Quarterly ccTLD WHOIS Database Reference Manual

Whois API Inc.

https://www.whoisxmlapi.com

Copyright ©2010-2023

About this document

This document describes quarterly WHOIS database releases containing data for ccTLDs. It contains information relevant for all database releases. The variable details of each release can be found in the files "README.txt" distributed with each database release.

File URL:

```
https://www.domainwhoisdatabase.com/docs/Quarterly_CCTLD_WHOIS_Database_Reference_Manual
```

File version 2.20
Approved on 2023-09-12.

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

```
https://www.domainwhoisdatabase.com/docs
```

Contents

1 Introduction

1.1 About

Our Whois Database Downloads provide archived historic whois databases in both parsed and raw format for download as database dumps(MYSQL or MYSSQL dump) or CSV files. In general a database is produced each quarter. The current version (v8) provides downloads for the major ccTLDs: .de,.tk,.co,.eu,.fr,.cn,.ru,.uk,.nl,.at plus hundreds of new cctlds. The database contains exactly one whois record per domain name.

1.2 Database releases

In each quarter a new database release is issued. Database releases are identified with a *version number* in the format of a letter "v" followed by a number. The numbers are incremented upon each release. E.g. "v7" was released on September 1, 2017, whereas v8 was released on December 1, 2017.

1.3 Release and update announcement RSS

New quarterly database releases, as well as occasional post-release updates are announced via a public RSS feed. The feed URL is

```
https://domainwhoisdatabase.com/docs/feeds/quarterly_release_announce.xml
```

An RSS item is published upon each release publication or update. The body of the RSS items is in JSON format to facilitate automated processing. A typical message body looks like this:

```
{"event":"release","available_from":"Fri Sep 1 08:03:51 UTC 2023",
"dbtype":"gTLD","dbversion":"v45"}
```

The fields are:

event Type of event. It can be 'release' (a new release is available) or 'update' (data in the release are updated after release.

available_from Date and time from which the released/updated data are available.

dbtype Database type, can be 'gTLD' or 'ccTLD', depending on whether it is a gTLD or ccTLD database release.

dbversion database version, e.g. 'v45'.

1.4 Lower level domains treated as TLDs

In case of some ccTLDs there are certain lower level domains which are separately managed from their top-level domains. For instance, certain countries tend to separate the academic sector in a separately managed SLD, e.g. ac.uk of the ccTLD uk. Even if not separately managed, sometimes SLDs (or even lower level domains) tend to have a number of domains commensurable to those in the rest of the TLD. The collection of the data of all these subdomains can require a separate technical treatment, different from that of the respective TLD.

In quarterly ccTLD data feeds with database versions v16 or earlier, certain lower-level domains are treated as if they were TLDs. This means:

- They have data files separate from those of their TLD,

- in the release statistics they are also separately treated,

- the "List of supported TLDs" include them, too,

- the present documentation uses the term "TLD" in the sense including these,

- the support scripts (like downloaders) treat them also as they were separate TLDs.

For instance, to get a complete list of domains in the `.uk` TLD from "v16" database, the files for the "TLDs":

```
ac.uk, co.uk, gov.uk, ltd.uk, me.uk, mod.uk, net.uk, nhs.uk, org.uk,
                parliament.uk, plc.uk, police.uk
```

have separate data files which have to be downloaded and merged into the " .uk" data.

A list of such lower-level domains can be deduced from the list of supported TLDs (in the `doc/vXX.tlds` text file of the release directory), and are also listed in the following appendix of the present document:

```
http://domainwhoisdatabase.com/docs/LowerLevelDomainsInCCTLDQuarterlies
```

Starting with release v17, this distinction is eliminated: for each ccTLD, all the subdomain data are in the file belonging to their gTLD, respective of the management and other characteristics of these subdomains. The daily ccTLD feeds never had this separation of any lower-level domain. Note also that all this implies that domain listings include domains below second or even lower levels.

1.5 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 separete 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.

1.6 Database release locations

The main URL of a given release (vXX, e.g. v22) is

```
https://domainwhoisdatabase.com/domain_list_quarterly/vXX
```

(replace "vXX" by the actual release number). At this URL a plain http authentication is used. Clients having ssl-authenticated access can find the data also under

```
https://direct.domainwhoisdatabase.com/domain_list_quarterly/vXX
```

Clients having ftp access will find the data under the subdirectory

```
domain_list_quarterly/vXX
```

in their ftp home directory.

1.7 Variable documentation of the releases

The present documentation is supplemented by a file named "README.txt" in the main directory in each release.
For releases v6-v8, the respective file is named "README_cctld_Q$ver.appendix", e.g. "README_cctld_Q22_appe
Releases before v19 have a textual README only, in which this information is also included.
The README file contains the following information:

Data sizes. Record counts by tld.

Directory listings. Directory tree with file sizes.

Record count details. The following listing contains details of unique record count by tlds and by fields. There are three important fields that we gather unique record count on:

- contactEmail,
- registrant_country and
- whoisServer.

Release statistics. The following data are provided:

- Top 5 registrant countries,
- Top 5 WHOIS servers,
- Top 5 contact e-mails.
- ccTLD coverage statistics

Records count details. The record counts can be found in the README of the actual release.

1.8 Data formats

The download comes in 2 formats: CSVs and Database dumps

- The files are generally compressed in .tar.gz, use the following commands/tools to uncompress:

    - on linux: `tar -zxvf input.tar.gz`
    - on Windows: use a software tool such as winzip, winrar
    - on Mac OS X: you may do `tar -zxvf input.tar.gz` or use a suitable software GUI tool.

- Some databases (for instance, those in the `csv/tlds_combined` or in the `database_dump/mysqldump_co` subdirectories) are compressed into multipart compressed files. The parts are all numbered with a four digit serial number at the end of the file name. These files can be simply uncompressed by joining them together and sending them to the tar program for example:

    ```
    cat simple-v8.tar.gz.[0-9][0-9][0-9][0-9] | tar xzf -
    ```

    Please make sure you have all the parts downloaded and no other files with the same name pattern are present.

- The directory csv/tlds contains 3 subdirectories: simple, regular and full, each represents a version of csv-s, See Section 2 for the description of the formats of the csv files.

- For convenience we also provide multi-part tarball for each type of csv/mysqldump filesets. They are under `csv/csv_tlds_combined` and `database_dump/mysqldump_tld_combined`.

- The directory `csv/csv_tlds_combined` contains 3 directories: simple, regular and full, each represents a version of csvs with multi-part tarballs.

- If you want to download the data of all tlds in one shot, this is the recommended method. Once you download the tarball, on Unix type systems you may run command below to extract it:

    ```
    cat somename.tar.gz.* | tar xzf -
    ```

1.9 Directory structure

The release directory contains the present documentation in html, text and pdf formats. Besides, it contains the follwowing subdirectories:

csv This is the directory with the csv files.

The `csv/tlds` **subdirectory** contains 3 subdirectories, `simple`, `regular`, `full`, containing tar.gz archives of the respective csv files, and their md5 and sha checksums.

The `csv/tlds_combined` **subdirectory** contains the same data organized into three multipart tar.gz archives (for simple, regular, and full) for sake of simpler downloading.

The `csv/domains` **subdirectory (available from releases v8 and later)** contains gzipped csv files listing the domains only by tld, that is, text files with a domain name per line. The subdirectory has a subdirectory for each TLD. Within this subdirectory there are two types of files:

- The files

    ```
    domain_names_$version_$tld.csv.gz
    ```

    , e.g.

<div align="center">

`domain_names_v22_at.csv.gz`

</div>

for the TLD "at" in release v22, contains the list of all domains we sourced at the beginning of our data collection procedure and attempted to get data for, with or without success. Hence there are domains in these lists for which the release contains no WHOIS data as they were unavailable.

- The files

<div align="center">

`verified_domain_names_$version_$tld.csv.gz`

</div>

, e.g.

<div align="center">

`verified_domain_names_v21_at.csv.gz`

</div>

for the TLD "at" in release v22, contains the list of all the domains for which there is an actual WHOIS record available in the release.

- The files

<div align="center">

`missing_domain_names_$version_$tld.csv.gz`

</div>

, e.g.

<div align="center">

`missing_domain_names_v21_uk.csv.gz`

</div>

for the TLD "uk" in release v21, contains the list of the domains that were included in the list of domains we had sourced at the beginning of the data generation procuedure, but it was found that they did not exist when their data were queried. These files are provided starting with the v21 ccTLD release.

- The files

<div align="center">

`reserved_domain_names_$version_$tld.csv.gz`

</div>

, e.g.

<div align="center">

`reserved_domain_names_v21_uk.csv.gz`

</div>

for the TLD "uk" in release v35, contains the list of the domains of reserved domains of `.uk` like, e.g., `unitednations.uk` or `imf.uk`. Consult https://www.iana.org/domains/reserved for a more detailed explanation on reserved domain names. These files are provided starting with the v35 gTLD release.

