CentralNic Group PLC

SSL documentation

Reference Manual

2022-10-04

Table of Contents

1 About SSL certificates

1.1 What is an SSL certificate?

In the context of this reseller documentation “SSL certificate” is a common term for digital certificate that enables an encrypted connection between a server and a client. SSL is most publicly visible for its use in providing encrypted HTTPS connections between a web server and a web browser.

Transport Layer Security (TLS) is a cryptographic protocol designed to provide communications security over a computer network. The protocol is widely used in applications such as email, instant messaging, and voice over IP and of course HTTPS.

Secure Sockets Layer (SSL) is a protocol originally specified and developed in the mid 1990s by Netscape, and over the years has been deprecated by its successor TLS. However, the term SSL is sticky and still most commonly used and well understood widely for certificates, even if the newer underlying protocol has a different name.

1.2 Why encryption?

The TLS protocol primarily aims to provide cryptography, including privacy (confidentiality), integrity, and authenticity through the use of certificates, between two or more communicating computer applications.

The primary use for SSL certificates is to store encrypted information sent over the Internet so that only the intended recipient can read it. Encrypting commercial websites is important to protect sensitive customer information, payment details and further data and ensure it can only be read by the intended recipient.

1.3 TLS/SSL Server Certificates

TLS requires the server to present the client with a digital certificate that proves it’s the intended destination. The connecting client performs a certification path validation and ensures that:

1. The subject of the certificate is the hostname (not to be confused with the domain name) to which the client is attempting to connect.
2. A trusted certificate authority signed the certificate.

The Subject field of the certificate must specify the primary hostname of the server as the common name. A certificate can be valid for multiple hostnames (e.g. a domain and its subdomains). Such certificates are commonly referred to as Subject Alternative Name (SAN) certificates or Unified Communications Certificates (UCC). These certificates include the Subject Alternative Name field, although many CAs also include them in the Subject Common Name field for backwards compatibility. If some of the hostnames contain an asterisk (*), a certificate can also be called a wildcard certificate.

Once the certification path validation is successful, the client can establish an encrypted connection with the server.

Servers connected to the Internet, such as B. public web servers, must get their certificates from a trusted, public certification authority (CA).

1.4 Other Certificates

In addition to server certificates, a number of other commercial certificate products are provided by CAs, in particular email or S/MIME, code or app signing and document signing certificates.

These products are currently not offered in our system.

1.5 Public Key Infrastructure

A public key infrastructure (PKI) is a set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

1.5.1 Public/Private Key

Before a client and server can begin to exchange information protected by TLS, they must securely exchange or agree upon an encryption key and a cipher to use when encrypting data.

Public-key cryptography, or asymmetric cryptography, is a cryptographic system that uses pairs of keys. Each pair consists of a public key (which may be known to others) and a private key (which may not be known by anyone except the owner). The generation of such key pairs depends on cryptographic algorithms which are based on mathematical problems termed one-way functions. Effective security requires keeping the private key private; the public key can be openly distributed without compromising security.

The primary role of the CA is to digitally sign and publish the public key bound to a given user. This is done using the CA's own private key, so that trust in the user key relies on one's trust in the validity of the CA's key.

1.5.2 CSR

In public key infrastructure (PKI) systems, a certificate signing request (also CSR or certification request) is a message sent from an applicant to a certificate authority of the public key infrastructure in order to apply for a digital identity certificate. It usually contains the public key for which the certificate should be issued, identifying information (such as a domain name) and integrity protection (e.g., a digital signature).

A CSR for an SSL certificate needs to be encoded on the respective server. A common tool for CSR generation is OpenSSL.

A CSR may be represented as a Base64 encoded PKCS#10; an example of which is given below:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIEpjCCAo4CAQAwYTELMAkGA1UEBhMCR0IxDzANBgNVBAgMBkxvbmRvbjEZMBcG
A1UECgwQQWNtZSBDb3Jwb3JhdGlvbjEQMA4GA1UECwwHUHJvZHVjdDEUMBIGA1UE
AwwLZXhhbXBsZS5jb20wggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDs
e8vo7aGk7I9PUT2Fw07V6J7dB/g8R+99gHG3aZ7aCM1I1r22uyAHw28GoRJh1sW3
11wz+SJrlinLSY2jSUIFvARJZImByuqMAFII2E3f2hRaTHDrdAaKSw7itRVejfpI
PukVPyIT+DMMRjExWJ3ZEpbpugcTDxv0NxnJKsI3nNjIoMKopPn4tK9hbKQZQnbd
gwlILALNb7PNo/OstDf9rIG4Ciafpsr4BUlksw1T0LCwG/yRR4OVJvM2994ujl51
F6ghO7OqoSJICC35x8otLVqQgk11EpWIgO157devN8IJuLG2ysAbB9zP6OX1Hh2Y
EWY6eptmaNuw0F0LSWMkTUEtkwsp0PatDsWfGiThmjVLUOirDhSUsXNHTydGNxAI
ZuEMICOD+15as5iVfrbu+VRzBsbiqirDtVVS0Njt2JZ+xW8X9oraYgioY6wah1X0
3ahT0cc3NBpAntGJRLsc1EwDRR59c9rCoqym48Z4kbpKy19O9h9MWIc6fHepko2a
IyOZWhC67v/y0SXbXWK4oH4VmSyHL720c4hlV4fODeYdUrp58HWpTXY4gVmlgRJg
kT5YEPdiZdtvkFIJeTWFHVtNfr7iO3X+KtBeR96tVY2jyfSbkbmYuyiIP2OFk5yt
Z1Od95NUpHRDQi/60STaeZ3zpKOvxCLwXiMLUE84/wIDAQABoAAwDQYJKoZIhvcN
AQELBQADggIBAK6gnkXKwN4dii4QvTrp1TK/fthjWFA4qSr2r5zumvIJ81lCB8Yc
qCc8Vkye/xlyymENrzS4VS0lT+XDXINkH1Zi74/JcQCUJODxyom7+D5qOtwvur1/
Ue0RA6kpiWD5ktYdMdf++7TJwy+WAwxSImbb0tSmSSTUbSFQhx+fVCH90tbc+pMU
m4omm7RmhWXndGemaZc6Z7Bj8XASbwYrPTTD0j3Pgfspvn5pbC6kDbUF4R5ydXbg
v0cwX7KvlsqMiLQIyWrMrPFWPXCWJleyd8tSSW+eP2QJtJ31+nRsV29G646JcvQT
1E74pu26B0CgfbuGDV+9gJhqJB9uXq0z8HRen2jG/5eAhm41kB7Xu+gHRq1uGs5x
a0J68ArwnwuWYVv24WtOJZbNhp4LBluVTKv3beqdXbv89wPW3XkuvzrnYP+/eVJ6
60pyzqKlbvpLHpRzZ6h2bTOI6bd+Jx5MJ9/9AezgK/B/s3gJ9PgFMMyYh0NaxrvH
JZDZgkJvk9J4i+FcQheneuLyQAn+RGQKLc+hDJ+6GfB1Tc3K7FpERAGnX6VR2Ml3
oZduMlQBXWwlN2YypA/tVv0XO9OeJHf04Nvo/LEW8kGkszirGSecXeHo01APmggR
Ox3AKZgMxMxXMkhWqW0KH1lbnLRnREm/LrEZZABwm35e8qlbCImmrF85
-----END CERTIFICATE REQUEST-----
```

1.6 Key Sizes and Signature Algorithms in CSRs

1.6.1 Root Length

The bit-length of the key pair determines the strength of the key and how easily it can be cracked using brute force methods. A minimum of a 2048-bit key size is the industry standard, 3072- and 4096-bit is widely supported.

1.7 Signature Algorithm

Hashing algorithm are used by issuing Certificate Authorities to actually sign certificates and CRLs (Certificate Revocation List) to generate unique hash values from files. It is highly recommended that your certificate be signed with SHA-2 as this is the strongest signature algorithm adopted by the industry.

1.8 Integration in servers / clients (formats)

In general the following files are needed to install a certificate on a server:

- **SSL certificate**: The actual certificate is a CRT file in PEM format (the usual extensions are .pem, .crt, .txt).
- **Private key**: A code (or file with the code, the usual extensions are .key, .pem, .txt) which is generated along with the CSR. If it was generated on the server, it will be located in the same folder that the generation command was run in, unless a different location was specified manually.
- **CA Bundle**: A file with intermediate and root certificates of the SSL chain of trust, provided along with the SSL certificate file.

Depending on the server additional tools can be used to install a certificate.

2 Role of certificate authorities

In cryptography, a certificate authority or certification authority (CA) is an entity that stores, signs, and issues digital certificates and provides a Public key infrastructure (PKI).

The CA/Browser Forum publishes the Baseline Requirements,a list of policies and technical requirements for CAs to follow.

One particularly common use for certificate authorities is to sign certificates used in HTTPS, the secure browsing protocol for the World Wide Web. Another common use is in issuing identity cards by national governments for use in electronically signing documents.

2.1 General validation

The commercial CAs that issue the bulk of certificates for HTTPS servers typically use a technique called “domain validation” to authenticate the recipient of the certificate. The techniques used for domain validation vary between CAs, but in general domain validation techniques are meant to prove that the certificate applicant controls a given domain name, not any information about the applicant’s identity.

Certificate Authorities also offer Organisation Validation (OV) and Extended Validation (EV) certificates as a more rigorous alternative to domain validated certificates. Organisation and Extended validation is intended to verify not only control of a domain name, but additional identity information to be included in the certificate.

2.2 Root CAs

2.3 Intermediate Certificate

The basic difference between root certificates and intermediate certificates is roots. A root CA has its own trusted roots in the trust stores of the major browsers. On the other hand, an intermediate certificate authority or sub certificate authority issues an intermediate root as they do not have roots in the trust stores of browsers. So, the roots of intermediate certificate authority point back to a trust-party root.

DigiCert and Sectigo operate Root CAs as well as subordinate CAs to provide their services. Their TLS/SSL certificates are provided including intermediate certificates

2.4 Chain of Trust

The SSL certificate chain order consists of root certificates, intermediate certificates, and the end-user certificate. Root CAs are a trusted source of certificates. Intermediate CAs are bridges that link the end-user certificate to the root CA. An SSL certificate chain order is the list of intermediate CAs leading back to a trusted root CA.

The certificate chain, also known as the certification path, is a list of certificates used to authenticate an entity. The chain, or path, begins with the certificate of that entity, and each certificate in the chain is signed by the entity identified by the next certificate in the chain. The chain terminates with a root CA certificate. The root CA certificate is always signed by the certificate authority (CA) itself. The signatures of all certificates in the chain must be verified until the root CA certificate is reached.

2.5 Transparency Logs

Certificate Transparency (CT) is an Internet security standard for monitoring and auditing the issuance of digital certificates. The standard creates a system of public logs that seek to eventually record all certificates issued by publicly trusted certificate authorities, allowing efficient identification of mistakenly or maliciously issued certificates.Version 2.0 of the Certificate Transparency mechanism, the latest, is described in the experimental RFC 9162, which obsoletes the earlier version 1.0 described in RFC 6962.

One of the problems with digital certificate management is that fraudulent certificates take a long time to be spotted, reported and revoked. An issued certificate not logged using Certificate Transparency may never be spotted at all. Certificate Transparency makes it possible for the domain owner (and anyone interested) to get in knowledge of any certificate issued for a domain.

CT strengthens the SSL/TLS certificate system by creating publicly auditable records of certificate issuance. Since 2015, Google has required CAs to log EV certificates to public CT logs. In April 2018, Google began requiring CAs to also log OV and DV certificates to public CT logs. As of October 15, 2018, Apple required CAs to log all SSL/TLS certificates (EV, OV, and DV).

A transparency required error means the website's SSL certificate was not properly logged when it was issued. Certificate Transparency is now a requirement for all trusted Certificate Authorities.

3 Certificate features

3.1 Classes

CAs offer certificates with different characteristics, requirements and configurations, resulting in different products offered. These features are e.g. SAN and/or wildcard support, verification types and possible validity periods. Details on this are in the following chapters.

3.2 Validation types

There are 3 different validation categories.

- **Domain Validated (DV)**: DV certificates only validate control over the domain(s). No further information from the user is used in the CRT and only the organization of the CA is listed in it.
- **Organization Validated (OV)**: These certificates briefly validate the organization specified in the request in addition to ownership of the domain(s) like the DV certificates do.
- **Extended Validated (EV)**: EV certificates support all features of OV certificates. In addition, a strict check is carried out as to the legitimacy of the company. EV certificates will not be accepted if they are not recorded in multiple transparency logs.

3.3 Multiple domains in certificates

A certificate consists of a (legacy) CommonName and so-called "Subject Alternative Names" (SAN). CommonName and SAN are the domains/FQDNs covered by each certificate and their number is limited by product/class. A domain in a certificate only covers a specific domain, not a variation of it. This also means that a certificate covering e.g. *example.com* does NOT cover *www.example.com* or other hosts within example.com (see wildcard domains). For most products, a defined SAN alternative can be added to the main domain free of charge. Thus it is possible to add a separate *www.example.com* SAN when *example.com* is the covered domain (and vice versa).

3.4 Wildcard domains

Some certificate products allow the addition of so-called wildcard domains. These are entries of the form **.example.com*, where * is a wildcard for a valid hostname. This means that all direct subdomains of *example.com* are covered, but *www.host.example.com*, for example, would not be covered. Since **.example.com* does not cover *example.com*, most products allow adding *example.com* for free.

3.5 Domain Control Validation (DCV) methods

Ownership of all the domains must be validated before they can be used in the certificates. There are methods of DCV to do so. Amongst them, Email validation is the most commonly used default method.

3.5.1 E-Mail

One way to prove ownership of a domain is to prove the control of significant email adresses under the domain. There are 5 standard email prefixes that are considered significant by the CA and no domain owner should ever give up control of any of those: admin@, administrator@, hostmaster@, webmaster@ & postmaster@. Any one of those can be used in the domain validation process using the email method. When validating the domain control of a sub domain, you can also choose an email adress under the main domain instead (eg. admin@example.com for api.example.com) This is relevant when validating control over a domain that was added to the certificate for free (as RRPproxy does by default) based on the free alternative domains.

In addition to those default emails, most CAs allow the use of email adresses listed in the whois of the related domain. This is becoming more unreliable due to increasing whois privacy and the use of the default adresses is recommended.

3.5.2 DNS

Another way to prove the ownership of a domain is to demonstrate control over the nameservers. For this, one receives a token in the ordering process that needs to be put in the nameserver for the domain either as CNAME or as a TXT record. The CA will periodically check for the existance of that record and will acknowledge your domain control once it finds the correct configuration.

3.5.3 HTTP(S)

Control over the webserver also demonstrates the ownership of a domain. Placing a file with a specific content on a pre-defined path on the server is the usual procedure. Both content and path are provided during the ordering process. This process is not available for wildcard domains.

3.6 Immediate issuance

All DCV is usually done post-ordering making the whole process non-real time and adding a bit of unpredictability to the timing. The immediate issuing process allows the preparation of DCV before placing the order of a certificate. For that the customer can retrieve the required information (eg. token) in advance and prepare the DNS/HTTP(s) validation. The certificate order then happens in real time as the CA can process the validation in real time. This option is often not available for email validation.

3.7 Periods

- Current limitations (teaser order)
- Prepare for upcoming multi-year and/or shorter periods

4 What can you do with certificates

4.1 Reissues

Due to the immutable character of certificates, any change results in the issuanced of a new certificate. The original certificate stays valid even if an updated certificate was issued. One common reason for a reissue could be a required change of the public/private key pair resulting in a changed CSR.

4.2 Revokes

There are several reasons why a certain certificate should be considered invalid by all parties. The most common reason would be a compromise of the private key. A revocation can be initiated either by the CA or by the user of the certificate. After a revocation, the CA adds the certificate to a Certificate Revocation List (CRL) and also responds negatively to queries about the validity of a certificate via its Online Certificate Status Protocol (OCSP) server. The option for a free reissue is usually still available (although a new CSR / key pair is recommended and sometimes even required).

5 Workflows

This chapter describes the most commonly used actions with certificates.

5.1 Creation

Probably the most important step in the life cycle of a certificate is the ordering process. There are two ways to get a new certificate. !Diagramm of the ordering process

5.1.1 Preparation

Creating a new key and CSR pair Normally, for security reasons, a user should always create a new private key and not reuse existing private keys. A CSR is generated from this private key.

An example of creating a new private key/CSR pair using openssl:

