

Daily Domain Name Whois Updates Reference Manual (ccTLDs)

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

Copyright ©2010-2021

Important: product end of life 2024

The present documentation describes legacy products.

- **The support for all products and services described here is scheduled to terminate on 1 March 2024.**
- **Their end of life and termination is scheduled to 31 December 2024.**

WhoisXML API, Inc. offers an improved service covering and extending the functionality of the here described ones.

For more information, visit

<https://newly-registered-domains.whoisxmlapi.com>

Please update your business processes on time.

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

The present document is available in html, pdf and unicode text format from the following locations.

Primary URL:

http://www.domainwhoisdatabase.com/docs/Daily_CCTLD_WHOIS_Updates_Reference_Manual

Additional URLs:

1. <http://bestwhois.org/cctld-domain-name-data/domain-names-whois>
2. <http://domainwhoisdatabase.com/domain-list/docs>

File version 2.24.

Approved on 2023-09-13.

A full list of available WhoisXML API data feed manuals is available at

<http://www.domainwhoisdatabase.com/docs>

Contents

PRODUCT END OF LIFE	1
1 Introduction	3
1.1 About the data feeds	3
1.2 Download schedule	3
1.2.1 When are the domain name data provided	3
1.2.2 Normal timings	4
1.2.3 Unpredictable and irregular delays	4
1.2.4 Schedule information	5
2 Feeds, download directory structures and data formats	5
2.1 Supported and unsupported TLDs	5
2.2 On the data feeds containing changes	5
2.3 Data feeds, URLs, directory structures and data formats	6
2.3.1 Feed: cctld_registered_domain_names_new	6
2.3.2 Feed: cctld_registered_domain_names_dropped	7
2.3.3 Feed: cctld_registered_domain_names_whois	7
2.3.4 Feed: cctld_registered_domain_names_dropped_whois	8
2.3.5 Feed: cctld_discovered_domain_names_new	9
2.3.6 Feed: cctld_discovered_domain_names_whois	10
2.3.7 Feed: cctld_discovered_domain_names_whois_archive	11
2.4 Supported tlds	11
2.5 Auxiliary data on actual time of data file creation	12
2.6 Data file hashes for integrity checking	13
3 CSV file formats	13
3.1 The use of CSV files	13
3.1.1 Loading CSV files into MySQL and other database systems	14
3.2 File formats	14
3.3 Data field details	16
3.4 Maximum data field lengths	17
3.5 Standardized country fields	17
4 JSON file availability	17
5 Database dumps	17
5.1 Software requirements for importing mysql dump files	17
5.2 Importing mysql dump files	18
5.2.1 Loading everything (including schema and data) from a single mysqldump file	18
5.3 Database schema	18
5.4 Further reading	21
6 Client-side scripts for downloading data, loading into databases, etc.	21
7 Tips for web-downloading data	22
7.1 When, how, and what to download	22
7.2 Downloaders with a GUI	24
7.3 Command-line downloaders	25

8	Handling large csv files	27
8.1	Line terminators in CSV files	27
8.2	Opening a large CSV file on Windows 8 Pro, Windows 7, Vista & XP	28
8.3	How can I open large CSV file on Mac OS X?	29
8.4	Tips for dealing with CSV files from a shell (any OS)	29
9	Daily data collection methodology	29
9.1	Domain life cycle and feed timings	30
9.2	Time data accuracy	30
10	Alternative data collection methodologies for ccTLDs	31
11	Data quality check	31
11.1	Quality check: csv files	32
11.2	Quality check: MySQL dumps	32
12	Access via SSL Certificate Authenticaon	32
12.1	Setup instructions	32
12.1.1	Microsoft Windows	32
12.1.2	Mac OS X	37
12.1.3	Linux	39
12.2	Accessible URLs	41
13	FTP access of WHOIS data	41
13.1	FTP clients	41
13.2	FTP access	41
13.3	FTP directory structure of legacy and quarterly subscriptions	42
13.4	FTP firewall settings for legacy subscriptions	42

1 Introduction

1.1 About the data feeds

Our daily data feeds provide whois data for newly registered domains in both parsed and raw formats for download as database dumps (MySQL or MySQL dump) or CSV files.

1.2 Download schedule

1.2.1 When are the domain name data provided

- Each file named after a given date holds data according to the zone file and WHOIS data published on the day in the filename. In the case of the “discovered” feeds, this is the date of the discovery.
- The typical availability times are to be found at the following link:

http://domainwhoisdatabase.com/docs/whoisxmlapi_daily_feed_schedule.html

The list of availability times contains the times when the given feeds’s data in the given format was available at least in 95% of the cases in the last 3 months back from the last update date.

In the following a detailed description is provided on the reasons of these timings and their possible fluctuations.

1.2.2 Normal timings

In order to understand when a newly registered domain will be visible in our WHOIS data feeds or through our API it is necessary to understand how WHOIS data are generated and get to our subscribers:

1. The domain gets registered at the domain name registrar. The some of the date fields in the WHOIS record such as `createdDate` or `expiresDate` normally contain time zone information, therefore these dates should be interpreted according to this information. It might be the case that the day or month of the same date is different in your time zone. We recommend to use the normalized fields we provide, such as `standardRegCreatedDate`". These are all given in UTC.
2. The registrar processes the registrations and publishes new WHOIS data. Normally the registrars publish WHOIS data of the registered domains once a day. Therefore the information on the registration can collect up to 24 hours delay compared to the date given in the WHOIS record.
3. We collect and process WHOIS data from the registrars. This typically takes up to 12 hours. Another source of the delay might be the difference between the time of publication mentioned at Step 2. and our collection schedule. (The information available on the time of publication of WHOIS data by the registrar is limited and it can vary even for the same registrar). As for the processing times of various data feed, typical upper bounds on the time of availability of different type of WHOIS data is available in the description of the data feeds in the present manual These estimates come from a time series analysis of the availability data of the last few years, ignoring irregular events, see the next section.
4. The user receives WHOIS data. It is also important to interpret the actual download time and some of the WHOIS records in the appropriate time zone and convert it to the desired time zone.

As a consequence, even under normal circumstances there is 12-36-hour real delay between the date in the WHOIS record and its availability in our database. Looking at the dates only, it can seemingly result in up to 3 days of delay, which might even be more if the date is not interpreted together with time in the appropriate time zone for some reason.

1.2.3 Unpredictable and irregular delays

In addition to the regular delays mentioned above, there might be additional reasons of the delay occurring occasionally. Some examples:

Problems with registrars. Some of the registrars introduce obstacles in getting data from them. Even some large registrars in large domains tend to be unreliable in providing bulk registration data. Sometimes the provided WHOIS data are incomplete, and it is not possible to obtain them from alternative, more accurate or reliable sources. For such external reasons, unfortunately, the WHOIS data of some domains we can provide are sometimes incomplete and some registrations appear with a more significant delay.

Technical obstacles. The set of domain WHOIS data is huge. Even though we employ cutting-edge software and hardware solutions to store, process and provide these data, the task sometimes reaches the de facto limitations of current hardware and software technologies. Therefore, in spite of all of our efforts to avoid any delay arising from this issue, we cannot, unfortunately, deny that in some cases there is some additional delay due to such issues, too.

1.2.4 Schedule information

An approximate schedule is provided in the detailed description of each feed. *Note, however, that the downloading and preprocessing of data is a compute-intensive task, hence, the schedule has to be considered as approximate.* As a rule of thumb: csv files are prepared mostly on time while WHOIS data and mysql dumps, whose preparation time depends on external resources and require more runtime, has usually more delay *and the preparation time may have a significant variance compared to the schedule given below.*

We provide an opportunity to precisely verify if certain files are already prepared and ready to be downloaded, both in the form of an RSS feed and other methods. This is described in Section 2.5.

2 Feeds, download directory structures and data formats

2.1 Supported and unsupported TLDs

By a “supported top-level domain (TLD)” it is meant that obtaining WHOIS data is addressed by the data collection procedure, and thus there are WHOIS data provided. (In some cases bigger second-level domains (SLDs) are treated separately from their TLDs in the data sources as if they were separate TLDs, hence, we refer to these also as “TLDs” in what follows.) The set of supported TLDs can vary in time, thus it is specified for each quarterly database version or day in case of quarterly and daily data sources, respectively. See the detailed documentation of the data feeds on how to find the respective list.

If a TLD is unsupported, it means that the given data source does not contain WHOIS data for the given TLD. There are many for reasons for which a domain is unsupported by our data sources; typically the reason behind is that it does not have a WHOIS server or any other source of WHOIS data or it is not available for replication for technical or legal reasons. A list of TLDs domains which are constantly unsupported by all feeds is to be found at

https://www.whoisxmlapi.com/support/unsupported_tlds.txt

For these domains we provide a file limited information that include just name server info in certain data sources; notably in quarterly feeds.

As of the list of supported TLDs, these are listed in auxiliary files for each data source separately. See the documentation of the auxiliary files for details.

2.2 On the data feeds containing changes

Many of the data feeds contain information about changes, such as “newly registered domains”. It is important to note that the change is detected by us via the analysis of the respective zone files: a domain appears in a daily feed of this type if there has been a change in the respective zone file, for instance, it has appeared and was not there directly before.

For this reason there might be a contradiction between the appearance of the domain as a newly registered one and the date appearing in the “createdDate” field of the actual WHOIS record. It occurs relatively frequently that the domain name disappears from the zone file, then appeared again. (Sometimes some domains are even not in the zone file and when we check by issuing a DNS request the domain is actually found by the name server.)

To illustrate this issue: given a domain with

Updated Date: 2018-04-23T07:00:00Z

Creation Date: 2015-03-09T07:00:00Z

may appear on 2018-04-23 as a newly registered one. And, unfortunately, sometimes the “updated date” in the WHOIS record is also inaccurate, and therefore the domain appears as new in contrast to any data content of the WHOIS record.

Looking closer at the reasons why a domain disappears temporarily from the zone file (and therefore it gets detected as new by us upon its reappearance), one finds that the typical reason is due to certain domain status values. Namely, a domain name is not listed in zone-file if it is in either of the following statuses:

Server-Hold
Client-Hold
PendingDelete
RedemptionPeriod

For instance, if a domain that has expired and gone into the `redemptionPeriod`, it will not show in the zone file, but if the current owner redeems it the next day, it will reappear. A registrar may also deem that a domain has violated their terms of service and put them on `clientHold` until they comply. This removes the domain at least temporarily from the zone file as well.

As a rule of thumb, if you really want to decide whether a domain which has just appeared in the zone file (and thus is included into our respective feeds) was a newly registered on a given day, please add a check of the contents of the “`createdDate`” field in the course of processing the data.

The discrepancies in other feeds related to changes, e.g. with data of dropped domains can be understood along the same lines.

For a more detailed explanation about this, see also Section 9.

2.3 Data feeds, URLs, directory structures and data formats

Important note: *The directories discussed in this section may contain subdirectories and/or files not described here. These are temporary, they are there for technical reasons only. Please ignore them.*

2.3.1 Feed: `cctld_registered_domain_names_new`

Newly registered domains for ccTLDs.

URL:

`https://bestwhois.org/cctld_domain_name_data/domain_names_new`

Directory structure. The directory and file naming convention is as follows:

`yyyy-MM-dd/add.$tld.csv`

for example, if you want to get newly registered `.ru` domain names on February 15th, 2017, the corresponding file is `2017-02-15/add.ru.csv`

`AvgDailyDomains.txt`

contains information about 30-day averages of the record count in this feed, that is, the 30-day average number of new domains in all ccTLDs, for informational purposes.

Important note on the schedule: the data for this feed are generated in two phases, whose completion time differs. The data for a part of the TLDs concludes typically on the given date, whereas the data for `fr`, `re`, `pm`, `tf`, `wf`, `yt` appear about one day later. Hence, the `status` directory has files

`YYYY-MM-DD-download-ready-csv-partial`

indicating that the data for all but the above domains are complete, whereas the presence of the

YYYY_MM_DD_download_ready.csv

indicates the ultimate completion. The partial completion is also reflected in the rss notifications.

Data file format. The domain names are listed one per line, with domain extension, for example 01info.ru in add.ru.csv.

2.3.2 Feed: cctld_registered_domain_names_dropped

Newly dropped domains for ccTLDs

URL:

https://bestwhois.org/cctld_domain_name_data/domain_names_dropped

Directory structure. The directory and file naming convention is as follows:

yyyy-MM-dd/dropped.\$tld.csv

for example, if you want to get newly dropped .ru domain names on February 15th, 2017, the corresponding file is 2017-02-15/dropped.ru.csv

Data file format. The domain names are listed one per line, with domain extension, for example 0media.ru.

2.3.3 Feed: cctld_registered_domain_names_whois

Whois data for newly registered domains for ccTLDs.