All these files have `.md5` and `.sha256` checksums next to them.

database_dump This is the directory with mysql dumps. `database_dump/mysqldump` contains mysql-dumps, schema and their checksums grouped by tld. Table-only files can be found under the `tables` subdirectory of each tld subdirectory. `database_dump/mysqldump_combined` is a single set of files that contains data for all tlds. `database_dump/perconna` contains binary dumps, if you use this, the import speed is faster, but it's less portable because it only supports certain minimum versions. It is only supported for MySQL server 5.6+.

docs This directory contains a link to the download scripts described in Section 5, the list of tlds in the release, and a brief pdf datasehet of the release.

The sample data directory of the release contains the aforementioned pdf datasheet and two subdirectories:

sample Sample csv data in the structure of the `csv/tlds` subdirectory of the release.

mysqldump_sample Sample sql dumps in the structure of the `database_dump/mysqldump` subdirectory of the release, without checksum files.

1.10 Feeds

The data described here can be downloaded in an automated way using Python and bash scripts describded in Section 5. The *feed* to be secified for the scripts is:

<div align="center">

7

</div>

- `domain_list_quarterly`,

and the version specification `v8` should be used.

1.11 Auxiliary files to support download automation

The subdirectory `docs/vXX.tlds` contains the actual list of supported TLDs in each release, e.g. `docs/v30.tlds` for v30. It is a comma-separated list of TLD names in a single line. This can be used in support of automated download process by reading in this list and then download each set of files for the TLD.

2 CSV file formats

2.1 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 7 you can find information on software tools to view the contents and handle large csv files on various platforms.

2.1.1 Loading CSV files into MySQL and other database systems

In Section 5 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 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`.

2.2 File formats

- 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 3 types of CSVs: simple, regualar and full.

  simple : 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", "Audit_auditUpdatedDate",
  ```

```
"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_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"
```

regular : in addition to the fields of "simple", it contains additional fields describing the billing contact, technical contact, and zone contact. Thus the fields are as follows:

```
"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","registarIANAID"
```

full: in addition to the fields of the simple 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","registarIANAID"
```

2.3 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.

2.4 Maximum data field lengths

```
domainName: 256, registrarName: 512,  contactEmail: 256,
whoisServer: 512, nameServers: 256, createdDate: 200,
updatedDate: 200, expiresDate: 200, standardRegCreatedDate: 200,
standardRegUpdatedDate: 200, standardRegExpiresDate: 200,
status: 65535, Audit_auditUpdatedDate: 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, registarIANAID: 65535
```

2.5 Standardized country fields

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

<div align="center">

`http://www.domainwhoisdatabase.com/docs/countries.txt`

</div>

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

2.6 CSV data schema

Below is a detailed comparison of the fields that are present different version of the csv files(simple, regular and full). The leftmost column reflects the fields of the MySQL schema.

WhoisRecord	Simple	Regular	Full
WhoisRecord			
domainName	domainName	domainName	domainName
createdDate	createdDate	createdDate	createdDate
updatedDate	updatedDate	updatedDate	updatedDate
expiresDate	expiresDate	expiresDate	expiresDate
domainNameExt	NA	NA	NA
nameServers	nameServers	nameServers	nameServers
nameServers/rawText	NA	NA	NA
nameServers/hostNames	NA	NA	NA
nameServers/Hostnames/Address	NA	NA	NA
nameServers/ips	NA	NA	NA
nameServers/ips/Address	NA	NA	NA
status	status	status	status
rawText	NA	NA	WhoisRecord_rawText
parseCode	NA	NA	NA
header	NA	NA	NA
strippedText	NA	NA	NA
footer	NA	NA	NA
audit	NA	NA	NA
audit/createdDate	NA	NA	NA
audit/updatedDate	Audit_auditUpdatedDate	Audit_auditUpdatedDate	Audit_auditUpdatedDate
registrarName	registrarName	registrarName	registrarName
registrarIANAID	NA	registrarIANAID	registrarIANAID
contactEmail	contactEmail	contactEmail	contactEmail
domainAvailability	NA	NA	NA
domainNameExt	NA	NA	NA
estimatedDomainAge	NA	NA	NA
registrant			
name	registrant_name	registrant_name	registrant_name
organization	registrant_organization	registrant_organization	registrant_organization
street1	registrant_street1	registrant_street1	registrant_street1
street2	registrant_street2	registrant_street2	registrant_street2
street3	registrant_street3	registrant_street3	registrant_street3
street4	registrant_street4	registrant_street4	registrant_street4
city	registrant_city	registrant_city	registrant_city
state	registrant_state	registrant_state	registrant_state
postalCode	registrant_postalCode	registrant_postalCode	registrant_postalCode
country	registrant_country	registrant_country	registrant_country
email	registrant_email	registrant_email	registrant_email
telephone	registrant_telephone	registrant_telephone	registrant_telephone
telephoneExt	registrant_telephoneExt	registrant_telephoneExt	registrant_telephoneExt
fax	registrant_fax	registrant_fax	registrant_fax
faxExt	registrant_faxExt	registrant_faxExt	registrant_faxExt
rawText	NA	registrant_rawText	registrant_rawText
unparsable	NA	NA	NA
administrativeContact			
name	administrativeContact_name	administrativeContact_name	administrativeContact_name
organization	administrativeContact_organization	administrativeContact_organization	administrativeContact_organization
street1	administrativeContact_street1	administrativeContact_street1	administrativeContact_street1
street2	administrativeContact_street2	administrativeContact_street2	administrativeContact_street2
street3	administrativeContact_street3	administrativeContact_street3	administrativeContact_street3
street4	administrativeContact_street4	administrativeContact_street4	administrativeContact_street4
city	administrativeContact_city	administrativeContact_city	administrativeContact_city
state	administrativeContact_state	administrativeContact_state	administrativeContact_state
postalCode	administrativeContact_postalCode	administrativeContact_postalCode	administrativeContact_postalCode
country	administrativeContact_country	administrativeContact_country	administrativeContact_country
email	administrativeContact_email	administrativeContact_email	administrativeContact_email
telephone	administrativeContact_telephone	administrativeContact_telephone	administrativeContact_telephone
telephoneExt	administrativeContact_telephoneExt	administrativeContact_telephoneExt	administrativeContact_telephoneExt
fax	administrativeContact_fax	administrativeContact_fax	administrativeContact_fax

WhoisRecord	Simple	Regular	Full
faxExt	administrativeContact.faxExt	administrativeContact.faxExt	administrativeContact.faxExt
rawText	administrativeContact.rawText	administrativeContact.rawText	administrativeContact.rawText
unparsable	NA	NA	NA
billingContact			
name	NA	billingContact.name	billingContact.name
organization	NA	billingContact.organization	billingContact.organization
street1	NA	billingContact.street1	billingContact.street1
street2	NA	billingContact.street2	billingContact.street2
street3	NA	billingContact.street3	billingContact.street3
street4	NA	billingContact.street4	billingContact.street4
city	NA	billingContact.city	billingContact.city
state	NA	billingContact.state	billingContact.state
postalCode	NA	billingContact.postalCode	billingContact.postalCode
country	NA	billingContact.country	billingContact.country
email	NA	billingContact.email	billingContact.email
telephone	NA	billingContact.telephone	billingContact.telephone
telephoneExt	NA	billingContact.telephoneExt	billingContact.telephoneExt
fax	NA	billingContact.fax	billingContact.fax
faxExt	NA	billingContact.faxExt	billingContact.faxExt
rawText	NA	billingContact.rawText	billingContact.rawText
unparsable	NA	NA	NA
technicalContact			
name	NA	technicalContact.name	technicalContact.name
organization	NA	technicalContact.organization	technicalContact.organization
street1	NA	technicalContact.street1	technicalContact.street1
street2	NA	technicalContact.street2	technicalContact.street2
street3	NA	technicalContact.street3	technicalContact.street3
street4	NA	technicalContact.street4	technicalContact.street4
city	NA	technicalContact.city	technicalContact.city
state	NA	technicalContact.state	technicalContact.state
postalCode	NA	technicalContact.postalCode	technicalContact.postalCode
country	NA	technicalContact.country	technicalContact.country
email	NA	technicalContact.email	technicalContact.email
telephone	NA	technicalContact.telephone	technicalContact.telephone
telephoneExt	NA	technicalContact.telephoneExt	technicalContact.telephoneExt
fax	NA	technicalContact.fax	technicalContact.fax
faxExt	NA	technicalContact.faxExt	technicalContact.faxExt
rawText	NA	technicalContact.rawText	technicalContact.rawText
unparsable	NA	NA	NA
zoneContact			
name	NA	zoneContact.name	zoneContact.name
organization	NA	zoneContact.organization	zoneContact.organization
street1	NA	zoneContact.street1	zoneContact.street1
street2	NA	zoneContact.street2	zoneContact.street2
street3	NA	zoneContact.street3	zoneContact.street3
street4	NA	zoneContact.street4	zoneContact.street4
city	NA	zoneContact.city	zoneContact.city
state	NA	zoneContact.state	zoneContact.state
postalCode	NA	zoneContact.postalCode	zoneContact.postalCode
country	NA	zoneContact.country	zoneContact.country
email	NA	zoneContact.email	zoneContact.email
telephone	NA	zoneContact.telephone	zoneContact.telephone
telephoneExt	NA	zoneContact.telephoneExt	zoneContact.telephoneExt
fax	NA	zoneContact.fax	zoneContact.fax
faxExt	NA	zoneContact.faxExt	zoneContact.faxExt
rawText	NA	zoneContact.rawText	zoneContact.rawText
unparsable	NA		
registryData			
createdDate	NA	NA	NA
updatedDate	NA	NA	NA
expiresDate	NA	NA	NA
registrant	NA	NA	NA
domainName	NA	NA	NA
nameServers	NA	NA	NA
nameServers/rawText	NA	NA	RegistryData.rawText
nameServers/hostNames	NA	NA	NA
nameServers/hostNames/Address	NA	NA	NA
nameServers/ips	NA	NA	NA
nameServers/ips/Address	NA	NA	NA
status	NA	NA	NA
rawText	NA	NA	NA
parseCode	NA	NA	NA
header	NA	NA	NA
strippedText	NA	NA	NA
footer	NA	NA	NA
audit	NA	NA	NA
audit/createdDate	NA	NA	NA
audit/updatedDate	NA	NA	NA
registrarName	NA	NA	NA
registrarIANAID	NA	NA	NA
createdDateNormalized	NA	NA	NA
updatedDateNormalized	NA	NA	NA
expiresDateNormalized	NA	NA	NA
whoisServer	NA	NA	NA
referralURL	NA	NA	NA
registryData	**registrant**		
name	NA	NA	NA
organization	NA	NA	NA
street1	NA	NA	NA
street2	NA	NA	NA
street3	NA	NA	NA
street4	NA	NA	NA
city	NA	NA	NA
state	NA	NA	NA
postalCode	NA	NA	NA