```
openssl req -new -newkey rsa:4096 -nodes -keyout your_domain.key -out your_domain.csr
```

The CSR will be used in the next steps during the ordering process.

Notes - The keyfile keeps your private key and should not be shown to anyone. Once it is compromised, anyone can impersonate the service secured by it. Once this happens, a certificate must be revoked (see Revokes). - The key size should be equal to or greater than 2048. - The command name should be filled with the domainname to be used (No wildcard allowed). - The phone number for the owner contact has to be in the following format: +[country code] [local area code] [line number] (separated by blank spaces) - If you want to use mutated vowels (Umlaute) in your CSR, you have to add the parameter -utf8 to the command or update your openssl.cnf with utf8=Yes. - Integration in servers / clients (formats).

5.1.2 Normal process

Placing the order After the CSR has been created, it must be submitted to the CA along with additional information about the type of certificate required. Depending on the type, the CA optionally requires additional contact information (see Add-Certificate).

Validation of the request Depending on the chosen domain validation method, different measures must be taken:

- **Email validation**: Upon receiving the request, the CA will send a link to each email address required to validate domain ownership. These links must be approved before a domain can be used in the certificate. Please see the email chapter for more details on valid email addresses that can be used.

- **DNS validation**: A request with DNS validation requires an entry in the DNS server of the corresponding domain. Typically, this entry is returned in the response to the purchase requisition. The TTL of this entry should be as short as possible. An example entry could look like this:

```
@ 3600 IN TXT bk5yp3dh9xrb94byhhny76dhxt0wdkn4
```

- **HTTP(S) validation**: This validation method requires a file to be available over HTTP(S) on the domain's web server. The exact location and content of this file is usually returned by the certificate authority in the checkout response. This information may not be changed. Wildcard domains may not be validated over HTTP(S) due to CA/B restrictions.

In addition to domain validation, OV and EV certificates require validation & verification of organizational contact information. This is done by the CA by e.g. checks if the company exists in a business register. The user can request a certificate for the company via telephone validation and more.

Fetch certificate & chain Once all the required validation is successful, the CA will create the certificate based on the CSR, signs it and provides this new certificate with an intermediate chain. The user is typically informed once the certificate is available (see Events) and can retrieve it (see StatusCertificate).

Install certificate The retrieved certificate needs to be installed and configured together with the key and intermediate chain in the server software (i.e. apache). You should consult the manual of the server software used.

5.1.3 Immediate issuance

Immediate issuance is a faster way to get a signed certificate for domain validated requests. This is not available for certificate types requiring email validation.

Creating a validation token Based on the CSR a dedicated validation token needs to be generated. This often includes a secure token provided by the CA in advance and requires to be integrated in the validation token. In some cases the end customer does not have access to this secure token and the provider will generate the validation token for the customer based on a given CSR (see CheckCertificate).

Installing validation token Before the order request is submitted, the token from the previous step needs to be configured in the DNS server or put in place on the web server (see Validation of the request).

Placing the order See Placing the order. A key difference between a normal order and an immediate issuance order is the necessity to provide the token in the request and getting the issued certificate and chain directly via the response of the order. No further information about the finished certificate is created.

Install certificate See Install certificate

5.2 Reissues

Reissues are only possible for already issued certificates. Most of the steps during a reissue are the same as for ordering a new certificate. Reissued certificates can contain the same information as the original certificate with minimal changes. Only the changed information needs to be submitted with the new request. Submitting at least a new CSR for a new private key is considered best practice.

5.3 Renewals

Since renewals do not exists on the CA's side, this process is mostly just a mapping to ordering a new certificate, with the exception of linking the old to the new certificate order.

6 Certificates in RRPproxy

One of the core principles of the API design for SSL certificates in RRPproxy is the ease of use for everyday users whilst allowing complex scenarios for power users. Basic processes do not require implementing most extra features.

It is possible to just work with Add/Status/Renew/Reissue/DeleteCertificate.

Extra handling of CertificateContacts is only required if contact verification needs to be reused.

The handling of CertificateOrders is only necessary if more details about the connection between related certificates are needed. All information necessary to handle the technical differences between certificate classes or CAs can be automatically queried via API commands (QueryCommandSyntax / GetCertificateInfo).

For further differences a product matrix is provided.

To facilitate the mixing of i.e. wildcard and non wildcard domains in a certificate, the calculation for prices of certificates is split up into multiple price types. These price types are combined on-the-fly based on the given request and can be checked in advance.

6.1 Configuration & Checks

As mentioned earlier in this documentation, every type of certificate was assigned a dedicated "CertificateClass" in RRPproxy. With the help of these classes all information about prices, configurations and similar handling can be retrieved.

- Details about classes and how they affect the possibilities during order (GetCertificateInfo)
- Why using commands like CheckCertificate to get tokens, emails etc.
- Using QueryCommandSyntax with the Certificate system

6.2 Certificates

Certificates are the main objects in the SSL system, which every other object will be linked to. CertificateOrders and CertificateContacts are possible linked objects, which will be explained in their own chapters.

These objects in RRPproxy are containers holding metadata about the certificate, the domains listed in the requests as well as the CSR, CRT and the intermediate chain with root.

Certificates are identified via a unique, auto-generated ID in RRPproxy. The ID consists of a CA identifier, class identifier as well as an 8 digit long number. During the order of a certificate, a class must be chosen. With this class the basic options / configuration are defined. Examples of options are wildcard support, number of supported SAN as well as verification & authentication methods.

Most classes allow the addition of alternative domains for free. Alternative domains are i.e. the main domain for a wildcard domain (example.com for *.example.com) or the special case "www" (www.example.com for example.com and vice versa). RPPproxy will add these free of charge alternatives automatically for as many domains as supported by the class. The command AddCertificate has an option (CHECKONLY = 1) to preview the generated domain list (as well as the cost). The generation of alternative domains can be skipped by supplying the parameter NOAUTOFILLDOMAINS = 1 with the command.

The certificate object allows for reissues, renewals or revokes of the associated CRT as well as the cancellation of the request (if supported by the CA).

6.3 CertificateOrders

CertificateOrders are containers for multiple Certificate objects. New entries will be added to a CertificateOrder upon reissues of a Certificate. The CertificateOrder holds the paid until date requested with the first Certificate. A refund can only happen if the complete CertificateOrder was cancelled (which can only happen as long as no certificate was issued for this CertificateOrder). This object can be ignored by most users, it is primarily an administrative container.

6.4 CertificateContacts

Ordering a certificate requires contact information. This contact information is stored in CertificateContact objects. Once a CertificateContact was created, it can no longer be modified. This is essential to minimize problems with renews or reissues as well as allow re-usage of already verified organization data.

CertificateContacts can be created in multiple ways. The easiest is to submit the necessary contact data while requesting a certificate. RRPproxy will create a new CertificateContact internally and return the new ID(s) in the response. Using P- or O- handles (P-ABCxxx) during a request will result in the creation of new CertificateContacts as well. Creating CertificateContacts in advance to ordering certificates is not necessary with these two options. The third option would be creating CertificateContacts explicitly via AddCertificateContact.

Once the validation expires and the CertificateContact is not linked to any active certificate, it will be deleted automatically after a few weeks. This automatic cleanup takes care of any unused objects, and thus a dedicated deletion is not required.

RRPproxy handles 3 basic types of CertificateContacts:

- **Tech**: A technical contact contains contact data the CA uses to communicate with in case there are problems during the order processing.
- **Organization**: The organization contact contains contact information about someone working for the organization included in the certificate order. The CA contacts them to validate the organization and verify the request for OV/EV TLS/SSL certificates.
- **EVapprover**: The EVapprover contact is someone who works for the organization included in the certificate order. The CA will contact the organization directly to verify this contact and confirm the individual's name, email, phone number, and job title.

The product matrix contains information which CertificateContact type is possible / required for the various certificate classes.

Once a CertificateContact is created, the only change possible is the verification status of the organization or the EVapprover contacts managed at the respective CA. Any other information is immutable. A CertificateContact can be used with multiple certificates.

6.5 Events

The SSL system uses the established event / notification system of RRPproxy and adds a few event-classes and event-subclasses. It informs about any updates to SSL objects. This usually happens during non-realtime processes like the ordering of a certificate.

Events can be queried via QueryEventList / StatusEvent. More details can be found in the standard RRPproxy documentation.

Possible events are listed in the appendix and all events contain at least the ID of the related object.

6.6 Billing

Certificates use dynamic pricing in RRPproxy, calculating the final price based on various components. Each CertificateClass has a set of possible price-components and price-types. There are two types of base prices: One (-wildcardbase) is used for certificates containing wildcard domains and one for every other certificate. A certificate uses exactly one of these base prices. Two different types of prices will be added to this base: -san and -wildcardsan. This additional price is multiplied by the number of (wildcard) SANs for every paid additional domain. Since the exact calculation of this varies depending on the CertificateClass and possible combinations of domains, RRPproxy offers a feature to pre-calculate the final price (AddCertificate with parameter checkonly=1, see example). Each of these price components is shown separately in the accountings.

If the CA automatically offers a refund when a certificate is canceled / revoked, this will also be forwarded by RRPproxy.

6.7 Web Interface

The web interface integration is in the roadmap. Stay tuned for our newsletter update!

7 Differences between providers

Most of the differences between the currently supported providers are worked around within RRPproxy, however, some procedural variations need to be handled outside of the API. The details of these differences are listed below.

7.1 Contacts

The verification of a CertificateContact can be reused for multiple orders at DigiCert. Since Sectigo does not support contact objects, each CertificateContact has to be re-verified for Sectigo OV/EV certificates and the verification cannot be explicitly reused.

7.2 Multi domain

Both DigiCert and Sectigo support SAN for some of their products as well as adding “free alternatives” in certificates. However, Sectigo only offers the “free alternative” for single domain certificates. Each Sectigo certificate with multiple domain support includes 3 domains in the base price, these domains do not have to be alternatives to each other (AddCertificate with checkonly=1 returns the billed number of domains for verification).

7.3 Validation methods

The CAs support domain - validation via multiple methods, however the details how they are handled differs slightly.

- **DNS**: Sectigo only supports records via CNAME while DigiCert supports TXT records as well. Both completed Bind resource records will be returned by the RRPproxy API.
- **HTTP(S)**: The filename for DigiCert products will always be [http://your-domain]/.well-known/pki-validation/fileauth.txt. Sectigo uses a random filename within the .well-known/pki-validation/ directory. These filenames will always be returned from the RRPproxy API.
- **Email**: No difference in email handling

7.4 Immediate issuance

Only some products of DigiCert support immediate issuance, none of Sectigo do. The product matrix lists the possible DigiCert products with immediate issuance.

7.5 Refunds

Sectigo does not offer automatic refunds, therefore RRPproxy cannot support refunds for Sectigo certificates. DigiCert refunds are automatically processed where applicable.

8 Command reference

8.1 Certificates

8.1.1 CheckCertificate

The command checks if the certificate request is valid and which information is inside.

Command

```
[COMMAND]
command = CheckCertificate
csr0..256 = <TEXT>|<NULL>
crt0..256 = <TEXT>|<NULL>
domain0..250 = <TEXT>|<NULL>
class = geotrustflexdv|rapidssldv|securesiteflexov|securesiteflexev|securesiteproov|
    securesiteproev|ssl123dv|webserverov|webserverev|geotrusttruebizidov|geotrusttruebizidev|
    instantsslov|premiumov|singledomainev|positivessldv|unifiedcommunicationsdv|
    unifiedcommunicationsov|multidomainev|multidomainov|multidomaindv|<NULL>
generatedcvtoken = 1|0|<NULL>
apiversion = 1|2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[csr public key algorithm][n] = <TEXT>
property[csr signature algorithm][n] = <TEXT>
property[csr size][n] = <INT>
property[csr subject][n] = <TEXT>
property[csr san#][n] = <TEXT>
property[csr key usage#][n] = <TEXT>
property[csr emailaddress][n] = <TEXT>
property[csr location][n] = <TEXT>
property[csr commonname][n] = <TEXT>
property[csr state][n] = <TEXT>
property[csr country][n] = <TEXT>
property[csr organization][n] = <TEXT>
property[csr organizational unit][n] = <TEXT>
property[crt public key algorithm][n] = <TEXT>
property[crt signature algorithm][n] = <TEXT>
property[crt validity not before][n] = <DATE>
property[crt validity not after][n] = <DATE>
property[crt size][n] = <INT>
property[crt version][n] = <TEXT>
property[crt serial][n] = <TEXT>
property[crt subject][n] = <TEXT>
property[crt issuer][n] = <TEXT>
```

```
property[crt key usage#][n] = <TEXT>
property[crt extended key usage#][n] = <TEXT>
property[crt san#][n] = <TEXT>
property[crt subject emailaddress][n] = <TEXT>
property[crt subject location][n] = <TEXT>
property[crt subject commonname][n] = <TEXT>
property[crt subject state][n] = <TEXT>
property[crt subject country][n] = <TEXT>
property[crt subject organization][n] = <TEXT>
property[crt subject organizational unit][n] = <TEXT>
property[crt issuer emailaddress][n] = <TEXT>
property[crt issuer location][n] = <TEXT>
property[crt issuer commonname][n] = <TEXT>
property[crt issuer state][n] = <TEXT>
property[crt issuer country][n] = <TEXT>
property[crt issuer organization][n] = <TEXT>
property[crt issuer organizational unit][n] = <TEXT>
EOF
```

Command Parameters

- **csr0..256**

  Type: <TEXT> | <NULL>

  A base64 encoded certificate request

- **crt0..256**

  Type: <TEXT> | <NULL>

  A base64 encoded X509 certificate

- **domain0..250**

  Type: <TEXT> | <NULL>

  Optional domain to check approver email addresses

- **class**

  Type: geotrustflexdv | rapidssldv | securesiteflexov | securesiteflexev | securesiteproov | securesiteproev | ssl123dv | webserverov | webserverev | geotrusttruebizidov | geotrusttruebizidev | instantsslov | premiumov | singledomainev | positivessldv | unifiedcommunicationsdv | unifiedcommunicationsov | multidomainev | multidomainov | multidomaindv | <NULL>

  Optional class to check approver email addresses

- **generatedcvtoken**

  Type: 1 | 0 | <NULL>

  Generate a dcv token. This requires a CSR and CLASS

- **apiversion**

  Select the used API version

Option	Description
1	Pre - 2022 API
2	2022 API

Response Parameters

- **csr public key algorithm**

  Type: <TEXT>

  Public key algorithm used for the CSR

- **csr signature algorithm**

  Type: <TEXT>

  Signature algorithm used for the CSR

- **csr size**

  Type: <INT>

  Size of the CSR

- **csr subject**

  Type: <TEXT>

  Subject encoded in the CSR

- **csr san#**

  Type: <TEXT>

  Domains encoded in the CSR

- **csr key usage#**

  Type: <TEXT>

  Allowed usage of the CSR

- **csr emailaddress**

  Type: <TEXT>

  Email given in the CSR

- **csr location**

  Type: <TEXT>

  City given in the CSR

- **csr commonname**

  Type: <TEXT>

  Common name given in the CSR

- **csr state**

  Type: <TEXT>

  State given in the CSR

- **csr country**

  Type: <TEXT>

  Country given in the CSR

- **csr organization**

  Type: <TEXT>

  Organization given in the CSR

- **csr organizational unit**

  Type: <TEXT>

  Unit of the organization given in the CSR

- **crt public key algorithm**

  Type: <TEXT>

  Public key algorithm used for the CRT

- **crt signature algorithm**

  Type: <TEXT>

  Signature algorithm used for the CRT

- **crt validity not before**

  Type: <DATE>

  CRT only valid after this date

- **crt validity not after**

  Type: <DATE>

  CRT only valid before this date

- **crt size**

  Type: <INT>

  Size of the CRT

- **crt version**

  Type: <TEXT>

  Version encoded in the CRT

- **crt serial**

  Type: <TEXT>

  Serial of the CRT

- **crt subject**

  Type: <TEXT>

  Subject encoded in the CRT

- **crt issuer**

  Type: <TEXT>

  Issuer of the CRT

- **crt key usage#**

  Type: <TEXT>

  Allowed usage for the CRT

- **crt extended key usage#**

  Type: <TEXT>

  Details of the allowed usage for the CRT

- **crt san#**

  Type: <TEXT>

  Domains encoded in the CRT

- **crt subject emailaddress**

  Type: <TEXT>

  Email given in the CRT

- **crt subject location**

  Type: <TEXT>

  City given in the CRT

- **crt subject commonname**

  Type: <TEXT>

  Common name given in the CRT

- **crt subject state**

  Type: <TEXT>

  State given in the CRT

- **crt subject country**

  Type: <TEXT>

  Country given in the CRT

- **crt subject organization**

  Type: <TEXT>

  Organization given in the CRT

- **crt subject organizational unit**

  Type: <TEXT>

  Unit of the organization given in the CRT

- **crt issuer emailaddress**

  Type: <TEXT>

  Email given in the CRT

- **crt issuer location**

  Type: <TEXT>

  City given in the CRT

- **crt issuer commonname**

  Type: <TEXT>

  Common name given in the CRT

- **crt issuer state**

  Type: <TEXT>

  State given in the CRT

- **crt issuer country**

  Type: <TEXT>

  Country given in the CRT

- **crt issuer organization**

  Type: <TEXT>

  Organization given in the CRT

- **crt issuer organizational unit**

  Type: <TEXT>

  Unit of the organization given in the CRT

8.1.2 AddCertificate

Failed requests will return an ID in the format "FAILxxxxxxxx" which can't be used for any command. This ID can be used by the RRPproxy support if assistance is required. Request a new SSL certificate

Command