URL:

https://bestwhois.org/cctld_domain_name_data/domain_names_whois

Directory structure. The directory and file naming convention is as follows:

yyyy_MM_dd_\$tld.csv.gz

A compressed(gzipped) file containing parsed whois data for a tld on a date. For example: 2017_02_15_ru.csv.gz contains the compressed whois data csv file for newly registered .ru from February 15th, 2017.

add_yyyy_MM_dd_\$tld

The uncompressed directory containing csv files representing parsed whois data of newly registered domains for the date. For example: add_2017_02_15/ru/ contains the whois data csv files for newly registered .ru from February 15th, 2017. The file names have a format \$p_\$i.csv where \$p is the prefix of the domain name(0-9,a-z) and i is an 1-based index. For example: a_1.csv is the 1st file contains all domain names starting with the letter 'a' and their whois records

full_yyyy_MM_dd_\$tld.csv.gz

A compressed(gzipped) file containing full whois data (parsed and raw text) for a tld on a date. For example: full_2017_02_15_ru.csv.gz contains the compressed full whois data csv file for newly registered .ru from February 15th, 2017

`add.full-yyy-MM-dd/$tld`

The uncompressed directory containing csv files representing full whois data (parsed and raw text) of newly registered domains for the date. For example: `add.full-2017-02-15/ru/` contains the compressed whois data csv files for newly registered .ru from February 15th, 2017. The file names have a format `$p-$i.csv` where `$p` is the prefix of the domain name(0-9,a-z) and `i` is an 1-based index. For example: `a-1.csv` is the 1st file contains all domain names starting with the letter 'a' and their whois records.

`add.mysql-dump-yyy-MM-dd-$tld.sql.gz`

The compressed (gzipped) SQL database dump(mysql-dump) file containing parsed and raw whois data for a tld on a date. For example:

`add.mysql-dump-2017-02-15-ru.sql.gz`

contains the compressed mysql-dump file for newly registered .ru from February 15th, 2017.

Data file format. The csv and sql files whose name end with gz are compressed with gzip. The detailed csv file format is described in Section 3, while the database schema are to be found in Section 5.3.

2.3.4 Feed: `cctld-registered-domain-names-dropped-whois`

Whois data for newly dropped domains for ccTLDs. *Note: in case of some of the ccTLDs the whois information becomes unavailable very shortly after the act of dropping the domain. Hence, it is possible that in this feed there are data for much less domains than in the feed `cctld-registered-domain-names-dropped`, and this is normal.*

URL:

`https://bestwhois.org/cctld-domain-name-data/domain-names-dropped-whois`

Directory structure. The directory and file naming convention is as follows:

`yyy-MM-dd-$tld.csv.gz`

A compressed(gzipped) file containing parsed whois data for a tld on a date. For example: `2017-02-15-ru.csv.gz` contains the compressed whois data csv file for newly dropped .ru from February 15th, 2017.

`dropped-yyy-MM-dd/$tld`

The uncompressed directory containing csv files representing parsed whois data of newly dropped domains for the date. For example: `dropped-2017-02-15/ru/` contains the compressed whois data csv files for newly dropped .ru from February 15th, 2017. The file names have a format `$p-$i.csv` where `$p` is the prefix of the domain name(0-9,a-z) and `i` is an 1-based index. For example: `a-1.csv` is the 1st file contains all domain names starting with the letter 'a' and their whois records

`full-yyy-MM-dd-$tld.csv.gz`

A compressed(gzipped) file containing full whois data (parsed and raw text) for a tld on a date. For example: `full-2017-02-15-ru.csv.gz` contains the compressed full whois data csv file for newly dropped .ru from February 15th, 2017

`dropped_full_YYYY_MM_dd/$tld`

The uncompressed directory containing csv files representing full whois data (parsed and raw text) of newly dropped domains for the date. For example: `dropped_full_2017_02_15/ru/` contains the compressed whois data csv files for newly dropped .ru from February 15th, 2017. The file names have a format `$p_$i.csv` where `$p` is the prefix of the domain name(0-9,a-z) and `i` is an 1-based index. For example: `a_1.csv` is the 1st file contains all domain names starting with the letter 'a' and their whois records.

`dropped_mysqldump_YYYY_MM_dd.$tld.sql.gz`

The compressed (gzipped) SQL database dump(mysqldump) file containing parsed and raw whois data for a tld on a date. For example:

`dropped_mysqldump_2017_02_15_ru.sql.gz`

contains the compressed mysqldump file for newly dropped .ru from February 15th, 2017.

Data file format. The csv and sql files whose name end with gz are compressed with gzip. The detailed csv file format is described in Section 3, while the database schema are to be found in Section 5.3.

2.3.5 Feed: `cctld_discovered_domain_names_new`

Newly discovered domains for ccTLDs. These are newly discovered domains from our world-wide third party DNS sensors (some of which tend to be newly registered). See also Section 10.

URL:

`https://domainwhoisdatabase.com/domain_list/domain_names_new`

Directory structure. The directory and file naming convention is as follows:

`YYYY-MM-dd.txt`

a text file containing raw data, both from ccTLDs and gTLDs which were discovered on the given day.

`YYYY-MM-dd/$tld`

the newly discovered domains for the given date, one text file per tld.

`daily_tlds/YYYY-MM-dd.domains.tlds`

ccTLDs for the given day for which data are provided in the feed. Historic files may contain data for gTLDs, too. Provided for each day.

`daily_tlds/YYYY-MM-dd.whois.tlds`

ccTLDs for the given day for which whois data can be collected. As this is not possible for all of them, this list is a subset of the one in the respective `.domains.tlds` file. Not provided for each day.

`by_tld/$tld/YYYY_MM_dd.$tld.txt`

Data categorized by tld, e.g. `by_tld/tel/2017_02_15_tel.txt` contains data for .tel on February 15, 2017. *Note: your subscription may be limited to certain tlds. In this case your access is restricted to one of these subdirectories.* *Note: In case of this feed, the `supported_tlds` file in the `status` subdirectory bears no relevance.*

Data file format. The domain names are listed one per line, with domain extension.

ccTLDs and gTLDs in the `daily_tlds` subdirectory listed in the `.tlds` files in a single line, separated by commas.

2.3.6 Feed: `cctld_discovered_domain_names_whois`

Whois data for newly discovered domains for ccTLDs. These are newly discovered domains from our world-wide third party DNS sensors (some of which tend to be newly registered). See also Section 10. The data are available here for 65 days. The files older than 65 days are moved to the feed

`cctld_discovered_domain_names_whois_archive`

URL:

`https://domainwhoisdatabase.com/domain_list/domain_names_whois`

Directory structure. The directory and file naming convention is as follows:

`yyyy_MM_dd_$tld.csv.gz`

A compressed(gzipped) file containing parsed whois data for a tld on a date. For example:

`2017_02_15_ru.csv.gz` contains the compressed whois data csv file for newly discovered .ru from February 15th, 2017.

`add_YYYY_MM_dd/$tld`

The uncompressed directory containing csv files representing parsed whois data of newly discovered domains for the date. For example: `add_2017_12_15/ru/` contains the whois data csv files for newly discovered .ru from February 15th, 2017. The file names have a format `$p_$i.csv` where `$p` is the prefix of the domain name(0-9,a-z) and `i` is an 1-based index. For example: `a_1.csv` is the 1st file contains all domain names starting with the letter 'a' and their whois records

`full_YYYY_MM_dd_$tld.csv.gz`

A compressed(gzipped) file containing full whois data (parsed and raw text) for a tld on a date. For example: `full_2017_02_15_ru.csv.gz` contains the compressed full whois data csv file for newly discovered .ru from February 15th, 2017

`add_full_YYYY_MM_dd/$tld`

The uncompressed directory containing csv files representing full whois data (parsed and raw text) of newly discovered domains for the date For example: `add_full_2017_02_15/ru/` contains the compressed whois data csv files for newly discovered .ru from February 15th, 2017. The file names have a format `$p_$i.csv` where `$p` is the prefix of the domain name(0-9,a-z) and `i` is 0-based index. For example: `a_1.csv` is the 1st file contains all domain names starting with the letter 'a' and their whois records.

`add_mysqldump_YYYY_MM_dd/$tld/add_mysqldump_YYYY_MM_dd_$tld.sql.gz`

The compressed (gzipped) SQL database dump(mysqldump) file containing parsed and raw whois data for a tld on a date. For example:

`add_mysqldump_2017_02_15/ru/add_mysqldump_2017_02_15_ru.sql.gz`

contains the compressed mysqldump file for newly discovered .ru from February 15th, 2017.

Data file format. The csv and sql files whose name end with gz are compressed with gzip. The detailed csv file format is described in Section 3, while the database schema are to be found in Section 5.3.

2.3.7 Feed: `cctld_discovered_domain_names_whois_archive`

Historic data from the data feed `cctld_discovered_domain_names_whois`.

URL:

`https://domainwhoisdatabase.com/domain_list/domain_names_whois_archive`

Directory structure. The directory and file naming convention is as follows: In the root directory there is a directory for each available year. Within each year-named subdirectory there is with a name of the two-digit month. Thus the information in the root directory of the feed is divided into subdirectories

`YYYY/MM`

e.g.

`2017/08`

for August 2017. Within these subdirectories the naming conventions are the same as in the non-archive version of the feed, `cctld_discovered_domain_names_whois`, with the following differences:

- There are no md5 sums provided, there are hashes subdirectories.
- Within the `status` subdirectories (c.f. Section 2.4), the `supported_tlds` files are available from the date 2016-10-10 on. For earlier dates the `added_tlds` files can be used to determine the domains for which data are provided. The `added_tlds` files are available for the later dates, too.
- For 2016-01 to 2016-03, the format of data was different: there are no csv files but MySQL dumps only. They are organized into TLD-named subdirectories, e.g. the subdirectory

`2016/01/ac/`

holds the file

`add_mysqldump_2016_01_31_ac.sql.gz`

, the data in a gzipped MySQL dump for the TLD “ac”. This extraordinary format is not supported by our downloader scripts, please download these files manually.

- For 2016-04, there are only cumulative data for the whole month, they are in non-compressed csv-s, e.g.

`2016/04/uk/c_1.csv`

. This extraordinary format is not supported by our downloader scripts, please download these files manually.

Data file format. The csv and sql files whose name end with gz are compressed with gzip. The detailed csv file format is described in Section 3, while the database schema are to be found in Section 5.3.

2.4 Supported tlds

The files

`supported_tlds_YYYY_mm_dd`

in the subdirectory of the feeds, e.g.

`http://bestwhois.org/cctlds_domain_name_data/domain_names_whois/status/supported_tlds_YYYY_mm_dd`

contain information on tlds supported on a particular day. Similarly, the files

`added_tlds_YYYY-mm-dd`

contain a list of those tlds which have new data on a particular day.

All data formats are identical.

2.5 Auxiliary data on actual time of data file creation

Important note: *the time information seen on the web server in the file listings is always to be understood in GMT/UTC.* Users who do automated processing with scripts should note that when downloaded with the `wget` utility to a local file, this file is saved under the same datetime as it was found on the server, but it appears locally according to the local user's locale settings.

For example, a file seen on a subdirectory of `bestwhois.org` displayed in the listing having a time 21:30, this is GMT. However, for instance, in Central European Summer Time (CEST) it is 23:30, so if you reside in this latter timezone and your computer is set accordingly, this will appear in the local file listing.

Each feed subdirectory contains a `status` subdirectory, e.g. the feed `domain_names_whois` has

`http://bestwhois.org/cctld_domain_name_data/domain_names_whois/status`

Within the `status` directory each daily non-archive feed has a file named `download_ready_rss.xml`, which is an RSS feed providing immediate information if the data in the feed in a given format are finalized and ready for downloading. For instance, if the regular csv data of the above mentioned `domain_names_whois` feed are ready for downloading, the in RSS feed

`https://bestwhois.org/cctld_domain_name_data/domain_names_whois/status/download_ready_rss.xml`

the following entry will appear:

```
{"data_feed": "cctld_registered_domain_names_whois",  
"format": "regular_csv",  
"day": "2021-08-24",  
"available_from": "2021-08-25 18:09:52 UTC"}
```

indicating that the regular csv data of the `cctld_registered_domain_names_whois` feed for the day 2021-01-30 are ready for downloading from 2021-01-30 23:05:42 UTC. The entry is in JSON format so it is suitable for a machine-based processing: the maybe most efficient way to download complete data as soon as they are available is to observe this feed and initiate the download process as soon as the RSS entry appears. (Premature downloading on the other hand can produce incomplete data.)

As another indication of the readiness of a given data set, the `status` subdirectories in each feed's directory contain files which indicate the actual completion time of the preparation of the data files described in Section 1.2. These can be used to verify if a file to be downloaded according to the schedule is really complete and ready to be downloaded. Namely, if a file

`yyyy_MM_dd_download_ready_csv`

exists in the `status` subdirectory then the generation of

`yyyy_MM_dd_$tld.csv.gz`

has been completed by the time of the creation datetime of the file

`yyyy_MM_dd_download_ready_csv`

and it is ready to be downloaded since. The contents of `yyyy_MM_dd_download_ready_csv` are irrelevant, only their existence and creation datetime are informative. Similarly, the files

`yyyy_MM_dd_download_ready_csv`

correspond to the data files

`yyyy_MM_dd_${tld}.csv.gz`

while the

`yyyy_MM_dd_download_ready_mysql`

correspond to the

`add_mysql_yyyy_MM_dd_${tld}.csv.gz`

data files.