WhoisRecord	Simple	Regular	Full
country	NA	NA	NA
email	NA	NA	NA
telephone	NA	NA	NA
telephoneExt	NA	NA	NA
fax	NA	NA	NA
faxExt	NA	NA	NA
rawText	NA	NA	NA
unparsable	NA	NA	NA
registryData	**administrativeContact**		
name	NA	NA	NA
organization	NA	NA	NA
street1	NA	NA	NA
street2	NA	NA	NA
street3	NA	NA	NA
street4	NA	NA	NA
city	NA	NA	NA
state	NA	NA	NA
postalCode	NA	NA	NA
country	NA	NA	NA
email	NA	NA	NA
telephone	NA	NA	NA
telephoneExt	NA	NA	NA
fax	NA	NA	NA
faxExt	NA	NA	NA
rawText	NA	NA	NA
unparsable	NA	NA	NA
registryData	**billingContact**		
name	NA	NA	NA
organization	NA	NA	NA
street1	NA	NA	NA
street2	NA	NA	NA
street3	NA	NA	NA
street4	NA	NA	NA
city	NA	NA	NA
state	NA	NA	NA
postalCode	NA	NA	NA
country	NA	NA	NA
email	NA	NA	NA
telephone	NA	NA	NA
telephoneExt	NA	NA	NA
fax	NA	NA	NA
faxExt	NA	NA	NA
rawText	NA	NA	NA
unparsable	NA	NA	NA
registryData	**technicalContact**		
name	NA	NA	NA
organization	NA	NA	NA
street1	NA	NA	NA
street2	NA	NA	NA
street3	NA	NA	NA
street4	NA	NA	NA
city	NA	NA	NA
state	NA	NA	NA
postalCode	NA	NA	NA
country	NA	NA	NA
email	NA	NA	NA
telephone	NA	NA	NA
telephoneExt	NA	NA	NA
fax	NA	NA	NA
faxExt	NA	NA	NA
rawText	NA	NA	NA
unparsable	NA	NA	NA
registryData	**zoneContact**		
name	NA	NA	NA
organization	NA	NA	NA
street1	NA	NA	NA
street2	NA	NA	NA
street3	NA	NA	NA
street4	NA	NA	NA
city	NA	NA	NA
state	NA	NA	NA
postalCode	NA	NA	NA
country	NA	NA	NA
email	NA	NA	NA
telephone	NA	NA	NA
telephoneExt	NA	NA	NA
fax	NA	NA	NA
faxExt	NA	NA	NA
rawText	NA	NA	NA
unparsable	NA	NA	NA
NA	standardRegCreatedDate	standardRegCreatedDate	standardRegCreatedDate
NA	standardRegUpdatedDate	standardRegUpdatedDate	standardRegUpdatedDate
NA	standardRegExpiresDate	standardRegExpiresDate	standardRegExpiresDate

3   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.

4 Database dumps

4.1 Hardware requirements for importing mysql dump files

Disk space: at least one single 2 TB partition is required to store mysql data file once it's loaded into mysql server

Memory: at least 16 GB of RAM

The server that collects the whois database has the following spec, it's recommended that your server is comparable to our server:

- Core i7 Quad Core i7-2600 3.4 GHz

- 16 GB DDR3-1333 UDIMM

- First Hard Drive: 2 TB SATA HDD (7200 RPM)

- Second Hard Drive: 2 TB SATA HDD (7200 RPM)

4.2 Software requirements for importing mysql dump files

- Mysql server 5.1+ is recommended although it should work for versions of mysql-server lower than 5.1

- Mysql server 5.6+ is required for importing through binary files

4.3 Importing mysql dump files

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

Due to the large database size especially for `.com`, it is recommended to use load schema first, then load data for each table separately. The bash scripts under the subdirectory `mysql/` can be used as a starting point to help with loading mysqldump files incrementally. It is also recommended to test the load procedure on a small sample dataset first before loading the complete dataset.

4.3.1 Sample data

Complete sample data and schema for a tld can be found in the subdirectories

```
mysqldump_sample/$tld/
```

of the sample data directory (not in the production release) for example, for `.com`, the complete sample data (including schema) are in the subdirectory

```
mysqldump_sample/com/whoiscrawler_v7_com_subset_mysql.sql.gz
```

The schema-only file is

15

```
mysqldump_sample/com/
whoiscrawler_v7_com_subset_mysql_schema.sql.gz
```

Table-only files can be found under

```
mysqldump_sample/com/tables
```

4.3.2 Production data

Complete production data and schema for a tld can be found under

```
database_dump/mysqldump/$tld
```

for example, for .com complete sample data (including schema) are to be found in the compressed file

```
database_dump/mysqldump/com/whoiscrawler_v7_com_subset_mysql.sql.gz
```

The schema-only file is

```
database_dump/mysqldump/com/whoiscrawler_v7_com_mysql_schema.sql.gz
```

Table-only files can be found under

```
database_dump/mysqldump/com/tables/
```

4.3.3 Loading mysqldump files

There are two ways to load mysqldump files:

1. Loading schema first, then load each table's data separately. This is recommened for a large database such as .com.

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

We provide BASH scripts for both approaches. The scripts and their documentations are available from our Github repositiory:

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

under the subdirectory "whoisxmlapi_mysqldump_loaders''. The procedure can be easily performed and understood according to the scripts.

4.3.4 Mysql settings

Please consider tweaking the following parameter in my.cnf to speedup the import. This is what we have one our server, be careful how you tweak yours:

```
innodb_flush_log_at_trx_commit = 2
innodb_log_file_size = 256M
innodb_flush_method = O_DIRECT
```

4.3.5 Loading performance

Using provided scripts on our reference server, importing contact table takes 7 hours, importing the domain_names_wh table takes 6 hours, importing the registry_data table takes 24 hours, importing whois_record takes 20 hours, adding indices takes 4 hours. In total it takes about 61 hours to import the mysqldumps into the whole database with the following hardware and software:
   Intel® Xeon® CPU E5-1650 v2 @ 3.50GHz with 64 GB of RAM, 2TB SATA HDD, Mysql 5.6

4.4 Importing mysql binary files

This Section does not apply to all the database releases, as the binary dumps were not found useful in some cases. E.g. for the GTLD release v23 there are no binary dumps provided. So read this only if the subdirectory `database_dump/percona` directory exists in the release you are using.

Using mysql binary files is a fast way to import the database although a less portable one. It's only supported for mysql server 5.6+.