```
[COMMAND]
command = AddCertificate
apiversion = 1|2|<NULL>
organizationcontact = <CONTACT>|<NULL>
evapprovercontact0..1 = <CONTACT>|<NULL>
techcontact0..1 = <CONTACT>|<NULL>
csr0..256 = <TEXT>|<NULL>
class = geotrustflexdv|rapidssldv|securesiteflexov|securesiteflexev|securesiteproov|
    securesiteproev|ssl123dv|webserverov|webserverev|geotrusttruebizidov|geotrusttruebizidev|
    instantsslov|premiumov|singledomainev|positivessldv|unifiedcommunicationsdv|
    unifiedcommunicationsov|multidomainev|multidomainov|multidomaindv
dcvmethod = EMAIL|DNS-TXT|DNS-CNAME|HTTP|HTTPS|<NULL>
dcvtoken = <TEXT>|<NULL>
domain0..250 = <TEXT>|<NULL>
dcvemail0..250 = <EMAIL>|<NULL>
servertype = apache|barracuda|weblogic|cisco|citrix|cpanel|f5|ibm|java|lighttpd|lotus|macos|
    exchange.*2007|exchange.*2010|exchange.*2013|exchange.*2016|forefront|iis56|iis7|iis8|
    iis10|netscape|iplanet|nginx|novellichain|novellnetware|oracle|qmail|sunone|tomcat|
    webstar|zeus|other|<NULL>
period = 1|1y
techfirstname0..1 = <TEXT>|<NULL>
techlastname0..1 = <TEXT>|<NULL>
```

```
techphone0..1 = <PHONE>|<NULL>
techemail0..1 = <EMAIL>|<NULL>
techjobtitle = <TEXT>|<NULL>
evapproverfirstname = <TEXT>|<NULL>
evapproverlastname = <TEXT>|<NULL>
evapproverphone = <PHONE>|<NULL>
evapproveremail = <EMAIL>|<NULL>
evapproverjobtitle = <TEXT>|<NULL>
organizationname = <TEXT>|<NULL>
organizationdba = <TEXT>|<NULL>
organizationstreet = <TEXT>|<NULL>
organizationzip = <TEXT>|<NULL>
organizationcity = <TEXT>|<NULL>
organizationstate = <TEXT>|<NULL>
organizationcountry = <COUNTRY>|<NULL>
organizationphone = <PHONE>|<NULL>
organizationemail = <EMAIL>|<NULL>
checkonly = 1|0|<NULL>
noautofilldomains = 1|0|<NULL>
servicetag0..100 = <TEXT>|<NULL>
addservicetag0..100 = <TEXT>|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[certificate][n] = <TEXT>
property[sub][n] = <TEXT>
property[status][n] = <TEXT>
property[sub status][n] = <TEXT>
property[fileauth name][n] = <TEXT>
property[fileauth contents][n] = <TEXT>
property[dnsauth name][n] = <TEXT>
property[billingclass][n] = <TEXT>
property[order][n] = <TEXT>
property[domain#][n] = <TEXT>
property[dcvtoken#][n] = <TEXT>
property[dcvdomain#][n] = <TEXT>
EOF
```

Command Parameters

- **apiversion**

Option	Description
1	Legacy System
2	2022 system

- **organizationcontact**

  Type: <CONTACT>|<NULL>

  Organization - CertificateContact

- **evapprovercontact0..1**

  Type: <CONTACT>|<NULL>

  EV approver - CertificateContact

- **techcontact0..1**

Type: <CONTACT> | <NULL>

Tech - CertificateContact

- **csr0..256**

  Type: <TEXT> | <NULL>

  The CSR to be used to generate the certificate

- **class**

  Type of certificate to be added

Option	Description
geotrustflexdv	GeoTrust Flex DV
rapidssldv	RapidSSL
securesiteflexov	DigiCert Secure Site Flex OV
securesiteflexev	DigiCert Secure Site Flex EV
securesiteproov	DigiCert Secure Site Pro OV
securesiteproev	DigiCert Secure Site Pro EV
ssl123dv	SSL123
webserverov	webserver OV
webserverev	webserver EV
geotrusttruebizidov	GeoTrust True Business ID OV
geotrusttruebizidev	GeoTrust True Business ID EV
instantsslov	PremiumSSL
premiumov	SectigoSSL Premium OV
singledomainev	EV SSL
positivessldv	PositiveSSL
unifiedcommunicationsdv	DV SSL Unified Communications Certificate
unifiedcommunicationsov	OV SSL Unified Communications Certificate
multidomainev	SectiGo Multi-Domain SSL EV
multidomainov	SectiGo Multi-Domain SSL OV
multidomaindv	SectiGo Multi-Domain SSL DV

- **dcvmethod**

  Method for authentication of the domain name(s)

Option	Description
EMAIL	EMAIL
DNS-TXT	DNS-TXT
DNS-CNAME	DNS-CNAME
HTTP	HTTP
HTTPS	HTTPS

- **dcvtoken**

  Type: <TEXT> | <NULL>

Token generated by CheckCertificate

- **domain0..250**

  Type: <TEXT> | <NULL>

  Domains for certificate, first one is the CommonName

- **dcvemail0..250**

  Type: <EMAIL> | <NULL>

  Email to validate the request if dcvmethod = email for domain#

- **servertype**

  Servertype

Option	Description
apache	Apache
barracuda	Barracuda
weblogic	BEA Weblogic
cisco	Cisco
citrix	Citrix
cpanel	cPanel
f5	F5
ibm	IBM HTTP Server
java	Java Web Server (Javasoft / Sun)
lighttpd	Lighttpd
lotus	Lotus Domino
macos	Mac OS X Server
exchange.*2007	Microsoft Exchange Server 2007
exchange.*2010	Microsoft Exchange Server 2010
exchange.*2013	Microsoft Exchange Server 2013
exchange.*2016	Microsoft Exchange Server 2016
forefront	Microsoft Forefront Unified Access Gateway
iis56	Microsoft IIS 5 or 6
iis7	Microsoft IIS 7
iis8	Microsoft IIS 8
iis10	Microsoft IIS 10
netscape	Netscape Enterprise Server
iplanet	Netscape iPlanet
nginx	nginx
novellichain	Novell iChain
novellnetware	Novell NetWare
oracle	Oracle
qmail	Qmail
sunone	SunOne
tomcat	Tomcat
webstar	WebStar
zeus	Zeus Web Server
other	Other

- **period**

  Type: `1` | `1y`

  How many years the certificate should be valid

- **techfirstname0..1**

  Type: <TEXT> | <NULL>

  The given name of the new contact (may be empty for an organization handle)

- **techlastname0..1**

  Type: <TEXT> | <NULL>

  The family name of the new contact (may be empty for an organization handle)

- **techphone0..1**

  Type: <PHONE> | <NULL>

  The telephone number for this new contact handle

- **techemail0..1**

  Type: <EMAIL> | <NULL>

  The email address for this new contact handle

- **techjobtitle**

  Type: <TEXT> | <NULL>

  The job title for this new contact handle

- **evapproverfirstname**

  Type: <TEXT> | <NULL>

  The given name of the new contact (may be empty for an organization handle)

- **evapproverlastname**

  Type: <TEXT> | <NULL>

  The family name of the new contact (may be empty for an organization handle)

- **evapproverphone**

  Type: <PHONE> | <NULL>

  The telephone number for this new contact handle

- **evapproveremail**

  Type: <EMAIL> | <NULL>

  The email address for this new contact handle

- **evapproverjobtitle**

  Type: <TEXT> | <NULL>

  The job title for this new contact handle

- **organizationname**

  Type: <TEXT> | <NULL>

  The organization the new contact will be created for

- **organizationdba**

  Type: <TEXT> | <NULL>

  DBA of the organization (if necessary)

- **organizationstreet**

  Type: <TEXT> | <NULL>

  The street for this new contact handle

- **organizationzip**

  Type: <TEXT> | <NULL>

  The postal code for this new contact handle

- **organizationcity**

  Type: <TEXT> | <NULL>

  The city for this new contact handle

- **organizationstate**

  Type: <TEXT> | <NULL>

  The federal state for this new contact handle

- **organizationcountry**

  Type: <COUNTRY> | <NULL>

  The country for this new contact handle (please use the country code, e.g. “DE” or “US”)

- **organizationphone**

  Type: <PHONE> | <NULL>

  The telephone number for this new contact handle

- **organizationemail**

  Type: <EMAIL> | <NULL>

  The email address for this new contact handle

- **checkonly**

  Type: 1 | 0 | <NULL>

  Only verify that the request parameters are valid for the specified class

- **noautofilldomains**

  Type: 1 | 0 | <NULL>

  Prevent auto-population of the domain list with free alternatives

- **servicetag0..100**

  Type: <TEXT> | <NULL>

  The service tags that will be associated with this service.

- **addservicetag0..100**

  Type: <TEXT> | <NULL>

  The service tags that will be associated with this service.

Response Parameters

- **certificate**

  Type: <TEXT>

  ID of the new certificate

- **sub**

  Type: <TEXT>

  ID of the new certificate sub

- **status**

  Type: <TEXT>

  Status of the request process for the certificate

- **sub status**

  Type: <TEXT>

  Status of the request process for the sub

- **fileauth name**

  Type: <TEXT>

  Name for the file needed for file based authentication

- **fileauth contents**

  Type: <TEXT>

  Content for the file needed for file based authentication

- **dnsauth name**

  Type: <TEXT>

  Name of the RR entry needed for DNS based authentication

- **billingclass**

  Type: <TEXT>

  Class used to bill this certificate in case of a SAN certificate

- **order**

  Type: <TEXT>

  ID of the new certificate order

- **domain#**

  Type: <TEXT>

  Domains associated with the certificate

- **dcvtoken#**

  Type: <TEXT>

  DCV token for the domain (if requested)

- **dcvdomain#**

  Type: <TEXT>

  DCV domain (if requested)

8.1.3 RenewCertificate

Failed requests will return an ID in the format "FAILxxxxxxxx" which can't be used for any command. This ID can be used by the RRPproxy support if assistance is required. Renew a SSL - certificate

Command

```
[COMMAND]
command = RenewCertificate
certificate = <TEXT>
period = 1
organizationcontact0..1 = <CONTACT>|<NULL>
evapprovercontact0..1 = <CONTACT>|<NULL>
techcontact0..1 = <CONTACT>|<NULL>
csr0..256 = <TEXT>|<NULL>
dcvmethod = EMAIL|DNS-TXT|DNS-CNAME|FILE|<NULL>
dcvtoken = <TEXT>|<NULL>
domain0..250 = <TEXT>|<NULL>
dcvemail0..250 = <EMAIL>|<NULL>
servertype = <TEXT>|<NULL>
period = 1
techfirstname0..1 = <TEXT>|<NULL>
techlastname0..1 = <TEXT>|<NULL>
techphone0..1 = <PHONE>|<NULL>
techemail0..1 = <EMAIL>|<NULL>
techjobtitle = <TEXT>|<NULL>
evapproverfirstname = <TEXT>|<NULL>
evapproverlastname = <TEXT>|<NULL>
evapproverphone = <PHONE>|<NULL>
evapproveremail = <EMAIL>|<NULL>
evapproverjobtitle = <TEXT>|<NULL>
organizationname = <TEXT>|<NULL>
organizationdba = <TEXT>|<NULL>
organizationstreet = <TEXT>|<NULL>
organizationzip = <TEXT>|<NULL>
organizationcity = <TEXT>|<NULL>
organizationstate = <TEXT>|<NULL>
organizationcountry = <COUNTRY>|<NULL>
organizationphone = <PHONE>|<NULL>
organizationemail = <EMAIL>|<NULL>
checkonly = 1|0|<NULL>
apiversion = 1|2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
```

```
description = <TEXT>
property[certificate][n] = <TEXT>
property[sub][n] = <TEXT>
property[status][n] = <TEXT>
property[sub status][n] = <TEXT>
property[fileauth name][n] = <TEXT>
property[fileauth contents][n] = <TEXT>
property[dnsauth name][n] = <TEXT>
EOF
```

Command Parameters

- **certificate**

  Type: <TEXT>

  The ID of the certificate order to be renewed

- **period**

  Type: 1

  How many years the certificate should be valid

- **organizationcontact0..1**

  Type: <CONTACT> | <NULL>

  Organization - CertificateContact

- **evapprovercontact0..1**

  Type: <CONTACT> | <NULL>

  EV approver - CertificateContact

- **techcontact0..1**

  Type: <CONTACT> | <NULL>

  Tech - CertificateContact

- **csr0..256**

  Type: <TEXT> | <NULL>

  The CSR to be used to generate the certificate

- **dcvmethod**

  Method for authentication of the domainname(s)

Option	Description
EMAIL	EMAIL
DNS-TXT	DNS-TXT
DNS-CNAME	DNS-CNAME
FILE	FILE

- **dcvtoken**

  Type: <TEXT> | <NULL>

Token generated by CheckCertificate

- **domain0..250**

  Type: <TEXT> | <NULL>

  Domains for certificate, first one is the CommonName

- **dcvemail0..250**

  Type: <EMAIL> | <NULL>

  Email to validate the request if dcvmethod = email for domain#

- **servertype**

  Type: <TEXT> | <NULL>

  Servertype

- **period**

  Type: 1

  How many years the certificate should be valid

- **techfirstname0..1**

  Type: <TEXT> | <NULL>

  The given name of the new contact (may be empty for an organization handle)

- **techlastname0..1**

  Type: <TEXT> | <NULL>

  The family name of the new contact (may be empty for an organization handle)

- **techphone0..1**

  Type: <PHONE> | <NULL>

  The telephone number for this new contact handle

- **techemail0..1**

  Type: <EMAIL> | <NULL>

  The email address for this new contact handle

- **techjobtitle**

  Type: <TEXT> | <NULL>

  The job title for this new contact handle

- **evapproverfirstname**

  Type: <TEXT> | <NULL>

  The given name of the new contact (may be empty for an organization handle)

- **evapproverlastname**

  Type: <TEXT> | <NULL>

  The family name of the new contact (may be empty for an organization handle)

- **evapproverphone**

  Type: <PHONE> | <NULL>

  The telephone number for this new contact handle

- **evapproveremail**

  Type: <EMAIL> | <NULL>

  The email address for this new contact handle

- **evapproverjobtitle**

  Type: <TEXT> | <NULL>

  The job title for this new contact handle

- **organizationname**

  Type: <TEXT> | <NULL>

  The organization the new contact will be created for

- **organizationdba**

  Type: <TEXT> | <NULL>

  DBA of the organization (if necessary)

- **organizationstreet**

  Type: <TEXT> | <NULL>

  The street for this new contact handle

- **organizationzip**

  Type: <TEXT> | <NULL>

  The postal code for this new contact handle

- **organizationcity**

  Type: <TEXT> | <NULL>

  The city for this new contact handle

- **organizationstate**

  Type: <TEXT> | <NULL>

  The federal state for this new contact handle

- **organizationcountry**

  Type: <COUNTRY> | <NULL>

  The country for this new contact handle (please use the country code, e.g. “DE” or “US”)

- **organizationphone**

  Type: <PHONE> | <NULL>

  The telephone number for this new contact handle

- **organizationemail**

  Type: <EMAIL> | <NULL>

  The email address for this new contact handle

- **checkonly**

  Type: 1 | 0 | <NULL>

  Only verify that the request parameters are valid for the specified class

- **apiversion**

  Select the used API version

Option	Description
1	Pre - 2022 API
2	2022 API

Response Parameters

- **certificate**

  Type: <TEXT>

  ID of the new certificate order

- **sub**

  Type: <TEXT>

  ID of the new certificate sub

- **status**

  Type: <TEXT>

  Status of the request process for the certificate order

- **sub status**

  Type: <TEXT>

  Status of the request process for the sub

- **fileauth name**

  Type: <TEXT>

  Name for the file needed for file based authentication

- **fileauth contents**

  Type: <TEXT>

  Content for the file needed for file based authentication

- **dnsauth name**

  Type: <TEXT>

  Name of the RR entry needed for DNS based authentication

8.1.4 ReissueCertificate

Failed requests will return an ID in the format “FAILxxxxxxxx” which can’t be used for any command. This ID can be used by the RRPproxy support if assistance is required. Reissue a SSL certificate with a new CSR. This is currently not available for Comodo certificates.

Command