The text files

`exported_files`

in the `status` subdirectories, wherever they exist, provide information about the filename, file size and modification time for each of the relevant data files. This file is updated whenever a file is regenerated.

2.6 Data file hashes for integrity checking

Each feed subdirectory contains a

`hashes`

subdirectory, e.g. the feed `domain_names_whois` contains

`http://bestwhois.org/domain-name-data/domain-names_whois/hashes`

These subdirectories contain contain md5 and sha hashes of the downloadable data files accessible from their parent directories. These can be used to check the integrity of downloaded files.

3 CSV file formats

3.1 The use of CSV files

CSV files (Comma-Separated Values) are text files whose lines are records whose fields are separated by the field separator character. Our CSV files use Unicode encoding. The line terminators may vary: some files have DOS-style CR+LF terminators, while some have Unix-style LF-s. *It is recommended to check the actual file's format before use.* The field separator character is a comma (“,”), and the contents of the text fields are between quotation mark characters.

CSV-s are very portable. They can also be viewed directly. In Section 8 you can find information on software tools to view the contents and handle large csv files on various platforms.

3.1.1 Loading CSV files into MySQL and other database systems

In Section 6 we describe client-side scripts provided for end-users. The available scripts include those which can load csv files into MySQL databases. In particular, a typical usecase is to load data from CSV files daily with the purpose of updating an already existing MySQL WHOIS database. This can be also accomplished with our scripts.

CSV files can be loaded into virtually any kind of SQL or noSQL database, including PostgreSQL, Firebird, Oracle, MongoDB, or Solr, etc. Some examples are presented in the technical blog available at

<https://www.whoisxmlapi.com/blog/setting-up-a-whois-database-from-whoisxml-api-data>.

3.2 File formats

There are 2 types of CSVs and 1 type of Database dump for whois records.

- The files are generally compressed in .tar.gz, use the following commands/tools to uncompress
 - on Linux and other UNIX-style systems, use `tar -zxvf input.tar.gz` in your shell.
 - on Windows, use a software tool such as winzip, winrar
 - on Mac OS X, `tar -zxvf input.tar.gz` shall work in a shell, but you may also use other suitable software tools.
- There are 2 types of csv files: regular and full

regular : these contain the following core set of data fields (without raw texts), this is the most commonly used format:

```
"domainName", "registrarName", "contactEmail", "whoisServer",  
"nameServers", "createdDate", "updatedDate", "expiresDate",  
"standardRegCreatedDate", "standardRegUpdatedDate",  
"standardRegExpiresDate", "status", "RegistryData_rawText",  
"WhoisRecord_rawText", "Audit_auditUpdatedDate", "registrant_rawText",  
"registrant_email", "registrant_name", "registrant_organization",  
"registrant_street1", "registrant_street2", "registrant_street3",  
"registrant_street4", "registrant_city", "registrant_state",  
"registrant_postalCode", "registrant_country", "registrant_fax",  
"registrant_faxExt", "registrant_telephone", "registrant_telephoneExt",  
"administrativeContact_rawText", "administrativeContact_email",  
"administrativeContact_name", "administrativeContact_organization",  
"administrativeContact_street1", "administrativeContact_street2",  
"administrativeContact_street3", "administrativeContact_street4",  
"administrativeContact_city", "administrativeContact_state",  
"administrativeContact_postalCode", "administrativeContact_country",  
"administrativeContact_fax", "administrativeContact_faxExt",  
"administrativeContact_telephone", "administrativeContact_telephoneExt",  
"billingContact_rawText", "billingContact_email", "billingContact_name",  
"billingContact_organization", "billingContact_street1",  
"billingContact_street2", "billingContact_street3",  
"billingContact_street4", "billingContact_city", "billingContact_state",  
"billingContact_postalCode", "billingContact_country",  
"billingContact_fax", "billingContact_faxExt",  
"billingContact_telephone", "billingContact_telephoneExt",  
"technicalContact_rawText", "technicalContact_email",
```

"technicalContact_name", "technicalContact_organization",
 "technicalContact_street1", "technicalContact_street2",
 "technicalContact_street3", "technicalContact_street4",
 "technicalContact_city", "technicalContact_state",
 "technicalContact_postalCode", "technicalContact_country",
 "technicalContact_fax", "technicalContact_faxExt",
 "technicalContact_telephone", "technicalContact_telephoneExt",
 "zoneContact_rawText", "zoneContact_email", "zoneContact_name",
 "zoneContact_organization", "zoneContact_street1", "zoneContact_street2",
 "zoneContact_street3", "zoneContact_street4", "zoneContact_city",
 "zoneContact_state", "zoneContact_postalCode", "zoneContact_country",
 "zoneContact_fax", "zoneContact_faxExt", "zoneContact_telephone",
 "zoneContact_telephoneExt", "registrarIANAID"

full: in addition to the fields of the regular format, these contain 2 additional fields:

- RegistryData_rawText: raw text from the whois registry
- WhoisRecord_rawText: raw text from the whois registrar

The full data fields are shown in the following lines:

"domainName", "registrarName", "contactEmail", "whoisServer",
 "nameServers", "createdDate", "updatedDate", "expiresDate",
 "standardRegCreatedDate", "standardRegUpdatedDate",
 "standardRegExpiresDate", "status", "RegistryData_rawText",
 "WhoisRecord_rawText", "Audit_auditUpdatedDate", "registrant_rawText",
 "registrant_email", "registrant_name", "registrant_organization",
 "registrant_street1", "registrant_street2", "registrant_street3",
 "registrant_street4", "registrant_city", "registrant_state",
 "registrant_postalCode", "registrant_country", "registrant_fax",
 "registrant_faxExt", "registrant_telephone", "registrant_telephoneExt",
 "administrativeContact_rawText", "administrativeContact_email",
 "administrativeContact_name", "administrativeContact_organization",
 "administrativeContact_street1", "administrativeContact_street2",
 "administrativeContact_street3", "administrativeContact_street4",
 "administrativeContact_city", "administrativeContact_state",
 "administrativeContact_postalCode", "administrativeContact_country",
 "administrativeContact_fax", "administrativeContact_faxExt",
 "administrativeContact_telephone", "administrativeContact_telephoneExt",
 "billingContact_rawText", "billingContact_email", "billingContact_name",
 "billingContact_organization", "billingContact_street1",
 "billingContact_street2", "billingContact_street3",
 "billingContact_street4", "billingContact_city", "billingContact_state",
 "billingContact_postalCode", "billingContact_country",
 "billingContact_fax", "billingContact_faxExt",
 "billingContact_telephone", "billingContact_telephoneExt",
 "technicalContact_rawText", "technicalContact_email",
 "technicalContact_name", "technicalContact_organization",
 "technicalContact_street1", "technicalContact_street2",
 "technicalContact_street3", "technicalContact_street4",
 "technicalContact_city", "technicalContact_state",
 "technicalContact_postalCode", "technicalContact_country",
 "technicalContact_fax", "technicalContact_faxExt",
 "technicalContact_telephone", "technicalContact_telephoneExt",
 "zoneContact_rawText", "zoneContact_email", "zoneContact_name",

```
"zoneContact_organization", "zoneContact_street1", "zoneContact_street2",  
"zoneContact_street3", "zoneContact_street4", "zoneContact_city",  
"zoneContact_state", "zoneContact_postalCode", "zoneContact_country",  
"zoneContact_fax", "zoneContact_faxExt", "zoneContact_telephone",  
"zoneContact_telephoneExt", "registrarIANAID"
```

3.3 Data field details

The csv data fields are mostly self-explanatory by name except for the following:

createdDate: when the domain name was first registered/created

updatedDate: when the whois data were updated

expiresDate: when the domain name will expire

standardRegCreatedDate: created date in the standard format(YYYY-mm-dd), e.g. 2012-02-01

standardRegUpdatedDate: updated date in the standard format(YYYY-mm-dd), e.g. 2012-02-01

standardRegExpiresDate: expires date in the standard format(YYYY-mm-dd), e.g. 2012-02-01

Audit_auditUpdatedDate: the timestamp of when the whois record is collected in the standardFormat(YYYY-mm-dd), e.g. 2012-02-01

status: domain name status code; see

<https://www.icann.org/resources/pages/epp-status-codes-2014-06-16-en>
for details

registrant: The domain name registrant is the owner of the domain name. They are the ones who are responsible for keeping the entire WHOIS contact information up to date.

administrativeContact: The administrative contact is the person in charge of the administrative dealings pertaining to the company owning the domain name.

billingContact: the billing contact is the individual who is authorized by the registrant to receive the invoice for domain name registration and domain name renewal fees.

technicalContact: The technical contact is the person in charge of all technical questions regarding a particular domain name.

zoneContact: The domain technical/zone contact is the person who tends to the technical aspects of maintaining the domain's name server and resolver software, and database files.

registrarIANAID: The IANA ID of the registrar.

Consult <https://www.iana.org/assignments/registrar-ids/registrar-ids.xhtml>
to resolve IANA ID-s.

3.4 Maximum data field lengths

domainName: 256, registrarName: 512, contactEmail: 256,
whoisServer: 512, nameServers: 256, createdAt: 200,
updatedAt: 200, expiresDate: 200, standardRegCreatedAt: 200,
standardRegUpdatedAt: 200, standardRegExpiresDate: 200,
status: 65535, Audit_auditUpdatedAt: 19, registrant_email: 256,
registrant_name: 256, registrant_organization: 256,
registrant_street1: 256, registrant_street2: 256,
registrant_street3: 256, registrant_street4: 256,
registrant_city: 64, registrant_state: 256, registrant_postalCode: 45,
registrant_country: 45, registrant_fax: 45, registrant_faxExt: 45,
registrant_telephone: 45, registrant_telephoneExt: 45,
administrativeContact_email: 256, administrativeContact_name: 256,
administrativeContact_organization: 256, administrativeContact_street1: 256,
administrativeContact_street2: 256, administrativeContact_street3: 256,
administrativeContact_street4: 256, administrativeContact_city: 64,
administrativeContact_state: 256, administrativeContact_postalCode: 45,
administrativeContact_country: 45, administrativeContact_fax: 45,
administrativeContact_faxExt: 45, administrativeContact_telephone: 45,
administrativeContact_telephoneExt: 45, registrarIANAID: 65535

3.5 Standardized country fields

The [contact]_country fields are standardized. The possible values are listed in the first column of the file

<http://www.domainwhoisdatabase.com/docs/countries.txt>

The possible country names are in the first column of this file; the field separator character is “|”.

4 JSON file availability

Even though CSV is an extremely portable format accepted by virtually any system, in many applications, including various NoSQL solutions as well as custom solutions to analyze WHOIS data, the JSON format is preferred.

The data files which can be downloaded from WhoisXML API can be converted to JSON very simply. We provide Python scripts which can be used to turn the downloaded CSV WHOIS data into JSON files. These are available in our Github repository under

https://github.com/whois-api-llc/whois_database_download_support/tree/master/whoisxmlapi_csv2json

We refer to the documentation of the scripts for details.

5 Database dumps

5.1 Software requirements for importing mysql dump files

- Mysql server 5.1+ is recommended although it should work also with mysql-server of version lower than 5.1

5.2 Importing mysql dump files

Using `mysqldump` is a portable way to import the database.