4.4.1 Input data

Complete input binary data and schema for a tld can be found under the subdirectory

`database_dump/percona`

in 7zipped files named after the given tld. For example, for `.coop` the complete sample data (including schema) is to be found in

`database_dump/percona/coop.7z`

Please find md5 and sha256 sums next to each file for download verification purposes.

4.4.2 Scripts for loading mysql binary files

We provide example BASH scripts to load binary mysql data. The scripts and their documentations are available from our Github repositiory:

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

under the subdirectory "`whoisxmlapi_percona_loader_scripts`". **We recommend the use of our scripts primarily to those who want to import a subset of domains only.**

Those who want to load all data for all domains are advised to use the xtrabackup scripts provided by Percona, downloadable from this link.

We recommend to first familiarize yourself with the operation of Percona scripts studying the Percona xtrabackup Documentation available from Percona, especially its Section 10.1 describing the options of the `innobackupe` script you will use.

The outline of the workflow using the script is as follows:

1. Install mysql and Percona xtrabackup on your platform.

2. Ensure that you have no other databases, preferably use a fresh mysql installation. Percona xtrabackup will recover the status of the database exactly as we have saved it, and denies to do so if your MySQL server is a fresh installation which has not yet been used. You may, however, by using the `--force-non-empty-directories` option of the `innobackupex` script by Percona you will use.

3. Download all the .7z files in the `database_dump/percona` subdirectory of the release. Apart from the domain specific files you will also need the file `MAIN.7z` containing additional files required to restore the full backup. (This is not needed when you are using our scripts.)

4. Uncompress all the 7z files (On linux, `p7zip -d file.7z` for each file will do the job, use a 7z compatible utility on other systems.) Assume that you have all the uncompressed data in the local directory named "percona" now.

5. Stop the MySQL server.

6. As root, run

```
innobackupex --copy-back percona
```

(Replace "percona" with the appropriate directory name if it is in some other directory.) Note: this will take a long time.

7. Start your MySQL server.

You can make a partial backup by specifying the `--databases=LIST` option of `innobackupex`, where "LIST" is a space-separated list of databases for particular domains to be restored. For the domain `aaa` the respective database name is `whoiscrawler_v20_aaa`, you can see these as subdirectory names in the downloaded backup (as well as the names of the 7z files).

4.5 Database schema

The following diagram shows the structure of the three relevant tables.
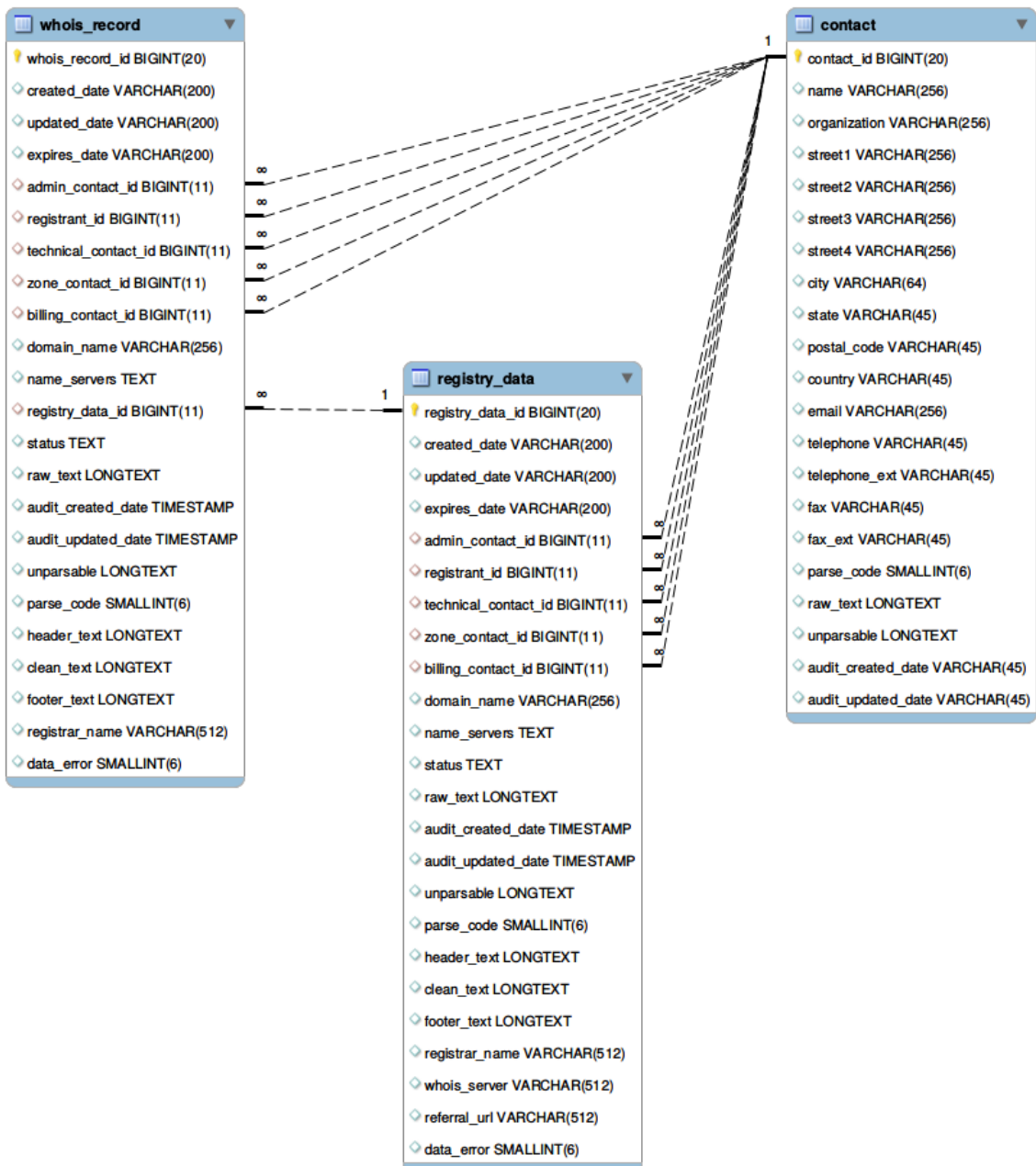
The detailed description of the tables is the following:

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 adminstrative 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 datafrom 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, createdDate).

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 → createdDate or WhoisRecord → registryData → createdDate

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

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: createdDate, expiresDate, referralURL(exists in registryData only), registrarName, status, updatedDate, 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 createdDate 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`

4.6   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-slide scripts are provied 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
```

5 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.

`whoismxlapi_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.

6 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 5 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.

6.1 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 5) 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 1.5 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.

6.2 Downloaders with a GUI

GUI-based downloading is mainly an option for those who download data occasionally as it less efficient than the command-line approach and cannot be automated. Primarily we recommend to use or python downloader (Section 5) 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:
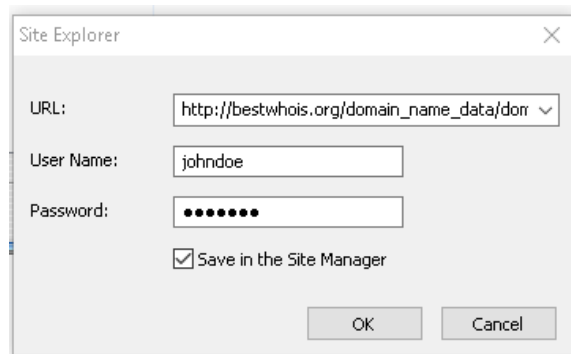
1. Open iGetter

2. Click the right button on "Site explorer", and choose "Enter new URL":
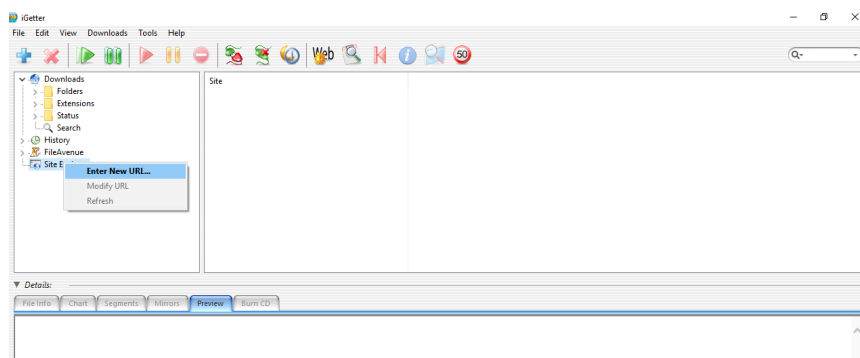
3. A window pops up, paste the feed URL, this time it is