```
[COMMAND]
command = ReissueCertificate
certificate = <TEXT>
csr0..256 = <TEXT>|<NULL>
dcvtoken = <TEXT>|<NULL>
apiversion = 1|2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[certificate][n] = <TEXT>
property[sub][n] = <TEXT>
property[status][n] = <TEXT>
property[sub status][n] = <TEXT>
property[fileauth name][n] = <TEXT>
property[fileauth contents][n] = <TEXT>
property[dnsauth name][n] = <TEXT>
EOF
```

Command Parameters

- **certificate**

  Type: <TEXT>

  ID of the certificate order to be reissued

- **csr0..256**

  Type: <TEXT> | <NULL>

  The CSR to be used to generate the certificate

- **dcvtoken**

  Type: <TEXT> | <NULL>

  If a new CSR was provided, token generated by CheckCertificate

- **apiversion**

  Select the used API version

Option	Description
1	Pre - 2022 API
2	2022 API

Response Parameters

- **certificate**

Type: <TEXT>

ID of the new certificate order

- **sub**

  Type: <TEXT>

  ID of the new certificate sub

- **status**

  Type: <TEXT>

  Status of the request process for the certificate order

- **sub status**

  Type: <TEXT>

  Status of the request process for the sub

- **fileauth name**

  Type: <TEXT>

  Name for the file needed for file based authentication

- **fileauth contents**

  Type: <TEXT>

  Content for the file needed for file based authentication

- **dnsauth name**

  Type: <TEXT>

  Name of the RR entry needed for DNS based authentication

8.1.5 DeleteCertificate

Command

```
[COMMAND]
command = DeleteCertificate
certificate = <TEXT>
action = REVOKE|CANCELORDER|REVOKEORDER|<NULL>
apiversion = 1|2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
EOF
```

Command Parameters

- **certificate**

  Type: <TEXT>

  The ID of the certificate order to be deleted

- **action**

  Type of cancelation

Option	Description
REVOKE	Revoke a single certificate in an order
CANCELORDER	Cancel the order before the certificate was issued
REVOKEORDER	Revoke an issued CertificateOrder

- **apiversion**

  Select the used API version

Option	Description
1	Pre - 2022 API
2	2022 API

Response Parameters

8.1.6 RevokeCertificate

Note: This does not issue a refund. For DigiCert the corresponding CertificateOrder can still be used to reissue a Certificate.

Command

```
[COMMAND]
command = RevokeCertificate
certificate = <TEXT>
apiversion = 2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
EOF
```

Command Parameters

- **certificate**

  Type: <TEXT>

  The ID of the certificate to be cancelled

- **apiversion**

  Type: 2 | <NULL>

  Select the used API version

Response Parameters

8.1.7 StatusCertificate

Command

```
[COMMAND]
command = StatusCertificate
certificate = <TEXT>
apiversion = 1|2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[certificate][n] = <TEXT>
property[sub id][n] = <TEXT>
property[sub created date][n] = <DATE>
property[sub][n] = <TEXT>
property[sub status][n] = <TEXT>
property[status][n] = ACTIVE|REQUESTED|PROCESSING|REVOKED|EXPIRED|CANCELLED
property[approveremail][n] = <EMAIL>
property[auth message][n] = <TEXT>
property[fileauth name][n] = <TEXT>
property[fileauth contents][n] = <TEXT>
property[dnsauth name][n] = <TEXT>
property[ownercontact][n] = <CONTACT>
property[admincontact][n] = <CONTACT>
property[techcontact][n] = <CONTACT>
property[billingcontact][n] = <CONTACT>
property[webservertype][n] = apachessl|apacheraven|apachessleay|c2net|Ibmhttp|Iplanet|
    Dominogo4625|Dominogo4626|Domino|iis4|iis5|Netscape|zeusv3|Other|apacheopenssl|apache2|
    apacheapachessl|cobaltseries|cpanel|ensim|hsphere|ipswitch|plesk|tomcat|WebLogic|website|
    webstar|iis
property[class][n] = SSL123|SSLWebServer|SSLWebServerWC|SSLWebServerEV|SGCSuperCerts|
    SecureSite|SecureSitePro|SecureSiteEV|SecureSiteProEV|QuickSSLPremium|TrueBizID|
    TrueBizIDWC|TrueBizIDEV|RapidSSL|RapidSSLWC|InstantSSL|SGCWildcardSSL|PositiveSSL|
    PremiumSSL|ExtendedValidatedSSL|PositiveWildcardSSL|unifiedcommunicationsdv|
    unifiedcommunicationsov|multidomainov|multidomainev
property[domain][n] = <DOMAIN>
property[csr][n] = <TEXT>
property[crt][n] = <TEXT>
property[created date][n] = <DATE>
property[created by][n] = <TEXT>
property[certificate expiration date][n] = <DATE>
property[csr public key algorithm][n] = <TEXT>
property[csr signature algorithm][n] = <TEXT>
property[csr size][n] = <INT>
property[csr subject][n] = <TEXT>
property[csr san][n] = <TEXT>
property[csr key usage][n] = <TEXT>
property[csr emailaddress][n] = <TEXT>
property[csr location][n] = <TEXT>
property[csr commonname][n] = <TEXT>
property[csr state][n] = <TEXT>
property[csr country][n] = <TEXT>
property[csr organization][n] = <TEXT>
property[csr organizational unit][n] = <TEXT>
property[crt public key algorithm][n] = <TEXT>
property[crt signature algorithm][n] = <TEXT>
property[crt validity not before][n] = <DATE>
property[crt validity not after][n] = <DATE>
property[crt size][n] = <INT>
property[crt version][n] = <TEXT>
property[crt serial][n] = <TEXT>
property[crt subject][n] = <TEXT>
property[crt issuer][n] = <TEXT>
property[crt key usage][n] = <TEXT>
property[crt extended key usage][n] = <TEXT>
```

```
property[crt san][n] = <TEXT>
property[crt subject emailaddress][n] = <TEXT>
property[crt subject location][n] = <TEXT>
property[crt subject commonname][n] = <TEXT>
property[crt subject state][n] = <TEXT>
property[crt subject country][n] = <TEXT>
property[crt subject organization][n] = <TEXT>
property[crt subject organizational unit][n] = <TEXT>
property[crt issuer emailaddress][n] = <TEXT>
property[crt issuer location][n] = <TEXT>
property[crt issuer commonname][n] = <TEXT>
property[crt issuer state][n] = <TEXT>
property[crt issuer country][n] = <TEXT>
property[crt issuer organization][n] = <TEXT>
property[crt issuer organizational unit][n] = <TEXT>
property[root crt][n] = <TEXT>
property[intermediate crt][n] = <TEXT>
EOF
```

Command Parameters

- **certificate**

  Type: <TEXT>

  ID of the certificate order

- **apiversion**

  Select the used API version

Option	Description
1	Pre - 2022 API
2	2022 API

Response Parameters

- **certificate**

  Type: <TEXT>

  Unique ID of the certificate order

- **sub id**

  Type: <TEXT>

  Unique ID of the certificate sub

- **sub created date**

  Type: <DATE>

- **sub**

  Type: <TEXT>

  Subs available in this certificate order

- **sub status**

  Type: <TEXT>

  Subs status in this certificate order

- **status**

  Status of the certificate

Option	Description
ACTIVE	ACTIVE
REQUESTED	REQUESTED
PROCESSING	PROCESSING
REVOKED	REVOKED
EXPIRED	EXPIRED
CANCELLED	CANCELLED

- **approveremail**

  Type: <EMAIL>

  Address, where the approval email was sent to

- **auth message**

  Type: <TEXT>

  Information from the provider regarding the authentication process

- **fileauth name**

  Type: <TEXT>

  Name for the file needed for file based authentication

- **fileauth contents**

  Type: <TEXT>

  Content for the file needed for file based authentication

- **dnsauth name**

  Type: <TEXT>

  Name of the RR entry needed for DNS based authentication

- **ownercontact**

  Type: <CONTACT>

  Owner contact of the certificate

- **admincontact**

  Type: <CONTACT>

  Admin contact of the certificate

- **techcontact**

  Type: <CONTACT>

  Technical contact of the certificate

- **billingcontact**

  Type: <CONTACT>

  Billing contact of the certificate

- **webservertype**

  Type of certificate.

Option	Description
apachessl	Apache + MOD SSL
apacheraven	Apache + Raven
apachessleay	Apache + SSLeay
c2net	C2Net Stronghold
Ibmhttp	IBM HTTP
Iplanet	iPlanet Server 4.1
Dominogo4625	Lotus Domino Go 4.6.2.51
Dominogo4626	Lotus Domino Go 4.6.2.6+
Domino	Lotus Domino 4.6+
iis4	Microsoft IIS 4.0
iis5	Microsoft IIS 5.0
Netscape	Netscape Enterprise/FastTrack
zeusv3	Zeus v3+
Other	Other
apacheopenssl	Apache + OpenSSL
apache2	Apache 2
apacheapachessl	Apache + ApacheSSL
cobaltseries	Cobalt Series
cpanel	Cpanel
ensim	Ensim
hsphere	Hsphere
ipswitch	Ipswitch
plesk	Plesk
tomcat	Jakart-Tomcat
WebLogic	WebLogic - all versions
website	O'Reilly WebSite Professional
webstar	WebStar
iis	Microsoft Internet Information Server

- **class**

  Type of certificate

Option	Description
SSL123	Thawte SSL123
SSLWebServer	Thawte SSL Web Server
SSLWebServerWC	Thawte SSL Web Server (Wildcard)
SSLWebServerEV	Thawte SSL Web Server with extended validation
SGCSuperCerts	Thawte SGC SuperCerts

Option	Description
SecureSite	Symantec Secure Site
SecureSitePro	Symantec Secure Site Pro
SecureSiteEV	Symantec Secure Site with extended validation
SecureSiteProEV	Symantec Secure Site Pro with extended validation
QuickSSLPremium	Geotrust QuickSSL Premium
TrueBizID	Geotrust True BusinessID
TrueBizIDWC	Geotrust True BusinessID (Wildcard)
TrueBizIDEV	Geotrust True BuisnessID with extended validation
RapidSSL	RapidSSL
RapidSSLWC	RapidSSL Wildcard
InstantSSL	InstantSSL
SGCWildcardSSL	SGC Wildcard SSL
PositiveSSL	PositiveSSL
PremiumSSL	PremiumSSL
ExtendedValidatedSSL	SectigoSSL EV
PositiveWildcardSSL	PositiveSSL Wildcard
unifiedcommunicationsdv	SectigoSSL UCC
unifiedcommunicationsov	InstantSSL UCC
multidomainov	EnterpriseSSL Pro Multi-Domain
multidomainev	SectigoSSL EV Multi-Domain

- **domain**

  Type: <DOMAIN>

  Domain the certificate may be used for

- **csr**

  Type: <TEXT>

  The certificate signing request used to create the certificate

- **crt**

  Type: <TEXT>

  The certificate signed by the CA

- **created date**

  Type: <DATE>

- **created by**

  Type: <TEXT>

  Date the certificate was issued

- **certificate expiration date**

  Type: <DATE>

  Date the certificate will expire

- **csr public key algorithm**

  Type: <TEXT>

  Public key algorithm used for the CSR

- **csr signature algorithm**

  Type: <TEXT>

  Signature algorithm used for the CSR

- **csr size**

  Type: <INT>

  Size of the CSR

- **csr subject**

  Type: <TEXT>

  Subject encoded in the CSR

- **csr san**

  Type: <TEXT>

  Domains encoded in the CSR

- **csr key usage**

  Type: <TEXT>

  Allowed usage of the CSR

- **csr emailaddress**

  Type: <TEXT>

  Email given in the CSR

- **csr location**

  Type: <TEXT>

  City given in the CSR

- **csr commonname**

  Type: <TEXT>

  Common name given in the CSR

- **csr state**

  Type: <TEXT>

  State given in the CSR

- **csr country**

  Type: <TEXT>

  Country given in the CSR

- **csr organization**

  Type: <TEXT>

  Organization given in the CSR

- **csr organizational unit**

  Type: <TEXT>

  Unit of the organization given in the CSR

- **crt public key algorithm**

  Type: <TEXT>

  Public key algorithm used for the CRT

- **crt signature algorithm**

  Type: <TEXT>

  Signature algorithm used for the CRT

- **crt validity not before**

  Type: <DATE>

  CRT only valid after this date

- **crt validity not after**

  Type: <DATE>

  CRT only valid before this date

- **crt size**

  Type: <INT>

  Size of the CRT

- **crt version**

  Type: <TEXT>

  Version encoded in the CRT

- **crt serial**

  Type: <TEXT>

  Serial of the CRT

- **crt subject**

  Type: <TEXT>

  Subject encoded in the CRT

- **crt issuer**

  Type: <TEXT>

  Issuer of the CRT

- **crt key usage**

  Type: <TEXT>

  Allowed usage for the CRT

- **crt extended key usage**

  Type: <TEXT>

  Details of the allowed usage for the CRT

- **crt san**

  Type: <TEXT>

  Domains encoded in the CRT

- **crt subject emailaddress**

  Type: <TEXT>

  Email given in the CRT

- **crt subject location**

  Type: <TEXT>

  City given in the CRT

- **crt subject commonname**

  Type: <TEXT>

  Common name given in the CRT

- **crt subject state**

  Type: <TEXT>

  State given in the CRT

- **crt subject country**

  Type: <TEXT>

  Country given in the CRT

- **crt subject organization**

  Type: <TEXT>

  Organization given in the CRT

- **crt subject organizational unit**

  Type: <TEXT>

  Unit of the organization given in the CRT

- **crt issuer emailaddress**

  Type: <TEXT>

  Email given in the CRT

- **crt issuer location**

  Type: <TEXT>

  City given in the CRT

- **crt issuer commonname**

  Type: <TEXT>

  Common name given in the CRT

- **crt issuer state**

  Type: <TEXT>

  State given in the CRT

- **crt issuer country**

  Type: <TEXT>

  Country given in the CRT

- **crt issuer organization**

  Type: <TEXT>

  Organization given in the CRT

- **crt issuer organizational unit**

  Type: <TEXT>

  Unit of the organization given in the CRT

- **root crt**

  Type: <TEXT>

  The certificate used by the root CA

- **intermediate crt**

  Type: <TEXT>

  The certificate chain from the root certificate to the certificate

8.1.8 QueryCertificateList

Command

```
[COMMAND]
command = QueryCertificateList
wide = 0|1|<NULL>
certificate = <PATTERN>
certificateorder = <PATTERN>
createddate = <PATTERN>
status = <PATTERN>
status = <PATTERN>
servertype = <PATTERN>
updateddate = <PATTERN>
expirationdate = <PATTERN>
class = <PATTERN>
apiversion = 1|2|<NULL>
```