5.2.1 Loading everything (including schema and data) from a single mysqldump file

This is equivalent to running the following in mysql

- create a database for the tld for example:

```
mysql -uroot -ppassword -e "create database whoiscrawler_com"
```

- import the mysqldump file into the database for example:

```
zcat add_mysqldump_2015_01_12_com.sql.gz | \
mysql -uroot -ppassword whoiscrawler_com --force
```

5.3 Database schema

There are 3 important tables in the database:

Table: whois_record Fields:

whois_record_id BIGINT(20) PRIMARY KEY NOT NULL Primary key of whois_record.

created_date VARCHAR(200) When the domain name was first registered/created.

updated_date VARCHAR(200) When the whois data was updated.

expires_date VARCHAR(200) When the domain name will expire.

admin_contact_id BIGINT(20) FOREIGN KEY Foreign key representing the id of the administrative contact for this whois_record. It references the primary key in contact table. The administrative contact is person in charge of the administrative dealings pertaining to the company of the domain name.

registrant_id BIGINT(20) FOREIGN KEY Foreign key representing the id of the registrant for this whois_record. It references the primary key in contact table. The domain name registrant is the owner of the domain name. They are the ones who are responsible for keeping the entire WHOIS contact information up to date.

technical_contact_id BIGINT(20) FOREIGN KEY Foreign key representing the id of the technical contact for this whois_record. It references the primary key in contact table. The technical contact is the person in charge of all technical questions regarding a particular domain name.

zone_contact_id BIGINT(20) FOREIGN KEY Foreign key representing the id of the zone contact for this whois_record. is the person who tends to the technical aspects of maintaining the domain's name server and resolver software, and database files.

billing_contact_id BIGINT(20) FOREIGN KEY Foreign key representing the id of the billing contact for this whois_record. It references the primary key in contact table. the billing contact is the individual who is authorized by the registrant to receive the invoice for domain name registration and domain name renewal fees.

domain_name VARCHAR(256) FOREIGN KEY Domain Name

name_servers TEXT Name servers or DNS servers for the domain name. The most important function of DNS servers is the translation (resolution) of human-memorable domain names and hostnames into the corresponding numeric Internet Protocol (IP) addresses.

registry_data_id BIGINT(20) FOREIGN KEY Foreign key representing the id of the registry data. It references the primary key in registry_data table. Registry Data is typically a whois record from a domain name registry. Each domain name has potentially up to 2 whois record, one from the registry and one from the registrar. Whois_record(this table) represents the data from the registrar and registry_data represents whois data collected from the whois registry. Note that registryData and WhoisRecord has almost identical data structures. Certain gtlds(eg. most of.com and .net) have both types of whois data while most cctlds have only registryData. Hence it's recommended to look under both WhoisRecord and registryData when searching for a piece of information(eg. registrant, createdAt).

status TEXT domain name status code; see details at <https://www.icann.org/resources/pages/epp-status-codes-2014-06-16-en>

raw_text LONGTEXT the complete raw text of the whois record

audit_created_date TIMESTAMP FOREIGN KEY the date this whois record is collected on whoisxmlapi.com, note this is different from WhoisRecord → createdAt or WhoisRecord → registryData → createdAt

audit_updated_date TIMESTAMP FOREIGN KEY the date this whois record is updated on whoisxmlapi.com, note this is different from WhoisRecord → updatedAt or WhoisRecord → registryData → updatedAt

unparsable LONGTEXT the part of the raw text that is not parsable by our whois parser

parse_code SMALLINT(6) a bitmask indicating which fields are parsed in this whois record. A binary value of 1 at index i represents a non empty value field at that index. The fields that this parse code bitmask represents are, from the least significant to most significant bit in this order: createdAt, expiresDate, referralURL(exists in registryData only), registrarName, status, updatedAt, whoisServer(exists in registryData only), nameServers, administrativeContact, billingContact, registrant, technicalContact, and zoneContact. For example, a parseCode of 3 (binary: 11) means that the only non-empty fields are createdAt and expiresDate. a parseCode of 8(binary:1000) means that the only non-empty field is registrarName. Note: the fields represented by the parseCode do not represent all fields exist in the whois record.

header_text LONGTEXT the header of the whois record is part of the raw text up until the first identifiable field.

clean_text LONGTEXT the stripped text of the whois record includes part of the raw excluding header and footer, this should only include identifiable fields.

footer_text LONGTEXT the footer of the whois record is part of the raw after the last identifiable field.

registrar_name VARCHAR(512) A domain name registrar is an organization or commercial entity that manages the reservation of Internet domain names.

data_error SMALLINT(6) FOREIGN KEY an integer with the following meaning: 0=no data error 1=incomplete data; 2=missing whois data, it means that the domain name has no whois record in the registrar/registry 3=this domain name is a reserved word

Table: registry_data Fields:

registry_data_id BIGINT(20) PRIMARY KEY NOT NULL

created_date VARCHAR(200)

updated_date VARCHAR(200)

expires_date VARCHAR(200)

admin_contact_id BIGINT(20) FOREIGN KEY

registrant_id BIGINT(20) FOREIGN KEY
technical_contact_id BIGINT(20) FOREIGN KEY
zone_contact_id BIGINT(20) FOREIGN KEY
billing_contact_id BIGINT(20) FOREIGN KEY
domain_name VARCHAR(256) FOREIGN KEY
name_servers TEXT
status TEXT
raw_text LONGTEXT
audit_created_date TIMESTAMP
audit_updated_date TIMESTAMP FOREIGN KEY
unparsable LONGTEXT
parse_code SMALLINT(6)
header_text LONGTEXT
clean_text LONGTEXT
footer_text LONGTEXT
registrar_name VARCHAR(512)
whois_server VARCHAR(512)
referral_url VARCHAR(512)
data_error SMALLINT(6) FOREIGN KEY

Table: contact Fields:

contact_id BIGINT(20) PRIMARY KEY NOT NULL
name VARCHAR(512)
organization VARCHAR(512)
street1 VARCHAR(256)
street2 VARCHAR(256)
street3 VARCHAR(256)
street4 VARCHAR(256)
city VARCHAR(256)
state VARCHAR(256)
postal_code VARCHAR(45)
country VARCHAR(45)
email VARCHAR(256)
telephone VARCHAR(128)
telephone_ext VARCHAR(128)
fax VARCHAR(128)
fax_ext VARCHAR(128)
parse_code SMALLINT(6)

```
raw_text LONGTEXT
unparsable LONGTEXT
audit_created_date VARCHAR(45)
audit_updated_date VARCHAR(45) FOREIGN KEY
```

Remark about maximum field lengths: in some database dump files, especially dailies, the maximum size of the VARCHAR and BIGINT files is smaller than what is described in the above schema. When using such database dumps together with others, it is recommended to set the respective field length to the “failsafe” values, according to the here documented schema. For instance, in case of daily WHOIS database dump from the domain_names_whois data feed, the recommended modifications of the maximum lengths of VARCHAR or BIGINT fields are:

- whois_record table:
 - domain_name: 256 instead of 70
 - all foreign key _id fields: 20 instead of 11
- contact table:
 - name: 512 instead of 256
 - organization: 512 instead of 256
 - city: 256 instead of 64
 - state: 256 instead of 45
 - telephone, telephone_ext: 128 instead of 45
 - fax, fax_ext: 128 instead of 45

5.4 Further reading

There can be many approaches for creating and maintaining a MySQL domain WHOIS database depending on the goal. In some cases the task is cumbersome as we are dealing with big data. Our client-side scripts are provided as samples to help our clients to set up a suitable solution; they can be used as they are in many cases. All of them come with a detailed documentation.

Some of our blogs can be also good reads with this respect, for instance, this one:

<https://www.whoisxmlapi.com/blog/setting-up-a-whois-database-from-whoisxml-api-data>

6 Client-side scripts for downloading data, loading into databases, etc.

Scripts are provided in support of downloading WHOIS data through web-access and maintaining a WHOIS database. These are available on github:

https://github.com/whois-api-llc/whois_database_download_support

The actual version can be downloaded as a zip package or obtained via git or svn.

There are scripts in Bourne Again Shell (BASH) as well as in Python (natively supported also on Windows systems).

The subdirectories of the repository have the following contents:

`whoisxmlapi_download_whois_data`: a Python2 script for downloading bulk data from daily and quarterly WHOIS data feeds in various formats. It can be used from command line, but also supports a simple GUI. For all platforms.

`whoisxmlapi_whoisdownload_bash`: a bash script for downloading bulk data from daily and quarterly WHOIS data feeds.

`whoisxmlapi_bash_csv_to_mysqldb`: bash scripts to create and maintain WHOIS databases in MySQL based on csv files downloaded from WhoisXML API. If you do not insist on bash, check also

`whoisxmlapi_flexible_csv_to_mysqldb`

which is in Python 3 and provides extended functionality.

`whoisxmlapi_flexible_csv_to_mysqldb`: a flexible and portable script in Python to create and maintain WHOIS databases in MySQL based on csv files downloaded from WhoisXML API.

`whoisxmlapi_mysqldump_loaders`: Python2 and bash scripts to set up a WHOIS database in MySQL, using the data obtained from WhoisXML API quarterly data feeds.

`whoismxmlapi_percona_loaders`: bash scripts for loading binary MySQL dumps of quarterly releases where available

`legacy_scripts`: miscellaneous legacy scripts not developed anymore, published for compatibility reasons.

In addition, the scripts can be used as a programming template for developing custom solutions. The script package includes a detailed documentation.

7 Tips for web-downloading data

In this Section we provide additional information in support of web-downloading the feeds. This includes recommendations about organizing and scheduling the download process as well as some tips for those who want to download multiple files from the data feeds via web access by either using generic software tools, either command-line based or GUI for some reason. We remark, however, that our downloader scripts are at our clients' disposal, see Section 6 on their details. Our scripts provide a specialized solution for this task, and the Python version can be run in GUI mode, too.

Note: this information describes both the case of quarterly releases and daily data feeds, as most users who do this process will use both.

7.1 When, how, and what to download

While the data feeds' web directories are suitable for downloading a few files interactively, in most cases the download is to be carried out with an automated process. To implement this,

- the URLs of individual files for a given database release or day have to be determined,
- and the files have to be downloaded according to a suitable schedule.

File URLs. The organization of the web directories is described at each data feed in the present manual. Given a day (e.g. 2020-03-15) or database release (e.g. v31), a TLD name (e.g. .com), the URL of the desired files can be put together easily after going through the data feed's docs. E.g. the regular csv data for .com in the v31 quarterly release will be at

`http://www.domainwhoisdatabase.com/whois_database/v31/csv/tlds/regular`

whereas the daily data for 2020-03-15 of this domain will be at

`http://bestwhois.org/domain_name_data/domain_names_whois/2020_03_15_com.csv.gz`

The downloader scripts supplied with our products (c.f. Section 6) given the feed's name and the data format's name. But what should be the TLD name.

TLDs to download. The broadest list a data feed can have data for is that of the *supported* TLDs, consult Section 2.1 for the explanation. Their actual list depends on the database release in case of quarterlies, and on the data feed and day in case of daily feeds. To use an accurate list, check the auxiliary files provided to support download automation. In particular, the list will be in the subdirectory

- `docs/vXX.tlds` in the quarterly releases,
- `status/supported_tlds_YYYY_MM_DD` in case of most daily feeds; consult the actual feeds' description.

If a domain is supported, it is not necessary that it has data in all the daily feeds. E.g. if there were no domains added on a day in a give TLD, it will not have data files on a given day. *Hence, the lack of a file for a given supported TLD on a given day can be normal.*

Another option in case of daily feeds is to use another supplemental file provided with the data feed. E.g. in case of `domain_names_new`, the files

`status/added_tlds_YYYY_MM_DD`

will give a list of domains for which there are actual data on the given day.

Scheduling. The key question is: when a set of files are ready for downloading. In case of quarterly releases the availability is announced via e-mail to subscribers, and so are possible extensions or corrections.

In case of daily data the tentative schedule information is published here:

`http://domainwhoisdatabase.com/docs/whoisxmlapi_daily_feed_schedule.html`

As the actual availability times vary, there are supplemental files (typically named `status/*download_ready*`, consult the description of the feeds) whose existence indicates that the data are ready for downloading, and their file date reflects the time when they became available.

Redownloading missing or broken files. If a given data file was unavailable when a scheduled attempt was made, it has to be downloaded again. For most files we provide `md5` and `sha256` checksum files, see the detailed docs of the data feeds for their naming convention.

When attempting to redownload a file, a recommended method is to download its checksum. If there is an already downloaded version of the file which is inline with the checksum, no redownload is needed. If the check fails, or the file is not there, the file has to be redownloaded. This policy is implemented by the Python downloader provided with the products, which is also capable of continuing a broken download. The downloader script in BASH will repeat downloading if and only if the file is absent.

Implementing this policy, or using the provided scripts, a recommended approach is to repeat the download procedure multiple times, going back a few days, and keep the downloaded files in place. Thereby all the missing files will get downloaded, and the ones which are downloaded and are the same as the one on the web server will be skipped.

7.2 Downloaders with a GUI

GUI-based downloading is mainly an option for those who download data occasionally as it is less efficient than the command-line approach and cannot be automated. Primarily we recommend to use our python downloader (Section 6) which comes with a simple GUI specialized for downloading from our data feeds.

There are, however, several stand-alone programs as well as browser plugins intended for downloading several files at once from webpages. Unfortunately, however, most of these are not very suitable for the purpose of downloading from WhoisXML API feeds. There are some exceptions, though. In the following we describe one of them, iGetter, which we found suitable for the purpose.

