 200 megabytes except for the last one. These files together form the multiparty tar.gz archive. See Section 4 on how to uncompress them.

The subdirectory `latest` has the same contents the latest available `yyyy_MM_dd`.

2.3 Auxiliary files

The subdirectory `http://bestwhois.org/zone_file/status` contains files named `supported_tlds_YYYY_MM_dd`

(e.g. `supported_tlds_2016_08_22`) with a list of all the tlds supported by this data feed on the given day.

The rest of the files are scripts for downloading the feed and the parts of the present documentation in various file formats.

2.4 Data file formats

When the downloaded archives are uncompressed, they contain the following files:

The raw zone file for the domain.

`domain_names.$tld.sorted` is a sorted list of active domains deduced from the zone file on the given day.

For technical reasons, there is a slight difference between the subdirectory structure of the contents of the tar.gz archives:

For major gTLDs there is a file `domain_names.$tld.sorted`, with the sorted domain list, and a subdirectory `/zone/$tld/zonefile` containing the raw zone file named `aero.$tld.YYYY_mm.d.h.mm`, e.g. for the TLD `aero` on 2017-11-08, when the respective archive is uncompressed we have

```
./domain_names.aero_sorted
./zone/aero/zonefile/aero.zone.2017_11_8_8_33
```

.

For new gTLDs we also have the file `domain_names.$tld.sorted`, with the sorted domain list, but the raw zone file resides in a subdirectory named `zone_tldnamed/YYYY_MM_DD/$tld/zone`, under the name `$tld`. For instance, for the TLD `aaa` on 2017-11-08, when the respective archive is uncompressed we have

```
./domain_names.aaa_sorted
./zone_tldnamed/2017_11_08/aaa/zone/aaa
```

3 Script for automated downloading

We provide a downloader script which supports the automated download of zone file data, too. The latest version is available from `http://bestwhois.org/zone_file/download_scripts` in a zip file. (If you subscribe other daily or quarterly data feeds, you do not need to redownload it, it is a universal script for all of our feeds.)

Having downloaded and uncompressed the zip file (e.g. `download_whois_data_beta_0.0.2.zip`) please follow the installation and usage instructions in the included documentation.

You have to choose the feed “`zone_file`”, and the only available format, that is, “`raw`” to download data documented here.

4 Decompressing multipart archives

Here we describe how to uncompress the downloaded multipartite tar archives on various platforms. These are binary files just split into parts, so if you join them again they will become big archive files which can then be handled by the usual uncompression utilities.

Note: if you find a single file like “foo.tar.gz.0000” in some feeds, it is a “single-part” multipart archive, this is just a complete archive. This may seem counterintuitive, however, when the uncompression is automated it is more convenient: there is no need to check whether the file is a single archive or it is multipartite. So we follow this convention in case of some feeds. Unfortunately, however, some windows utilities are confused by this convention. But concatenating a single file into a file named “.tar.gz” consists in just renaming it; so if there is a single file with the same name with “0000”, you can just remove the “0000” from the end and obtain a valid archive.

In what follows we describe how to concatenate and uncompress multipart files in various platforms.

4.1 Linux/Mac OS X

You can do the job in a terminal with shell commands:

1. Download all tar.gz parts to your local disk.
2. To restore a given archive, the following command-line is suitable:

```
cat $INPUT_DIR/$ARCHIVE_NAME.* > $ARCHIVE_NAME;
```

e.g. for com.tar.gz.*, the command to be issued in the directory where the files reside is

```
cat com.tar.gz.* > ./com.tar.gz
```

resulting in the file named “com.tar.gz”