```
generatelist = 0|1|<NULL>
sendlistemail = <EMAILS>|<NULL>
first = <INT>
limit = <INT>
includesub = 0|1|<NULL>
servicetag0..100 = <TEXT>|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[certificate][n] = <TEXT>
property[class][n] = <TEXT>
property[domain][n] = <DOMAIN>
property[webservertype][n] = apachessl|apacheraven|apachessleay|c2net|Ibmhttp|Iplanet|
    Dominogo4625|Dominogo4626|Domino|iis4|iis5|Netscape|zeusv3|Other|apacheopenssl|apache2|
    apacheapachessl|cobaltseries|cpanel|ensim|hsphere|ipswitch|plesk|tomcat|WebLogic|website|
    webstar|iis
property[servertype][n] = <TEXT>
property[status][n] = <TEXT>
property[created date][n] = <DATE>
property[updated date][n] = <DATE>
property[expiration date][n] = <DATE>
property[owner contact][n] = <CONTACT>
property[admin contact][n] = <CONTACT>
property[tech contact][n] = <CONTACT>
property[billing contact][n] = <CONTACT>
property[approveremail][n] = <EMAIL>
property[registrar][n] = <TEXT>
property[count][n] = <INT>
property[first][n] = <INT>
property[last][n] = <INT>
property[limit][n] = <INT>
property[total][n] = <INT>
EOF
```

Command Parameters

- **wide**

  Type: 0|1|<NULL>

- **certificate**

  Type: <PATTERN>

- **certificateorder**

  Type: <PATTERN>

- **createddate**

  Type: <PATTERN>

- **status**

  Type: <PATTERN>

- **status**

  Type: <PATTERN>

- **servertype**

  Type: <PATTERN>

- **updateddate**

  Type: <PATTERN>

- **expirationdate**

  Type: <PATTERN>

- **class**

  Type: <PATTERN>

- **apiversion**

  Select the used API version

Option	Description
1	Pre - 2022 API
2	2022 API

- **generatelist**

  Generate a CSV File and upload to ftp

Option	Description
0	Show result on screen
1	

- **sendlistemail**

  Type: <EMAILS> | <NULL>

  Email address where the CSV will be sent to

- **first**

  Type: <INT>

  Start the output of results from this item

- **limit**

  Type: <INT>

  Show only this many items in the response

- **includesub**

  Includes all accountings for certificates on all subreseller acccounts in the list.

Option	Description
0	No
1	Yes

- **servicetag0..100**

Type: <TEXT> | <NULL>

Show only services that are associated with this service tag.

Response Parameters

- **certificate**

  Type: <TEXT>

  Unique ID of the certificate

- **class**

  Type: <TEXT>

  The class of the certificate

- **domain**

  Type: <DOMAIN>

  Domain the certificate may be used for

- **webservertype**

  Type: apachessl | apacheraven | apachessleay | c2net | Ibmhttp | Iplanet | Dominogo4625 | Dominogo4626 | Domino | iis4 | iis5 | Netscape | zeusv3 | Other | apacheopenssl | apache2 | apacheapachessl | cobaltseries | cpanel | ensim | hsphere | ipswitch | plesk | tomcat | WebLogic | website | webstar | iis

  Type of the webserver where the certificate can be used

- **servertype**

  Type: <TEXT>

  Type of the server where the certificate can be used

- **status**

  Type: <TEXT>

  Current status of the certificate

- **created date**

  Type: <DATE>

  Date the certificate was issued

- **updated date**

  Type: <DATE>

  Date the certificate was last changed

- **expiration date**

  Type: <DATE>

  Date the certificate expires

- **owner contact**

  Type: <CONTACT>

  Owner contact of the certificate

- **admin contact**

  Type: <CONTACT>

  Admin contact of the certificate

- **tech contact**

  Type: <CONTACT>

  Tech contact of the certificate

- **billing contact**

  Type: <CONTACT>

  Billing contact of the certificate

- **approveremail**

  Type: <EMAIL>

  Address, where the approval email will be send to

- **registrar**

  Type: <TEXT>

  The registrar of the certificate. Only shown when parameter includesub is used.

- **count**

  Type: <INT>

  Total number of certificates shown

- **first**

  Type: <INT>

  Pointer to the first shown ID

- **last**

  Type: <INT>

  Pointer to the last shown ID

- **limit**

  Type: <INT>

  The limit given in the command

- **total**

  Type: <INT>

  Total number of existing certificates

8.2 CertificateOrders

8.2.1 CancelCertificateOrder

Command

```
[COMMAND]
command = CancelCertificateOrder
certificateorder = <TEXT>
apiversion = 2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
EOF
```

Command Parameters

- **certificateorder**

  Type: <TEXT>

  The ID of the certificate order to be cancelled

- **apiversion**

  Type: 2 | <NULL>

  Select the used API version

Response Parameters

8.2.2 RevokeCertificateOrder

Command

```
[COMMAND]
command = RevokeCertificateOrder
certificateorder = <TEXT>
apiversion = 2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
EOF
```

Command Parameters

- **certificateorder**

  Type: <TEXT>

  The ID of the certificate order to be cancelled

- **apiversion**

  Type: 2 | <NULL>

  Select the used API version

Response Parameters

8.2.3 StatusCertificateOrder

Command

```
[COMMAND]
command = StatusCertificateOrder
certificateorder = <TEXT>
apiversion = 1|2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[ca][n] = <TEXT>
property[certificate][n] = <TEXT>
property[certificate expiration date][n] = <DATE>
property[certificate status][n] = active|processing|expired|revoked|failed|canceled|
    cancel_processing|revoke_processing
property[certificateorder][n] = <TEXT>
property[class][n] = <TEXT>
property[created by][n] = <TEXT>
property[created date][n] = <DATE>
property[paid until][n] = <DATE>
property[status][n] = active|processing|expired|revoked|failed|canceled|cancel_processing|
    revoke_processing
property[updated by][n] = <TEXT>
property[updated date][n] = <DATE>
EOF
```

Command Parameters

- **certificateorder**

  Type: <TEXT>

- **apiversion**

  Select the API version used (Command only available in 2022 version)

Option	Description
1	Pre - 2022 API
2	2022 API

Response Parameters

- **ca**

  Type: <TEXT>

  The CA where this CertificateOrder was created

- **certificate**

  Type: <TEXT>

  A certificate within this CertificateOrder

- **certificate expiration date**

  Type: <DATE>

  Expiration date of the certificate

- **certificate status**

  Type: active|processing|expired|revoked|failed|canceled|cancel_processing|revoke_processing

  Status of the certificate

- **certificateorder**

  Type: <TEXT>

  ID of the CertificateOrder

- **class**

  Type: <TEXT>

  Class of the CertificateOrder

- **created by**

  Type: <TEXT>

  The user who created this CertificateOrder

- **created date**

  Type: <DATE>

  Creation date of this CertificateOrder

- **paid until**

  Type: <DATE>

  Date until a certifictae in this CertificateOrder may be valid

- **status**

  Type: active|processing|expired|revoked|failed|canceled|cancel_processing|revoke_processing

  Status of the order

- **updated by**

  Type: <TEXT>

  The user the last update of the CertificateOrder was made by

- **updated date**

  Type: <DATE>

  Last update of a certificate in the CertificateOrder

8.2.4 QueryCertificateOrderList

Search for certificate orders

Command

```
[COMMAND]
command = QueryCertificateOrderList
certificateorder = <PATTERN>
wide = 0|1|<NULL>
order = ASC|DESC|<NULL>
orderby = CERTIFICATEORDER|PAIDUNTIL|STATUS|CLASS|CREATEDDATE|UPDATEDDATE|<NULL>
paiduntil = <PATTERN>
status = <PATTERN>
class = <PATTERN>
createddate = <PATTERN>
updateddate = <PATTERN>
generatelist = 0|1|<NULL>
sendlistemail = <EMAILS>|<NULL>
first = <INT>
limit = <INT>
apiversion = 2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[certificateorder][n] = <TEXT>
property[wide][n] = 0|1|<NULL>
property[order][n] = ASC|DESC|<NULL>
property[orderby][n] = CERTIFICATEORDER|CREATED_DATE|UPDATED_DATE|STATUS|PAIDUNTIL|<NULL>
property[created date][n] = <DATE>
property[updated date][n] = <DATE>
property[paiduntil][n] = <DATE>
property[status][n] = <TEXT>
property[count][n] = <INT>
property[first][n] = <INT>
property[last][n] = <INT>
property[limit][n] = <INT>
property[total][n] = <INT>
EOF
```

Command Parameters

- **certificateorder**

  Type: <PATTERN>

  Search for orders with this ID

- **wide**

  Grade of details returned

Option	Description
0	Show more details
1	Show less details

- **order**

  Type: ASC | DESC | <NULL>

- **orderby**

  Type: CERTIFICATEORDER | PAIDUNTIL | STATUS | CLASS | CREATEDDATE | UPDATEDDATE | <NULL>

  Order the results for the corresponding column. Default is CERTIFICATECONTACT.

- **paiduntil**

  Type: <PATTERN>

  Search for orders with this paid-until - date

- **status**

  Type: <PATTERN>

  Search for orders with this status

- **class**

  Type: <PATTERN>

  Search for orders with this class

- **createddate**

  Type: <PATTERN>

  Search for orders with this created date

- **updateddate**

  Type: <PATTERN>

  Search for orders with this updated date

- **generatelist**

  Generate a CSV File and upload to ftp or send by email

Option	Description
0	Show result on screen
1	

- **sendlistemail**

  Type: <EMAILS> | <NULL>

  Email address where the CSV will be sent to

- **first**

  Type: <INT>

  Start the output of results from this item

- **limit**

  Type: <INT>

  Show only this many items in the response

- **apiversion**

  Type: 2 | <NULL>

  Select the used API version (Command only available in 2022 version)

Response Parameters

- **certificateorder**

  Type: <TEXT>

  ID of the certificate order

- **wide**

  Grade of details returned

Option	Description
0	Show more details
1	Show less details

- **order**

  Type: ASC | DESC | <NULL>

- **orderby**

  Type: CERTIFICATEORDER | CREATED_DATE | UPDATED_DATE | STATUS | PAIDUNTIL | <NULL>

  Order the results for the corresponding column. Default is CERTIFICATECONTACT.

- **created date**

  Type: <DATE>

  Date the order was created

- **updated date**

  Type: <DATE>

  Date of the last modification of the certificate order

- **paiduntil**

  Type: <DATE>

  Date the order will expire

- **status**

  Type: <TEXT>

  Status of the certificate orders.

- **count**

  Type: <INT>

  Total number of contacts shown

- **first**

  Type: <INT>

  Pointer to the first shown ID

- **last**

  Type: <INT>

  Pointer to the last shown ID

- **limit**

  Type: <INT>

  The limit given in the command

- **total**

  Type: <INT>

  Total number of existing contacts

8.3 CertificateContacts

8.3.1 AddCertificateContact

The AddCertificateContact command allows you to add a new contact in our certificate system.

Command

```
[COMMAND]
command = AddCertificateContact
type = organization|tech|evapprover
phandle = <TEXT>|<NULL>
ohandle = <TEXT>|<NULL>
firstname = <TEXT>|<NULL>
lastname = <TEXT>|<NULL>
phone = <PHONE>
email = <EMAIL>
jobtitle = <TEXT>|<NULL>
organizationname = <TEXT>|<NULL>
organizationdba = <TEXT>|<NULL>
organizationstreet = <TEXT>
organizationzip = <TEXT>|<NULL>
organizationcity = <TEXT>
organizationstate = <TEXT>|<NULL>
organizationcountry = <COUNTRY>
organizationphone = <PHONE>
organizationemail = <EMAIL>
apiversion = 2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[certificatecontact][n] = <TEXT>
EOF
```

Command Parameters

- **type**

  Type: `organization` | `tech` | `evapprover`

  The type of contact to be added

- **phandle**

  Type: <TEXT> | <NULL>

  An optional P-Handle to create the contact from

- **ohandle**

  Type: <TEXT> | <NULL>

  An optional O-Handle to create an organization contact from

- **firstname**

  Type: <TEXT> | <NULL>

  The given name of the new contact (may be empty for an organization handle)

- **lastname**

  Type: <TEXT> | <NULL>

  The family name of the new contact (may be empty for an organization handle)

- **phone**

  Type: <PHONE>

  The telephone number for this new contact handle

- **email**

  Type: <EMAIL>

  The email address for this new contact handle

- **jobtitle**

  Type: <TEXT> | <NULL>

  The jobtitle for this new contact handle

- **organizationname**

  Type: <TEXT> | <NULL>

  The organization the new contact will be created for

- **organizationdba**

  Type: <TEXT> | <NULL>

  DBA of the organization (if necessary)

- **organizationstreet**

  Type: <TEXT>

  The street for this new contact handle

- **organizationzip**

  Type: <TEXT> | <NULL>

  The postal code for this new contact handle

- **organizationcity**

  Type: <TEXT>

  The city for this new contact handle

- **organizationstate**

  Type: <TEXT> | <NULL>

  The federal state for this new contact handle

- **organizationcountry**

  Type: <COUNTRY>

  The country for this new contact handle (please use the country code, e.g. “DE” or “US”)

- **organizationphone**

  Type: <PHONE>

  The telephone number for this new contact handle

- **organizationemail**

  Type: <EMAIL>

  The email address for this new contact handle

- **apiversion**

  Type: 2 | <NULL>

  Select the used API version (Command only available in 2022 version)

Response Parameters

- **certificatecontact**

  Type: <TEXT>

  The ID of the new CertificateContact

8.3.2 StatusCertificateContact

Command

```
[COMMAND]
command = StatusCertificateContact
certificatecontact = <TEXT>
apiversion = 2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[created by][n] = <TEXT>
property[created date][n] = <DATE>
property[updated by][n] = <TEXT>
property[updated date][n] = <DATE>
property[registrar][n] = <TEXT>
property[certificatecontact][n] = <TEXT>
property[type][n] = organization|tech|evapprover
property[jobtitle][n] = <TEXT>
property[first name][n] = <TEXT>
property[last name][n] = <TEXT>
property[phone][n] = <PHONE>
property[email][n] = <EMAIL>
property[organizationname][n] = <TEXT>
property[organizationstreet][n] = <TEXT>
property[organizationcity][n] = <TEXT>
property[organizationstate][n] = <TEXT>
property[organizationzip][n] = <TEXT>
property[organizationcountry][n] = <COUNTRY>
property[organizationphone][n] = <PHONE>
property[organizationemail][n] = <EMAIL>
property[status][n] = <TEXT>
EOF
```

Command Parameters

- **certificatecontact**

  Type: <TEXT>

- **apiversion**

  Type: 2 | <NULL>

  Select the used API version (Command only available in 2022 version)

Response Parameters

- **created by**

  Type: <TEXT>

  Account where this contact was created in

- **created date**

  Type: <DATE>

  Date when this contact was created

- **updated by**

  Type: <TEXT>

  Account where the contact was last updated

- **updated date**

  Type: <DATE>

  Date when the contact was last updated

- **registrar**

  Type: <TEXT>

- **certificatecontact**

  Type: <TEXT>

  Contact ID

- **type**

  Type: organization|tech|evapprover

  Type of contact

- **jobtitle**

  Type: <TEXT>

  Title of this contact

- **first name**

  Type: <TEXT>

  First name of this contact

- **last name**

  Type: <TEXT>

  Last name of this contact

- **phone**

  Type: <PHONE>

  Phone number of this contact

- **email**

  Type: <EMAIL>

  Email address of this contact

- **organizationname**

  Type: <TEXT>

  Organization of this contact

- **organizationstreet**

  Type: <TEXT>

  Street of this contact

- **organizationcity**

  Type: <TEXT>

  City of this contact

- **organizationstate**

  Type: <TEXT>

  State of this contact

- **organizationzip**

  Type: <TEXT>

  Zip code of this contact

- **organizationcountry**

  Type: <COUNTRY>

  Country code of this handle

- **organizationphone**

  Type: <PHONE>

  Phone number of this contact

- **organizationemail**

  Type: <EMAIL>

  Email address of this contact

- **status**

  Type: <TEXT>

8.3.3 QueryCertificateContactList

Search for certificate contact handles. Wildcards are allowed for every parameter. Default will show all contacts.

Command