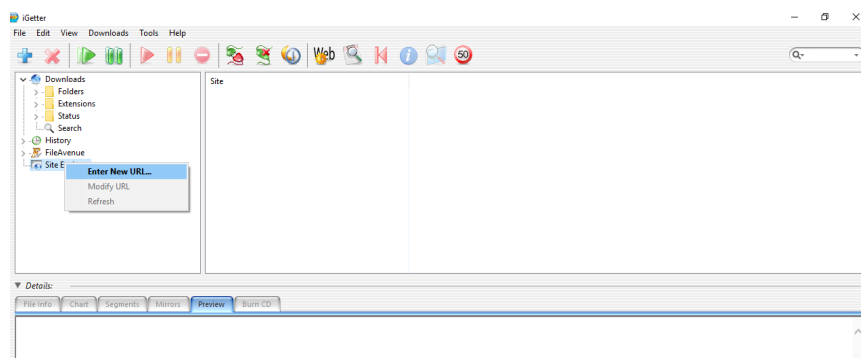
Installing iGetter. The program is available for Mac and Windows. It is a Shareware and can be downloaded from

<http://www.igetter.net/downloads.html>

After downloading it, simply follow the installation instructions.

An example. In the following description, the screenshots come from a Windows 10 system, under Mac OS X, the process is similar. The task is to download 3 days of data of the TLDs “aero” and “biz” from the feed “domain_names_new”. The dates will be from 2018.08.20 to 2018.08.22. (It is an example with a daily feed, but in case of quarterly feeds the process is very similar, as it is essentially about downloading a set of files from a web-hosted directory structure.) It can be carried out as follows:

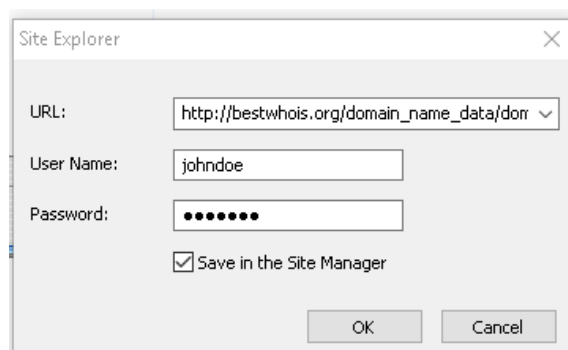
1. Open iGetter
2. Click the right button on “Site explorer”, and choose “Enter new URL”:



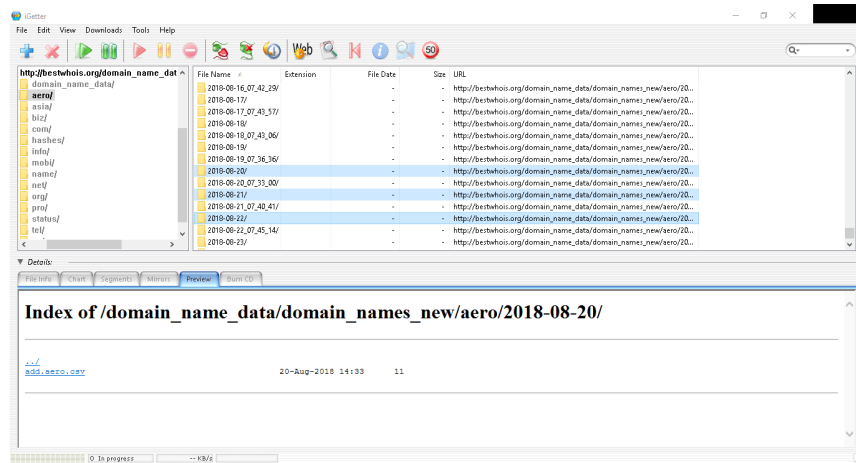
3. A window pops up, paste the feed URL, this time it is

`http://bestwhois.org/domain_name_data/domain_names_new`

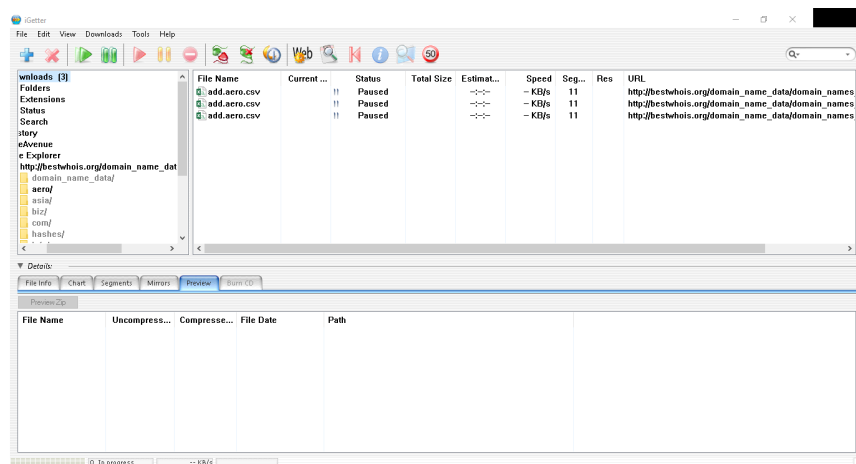
Also open the “Authenticate” part, enter your username and password, and check the “Save in the Site Manager” box:



4. After pressing “OK”, in the upper part of the screen the directory listing of the feed will appear. (Note: in all the cases, this switching to subdirectories with a large number of files may take a lot of time, please be patient.) Double click the directory “aero”. The upper panel shall divide into two parts. Select the directories of the given date in the right half:
5. Click the right button on “Site explorer”, and choose “Enter new URL”:



6. Press the right mouse button on this panel, and select “Add to queue”. Then say “Yes” to the question “Would you like to download web page contents?”. The right part of the upper half of the window will show the download queue now:
7. Click the right button on “Site explorer”, and choose “Enter new URL”:



8. Double click now “biz” on the left half of the upper part, and follow the same procedure as with “aero”. When the download queue is prepared, press the green arrow (“Set Auto downloading”) button. You can now follow the download procedure in the queue. Your files will be downloaded into the directory “best-whois.org” on the Desktop, under the same directory structure as on the server. You can see the details of completed downloads under “History”.

For further tips and tweaks, consult the documentation of the software.

7.3 Command-line downloaders

There are various command-line tools for downloading files or directories from web-pages. They provide an efficient way of downloading and can be used in scripts or batch files for automated downloading.

Most of these are available freely in some form on virtually any platform. These are, e.g. `curl`, `pavuk`, or <https://www.gnu.org/software/wget/wget>, to mention the maybe most popular ones. Here we describe the use of `wget` through some examples as this is the maybe most prevalent and it is very suitable for the purpose. We describe here a typical example of its use. Those who plan to write custom downloader scripts may take a look at the BASH downloader script we provide: it is also `wget`-based. We refer to its documentation for further details or tweaks.

Installing wget. To install `wget` you can typically use the package management of your system. For instance, on Debian-flavor Linux systems (including the Linux subsystem available on Windows 10 platforms) you can install it by the command-line

S

udo apt-get install wget A native Windows binary is available from
<http://gnuwin32.sourceforge.net/packages/wget.htm>

Command-line options. The program expects an URL as a positional argument and will replicate it into the directory where it is invoked from. The following options are the maybe most relevant for us:

`-r` Recursive download. Will download the pages linked from the starting page. These are the subdirectories of the directory in our case.

`-l` This should be used with `-r` followed by a number specifying the recursion depth of the download. E.g. with `-l 1` it will download the directory and its subdirectories, but not those below them.

`-c` Continue any broken downloads

`user=` Should be followed by the username for http authentication, that is, it should be the username of your subscription.

`password=` The password for the username, given with your subscription. If not specified, you will be prompted for it each time. If you use this option, bear in mind security considerations: your password will be readable e.g. from your shell history or from the process list of the system.

`--ca-certificate=` `--certificate=` `--private-key=` By writing the appropriate filenames after the “=” paths, you can use `wget` with ssl authentication instead of the basic password authentication if this option is available with your subscription. See Section 12 for more details.

An example. In the present example we shall download data of the “aero” TLD from the feed “`domain_names_new`” for 2018-08-20. (It is an example with a daily feed, but similar examples can be easily constructed also for quarterly feeds. In general it is about downloading a file replicating the directory structure of the web server.)

```
wget -r -ll --user=johndoe --password=johndoespassword  
"http://bestwhois.org/domain_name_data/domain_names_new/aero/2018-08-20/add.aero.csv"
```

This will leave us with a directory structure in the current working directory which is a replica of the one at the web server:

```
.  
|-bestwhois.org  
|---domain_name_data  
|----domain_names_new  
|-----aero  
|-----2018-08-20  
|-----add.aero.csv
```

Noe that we could have downloaded just the single file:

```
wget --user=johndoe --password=johndoespassword \
"http://bestwhois.org/domain_name_data/domain_names_new/aero/2018-08-20/add.aero.csv"
```

but this would leave us with a single file “add.aero.csv” which is hard to identify later. Albeit `wget` is capable of downloading entire directories recursively, the good strategy is to collect all the URLs of single files to get and download them with a single command-line each. This can be automated with script or batch files. Consult the BASH downloader script provided for downloading to get additional ideas, and the documentation of `wget` for more tweaks.

8 Handling large csv files

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

8.1 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 “tofromdos”. 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>

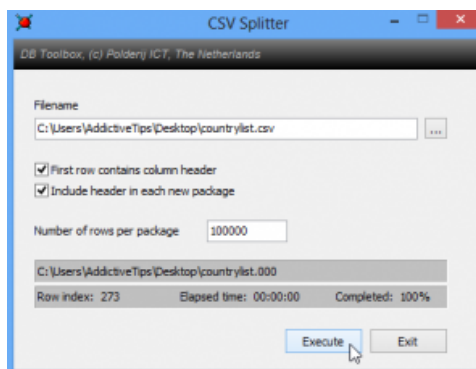
In Windows PowerShell you can use the commands “GetContent” and “SetContent” for the purpose, please consult their documentation.

8.2 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 8.4 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 8.4.

8.3 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 8.4

8.4 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

9 Daily data collection methodology

In this Section we describe how we describe in detail how and on which day a domain gets listed in a certain daily data feeds, that is, how the process behind the feeds detect the Internet domains have been registered, dropped or modified on a given day.

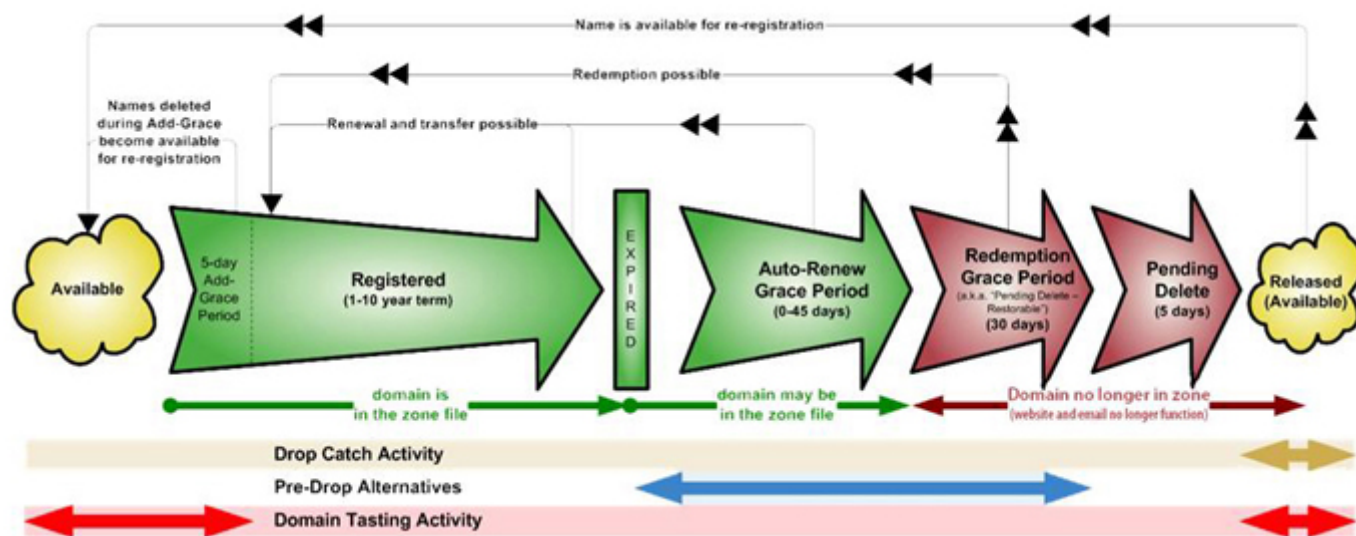
In principle, WHOIS records contain date fields that reflect the creation, modification, and deletion dates. It is not possible, however, to search the WHOIS system for such dates. In addition, there can be a delay between the actual date and the appearance of the WHOIS record: the presence of WHOIS information is not required by the technical operation of a domain, so registrars and registries are not very strict with updating the WHOIS system. Hence, it is not possible to efficiently obtain daily updates of WHOIS data entirely from the WHOIS system itself.

Most of the daily feeds thus follow another strategy. The technical operation of a domain requires its presence in the Domain Name System (DNS). So a domain starts operating if it appears in the zone file of the domain, and ceases to operate when it disappears from it. We refer to our white paper on DNS:

for further details. As for modifications of domains, the approach of the “delta” feeds’ data generation is to look for domains which have changed either of their name servers in the zone file: most of the relevant changes in a domain, like the change in the ownership imply such a change. In the following subsections we explain how the date when the domain appears in a daily feed is related to the one in its WHOIS record, and assess the accuracy of the data feeds.