```
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":
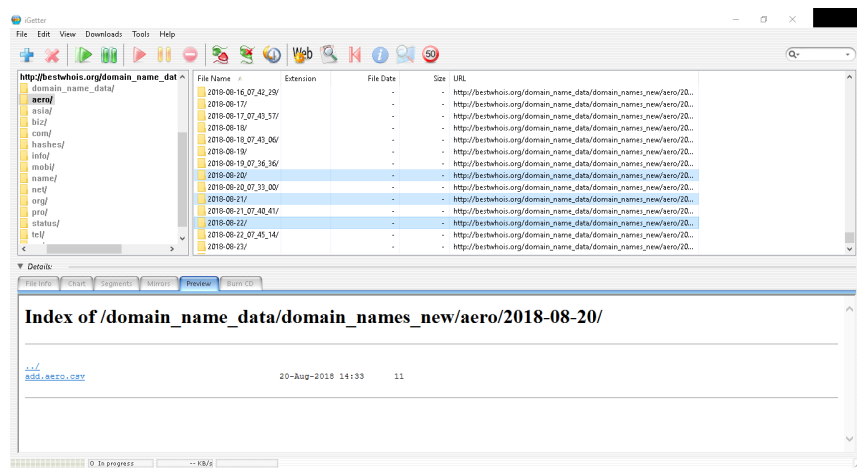
6. Press the right mouse button on this panel, and select "Add to queue". Then say "Yes" to the question "Would you like to download web page contents?". The right part of the upper half of the window will show the download queue now:

7. Click the right button on "Site explorer", and choose "Enter new URL":
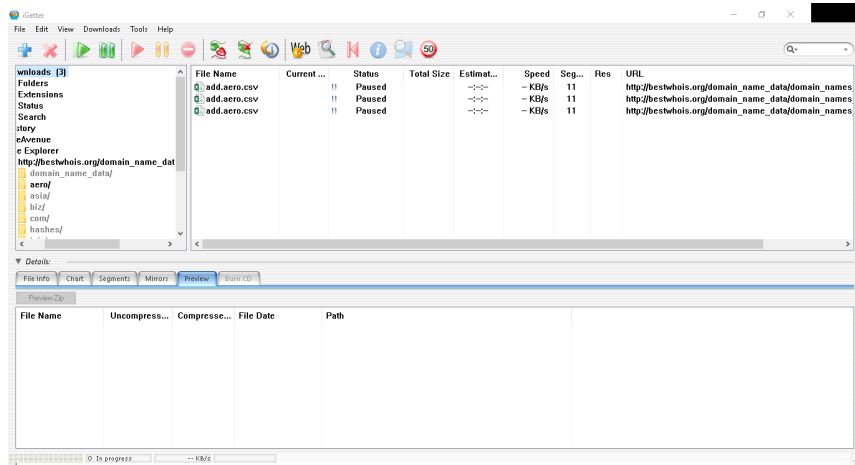
8. Double click now "`biz`" on the left half of the upper part, and follow the same procedure as with "`aero`". When the download queue is prepared, press the green arrow ("Set Auto downloading") button. You can now follow the download procedure in the queue. Your files will be downloaded into the directory "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.

6.3 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/wgetwget, 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 a 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

<div align="center">s</div>

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

<div align="center">26</div>

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 9 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 -l1 --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.

7 Handling large csv files

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

7.1 Line terminators in CSV files

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

1. Unix-style systems, including Linux and BSD use a single "LF"

2. DOS and Windows systems use two characters, "CR" + "LF"

3. Legacy classic Mac systems used to use "CR"

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

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

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

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

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

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

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

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

```
$ todos foo.txt
```

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

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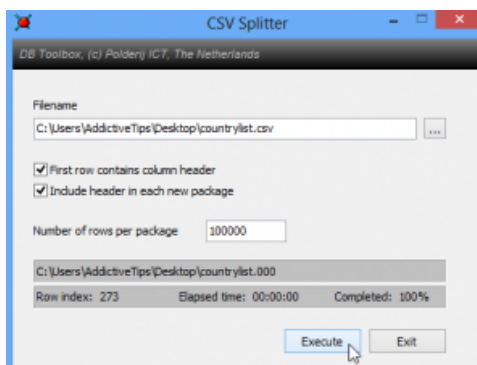
https://www.editpadpro.com/tricklinebreak.html

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

7.2 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 7.4 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 7.4.

7.3   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 7.4

7.4 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

8 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.

8.1 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.

8.2 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", "updatedDate", "whoisServer" (exists in "registryData" only), "nameServers", "administrativeContact", "billingContact", "registrant", "technicalContact", and "zoneContact". For example, a parse code $3_{10} = (11_2)$ means that the only non-empty fields are "createdDate" and "expiresDate", whereas the parse code $8_{10} = (1000_2)$ 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 $00100000000_2 = 512_{10}$ it should be 512. (The mask stands for the non-empty field "registrant").

9 Access via SSL Certifiate Authenticaton

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 neccessary 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.
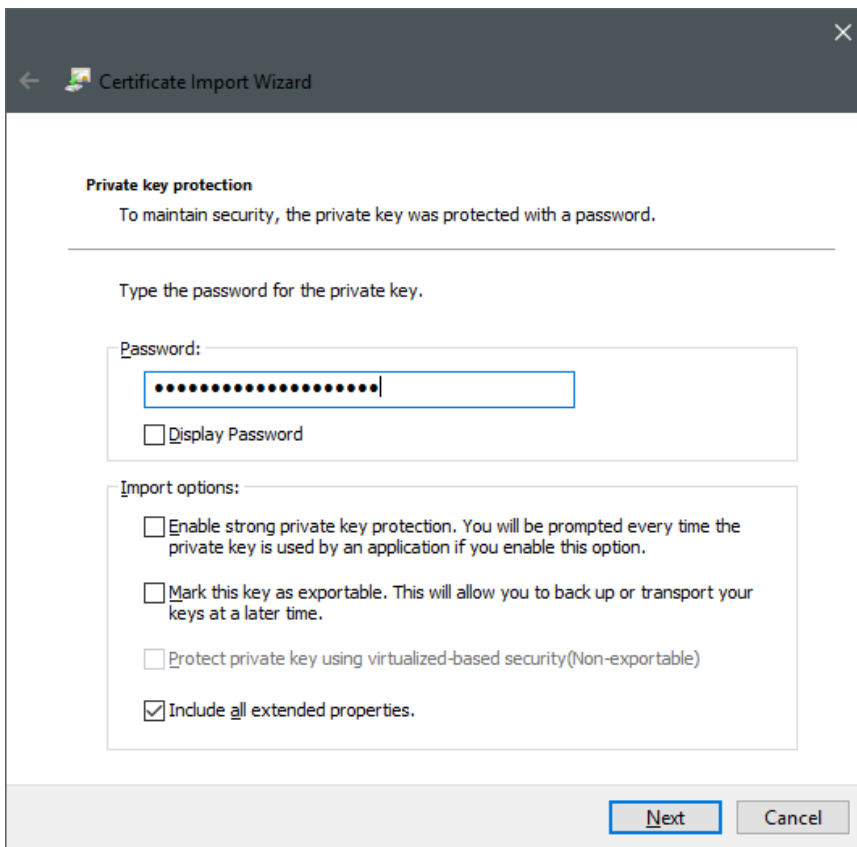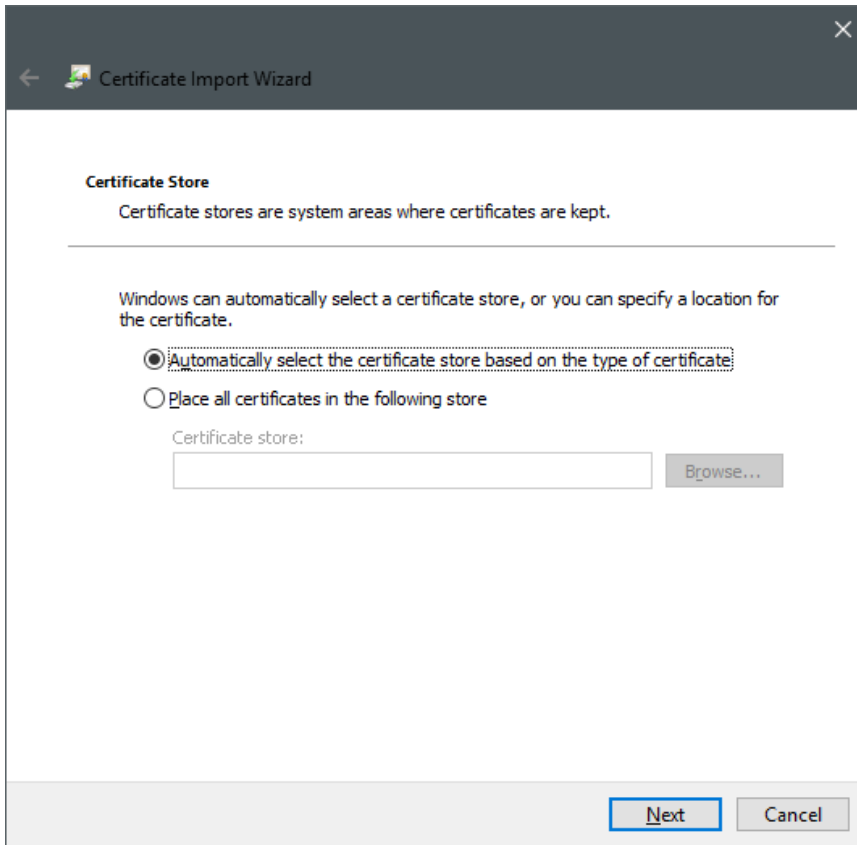
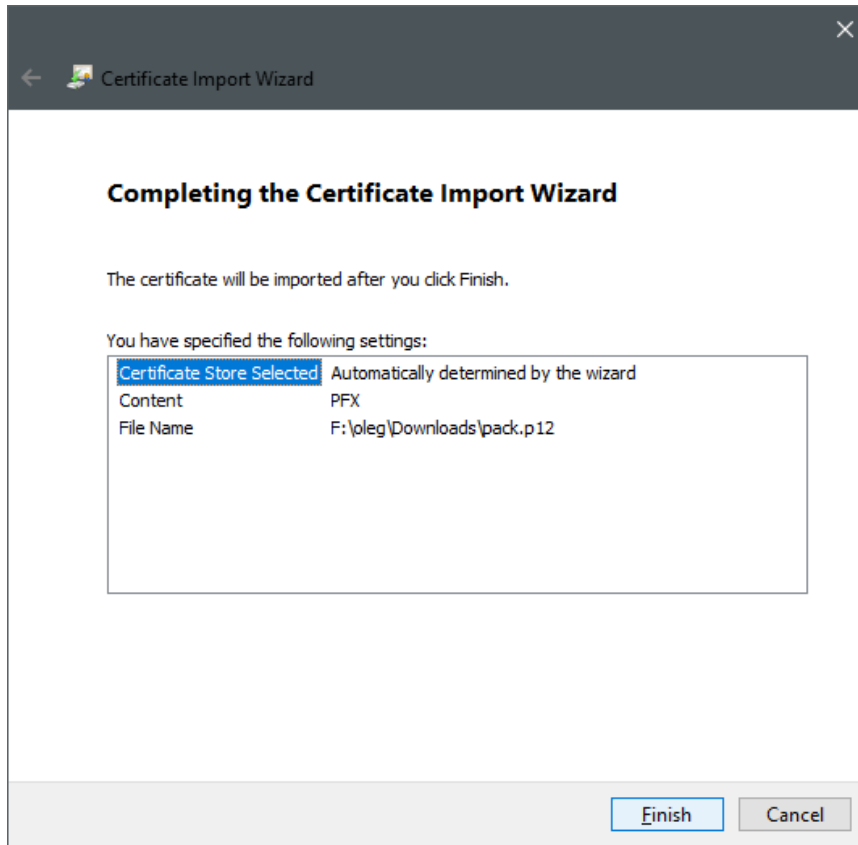## 9.1 Setup instructions