```
[COMMAND]
command = QueryCertificateContactList
certificatecontact = <PATTERN>
wide = 0|1|<NULL>
generatelist = 0|1|<NULL>
sendlistemail = <EMAILS>|<NULL>
order = ASC|DESC|<NULL>
orderby = CERTIFICATECONTACT|TYPE|JOBTITLE|FIRSTNAME|LASTNAME|PHONE|EMAIL|ORGANIZATION|
    ORGANIZATIONDBA|ORGANIZATIONSTREET|ORGANIZATIONCITY|ORGANIZATIONZIP|ORGANIZATIONSTATE|
    ORGANIZATIONCOUNTRY|ORGANIZATIONPHONE|ORGANIZATIONCATEGORY|CREATEDDATE|UPDATEDDATE|<NULL>
first = <INT>
limit = <INT>
apiversion = 2|<NULL>
organization = <TEXT>
email = <PATTERN>|<EMAIL>
type = <PATTERN>
jobtitle = <PATTERN>
firstname = <PATTERN>
lastname = <PATTERN>
phone = <PATTERN>
organizationdba = <PATTERN>
organizationstreet = <PATTERN>
organizationcity = <PATTERN>
organizationzip = <PATTERN>
organizationstate = <PATTERN>
organizationcountry = <PATTERN>
organizationphone = <PATTERN>
organizationcategory = <PATTERN>
createddate = <PATTERN>
updateddate = <PATTERN>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[contact][n] = <CONTACT>
property[type][n] = <TEXT>
property[status][n] = <TEXT>
property[organization][n] = <TEXT>
property[organizationdba][n] = <TEXT>
property[organizationstreet][n] = <TEXT>
property[organizationcity][n] = <TEXT>
property[organizationzip][n] = <TEXT>
property[organizationstate][n] = <TEXT>
property[organizationcountry][n] = <COUNTRY>
property[organizationphone][n] = <PHONE>
property[firstname][n] = <TEXT>
property[lastname][n] = <TEXT>
property[jobtitle][n] = <TEXT>
property[email][n] = <EMAIL>
property[phone][n] = <PHONE>
property[status][n] = <TEXT>
property[deletion date][n] = <DATE>
property[count][n] = <INT>
property[first][n] = <INT>
property[last][n] = <INT>
property[limit][n] = <INT>
property[total][n] = <INT>
EOF
```

Command Parameters

- **certificatecontact**

  Type: <PATTERN>

- **wide**

  Grade of details returned

Option	Description
0	Show more details
1	Show less details

- **generatelist**

  Generate a CSV File and upload to ftp or send by email

Option	Description
0	Show result on screen
1	

- **sendlistemail**

  Type: <EMAILS> | <NULL>

  Email address where the CSV will be sent to

- **order**

Type: ASC | DESC | <NULL>

- **orderby**

  Type: CERTIFICATECONTACT | TYPE | JOBTITLE | FIRSTNAME | LASTNAME | PHONE | EMAIL | ORGANIZATION | ORGANIZATIONDBA | ORGANIZATIONSTREET | ORGANIZATIONCITY | ORGANIZATIONZIP | ORGANIZATIONSTATE | ORGANIZATIONCOUNTRY | ORGANIZATIONPHONE | ORGANIZATIONCATEGORY | CREATEDDATE | UPDATEDDATE | <NULL>

  Order the results for the corresponding column. Default is CERTIFICATECONTACT.

- **first**

  Type: <INT>

  Start the output of results from this item

- **limit**

  Type: <INT>

  Show only this many items in the response

- **apiversion**

  Type: 2 | <NULL>

  Select the used API version (Command only available in 2022 version)

- **organization**

  Type: <TEXT>

  Search for contacts with this organization

- **email**

  Type: <PATTERN> | <EMAIL>

  Search for contacts with this email

- **type**

  Type: <PATTERN>

  Search for contacts of this type

- **jobtitle**

  Type: <PATTERN>

  Search for contacts with this jobtitle

- **firstname**

  Type: <PATTERN>

  Search for contacts with this first name

- **lastname**

  Type: <PATTERN>

  Search for contacts with this last name

- **phone**

  Type: <PATTERN>

  Search for contacts with this phone number

- **organizationdba**

  Type: <PATTERN>

  Search for contacts doing business as…

- **organizationstreet**

  Type: <PATTERN>

  Search for contacts with this street

- **organizationcity**

  Type: <PATTERN>

  Search for contacts in this city

- **organizationzip**

  Type: <PATTERN>

  Search for contacts in this zip code

- **organizationstate**

  Type: <PATTERN>

  Search for contacts in this state

- **organizationcountry**

  Type: <PATTERN>

  Search for contacts in this country

- **organizationphone**

  Type: <PATTERN>

  Search for contacts with this organizational phone

- **organizationcategory**

  Type: <PATTERN>

  Search for contacts with this business category

- **createddate**

  Type: <PATTERN>

  Search for contacts created on this date

- **updateddate**

  Type: <PATTERN>

  Search for contacts updated on this date

Response Parameters

- **contact**

  Type: <CONTACT>

  ID of the contact handle

- **type**

  Type: <TEXT>

- **status**

  Type: <TEXT>

  Status of the certificate contact

- **organization**

  Type: <TEXT>

  Organization of the certificate contact

- **organizationdba**

  Type: <TEXT>

  Organization doing business as

- **organizationstreet**

  Type: <TEXT>

  Street for the Organization of the certificate contact

- **organizationcity**

  Type: <TEXT>

  City for the Organization of the certificate contact

- **organizationzip**

  Type: <TEXT>

  Zip for the Organization of the certificate contact

- **organizationstate**

  Type: <TEXT>

  State for the Organization of the certificate contact

- **organizationcountry**

  Type: <COUNTRY>

  Country for the Organization of the certificate contact

- **organizationphone**

  Type: <PHONE>

  Phone number for the Organization of the certificate contact

- **firstname**

  Type: <TEXT>

  First name of the certificate contact

- **lastname**

  Type: <TEXT>

  Last name of the certificate contact

- **jobtitle**

  Type: <TEXT>

  Job title of the certificate contact

- **email**

  Type: <EMAIL>

  Email of the certificate contact

- **phone**

  Type: <PHONE>

  Phone number of the certificate contact

- **status**

  Type: <TEXT>

  Status of the contact. Can be linked, unlinked or clientAutoDeleteProhibited.

- **deletion date**

  Type: <DATE>

  Deletion date when the unlinked contact will be deleted.

- **count**

  Type: <INT>

  Total number of contacts shown

- **first**

  Type: <INT>

  Pointer to the first shown ID

- **last**

  Type: <INT>

  Pointer to the last shown ID

- **limit**

  Type: <INT>

  The limit given in the command

- **total**

  Type: <INT>

  Total number of existing contacts

8.4 General

8.4.1 ResendNotification

Command

```
[COMMAND]
command = ResendNotification
type = domaintransfer|certificate|trademark|claimsnotice|contactverification|ownerchange
object = <TEXT>
reason = <TEXT>|<NULL>
recipient = owner|admin|gaining|losing|both|<NULL>
sub = <TEXT>|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[owner][n] = <TEXT>
property[admin][n] = <TEXT>
EOF
```

Command Parameters

- **type**

  Type of notification to resend

Option	Description
domaintransfer	Domaintransfer (incl.FOA)
certificate	Certificate
trademark	Trademark (incl. SMD file)
claimsnotice	Claimsnotice
contactverification	Contactverification
ownerchange	

- **object**

  Type: <TEXT>

  Domain or Certificate ID or Trademark or Email address

- **reason**

  Type: <TEXT> | <NULL>

  Reason message for resending the notification

- **recipient**

Who should receive the notify again

Option	Description
owner	Owner
admin	Admin
gaining	Both
losing	
both	

- **sub**

  Type: <TEXT> | <NULL>

  The ID of the certificate sub where the notification should be send again

Response Parameters

- **owner**

  Type: <TEXT>

  FOA mail Owner

- **admin**

  Type: <TEXT>

  FOA mail Admin

8.4.2 GetCertificateInfo

Query information about a certificate class

Command

```
[COMMAND]
command = GetCertificateInfo
class = geotrustflexdv|rapidssldv|securesiteflexov|securesiteflexev|securesiteproov|
    securesiteproev|ssl123dv|webserverov|webserverev|geotrusttruebizidov|geotrusttruebizidev|
    instantsslov|premiumov|singledomainev|positivessldv|unifiedcommunicationsdv|
    unifiedcommunicationsov|multidomainev|multidomainov|multidomaindv|<NULL>
apiversion = 1|2|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[class][n] = <TEXT>
property[dns auth][n] = 0|1
property[encryption][n] = <TEXT>
property[extended validation][n] = 0|1
property[file auth][n] = 0|1
property[name][n] = <TEXT>
property[periods][n] = <TEXT>
property[prefix][n] = <TEXT>
property[provider][n] = <TEXT>
property[reissue][n] = 0|1
property[revoke][n] = 0|1
property[san][n] = 0|1
property[validation method][n] = domain|organization|extended
```

```
property[wildcard][n] = 0|1
EOF
```

Command Parameters

- **class**

  Return information about this zone

Option	Description
geotrustflexdv	GeoTrust Flex DV
rapidssldv	RapidSSL
securesiteflexov	DigiCert Secure Site Flex OV
securesiteflexev	DigiCert Secure Site Flex EV
securesiteproov	DigiCert Secure Site Pro OV
securesiteproev	DigiCert Secure Site Pro EV
ssl123dv	SSL123
webserverov	webserver OV
webserverev	webserver EV
geotrusttruebizidov	GeoTrust True Business ID OV
geotrusttruebizidev	GeoTrust True Business ID EV
instantsslov	PremiumSSL
premiumov	SectigoSSL Premium OV
singledomainev	EV SSL
positivessldv	PositiveSSL
unifiedcommunicationsdv	DV SSL Unified Communications Certificate
unifiedcommunicationsov	OV SSL Unified Communications Certificate
multidomainev	SectiGo Multi-Domain SSL EV
multidomainov	SectiGo Multi-Domain SSL OV
multidomaindv	SectiGo Multi-Domain SSL DV

- **apiversion**

  Select the used API version

Option	Description
1	Pre - 2022 API
2	2022 API

Response Parameters

- **class**

  Type: <TEXT>

  Product name of the certificate class

- **dns auth**

  This class has DNS authentication

Option	Description
0	This class has no DNS authentication
1	

- **encryption**

  Type: <TEXT>

  The encryption possible with this certificate class

- **extended validation**

  This class has no extended validation procedures

Option	Description
0	This class has extended validation procedures
1	

- **file auth**

  This class has FILE authentication

Option	Description
0	This class has no FILE authentication
1	

- **name**

  Type: <TEXT>

  Productname of the certificate class

- **periods**

  Type: <TEXT>

  Years possible with this class, comma-separated list

- **prefix**

  Type: <TEXT>

  2 characters prefix for this class in RRPproxy

- **provider**

  Type: <TEXT>

  Provider of the certificates

- **reissue**

  Automated reissue is possible

Option	Description
0	Automated reissue is not possible
1	

- **revoke**

  Automated revoke is possible

Option	Description
0	Automated revoke is not possible
1	

- **san**

  Subject Alternative Names are allowed

Option	Description
0	No Subject Alternative Names allowed
1	

- **validation method**

  Extended validation will be performed

Option	Description
domain	Domain validation only
organization	Domain and Organization will be vetted
extended	

- **wildcard**

  Wildcards (*.example.com) are possible

Option	Description
0	No wildcards possible
1	

8.4.3 QueryCommandSyntax

Please note: By default, QueryCommandSyntax does not return any information based on this SSL API version (apiversion=2). This information is returned when one of the current certificate classes is used for the CLASS parameter.

Command

```
[COMMAND]
command = QueryCommandSyntax
```

```
parent = 0|1|<NULL>
commandname = <TEXT>
type = REQUEST|RESPONSE|<NULL>
showonlyparent = 1|0|<NULL>
domain = <TEXT>
class = <TEXT>|<NULL>
ignorezone = 0|1|<NULL>
generatelist = 0|1|<NULL>
sendlistemail = <EMAILS>|<NULL>
EOF
```

Response

```
[RESPONSE]
code = <INT>
description = <TEXT>
property[commanddescription][n] = <TEXT>
property[commandname][n] = <TEXT>
property[description][n] = <TEXT>
property[domain][n] = <TEXT>
property[optional][n] = <TEXT>
property[parameter][n] = <TEXT>
property[paramgroup][n] = <TEXT>
property[position][n] = <TEXT>
property[property][n] = <TEXT>
property[quantity][n] = <TEXT>
property[range][n] = <TEXT>
property[title][n] = <TEXT>
property[type][n] = <TEXT>
property[wide][n] = <TEXT>
property[zone][n] = <TEXT>
EOF
```

Command Parameters

- **parent**

  Type: 0|1|<NULL>

- **commandname**

  Type: <TEXT>

  The name of the command where the syntax is unknown

- **type**

  Type: REQUEST|RESPONSE|<NULL>

  Type of syntax to be shown

- **showonlyparent**

  Type: 1|0|<NULL>

  Shows only parent values

- **domain**

  Type: <TEXT>

- **class**

  Type: <TEXT>|<NULL>

  A class to query (may be a TLD, too)

- **ignorezone**

  Type: `0` | `1` | <NULL>

  Shows all entries of all zones

- **generatelist**

  Generate a CSV File and upload to ftp

Option	Description
0	Show result on screen
1	

- **sendlistemail**

  Type: <EMAILS> | <NULL>

  Email address where the CSV will be sent to

Response Parameters

- **commanddescription**

  Type: <TEXT>

- **commandname**

  Type: <TEXT>

- **description**

  Type: <TEXT>

- **domain**

  Type: <TEXT>

- **optional**

  Type: <TEXT>

- **parameter**

  Type: <TEXT>

- **paramgroup**

  Type: <TEXT>

- **position**

  Type: <TEXT>

- **property**

  Type: <TEXT>

- **quantity**

  Type: <TEXT>

- **range**

  Type: <TEXT>

- **title**

  Type: <TEXT>

- **type**

  Type: <TEXT>

- **wide**

  Type: <TEXT>

- **zone**

  Type: <TEXT>

9 Appendices

9.1 Examples

9.1.1 Get list of all available CertificateClasses

```
[COMMAND]
command = GetCertificateInfo
apiversion = 2

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.007
runtime = 0.251
property[class][0] = geotrustflexdv
property[class][1] = geotrusttruebizidev
[...]
property[column][0] = class
```

9.1.2 Get details about a CertificateClass

```
[COMMAND]
command = GetCertificateInfo
apiversion = 2
class = SSL123DV

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.052
queuetime = 0.007
property[brand][0] = DigiCert
property[class][0] = ssl123dv
property[dcv dns cname][0] = 1
property[dcv dns txt][0] = 1
property[dcv email][0] = 1
property[dcv http][0] = 1
property[evapprovercontact][0] = NO
property[included domains with base][0] = 1
property[included nonwildcard domains with base][0] = 1
```

```
property[included wildcard domains with base][0] = 1
property[max domains][0] = 250
property[max domains including alternatives][0] = 250
property[max free alternatives][0] = 3
property[max nonwildcard domains][0] = 250
property[max wildcard domains][0] = 250
property[name][0] = SSL123DV
property[organizationcontact][0] = NO
property[periods][0] = 1y
property[prefix][0] = DK
property[server type][0] = barracuda
property[server type][1] = lighttpd
[...]
property[supplier][0] = DIGICERT
property[supports immediate issuance][0] = 1
property[techcontact][0] = MANDATORY
property[type][0] = DV
property[wildcard][0] = 1
```

9.1.3 Get parsed details about a CSR

```
[COMMAND]
command = CheckCertificate
apiversion = 2
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.133
queuetime = 0.341
property[csr commonname][0] = example.com
property[csr public key algorithm][0] = rsaEncryption
property[csr public key size][0] = 4096
property[csr subject][0] = C = US, ST = CA, L = San Francisco, O = Example Corp, CN = example
    .com
property[csr subject commonname][0] = example.com
property[csr subject country][0] = US
property[csr subject location][0] = San Francisco
property[csr subject organization][0] = Example Corp.
property[csr subject state][0] = CA
```

9.1.4 Get parsed details about a CRT

```
[COMMAND]
command = CheckCertificate
apiversion = 2
crt0 = -----BEGIN CERTIFICATE-----
crt1 = MIIHqjCCBZKgAwIBAgIQCLM5v+pkvhFRhZbjB4lKqTANBgkqhkiG9w0BAQsFADBc
[...]
crt42 = -----END CERTIFICATE-----