9.1 Domain life cycle and feed timings

Domains have their life cycle. For domains GTLDs this is well-defined, while in case of those in ccTLDs it can depend on the authoritative operator of the given TLD. So as a reference, in case of domains in a generic top-level domain, such as .com the life cycle can be illustrated in the following figure:



(Source: <https://www.icann.org/resources/pages/gtld-lifecycle-2012-02-25-en>)

Notice that in the auto-renew grace period, which *can* be *0-45* days, the domain *may* be in the zone file, so it may actively operate or not.

The deadline of introducing or removing the WHOIS record can vary, there is no very strict regulation of this. So it easily happens that the domain already works but has no WHOIS record yet, or the other way around: the domain does not work, it is not in the zone file, but it already (or still) has a WHOIS record.

9.2 Time data accuracy

Because of the nature of the described life cycle, the day of the appearance of a domain in a feed of new, modified or dropped domains (i.e. the day in the file name) is not exactly the day in the WHOIS record corresponding to the given event. In the daily feed there are the domains which start to technically function on that day, maybe not the first time even. (It might also happen that an entry in the zone file changes just because some error.) The date in the WHOIS record is, on the other hand the date when the domain was officially registered, but it is not forced to coincide with the time when it started to function.

The number of the record in the daily data feed, however, will show the same or similar main trends even if it will not coincide with the number of domains found with the given date in the WHOIS database, which can be found out later, via querying a complete database. But the maintenance of a complete database is definitely more resource expensive than counting the number of lines of some files, so it is a viable approach to study domain registration, modification, or deletion trends.

In relation to accuracy, a frequent misunderstanding is the notion of “today”. When talking about times, one should never forget about time zones. A date in the WHOIS record having a date for yesterday can be today in another time zone.

Another systematic feature of our methodology is that the WHOIS records for domains with status codes indication that in they are in the redemption grace period or pending delete period are not all captured. It is for the reason that if we detect that a domain is disappearing from the zone file it can have two meanings: it is somewhere in its auto-renew grace period or it has just started its redemption grace period. The uncertainty is because in the auto-renew grace period, “the domain may be in the zone file”. And it is very likely that it is just the status which changes when it disappears from the zone file, so we will probably not have too much new information from the rest of these records.

10 Alternative data collection methodologies for ccTLDs

As mentioned in Section , the data collection based on changes in DNS zone files is not always feasible in case of ccTLDs as the DNS zone files are not available. Hence, to provide data for maintaining an up-to-date WHOIS database, different approaches are required. We use various third-party DNS sensors as well as a web crawler to collect domain names under ccTLDs. The data feeds `cctld_discovered_domain_names_new` and `cctld_discovered_domain_names_whois` data feeds contain data from these sources.

Unfortunately these collection methods rely on network traffic or actual information on the World Wide Web, hence, the date of the data feed files are not related to the dates in the WHOIS record of the feed. The date just refers to the day when the domain was detected. This methodology is suitable for finding domains not seen before, but WHOIS record changes and dropped domains cannot be detected this way. The feed is cumulative in the sense that a domain detected recently is not repeated in the daily files if detected again.

Unfortunately, in case of many ccTLDs there is no other information available. Nevertheless, the two feeds, containing domains that have been detected by us on a given day and the respective WHOIS data are still useful for maintaining an up-to-date WHOIS database.

11 Data quality check

As WHOIS data come from very diverse sources with different policies and practices, their quality vary by nature. The data accuracy is strongly effected by data protection regulations, notably the GDPR of the European Union. Thus the question frequently arises: how to check the quality of a WHOIS record. In general, an assessment can be done in based on the following principles.

To decide if a record is acceptable at all, we recommend to check the following aspects:

- If the “createdDate”, “updatedDate”, or “expiresDate” fields are empty (and so are their version with their “standard” prefix), the record is invalid. These data are typically there even in the most GDPR-affected WHOIS records.
- If the “registrarName” field is empty, the record is invalid, except for some TLDs (typically ccTLDs) where the WHOIS server does not provide registrar information.

If these criteria are met, the record can be considered as *valid in principle*. Yet its quality is still in a broad range. To further assess the quality, the typical approaches

- The number of non-empty fields (the larger the better).
- The number of redacted fields. A field containing the word “redacted” with various capitalizations (e.g. also “Redacted” or “REDACTED”). The smaller the number of such fields, the better is the record.

- Check some fields relevant in the particular application. E.g. “registrant_name”, certain e-mail addresses are non-empty or can be validated (e.g. valid e-mail).

In what follows we describe how to check these aspects in case of the different download formats.

11.1 Quality check: csv files

In case of csv files the file has to be read and parsed. Then the empty or redacted fields can be identified, while the non-empty fields can possibly be validated against the respective criteria.

11.2 Quality check: MySQL dumps

The WHOIS databases recovered from MySQL dumps contain a field named “parseCode”, which makes the quality check more efficient. (It is not present in the csv files.) It is a bit mask indicating which fields have been parsed in the record; a binary value of 1 at position i points to a non-empty value field at that position.

The fields from the least significant bit to the most significant one are following: “createdDate”, “expiresDate”, “referralURL” (exists in “registryData” only), “registrarName”, “status”, “updatedAt”, “whoisServer” (exists in “registryData” only), “nameServers”, “administrativeContact”, “billingContact”, “registrant”, “technicalContact”, and “zoneContact”. For example, a parse code $310 = (11_2)$ means that the only non-empty fields are “createdDate” and “expiresDate”, whereas the parse code $810 = (1000_2)$ means that the only non-empty field is “registrarName”.

If you need to ascertain that a WHOIS record contains ownership information, calculate the binary AND of the parse code and $0010000000000_2 = 512_{10}$ it should be 512. (The mask stands for the non-empty field “registrant”).

12 Access via SSL Certificate Authenticon

We support SSL Certificate Authentication as an alternative to the plain login/password authentication when accessing some of our data feeds on the Web. This provides an encrypted communication between the client’s browser and the server when authenticating and downloading data. Here we describe how you can set up this kind of authentication.

In order to use this authentication, you as a client will need a personalized file provided to you by WhoisXML API, named **pack.p12**. This is a password-protected package file in PKCS12 format which can be easily installed on most systems. We typically send the package via e-mail and the respective password separately in an SMS message for security reasons. The package contains everything necessary for the authentication:

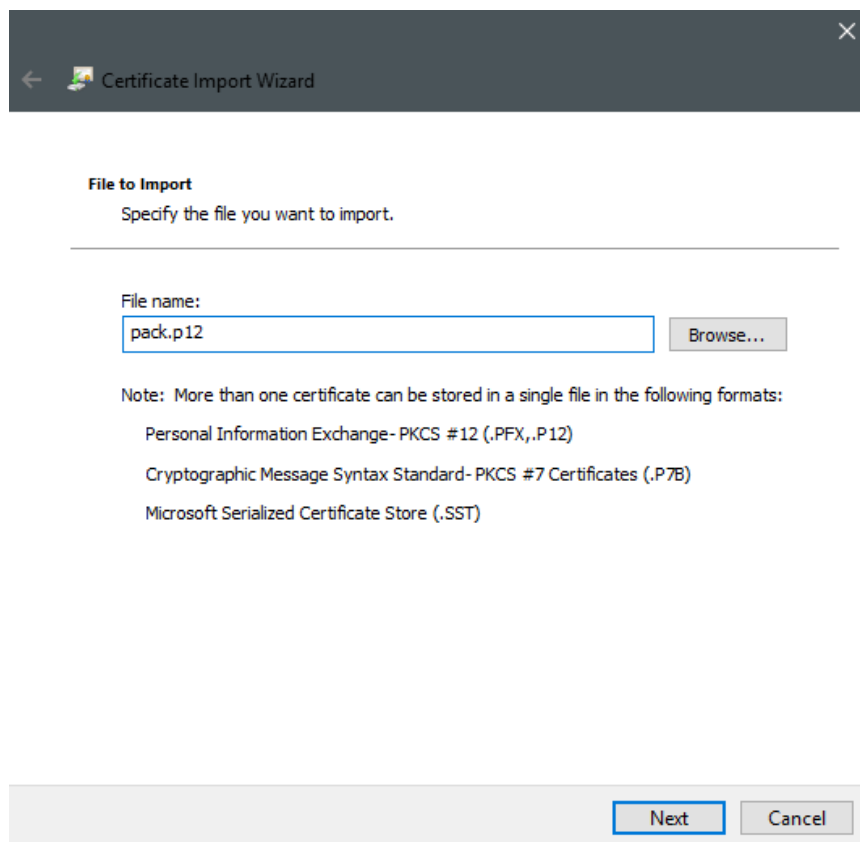
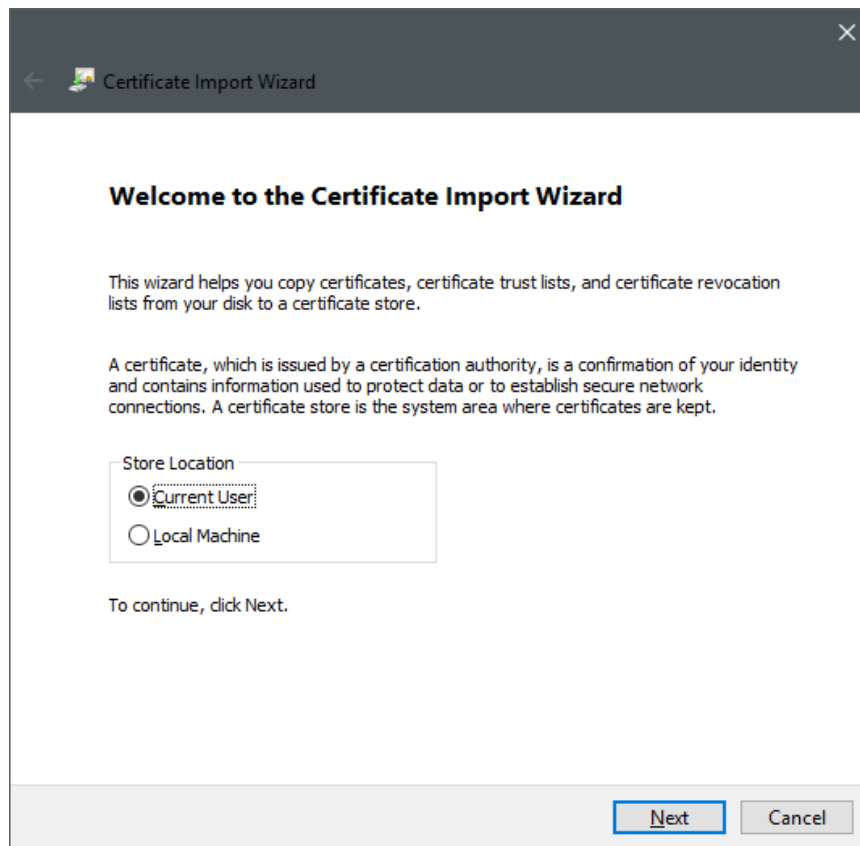
- The personal private key
- The personal certificate signed by server Certificate Authority (CA)
- The certificate of our CA server.

Assuming that you have obtained the package and the respective password, in what follows we describe how to install it on various platforms.


12.1 Setup instructions

12.1.1 Microsoft Windows

Double click on the **pack.p12** file. The following dialog windows will appear, you can proceed with “Next”:



In the next step you should provide the password you got for the package. Then you can just go through the next dialogs with the default settings:

←  Certificate Import Wizard

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

☒ Automatically select the certificate store based on the type of certificate


☐ Place all certificates in the following store

Certificate store:

Browse...

Next

Cancel

←  Certificate Import Wizard

Private key protection

To maintain security, the private key was protected with a password.

Type the password for the private key.

Password:

☐ Display Password

Import options:

☐ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.

☐ Mark this key as exportable. This will allow you to back up or transport your keys at a later time.