3. Uncompress the file with (tar):

```
tar -xvzf $ARCHIVE_NAME
```

e.g. for the previous example (data for .com):

```
tar -xzvf com.tar.gz
```

4.2 Windows

1. Download all tar.gz parts to your local disk. If there is just a single part, simply rename it to omit “.0000” from the end, and uncompress it with a suitable unarchiver. If there are multiple parts, proceed.
2. To restore a given tar.gz archive, you can use the DOS command-line. (If you do not want to use command-line, proceed to the next item.) So open a command-prompt and go to the directory where the downloaded files reside. Run the command of the form

```
copy /B $INPUT_DIR/$ARCHIVE_NAME.0000 + $INPUT_DIR/$ARCHIVE_NAME.0001 + $INPUT_DIR/$ARCHIVE_NAME.0002 + ... $ARCHIVE_NAME
```

where the “...” has to be replaced by the complete list of the remaining files in an increasing order, separated by the “+” character, whereas the last argument is the file to store the result in. For instance, for net.tar.gz.*, which are only two files, the command to be issued in the directory where the files reside is

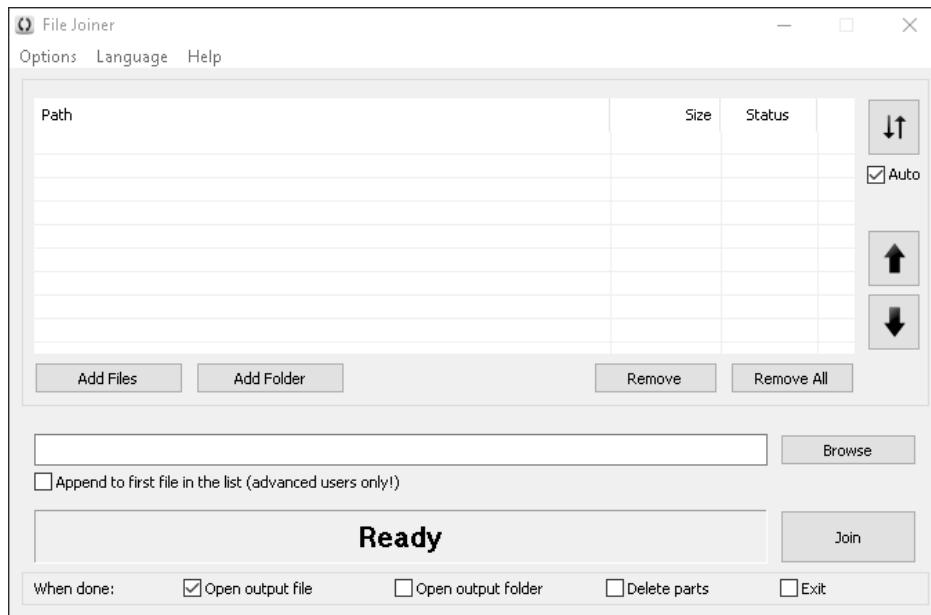
```
copy net.tar.gz.0000 + net.tar.gz.0001 net.tar.gz
```

which will create "net.tar.gz", a standard tar.gz archive. **Caution!** Do not use asterisk like `net.tar.gz.*`. The source files will be processed in a wrong order if you do so, resulting in an incorrect output.

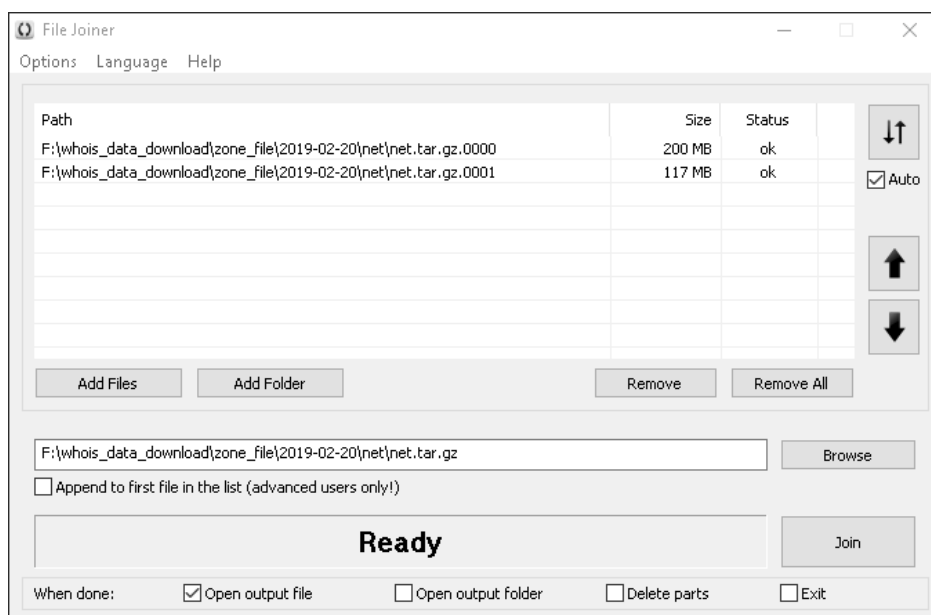
3. A GUI alternative to the previous step: if you do not want to use command-line, you need a program capable of joining files. A good option is "File Joiner", a free utility available for both 32 and 64 bit platforms from here:

<http://www.igorware.com/file-joiner>

After installing and starting the program you shall see the following window:



First press "Add Files" and select to add all the downloaded files which are a part of the same archive. In my case these are `net.tar.gz.0000` and `net.tar.gz.0001`, but you may have more of them. By default the program will put them into the appropriate order (increasing numbers from the top to the bottom), and propose a logical output file name in the entry box to the left of the "browse" button. In my case, the output will be `net.tar.gz` in the same directory where the downloaded files reside:



Finally you need to press the "Join" button, and the archive will be restored. In addition, it will open it with your default compression utility automatically. (To disable this deselect the "Open output file option" selected by default. You can find some additional useful options in the bottom line, "When done:").

4. Uncompress tar.gz archive resulting from the previous step with some uncompression utility. This can be done e.g. with WinRAR available from <https://www.win-rar.com/download.html> having a graphical user interface, or with 7-zip, <http://www.7-zip.org/download.html> from command-line. Follow the instructions of the chosen application.

5 Handling large csv files

In this Section we describe some possible ways how to view or edit large csv files on various operating systems.

5.1 Line terminators in CSV files

CSV files are plain text files by nature. Their character encoding is UTF8 Unicode, but even UTF8 files can have three different formats which differ in the line terminator characters:

1. Unix-style systems, including Linux and BSD use a single "LF"
2. DOS and Windows systems use two characters, "CR" + "LF"
3. Legacy classic Mac systems used to use "CR"

as the terminator character of lines. While the third option is obsolete, the first two types of files are both prevalent.

The files provided by WhoisXML API are generated with different collection mechanisms, and for historic reasons both formats can occur. Even if they were uniform with this respect, some download mechanisms can include automatic conversion, e.g. if you download them with FTP, some clients convert them to your system's default format. While most software, including the scripts provided by us handle both of these formats properly, in some applications it is relevant to have them in a uniform format. In what follows we give some hint on how to determine the format of a file and convert between formats.

To determine the line terminator the easiest is to use the "file" utility in your shell (e.g. BASH, also available on Windows 10 after installing BASH on Ubuntu on Windows): for a DOS file, e.g. "foo.txt" we have (" \$" stands for the shell prompt):

```
$ file foo.csv
foo.txt: UTF-8 Unicode text, with CRLF line terminators
```

whereas if "foo.txt" is Unix-terminated, we get

```
$ file foo.csv
foo.txt: UTF-8 Unicode text
```

or something alike, the relevant difference is whether "with CRLF line terminators" is included.

To convert between the formats, the command-line utilities "todos" and "fromdos" can be used. E.g.

```
$ todos foo.txt
```

will turn "foo.txt" into a Windows-style CR + LF terminated file (regardless of the original format of "foo.txt"), whereas using "fromdos" will do the opposite. The utilities are also capable of using STDIN and STDOUT, see their manuals.