9.1.1 Microsoft Windows

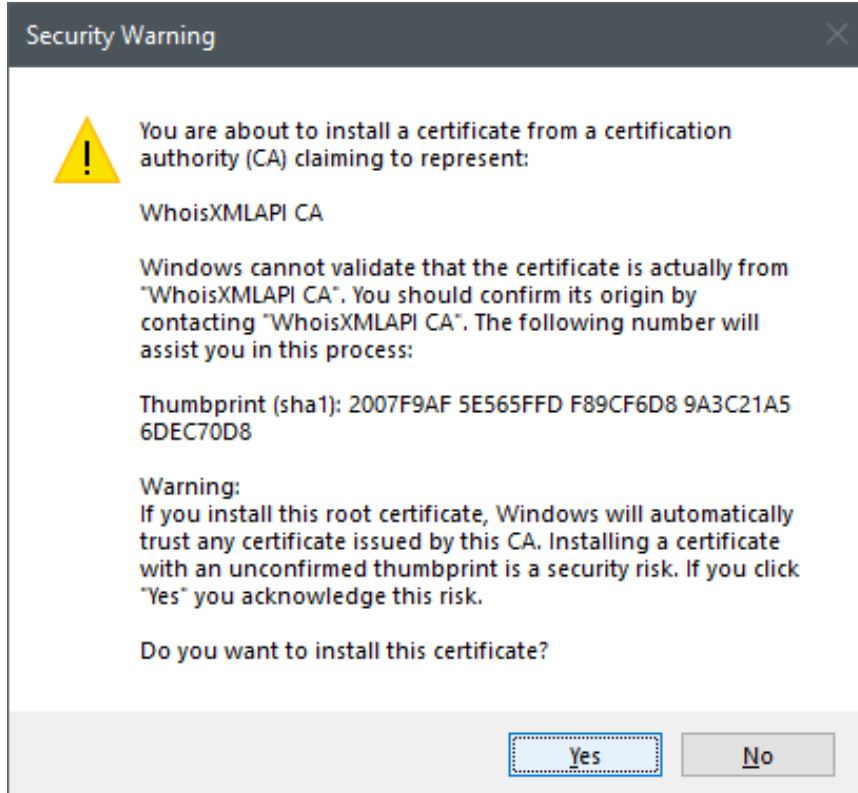Double click on the **pack.p12** file. The following dialog windows will appear, you can proceed with "Next":
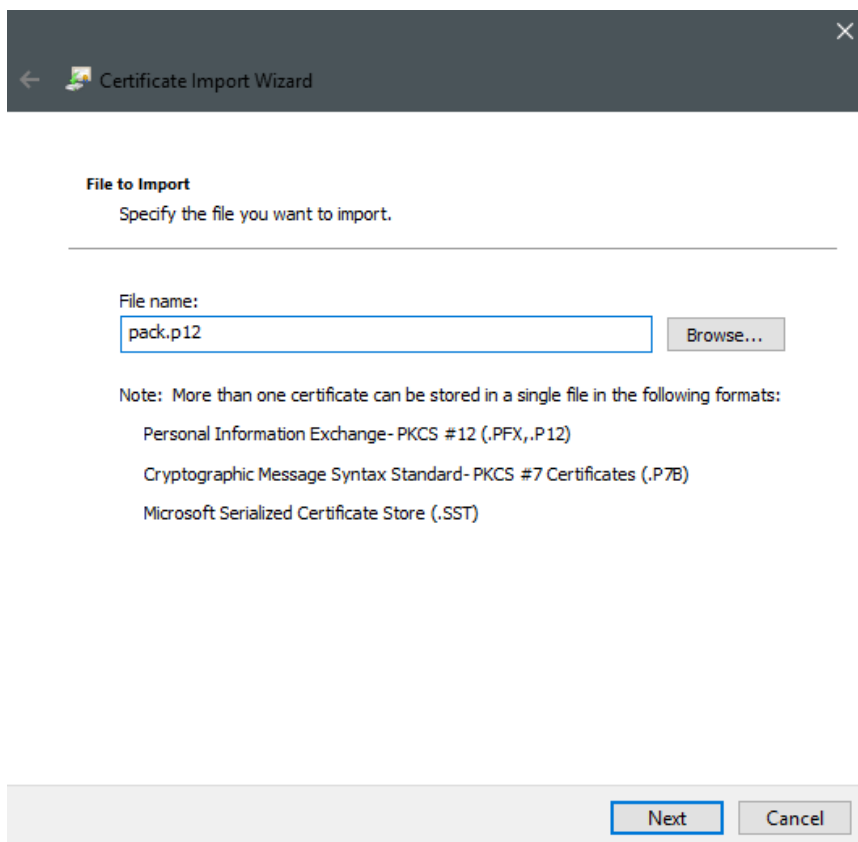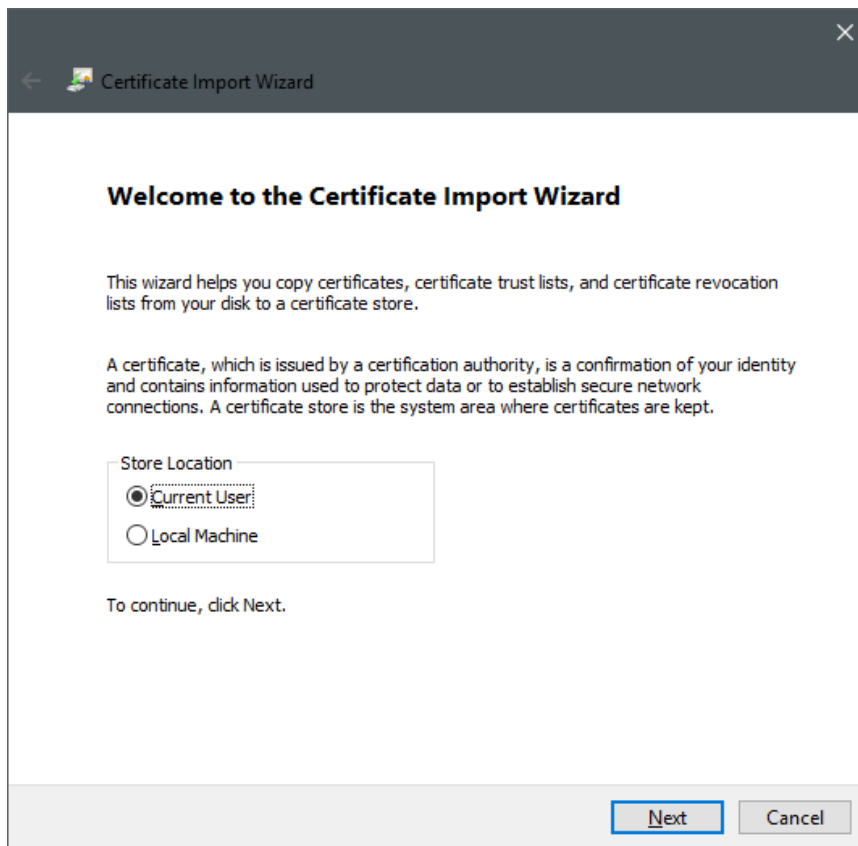
In the next step you should provide the password you got for the package. Then you can just go through the next dialogs with the default settings:
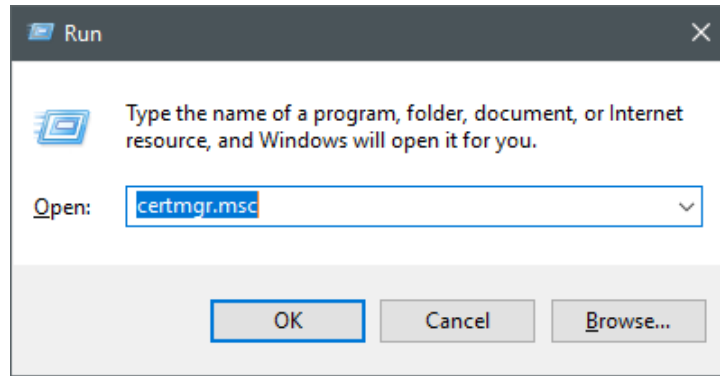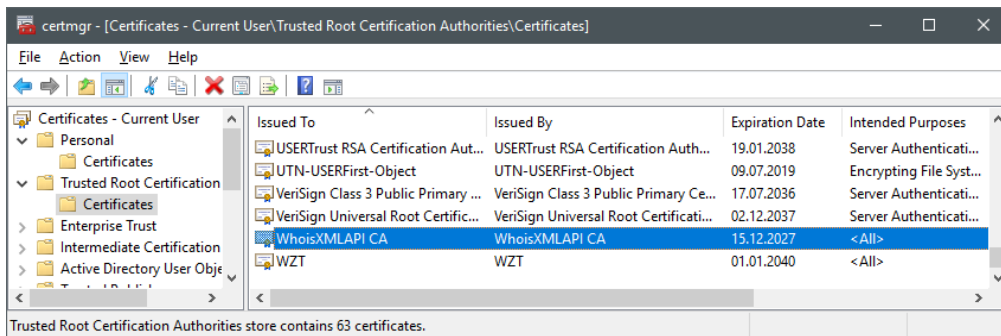
Certificate Store
Certificate stores are system areas where certificates are kept.