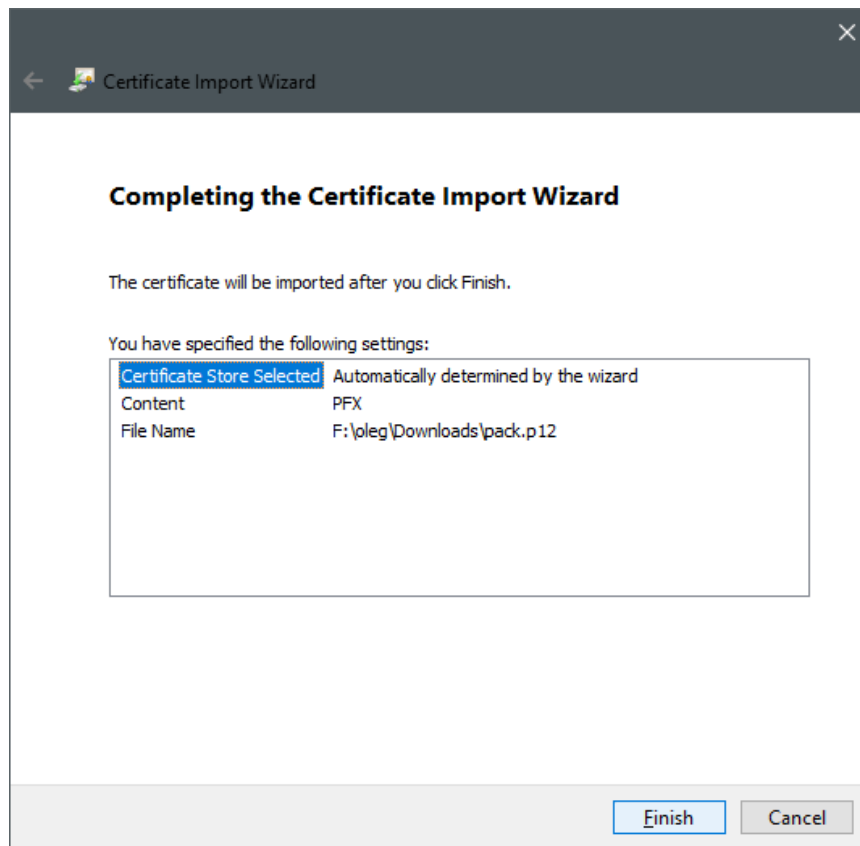
☐ Protect private key using virtualized-based security(Non-exportable)

☒ Include all extended properties.

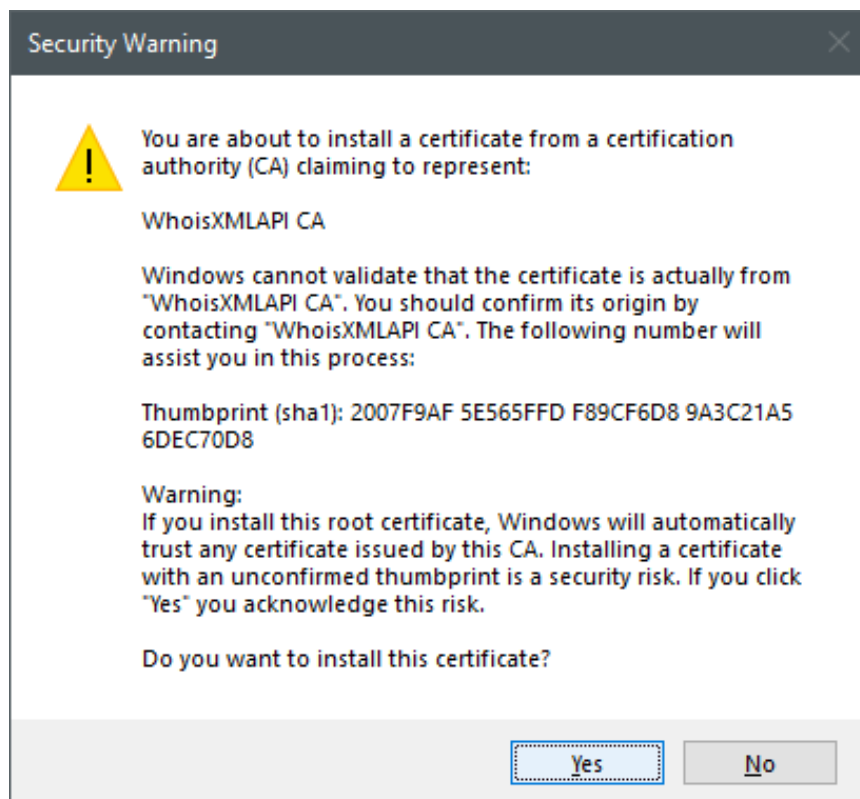
Next

Cancel

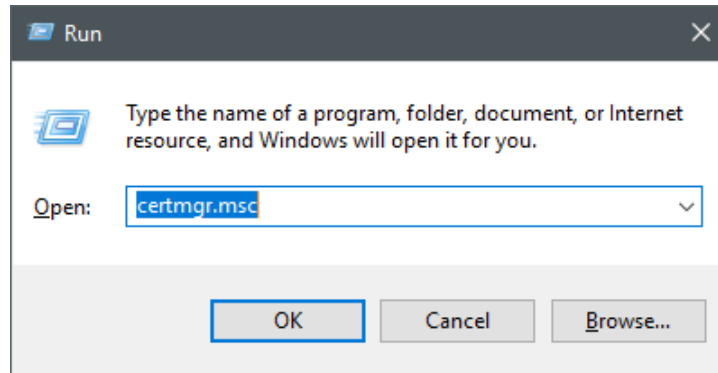
34



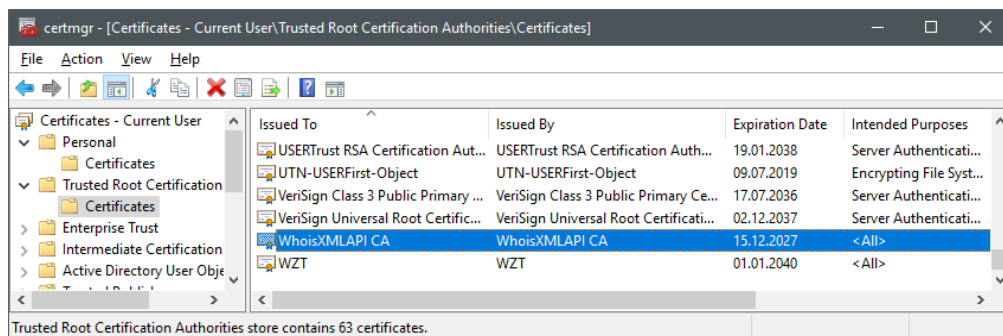
You can safely answer "Yes" to the following warning. It just says that you trust our CA server.



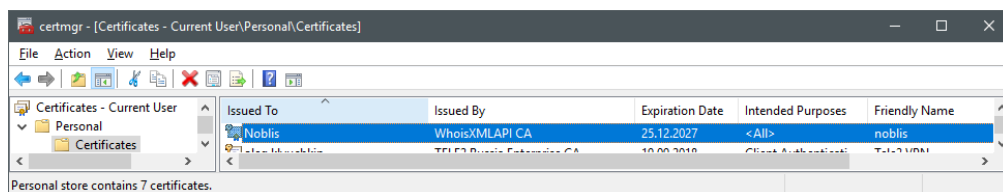
Your installation is complete now. You can verify or revise this or any of your certificates anytime with the **certmgr.msc** tool:



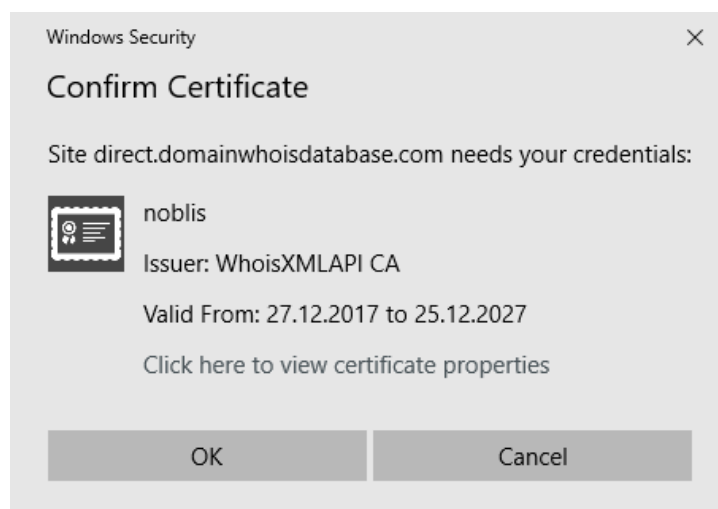
You should see the root certificate:

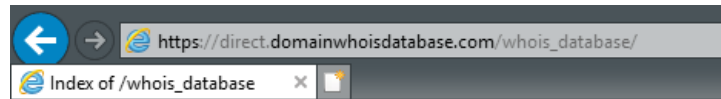


and your personal certificate



And as the main implication, after confirming the certificate you can open now the URLs you are eligible for, listed in Section 12.2, securely and without being prompted for passwords:





Index of /whois_database

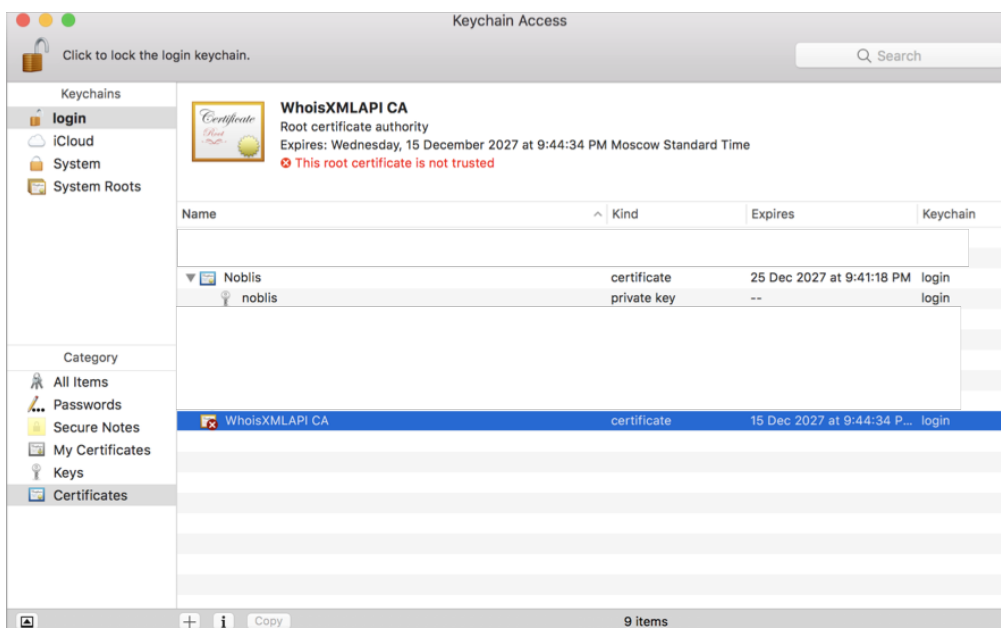
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
ae/	2016-10-02 05:34	-	
alexa 01 01 2017/	2017-03-24 18:32	-	
alexa 01 12 2016/	2017-09-08 08:00	-	
alexa 04 01 2016/	2016-04-25 03:38	-	
alexa 04 01 2017/	2017-09-08 07:42	-	
alexa 04 26 2016/	2016-04-30 11:31	-	
alexa 07 01 2016/	2016-07-05 00:11	-	
alexa 07 01 2017/	2017-07-17 17:05	-	
alexa 09 03 2017/	2017-12-09 19:42	-	
alexa 10 01 2016/	2016-10-06 18:54	-	

12.1.2 Mac OS X

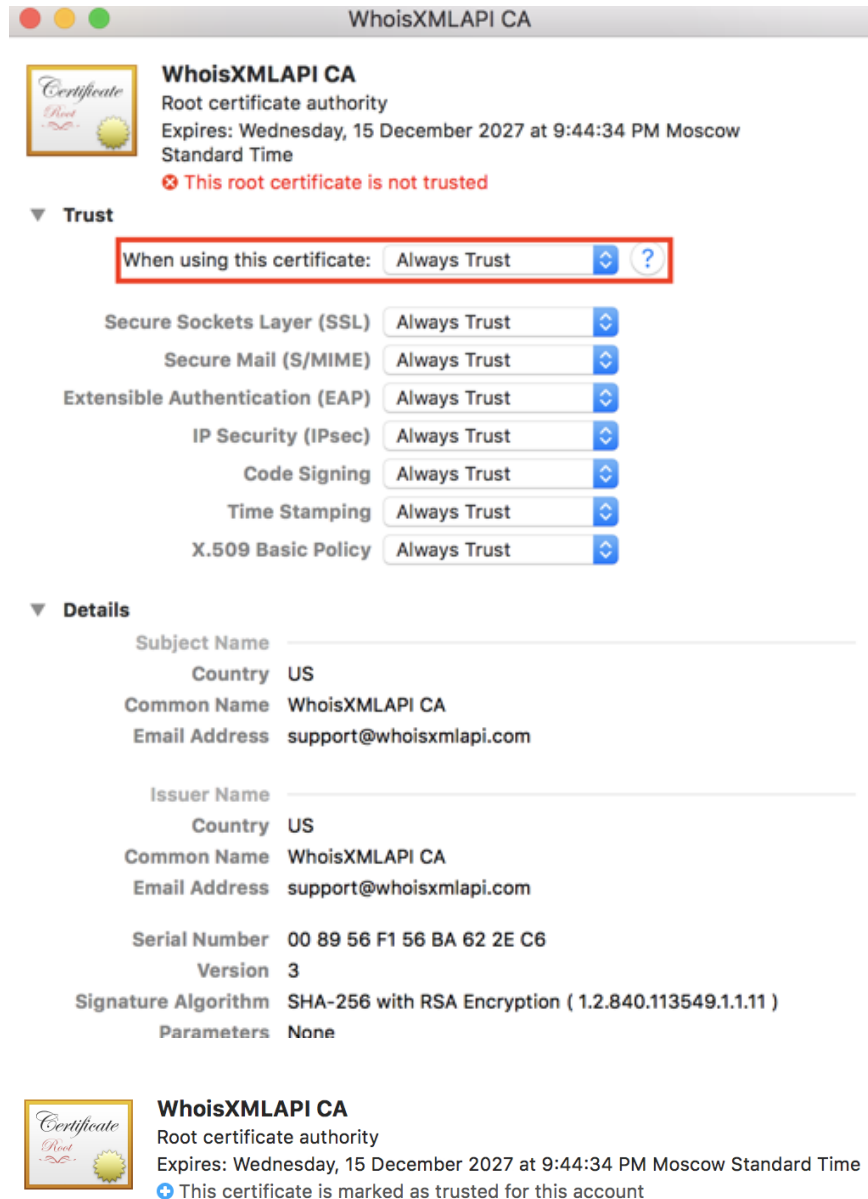
Double click the file **pack.p12**. The system will prompt for the password of the package, type it in, and press OK):