These utilities are not always installed by default, e.g. on Ubuntu you need to install the package "tofrodos". Formerly the relevant utilities were called "unix2dos" and "dos2unix", you may find them under this name on legacy systems. These are also available for DOS and Windows platforms from

<https://www.editpadpro.com/tricklinebreak.html>

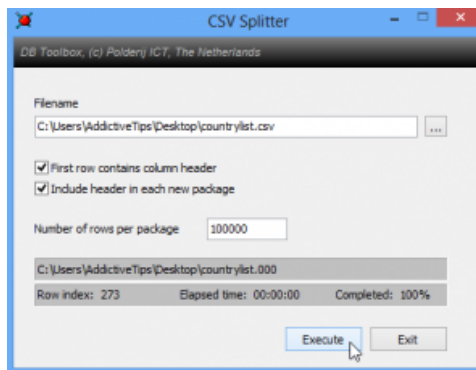
In Windows PowerShell you can use the commands “GetContent” and “SetContent” for the purpose, please consult their documentation.

5.2 Opening a large CSV file on Windows 8 Pro, Windows 7, Vista & XP

First solution: You can use an advanced editor that support handling large files, such as

- Delimit Editor: <http://delimitware.com>
- reCsvEdit: <http://recsveditor.sourceforge.net>

Second solution: You can split a CSV file into smaller ones with CSV Splitter (<http://erdconcepts.com/dbtoolbox.html>).



Third solution: You may import csv files into the spreadsheet application of your favorite office suite, such as Excel or LibreOffice Calc.

Note: If you want to use MS Excel, it would be advisable to use a newer version of Excel like 2010, 2013 and 2016.

Fourth solution: On Windows, you can also use the bash shell (or other UNIX-style shells) which enables several powerful operations on csv files, as we describe here in Section 5.4 of this document.

In order to do so,

- On Windows 10, the Anniversary Update brings “Windows subsystem for Linux” as a feature. Details are described e. g. in this article:

<https://www.howtogeek.com/265900/everything-you-can-do-with-windows-10s-new-bash-shell>

- In professional editions of earlier Windows systems the native solution to have an Unix-like shell was the package “Windows services for Unix”. A comprehensive description is to be found here:

https://en.wikipedia.org/wiki/Windows_Services_for_UNIX

- There are other Linux-style environments, compatible with a large variety of Windows OS-es, such as cygwin:

<https://www.cygwin.com>

or mingw:

<http://www.mingw.org>

Having installed the appropriate solution, you can handle your csv-s also as described in Section 5.4.

5.3 How can I open large CSV file on Mac OS X?

First solution: You can use one of the advanced text editors such as:

- BBEdit: <https://www.barebones.com/products/bbedit>
- MacVim: <http://macvim-dev.github.io/macvim>
- HexFiend: <http://ridiculousfish.com/hexfiend>
- reCsvEdit: <http://recsveditor.sourceforge.net>

Second solution: You may import csv files into the spreadsheet application of your favorite office suite, such as Excel or LibreOffice Calc.

Note: If you want to use MS Excel, it would be advisable to use a newer version of Excel like 2010, 2013 and 2016.

Third solution: Open a terminal and follow Subsection 5.4

5.4 Tips for dealing with CSV files from a shell (any OS)

You can split csv files into smaller pieces by using the shell command `split`, e. g.

```
split -l 2000 sa.csv
```

shall split `sa.csv` into files containing 2000 lines each (the last one maybe less). The “chunks” of the files will be named as `xaa`, `xab`, etc. To rename them you may do (in bash)

```
for i in x??; do mv "$i" "$i.csv"; done
```

so that you have `xaa.csv`, `xab.csv`, etc.

The `split` command is described in detail in its man-page or here:

http://www.gnu.org/software/coreutils/manual/html_node/split-invocation.html

We also recommend `awk`, especially GNU `awk`, which is a very powerful tool for many purposes, including the conversion and filtering csv files. It is available by default in most UNIX-style systems or subsystems. To get started, you may consult its manual:

https://www.gnu.org/software/gawk/manual/html_node/Getting-Started.html

End of manual.