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

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

33

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:
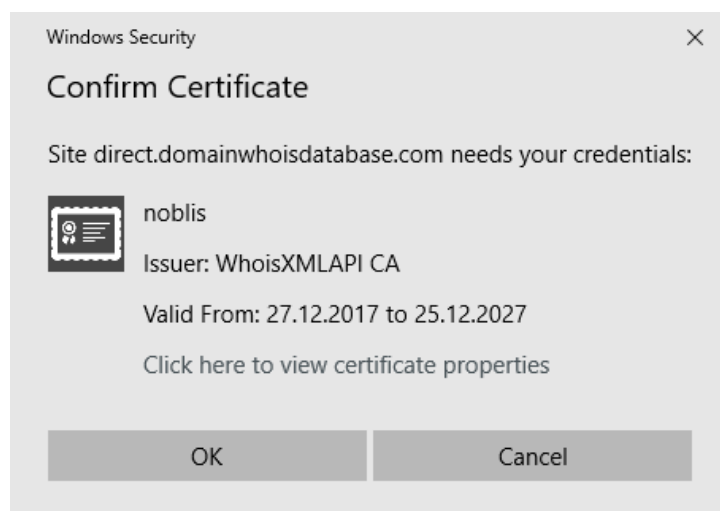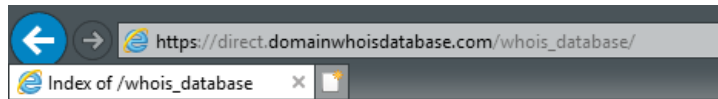
You should see the root certificate:

and your personal certificate
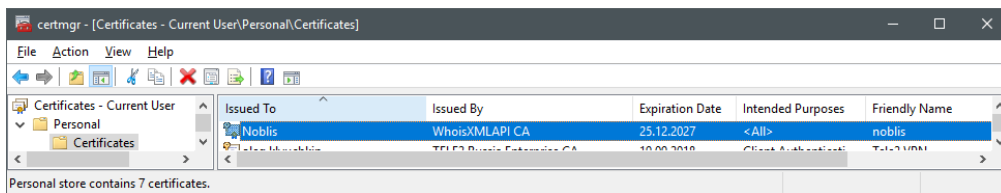
And as the main implication, after confirming the certificate you can open now the URLs you are eligible for, listed in Section 9.2, securely and without being prompted for passwords:

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	-	

9.1.2 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):
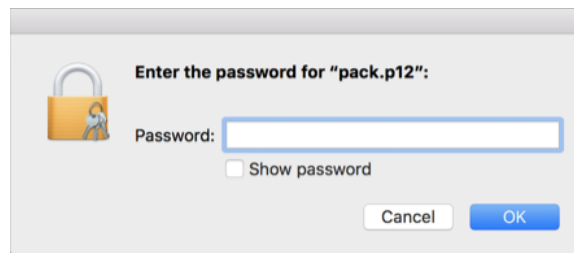

Note: you cannot paste the password into this latter dialog, so you need to type it. The Keychain Access tool window will open after the import:

36
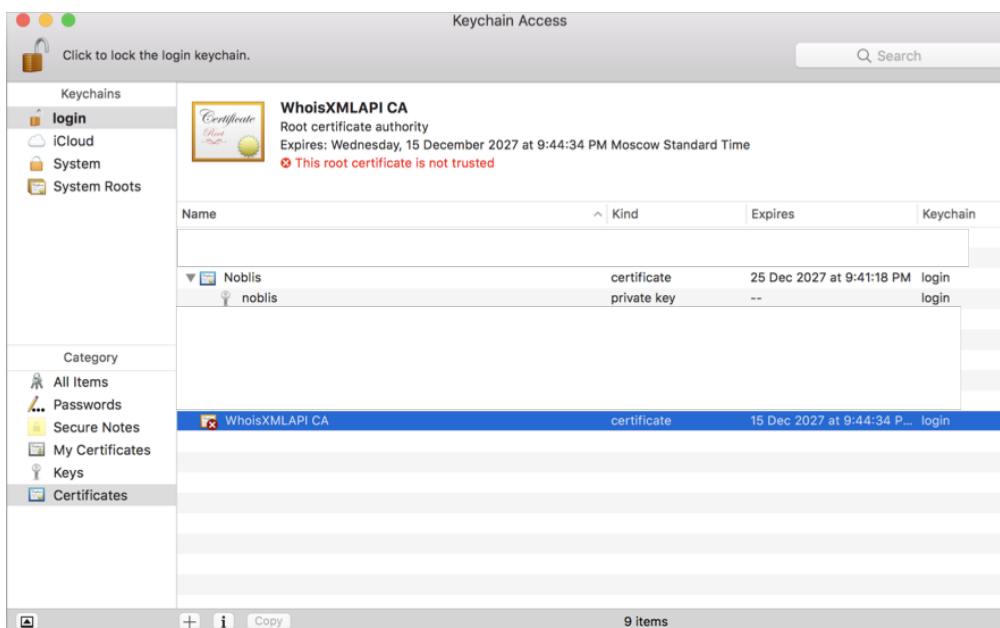
WhoisXMLAPI certificate is not trusted by default, so double click on WhoisXMLAPI ca certificate. Choose
"Always Trust" from the dropdown menu and close the window. The Administrator password is required to apply
this setting. Afterwards, our root certificate should appear as trusted:
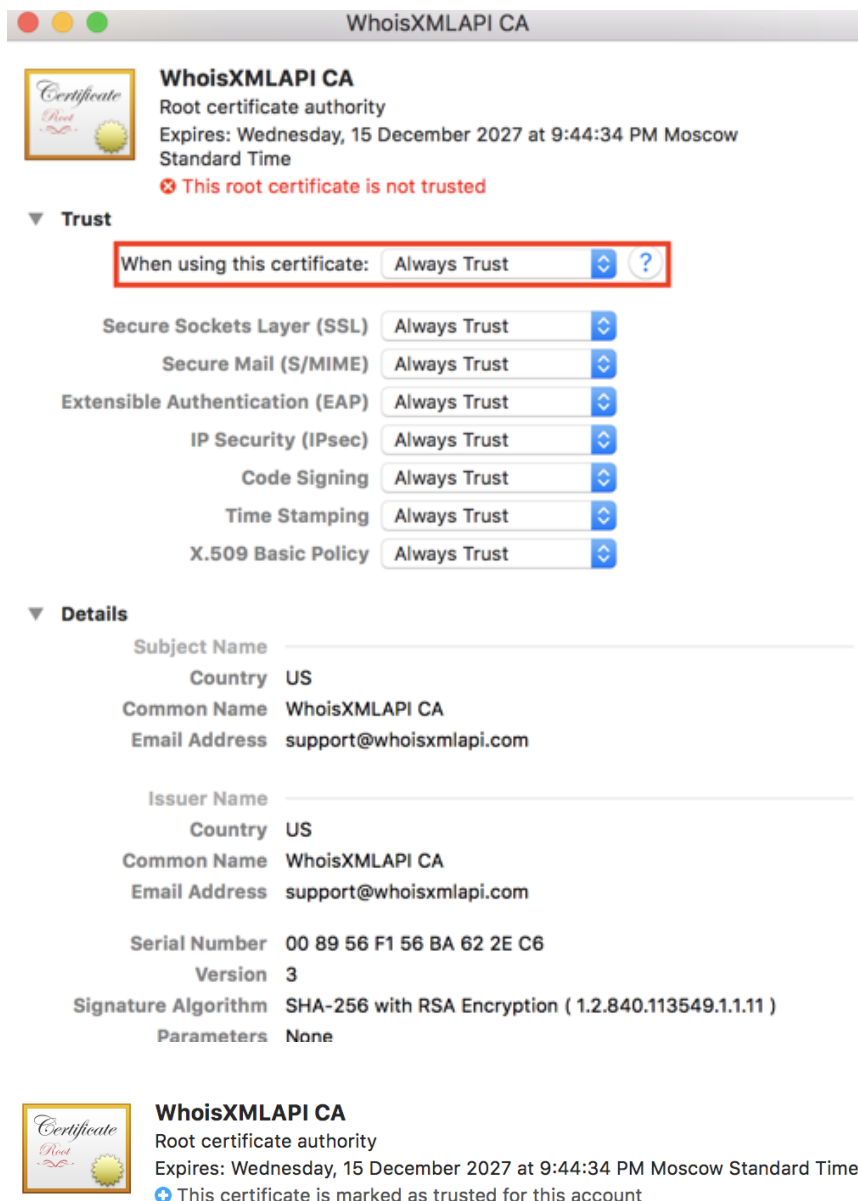
If you start the Safari web-browser and open any of the URLs listed in Section 9.2, it will ask for certificate to be
used for authentication: and the username-password pair to access the keychain:
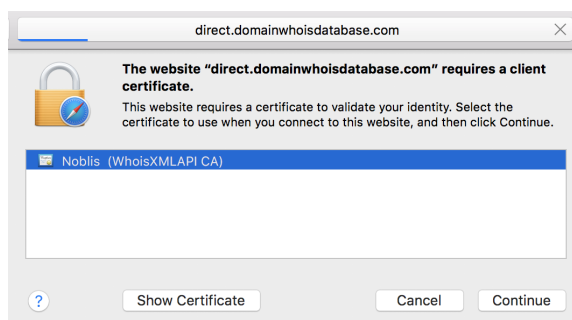
Then the requested page will open securely and without the basic http authentication.

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	-	

9.1.3   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:
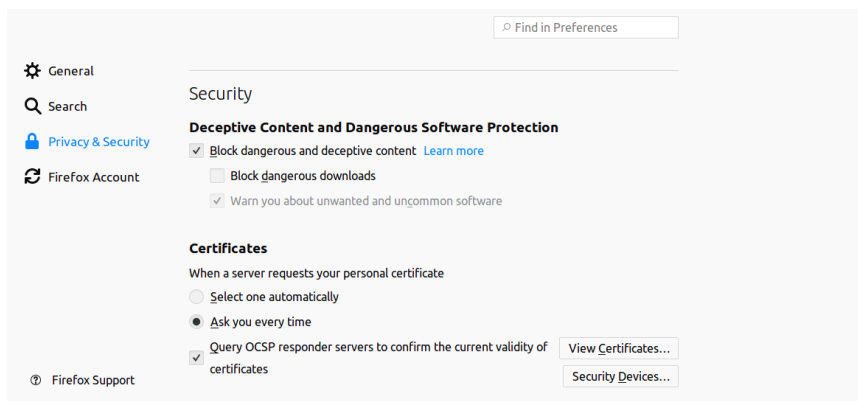
Press "View Certificates" and choose the "Your Certificates" tab. The certificate manager will appear:

38
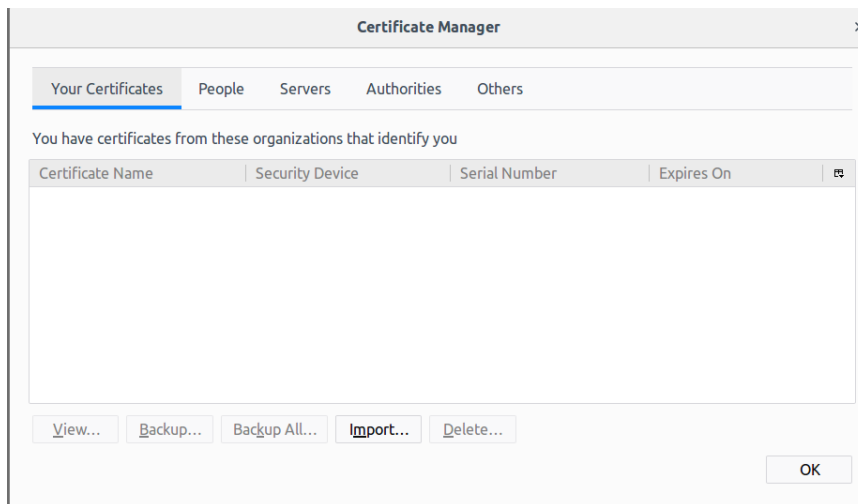
Press "Import", choose the file "package.p12", and enter the password you were given along with the certificate. You should see the certificate in the list. Now open any of the accessible URLs. You shall be warned as the browser considers the page as insecure:
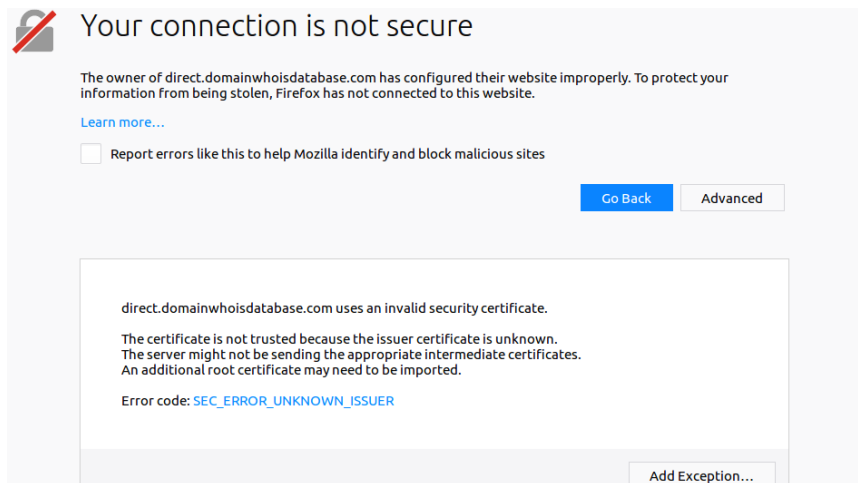
However, as you are using our trusted service, you can safely add an exception by pressing the button on the bottom. Add the exception permanently. Doing these steps you will be able to access the URLs mentioned in the last Section of the present document without the basic http authentication.

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.

9.2 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.domainwhois`" 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/`

10 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.

10.1 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.

10.2 FTP access

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

Host: 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.

10.3 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.

10.4 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.