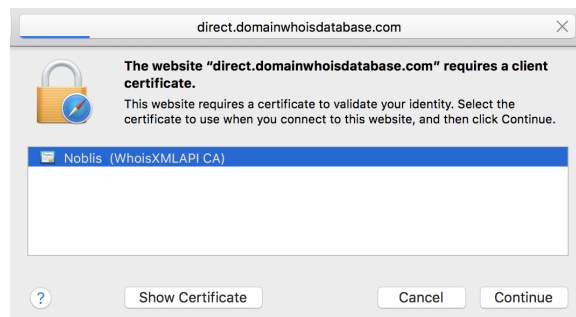
Note: you cannot paste the password into this latter dialog, so you need to type it. The Keychain Access tool window will open after the import:



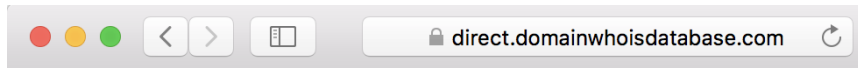
WhoisXMLAPI certificate is not trusted by default, so double click on WhoisXMLAPI ca certificate. Choose "Always Trust" from the dropdown menu and close the window. The Administrator password is required to apply this setting. Afterwards, our root certificate should appear as trusted:



If you start the Safari web-browser and open any of the URLs listed in Section 12.2, it will ask for certificate to be used for authentication: and the username-password pair to access the keychain:



Then the requested page will open securely and without the basic http authentication.



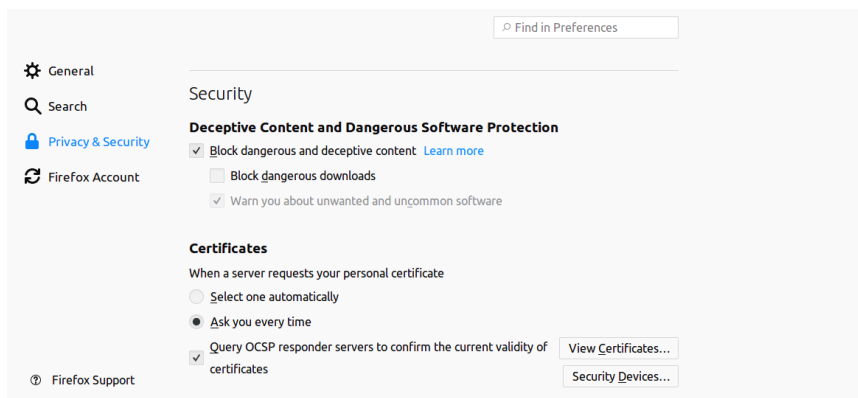
Index of /whois_database

Name	Last modified	Size	Description
Parent Directory		-	
ae/	2016-10-02 05:34	-	
alexa_01_01_2017/	2017-03-24 18:32	-	
alexa_01_12_2016/	2017-09-08 08:00	-	
alexa_04_01_2016/	2016-04-25 03:38	-	
alexa_04_01_2017/	2017-09-08 07:42	-	
alexa_04_26_2016/	2016-04-30 11:31	-	
alexa_07_01_2016/	2016-07-05 00:11	-	
alexa_07_01_2017/	2017-07-17 17:05	-	
alexa_09_03_2017/	2017-12-09 19:42	-	
alexa_10_01_2016/	2016-10-06 18:54	-	
at/	2016-10-02 05:42	-	
au/	2016-10-02 05:43	-	

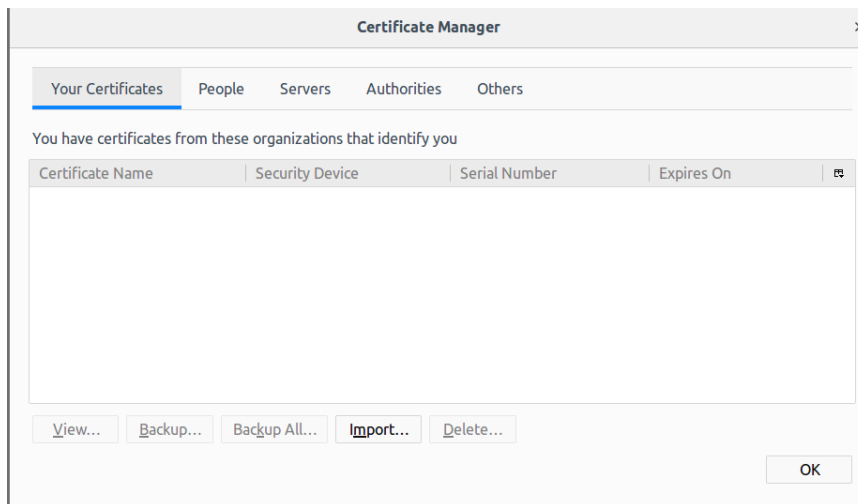
12.1.3 Linux

On Linux systems the procedure is browser dependent. Some browsers (e.g. Opera) use the standard database of the system, while others, such as Firefox, use their own certificate system. We show briefly how to handle both cases.

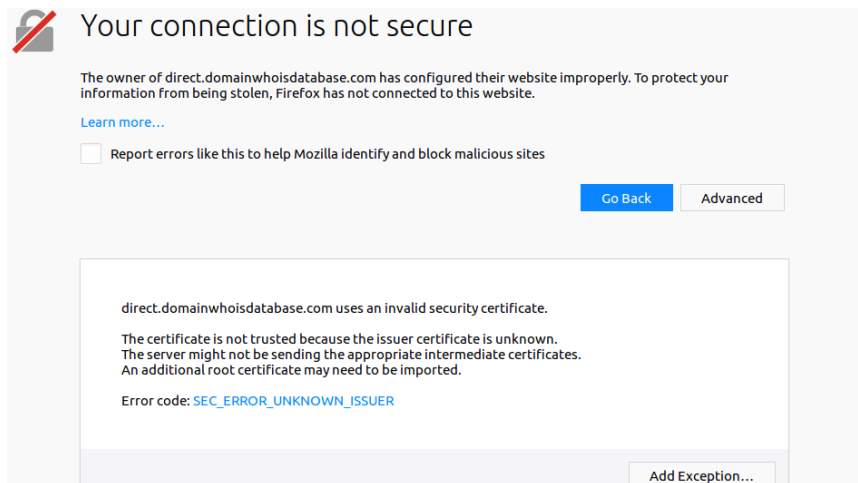
Firefox. Go to Edit → Preferences in the menu. Choose the "Privacy/Security" tab on the left. You should see the following:



Press "View Certificates" and choose the "Your Certificates" tab. The certificate manager will appear:



Press "Import", choose the file "package.p12", and enter the password you were given along with the certificate. You should see the certificate in the list. Now open any of the accessible URLs. You shall be warned as the browser considers the page as insecure:



However, as you are using our trusted service, you can safely add an exception by pressing the button on the bottom. Add the exception permanently. Doing these steps you will be able to access the URLs mentioned in the last Section of the present document without the basic http authentication.

Opera. Opera can use the certificates managed by the command-line tools available on Linux. To add the certificate, you need to install these tools.

On Debian/Ubuntu/Mint, you should do this by

```
sudo apt-get install libnss3-tools
```

while on Fedora and other yum-based systems:

```
yum install nss-tools
```

(Please consult the documentation of your distribution if you use a system in another flavor.) The command for adding the certificate is

```
pk12util -d sql:\$HOME/.pki/nssdb -i pack.p12
```

This will prompt you for the certificate password. You can list your certificates by


```
certutil -d sql:\$HOME/.pki/nssdb -L
```

Now if you open any of the accessible URLs listed at the end of this document, first of all you need to add an exception for the self-signed SSL certificate of the webpage. Then the browser will offer a list of your certificates to decide which one to use with this webpage. Having chosen the just installed certificate, you shall have the secure access to the page, without being prompted for a password.

12.2 Accessible URLs

Currently you can access the following URLs with this method. You shall find the feeds under these base URLs. This means, if you replace “<http://domainwhoisdatabase.com>” with “<https://direct.domainwhoisdatabase.com>” in the respective feed names, you shall be able to access all the feeds below the given base url, once you have set up the SSL authentication.

1. https://direct.domainwhoisdatabase.com/whois_database/
2. https://direct.domainwhoisdatabase.com/domain_list/
3. https://direct.bestwhois.org/domain_name_data/
4. https://direct.bestwhois.org/cctld_domain_name_data/
5. https://direct.bestwhois.org/ngtld_domain_name_data/

13 FTP access of WHOIS data

WHOIS data can be downloaded from our ftp servers, too. In case of newer subscribers the ftp access is described on the web page of the subscription.

13.1 FTP clients

You can use any software which supports the standard ftp protocol. On most systems there is a command-line ftp client. As a GUI client we recommend FileZilla (<https://filezilla-project.org>), which is a free, cross-platform solution. Thus it is available for most common OS environments, including Windows, Mac OS X, Linux and BSD variants.

On Windows systems, the default downloads of FileZilla contain adware, thus most virus protection software do not allow to run them. To overcome this issue, download FileZilla from the following URL:

https://filezilla-project.org/download.php?show_all=1

The files downloaded from this location do not contain adware.

13.2 FTP access

For the subscriptions after 2020, the ftp access to the data works with the following settings:

Host: [datafeeds.whoisxmlapi.com](ftp://datafeeds.whoisxmlapi.com)

Port: 21210

Username: 'user'

Password: the same as your personal API Key which you can obtain from the “My Products” page of the given service

Base path: `ftp://datafeeds.whoisxmlapi.com:21210`

Consult also the information pages of your subscription.

13.3 FTP directory structure of legacy and quarterly subscriptions

This applies to legacy and quarterly subscriptions, the data can be accessed as described below in case of legacy subscriptions (i.e. those who use `bestwhois.org` and `domainwhoisdatabase.com` for web-based access).

As a rule of thumb, if the feed you download has the base URL is

`https://domainwhoisdatabase.com`

you will find it on the ftp server

`ftp.domainwhoisdatabase.com`

while if it is under

`https://bestwhois.org`

you have to connect the ftp server

`ftp.bestwhois.org port 2021`

(please set the port information in your client) to access the data.

If you log in into the server, you will find the data in a subdirectory in your root ftp directory named after the feed. There are some exceptions, which are documented in the description of the given feed in the appropriate manual. You will see only those subdirectories which are accessible within any of your subscription plans there.

A word of caution: as most of the feeds contain a huge amount of data, some ftp operations can be slow. For instance, to obtain the directory listing of some of the feed directories may take a few minutes, so please be patient, do not cancel the operation after a shorter time. (Ftp does not have the feature to show a part of the directory listing or a listing stored in a cache, as in case of the web access.)

If your subscription covers a subset of quarterly releases only, you will find these under `quarterly_gtld` and `quarterly_cctld`, in a subdirectory named after the release version.

13.4 FTP firewall settings for legacy subscriptions

Our FTP servers use 4 ports: 21, 2021, 2121, and 2200. In order to use our ftp service, you need to ensure that the following ports:

- for `ftp.domainwhoisdatabase.com`: 21, 2121, and 2200,
- for `ftp.bestwhois.org`: 2021, 2121, and 2200

are open on both TCP and UDP on your firewall.

If the respective ports are not open, you will encounter either of the following behaviors: You cannot access the respective server. You can access the respective server, but after login, you can't even get the directory listing, it runs onto timeout. If you encounter any of these problems, please revise your firewall settings.

End of manual.