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.023
queuetime = 0.025
property[crt authority key identifier][0] = 00:aa:bb:cc:dd:ee:ff
    :11:22:33:44:55:66:77:88:99:11:22:33:44
property[crt commonname][0] = example.com
property[crt extended key usage][0] = TLS Web Server Authentication
```

```
property[crt extended key usage][1] = TLS Web Client Authentication
property[crt issuer][0] = C = US, O = "DigiCert, Inc.", CN = RapidSSL Global TLS RSA4096
    SHA256 2022 CA1
property[crt issuer commonname][0] = RapidSSL Global TLS RSA4096 SHA256 2022 CA1
property[crt issuer country][0] = US
property[crt issuer organization][0] = DigiCert, Inc.
property[crt key usage][0] = Digital Signature
property[crt key usage][1] = Key Encipherment
property[crt key usage type][0] = critical
property[crt public key algorithm][0] = rsaEncryption
property[crt public key size][0] = 2048
property[crt san][0] = example.com
property[crt san][1] = www.example.com
property[crt serial][0] = 00:aa:bb:cc:dd:ee:ff:11:22:33:44:55:66:77:88:99
property[crt signature algorithm][0] = sha256WithRSAEncryption
property[crt subject][0] = CN = example.com
property[crt subject commonname][0] = example.com
property[crt validity not after][0] = 2023-07-08 23:59:59
property[crt validity not before][0] = 2022-07-08 00:00:00
```

9.1.5 Get valid dcv - email addresses for a CSR & domains

```
[COMMAND]
command = CheckCertificate
apiversion = 2
class = ssl123dv
domain0 = example.com
domain1 = example.org

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.341
runtime = 3.741
property[dcv email][0] = postmaster@example.com
property[dcv email][1] = administrator@example.com
property[dcv email][2] = hostmaster@example.com
property[dcv email][3] = admin@example.com
property[dcv email][4] = webmaster@example.com
property[dcv email][5] = hostmaster@example.org
property[dcv email][6] = webmaster@example.org
property[dcv email][7] = admin@example.org
property[dcv email][8] = administrator@example.org
property[dcv email][9] = postmaster@example.org
property[domain][0] = example.com
property[domain][1] = example.com
property[domain][2] = example.com
property[domain][3] = example.com
property[domain][4] = example.com
property[domain][5] = example.org
property[domain][6] = example.org
property[domain][7] = example.org
property[domain][8] = example.org
property[domain][9] = example.org
```

9.1.6 A roundhouse check of CSR, CRT and dcv email addresses

```
[COMMAND]
command = CheckCertificate
apiversion = 2
class = ssl123dv
domain0 = example.com
domain1 = example.org
csr0 = -----BEGIN CERTIFICATE REQUEST-----
```

```
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----
crt0 = -----BEGIN CERTIFICATE-----
crt1 = MIIHqjCCBZKgAwIBAgIQCLM5v+pkvhFRhZbjB4lKqTANBgkqhkiG9w0BAQsFADBc
[...]
crt42 = -----END CERTIFICATE-----

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.008
runtime = 3.926
property[crt authority key identifier][0] = 00:aa:bb:cc:dd:ee:ff
    :11:22:33:44:55:66:77:88:99:11:22:33:44
property[crt commonname][0] = example.com
property[crt extended key usage][0] = TLS Web Server Authentication
property[crt extended key usage][1] = TLS Web Client Authentication
property[crt issuer][0] = C = US, O = "DigiCert, Inc.", CN = RapidSSL Global TLS RSA4096
    SHA256 2022 CA1
property[crt issuer commonname][0] = RapidSSL Global TLS RSA4096 SHA256 2022 CA1
property[crt issuer country][0] = US
property[crt issuer organization][0] = DigiCert, Inc.
property[crt key usage][0] = Digital Signature
property[crt key usage][1] = Key Encipherment
property[crt key usage type][0] = critical
property[crt public key algorithm][0] = rsaEncryption
property[crt public key size][0] = 2048
property[crt san][0] = example.com
property[crt san][1] = www.example.com
property[crt serial][0] = 00:aa:bb:cc:dd:ee:ff:11:22:33:44:55:66:77:88:99
property[crt signature algorithm][0] = sha256WithRSAEncryption
property[crt subject][0] = CN = example.com
property[crt subject commonname][0] = example.com
property[crt validity not after][0] = 2023-07-08 23:59:59
property[crt validity not before][0] = 2022-07-08 00:00:00
property[csr commonname][0] = example.com
property[csr public key algorithm][0] = rsaEncryption
property[csr public key size][0] = 4096
property[csr subject][0] = C = US, ST = CA, L = San Francisco, O = Example Corp, CN = example
    .com
property[csr subject commonname][0] = example.com
property[csr subject country][0] = US
property[csr subject location][0] = San Francisco
property[csr subject organization][0] = Example Corp.
property[csr subject state][0] = CA
property[dcv email][0] = postmaster@example.com
property[dcv email][1] = administrator@example.com
property[dcv email][2] = hostmaster@example.com
property[dcv email][3] = admin@example.com
property[dcv email][4] = webmaster@example.com
property[dcv email][5] = hostmaster@example.org
property[dcv email][6] = webmaster@example.org
property[dcv email][7] = admin@example.org
property[dcv email][8] = administrator@example.org
property[dcv email][9] = postmaster@example.org
property[domain][0] = example.com
property[domain][1] = example.com
property[domain][2] = example.com
property[domain][3] = example.com
property[domain][4] = example.com
property[domain][5] = example.org
property[domain][6] = example.org
property[domain][7] = example.org
property[domain][8] = example.org
property[domain][9] = example.org
```

9.1.7 Create a new technical CertificateContact

```
[COMMAND]
command = AddCertificateContact
apiversion = 2
type = tech
firstname = Max
lastname = Mustermann
email = max@example.org
phone = +1.5552345678
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.012
property[certificatecontact][0] = CC-12345678
property[status][0] = unverified
EOF
```

9.1.8 Create a new organizational CertificateContact

```
[COMMAND]
command = AddCertificateContact
apiversion = 2
type = organization
firstname = Max
lastname = Mustermann
email = max@example.org
phone = +1.2345678
organizationname = Example Corp
organizationdba = Example DBA
organizationstreet = Example Street 1
organizationcity = Exampletown
organizationstate = EX
organizationcountry = DE
organizationzip = 12345
organizationphone = +1.55512345678
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.601
property[certificatecontact][0] = CC-12345678
property[status][0] = unverified
EOF
```

9.1.9 Create a new organizational CertificateContact based on existing O-/P-Handles

```
[COMMAND]
command = AddCertificateContact
apiversion = 2
type = organization
ohandle = O-ABC456
phandle = P-ABC123
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.016
property[certificatecontact][0] = CC-12345678
property[status][0] = unverified
```

9.1.10 Get a list of all CertificateContacts

```
[COMMAND]
command = QueryCertificateContactList
apiversion = 2
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.037
queuetime = 0.774
property[certificatecontact][0] = CC-12345678
property[certificatecontact][1] = CC-23456789
property[certificatecontact][2] = CC-34567891
property[certificatecontact][3] = CC-45678912
property[certificatecontact][4] = CC-56789123
property[certificatecontact][5] = CC-67891234
property[certificatecontact][6] = CC-78912345
property[column][0] = certificatecontact
property[count][0] = 7
property[first][0] = 0
property[last][0] = 6
property[limit][0] = 1000
property[total][0] = 8
```

9.1.11 Get a detailed list of 2 CertificateContacts

```
[COMMAND]
command = QueryCertificateContactList
apiversion = 2
wide = 1
limit = 2

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.003
queuetime = 0.005
property[certificatecontact][0] = CC-12345678
property[type][0] = organization
property[job title][0] =
property[first name][0] = Max
property[last name][0] = Mustermann
property[organization name][0] = Example Corp.
property[organization dba][0] = Domain company Inc.
property[organization street][0] = Example street 1
property[organization city][0] = San Francisco
property[organization state][0] = CA
property[organization zip][0] = 94102
property[organization country][0] = US
property[organization phone][0] = +1.555123456
property[organization email][0] = contact@example.com
property[phone][0] = +1.555123456
property[email][0] = max.mustermann@example.com
property[created date][0] = 2022-01-18 14:20:09
property[updated date][0] = 2022-01-18 14:20:09
property[certificatecontact][1] = CC-23456789
property[type][1] = tech
property[job title][1] =
property[first name][1] = Max
property[last name][1] = Mustermann
property[organization name][1] =
property[organization dba][1] =
property[organization street][1] =
property[organization city][1] =
property[organization state][1] =
```

```
property[organization zip][1] =
property[organization country][1] =
property[organization phone][1] =
property[organization email][1] =
property[phone][1] = +1.55556789
property[email][1] = max@example.com
property[created date][1] = 2022-01-18 13:33:32
property[updated date][1] = 2022-01-18 13:33:32
property[column][0] = certificatecontact
property[column][1] = type
property[column][2] = job title
property[column][3] = first name
property[column][4] = last name
property[column][5] = organization name
property[column][6] = organization dba
property[column][7] = organization street
property[column][8] = organization city
property[column][9] = organization state
property[column][10] = organization zip
property[column][11] = organization country
property[column][12] = organization phone
property[column][13] = organization email
property[column][14] = phone
property[column][15] = email
property[column][16] = created date
property[column][17] = updated date
property[count][0] = 2
property[first][0] = 0
property[last][0] = 1
property[limit][0] = 2
property[total][0] = 8
```

9.1.12 Get details about a CertificateContact

Note: StatusCertificateContact will always return all fields, regardless of the contact type.

```
[COMMAND]
command = StatusCertificateContact
apiversion = 2
certificatecontact = CC-23456789
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.005
property[created by][0] = <registrar>
property[created date][0] = 2022-07-05 12:35:43.0
property[email][0] = max@example.org
property[first name][0] = Max
property[job title][0] =
property[last name][0] = Mustermann
property[organization category][0] =
property[organization city][0] =
property[organization country][0] =
property[organization dba][0] =
property[organization name][0] =
property[organization phone][0] =
property[organization state][0] =
property[organization street][0] =
property[organization zip][0] =
property[phone][0] = +1.5552345678
property[type][0] = tech
property[updated by][0] = <registrar>
property[updated date][0] = 2022-07-05 12:35:43.0
EOF
```

9.1.13 Check of billed types for an order of a single domain certificate

```
[COMMAND]
command = AddCertificate
apiversion = 2
techcontact0 = P-ABC123
class = SSL123DV
domain0 = example.com
checkonly = 1
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.007
runtime = 0.224
property[billingcount][0] = 1
property[billingtype][0] = ssl123dv-base
property[domain][0] = example.com
property[domain][1] = www.example.com
property[price][0] = 42
property[vat][0] = 6.72
```

9.1.14 Check of billed types for an order of a multi domain certificate

In this example the response displays the exact calculation of the final price and the resulting list of domains in the certificate.

```
[COMMAND]
command = AddCertificate
apiversion = 2
techcontact0 = P-ABC123
class = SSL123DV
domain0 = example.com
domain1 = example.net
domain2 = ftp.example.org
checkonly = 1
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.2
runtime = 0.186
property[billingcount][0] = 2
property[billingcount][1] = 1
property[billingtype][0] = ssl123dv-san
property[billingtype][1] = ssl123dv-base
property[domain][0] = example.com
property[domain][1] = example.net
property[domain][2] = ftp.example.org
property[domain][3] = www.example.com
property[domain][4] = www.example.net
property[domain][5] = www.ftp.example.org
property[price][0] = 98
property[vat][0] = 15.68
```

9.1.15 Check of billed types for an order of a multi domain certificate without automatically generated domains

```
[COMMAND]
command = AddCertificate
apiversion = 2
techcontact0 = P-ABC123
class = ssl123dv
domain0 = example.com
domain1 = example.net
domain2 = ftp.example.com
checkonly = 1
noautofilldomains = 1
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.006
runtime = 0.046
property[billingcount][0] = 2
property[billingcount][1] = 1
property[billingtype][0] = ssl123dv-san
property[billingtype][1] = ssl123dv-base
property[domain][0] = example.com
property[domain][1] = example.net
property[domain][2] = ftp.example.com
property[price][0] = 98
property[vat][0] = 15.68
```

9.1.16 Ordering a certificate with email validation

```
[COMMAND]
command = AddCertificate
apiversion = 2
techcontact0 = P-ABC123
class = ssl123dv
dcvmethod = email
domain0 = example.com
dcvemail0 = hostmaster@example.com
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 3.646
queuetime = 0.007
property[billingcount][0] = 1
property[billingtype][0] = ssl123dv-base
property[ca][0] = DigiCert
property[ca certificate id][0] = 234597282
property[ca order id][0] = 232946177
property[certificate][0] = DK67473099
property[class][0] = SSL123DV
property[created by][0] = [registrar]
property[created date][0] = 2022-06-27 13:05:54.0
property[csr][0] = -----BEGIN CERTIFICATE REQUEST-----
property[csr][1] = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
property[csr][26] = -----END CERTIFICATE REQUEST-----
property[dcv domain][0] = example.com
property[dcv domain][1] = example.com
```

```
property[dcv email][0] = hostmaster@example.com
property[dcv email][1] = hostmaster@example.com
property[dcv method][0] = email
property[dcv method][1] = email
property[domain][0] = example.com
property[domain][1] = www.example.com
property[order][0] = DKO98589617
property[order paid until][0] = 2023-06-27 13:05:54.0
property[price][0] = 42
property[server type][0] = apache
property[status][0] = processing
property[techcontact][0] = CC-63525687
property[updated by][0] = [registrar]
property[updated date][0] = 2022-06-27 13:05:54.0
property[vat][0] = 6.72
```

9.1.17 Ordering a certificate with dns validation

```
[COMMAND]
command = AddCertificate
apiversion = 2
techcontact0 = P-ABC123
class = ssl123dv
dcvmethod = dns-txt
domain0 = example.com
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.007
runtime = 2.754
property[billingcount][0] = 1
property[billingtype][0] = ssl123dv-base
property[ca][0] = DigiCert
property[ca certificate id][0] = 234597908
property[ca order id][0] = 232946801
property[certificate][0] = DK72503814
property[class][0] = SSL123DV
property[created by][0] = [registrar]
property[created date][0] = 2022-06-27 13:09:07.0
property[csr][0] = -----BEGIN CERTIFICATE REQUEST-----
property[csr][1] = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
property[csr][26] = -----END CERTIFICATE REQUEST-----
property[dcv domain][0] = example.com
property[dcv domain][1] = example.com
property[dcv method][0] = dns-txt
property[dcv method][1] = dns-txt
property[dns rr][0] = @ 3600 IN TXT bk5yp3dh9xrb94byhhny76dhxt0wdkn4
property[dns rr][1] = @ 3600 IN TXT bk5yp3dh9xrb94byhhny76dhxt0wdkn4
property[domain][0] = example.com
property[domain][1] = www.example.com
property[order][0] = DKO90768788
property[order paid until][0] = 2023-06-27 13:09:07.0
property[price][0] = 42
property[server type][0] = apache
property[status][0] = processing
property[techcontact][0] = CC-56558123
property[updated by][0] = [registrar]
property[updated date][0] = 2022-06-27 13:09:07.0
property[vat][0] = 6.72
```

9.1.18 Ordering a certificate with http validation

```
[COMMAND]
command = AddCertificate
apiversion = 2
techcontact0 = P-ABC123
class = ssl123dv
dcvmethod = http
domain0 = example.com
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 2.721
property[billingcount][0] = 1
property[billingtype][0] = ssl123dv-base
property[ca][0] = DigiCert
property[ca certificate id][0] = 234567890
property[ca order id][0] = 123456789
property[certificate][0] = DK12345678
property[class][0] = SSL123DV
property[created by][0] = <registrar>
property[created date][0] = 2022-07-04 12:35:40.0
property[csr][0] = -----BEGIN CERTIFICATE REQUEST-----
property[csr][1] = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
property[csr][26] = -----END CERTIFICATE REQUEST-----
property[dcv domain][0] = example.com
property[dcv domain][1] = example.com
property[dcv method][0] = http
property[dcv method][1] = http
property[domain][0] = example.com
property[domain][1] = www.example.com
property[file contents][0] = 5d00kd5l6w9yvvtp8k2srgj1kjy643fz
property[file contents][1] = 5d00kd5l6w9yvvtp8k2srgj1kjy643fz
property[file name][0] = https://example.com/.well-known/pki-validation/fileauth.txt
property[file name][1] = https://example.com/.well-known/pki-validation/fileauth.txt
property[order][0] = DKO123456678
property[order paid until][0] = 2023-07-04 12:35:40.0
property[price][0] = 42
property[server type][0] = apache
property[status][0] = processing
property[techcontact][0] = CC-12345678
property[updated by][0] = <registrar>
property[updated date][0] = 2022-07-04 12:35:40.0
property[vat][0] = 6.72
```

9.1.19 Ordering a new certificate with provided contact details

```
[COMMAND]
command = AddCertificate
apiversion = 2
class = ssl123dv
dcvmethod = email
domain0 = example.com
csr0 =-----BEGIN CERTIFICATE REQUEST-----
csr1 = MIICnzCCAYcCAQAwWjELMAkGA1UEBhMCREUxDDAKBgNVBAgMA05SVzEWMBQGA1UE
[...]
csr16 = -----END CERTIFICATE REQUEST-----
techfirstname0 = Max
techlastname0 = Mustermann
techemail0 = max@example.org
techphone0 = +1.2345678
```

```
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 3.646
queuetime = 0.007
property[billingcount][0] = 1
property[billingtype][0] = ssl123dv-base
property[ca][0] = DigiCert
property[ca certificate id][0] = 234597282
property[ca order id][0] = 232946177
property[certificate][0] = DK67473099
property[class][0] = SSL123DV
property[created by][0] = <registrar>
property[created date][0] = 2022-06-27 13:05:54.0
property[csr][0] = -----BEGIN CERTIFICATE REQUEST-----
property[csr][1] = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
property[csr][26] = -----END CERTIFICATE REQUEST-----
property[dcv domain][0] = example.com
property[dcv domain][1] = example.com
property[dcv email][0] = hostmaster@example.com
property[dcv email][1] = hostmaster@example.com
property[dcv method][0] = email
property[dcv method][1] = email
property[domain][0] = example.com
property[domain][1] = www.example.com
property[order][0] = DKO98589617
property[order paid until][0] = 2023-06-27 13:05:54.0
property[price][0] = 42
property[server type][0] = apache
property[status][0] = processing
property[techcontact][0] = CC-12345678
property[updated by][0] = <registrar>
property[updated date][0] = 2022-06-27 13:05:54.0
property[vat][0] = 6.72
```

9.1.20 Get a certificate with immediate issuance

Immediate issuance requires two steps. First the CSR and CertifcateClass need to be submitted to CheckCertificate. This will return an appropriate token valid for the domain(s) and CA requested. This token has an limited validity period of 30 days.

```
[COMMAND]
command = CheckCertificate
apiversion = 2
generatedcvtoken = 1
class = ssl123dv
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----
domain0 = example.com

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.195
queuetime = 1.057
[...]
property[dcv token][0] = <token>
```

This token needs to be added to the corresponding domain(s) DNS / Webserver. After the token is deployed, AddCertificate can be called. If the change was not propagated, AddCertificate will fail immediately. No further event is created and nothing is charged. The command can be tried again later.

```
[COMMAND]
command = AddCertificate
apiversion = 2
techcontact0 = P-ABC123
class = ssl123dv
dcvmethod = dns-txt
domain0 = example.com
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
csr26 = -----END CERTIFICATE REQUEST-----
dcvtoken=<token>
```

9.1.21 Getting information about a pending certificate (with DNS validation)

```
[COMMAND]
command = StatusCertificate
apiversion = 2
certificate = DD12345678

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.15
queuetime = 0.006
property[ca][0] = DigiCert
property[ca certificate id][0] = 123456780
property[ca order id][0] = 123456789
property[certificate][0] = DD12345678
property[class][0] = SSL123DV
property[created by][0] = <registrar>
property[created date][0] = 2022-01-18 14:20:09.0
property[csr][0] = -----BEGIN CERTIFICATE REQUEST-----
property[csr][1] = MIICnzCCAYcCAQAwWjELMAkGA1UEBhMCREUxDDAKBgNVBAgMA05SVzEWMBQGA1UE
[...]
property[csr][16] = -----END CERTIFICATE REQUEST-----
property[dcv domain][0] = example.com
property[dcv domain][1] = example.com
property[dcv method][0] = dns-cname
property[dcv method][1] = dns-cname
property[dns rr][0] = m278fkyc6xmxyf1x2zwh6st7ch0866f9.example.com. IN CNAME dcv.digicert.com
property[dns rr][1] = m278fkyc6xmxyf1x2zwh6st7ch0866f9.www.example.com. IN CNAME dcv.digicert
    .com
property[domain][0] = example.com
property[domain][1] = www.example.com
property[expiration date][0] =
property[order][0] = DDO12345678
property[order paid until][0] = 2023-01-18 14:20:09.0
property[server type][0] = apache
property[status][0] = processing
property[techcontact][0] = CC-12345678
property[updated by][0] = <registrar>
property[updated date][0] = 2022-01-18 14:20:09.0
```

9.1.22 Getting information about an issued certificate

```
[COMMAND]
command = StatusCertificate
apiversion = 2
certificate = DK58472664
```

```
[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.008
runtime = 0.141
property[ca][0] = DigiCert
property[ca certificate id][0] = 1234567890
property[ca order id][0] = 123456789
property[cert][0] = -----BEGIN CERTIFICATE-----
property[cert][1] = MIIElTCCA32gAwIBAgIQBlIPwjJO9vPJATGmS3Y5/zANBgkqhkiG9w0BAQsFADBc
[...]
property[cert][26] = -----END CERTIFICATE-----
property[certificate][0] = DK12345678
property[class][0] = SSL123DV
property[created by][0] = <registrar>
property[created date][0] = 2022-03-04 12:10:29.0
property[csr][0] = -----BEGIN CERTIFICATE REQUEST-----
property[csr][1] = MIIC4jCCAcoCAQAwajELMAkGA1UEBhMCREUxDDAKBgNVBAgMA05SVzENMAsGA1UE
[...]
property[csr][17] = -----END CERTIFICATE REQUEST-----
property[dcv domain][0] = example.com
property[dcv domain][1] = www.example.com
property[dcv email][0] = admin@example.com
property[dcv email][1] = admin@example.com
property[dcv method][0] = email
property[dcv method][1] = email
property[domain][0] = example.com
property[domain][1] = www.example.com
property[expiration date][0] = 2022-03-10 23:59:59
property[intermediate][0] = -----BEGIN CERTIFICATE-----
property[intermediate][1] = MIIEiTCCA3GgAwIBAgIQAlqK7xlvfg1sIQSyGuZwKzANBgkqhkiG9w0BAQsFADBh
[...]
property[intermediate][26] = -----END CERTIFICATE-----
property[order][0] = DKO12345678
property[order paid until][0] = 2023-03-04 12:10:29.0
property[root][0] = -----BEGIN CERTIFICATE-----
property[root][1] = MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh
[...]
property[root][21] = -----END CERTIFICATE-----
property[server type][0] = apache
property[status][0] = active
property[techcontact][0] = CC-12345678
property[updated by][0] = <registrar>
property[updated date][0] = 2022-06-28 12:22:42.0
```

9.1.23 Getting a list of all certificates (with details)

```
[COMMAND]
command = QueryCertificateList
apiversion = 2
wide = 1

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.003
queuetime = 0.004
property[certificate][0] = DD12345675
property[status][0] = processing
property[certificateorder][0] = DDO12345675
property[server type][0] = apache
property[created date][0] = 2022-01-18 14:20:09
property[updated date][0] = 2022-01-18 14:20:09
property[expiration date][0] =
property[class][0] = RAPIDSSLDV
property[certificate][1] = DD12345676
```

```
property[status][1] = canceled
property[certificateorder][1] = DD12345676
property[server type][1] = apache
property[created date][1] = 2022-01-18 13:33:32
property[updated date][1] = 2022-01-18 14:07:47
property[expiration date][1] =
property[class][1] = RAPIDSSLDV
property[certificate][2] = DK12345677
property[status][2] = processing
property[certificateorder][2] = DKO12345677
property[server type][2] = apache
property[created date][2] = 2022-06-27 13:05:54
property[updated date][2] = 2022-06-27 13:05:54
property[expiration date][2] =
property[class][2] = SSL123DV
property[column][0] = certificate
property[column][1] = status
property[column][2] = certificateorder
property[column][3] = server type
property[column][4] = created date
property[column][5] = updated date
property[column][6] = expiration date
property[column][7] = class
property[count][0] = 3
property[first][0] = 0
property[last][0] = 2
property[limit][0] = 1000
property[total][0] = 3
```

9.1.24 Renewing a certificate by the default period

```
[COMMAND]
command = RenewCertificate
apiversion = 2
certificate = DK12345678

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 2.912
property[billingcount][0] = 1
property[billingtype][0] = ssl123dv-base
property[ca][0] = DigiCert
property[ca certificate id][0] = 12345678
property[ca order id][0] = 123456789
property[certificate][0] = DK87654321
property[certificateorder][0] = DKO12345678
property[class][0] = SSL123DV
property[created by][0] = <registrar>
property[created date][0] = 2022-07-04 12:10:23.0
property[csr][0] = -----BEGIN CERTIFICATE REQUEST-----
property[csr][1] = MIIErDCCApQCAQAwZzELMAkGA1UEBhMCREUxETAPBgNVBAgMCFNhYXJsYW5kMRQw
[...]
property[csr][26] = -----END CERTIFICATE REQUEST-----
property[dcv domain][0] = example.com
property[dcv domain][1] = example.com
property[dcv method][0] = dns-txt
property[dcv method][1] = dns-txt
property[dns rr][0] = @ 3600 IN TXT vrw18t0fhr4vxrnvl1jmw1wswnkyfmyt
property[dns rr][1] = @ 3600 IN TXT vrw18t0fhr4vxrnvl1jmw1wswnkyfmyt
property[domain][0] = example.com
property[domain][1] = www.example.com
property[order paid until][0] = 2023-07-04 12:10:23.0
property[server type][0] = apache
property[status][0] = processing
property[techcontact][0] = CC-12345678
property[updated by][0] = <registrar>
```

```
property[updated date][0] = 2022-07-04 12:10:23.0
```

9.1.25 Reissuing a certificate with a new key

Reissung a certificate with a new key may result in re-validation of the domain(s). The command will return a new certificate id, but will attach the new certificate to the same CertificateOrder.

```
[COMMAND]
command = ReissueCertificate
apiversion = 2
certificate = DK12345678
csr0 = -----BEGIN CERTIFICATE REQUEST-----
csr1 = MIICnzCCAYcCAQAwWjELMAkGA1UEBhMCREUxDDAKBgNVBAgMA05SVzEWMBQGA1UE
[...]
csr16 = -----END CERTIFICATE REQUEST-----

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.008
runtime = 1.818
property[ca][0] = DigiCert
property[ca certificate id][0] = 123456789
property[ca order id][0] = 1234567890
property[certificate][0] = DK87654321
property[class][0] = SSL123DV
property[created by][0] = <registrar>
property[created date][0] = 2022-07-01 13:05:07.0
property[csr][0] = -----BEGIN CERTIFICATE REQUEST-----
property[csr][1] = MIICnzCCAYcCAQAwWjELMAkGA1UEBhMCREUxDDAKBgNVBAgMA05SVzEWMBQGA1UE
[...]
property[csr][16] = -----END CERTIFICATE REQUEST-----
property[dcv domain][0] = example.com
property[dcv domain][1] = example.com
property[dcv method][0] = dns-txt
property[dcv method][1] = dns-txt
property[dns rr][0] = 7gtszcs8r1sk5b9htlj50w39ypqh6p89
property[dns rr][1] = @ 3600 IN TXT z687qy852c2zwd2dp39cqcdrxpn73qyv
property[domain][0] = example.com
property[domain][1] = www.example.com
property[order][0] = DKO12345678
property[order paid until][0] = 2023-07-01 12:36:51.0
property[server type][0] = apache
property[status][0] = processing
property[techcontact][0] = CC-12345678
property[updated by][0] = <registrar>
property[updated date][0] = 2022-07-01 13:05:07.0
```

9.1.26 Revoke a certificate

```
[COMMAND]
command = RevokeCertificate
apiversion = 2
certificate = DK12345678
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 2.121
queuetime = 0.016
```

or

```
[COMMAND]
command = DeleteCertificate
apiversion = 2
certificate = DK12345678
action = revoke
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 2.121
queuetime = 0.016
```

9.1.27 Triggering DNS / HTTP validation

If the DNS or HTTP entry was not entered correctly at first, a re - validation can be triggered by calling ResendNotification:

```
[COMMAND]
command = ResendNotification
type = CERTIFICATE
object = D12345678

[RESPONSE]
code = 200
description = Command completed successfully
queuetime = 0.361
runtime = 3.751
```

9.1.28 Revoke a CertificateOrder

```
[COMMAND]
command = RevokeCertificateOrder
apiversion = 2
certificateorder = DKO12345678
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 5.128
queuetime = 0.016
```

or via the associated Certificate - ID:

```
[COMMAND]
command = DeleteCertificate
apiversion = 2
certificate = DK12345678
action = revokeorder
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 5.128
queuetime = 0.016
```

9.1.29 Cancel a CertificateOrder

```
[COMMAND]
```

```
command = CancelCertificateOrder
apiversion = 2
certificateorder = DKO12345678
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 3.768
queuetime = 0.043
```

or via the associated Certificate - ID:

```
[COMMAND]
command = DeleteCertificate
apiversion = 2
certificate = DK12345678
action = cancelorder
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 3.768
queuetime = 0.043
```

9.1.30 Get details about a CertificateOrder

```
[COMMAND]
command = StatusCertificateOrder
apiversion = 2
certificateorder = DDO12345678
EOF

[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.116
queuetime = 0.005
property[ca][0] = DigiCert
property[certificate][0] = DD12345678
property[certificate][1] = DD23456789
property[certificateorder][0] = DDO12345678
property[certificate expiration date][0] = 2022-01-18 13:33:32.0
property[certificate expiration date][0] =
property[certificate status][0] = expired
property[certificate status][1] = processing
property[class][0] = RapidSSL
property[created by][0] = <registrar>
property[created date][0] = 2021-01-18 13:33:32.0
property[paid until][0] = 2023-01-18 13:33:32.0
property[status][0] = active
property[updated by][0] = <registrar>
property[updated date][0] = 2022-01-18 14:07:47.0
```

9.1.31 Get a list of CertificateOrders with details

```
[COMMAND]
command = QueryCertificateOrderList
apiversion = 2
wide = 1
limit = 2
EOF
```

```
[RESPONSE]
code = 200
description = Command completed successfully
runtime = 0.003
queuetime = 0.006
property[certificateorder][0] = DDO12345678
property[paid until][0] = 2023-01-18 13:33:32
property[status][0] = processing
property[class][0] = RAPIDSSLDV
property[created date][0] = 2022-01-18 13:33:32
property[updated date][0] = 2022-01-18 14:07:47
property[certificateorder][1] = DDO23456789
property[paid until][1] = 2023-01-18 14:20:09
property[status][1] = processing
property[class][1] = RAPIDSSLDV
property[created date][1] = 2022-01-18 14:20:09
property[updated date][1] = 2022-07-01 12:39:00
property[column][0] = certificateorder
property[column][1] = paid until
property[column][2] = status
property[column][3] = class
property[column][4] = created date
property[column][5] = updated date
property[count][0] = 2
property[first][0] = 0
property[last][0] = 1
property[limit][0] = 2
property[total][0] = 7
```

9.2 List of server types

9.2.1 DigiCert server types

Server Type	Description
apache	Apache
barracuda	Barracuda
weblogic	BEA Weblogic 8 & 9
cisco	Cisco
citrix	Citrix (Other)
cpanel	cPanel
f5	F5 FirePass
ibm	IBM HTTP Server
java	Java Web Server (Javasoft / Sun)
lighttpd	Lighttpd
lotus	Lotus Domino
macos	Mac OS X Server
exchange.*2007	Microsoft Exchange Server 2007
exchange.*2010	Microsoft Exchange Server 2010
exchange.*2013	Microsoft Exchange Server 2013
exchange.*2016	Microsoft Exchange Server 2016
forefront	Microsoft Forefront Unified Access Gateway
iis56	Microsoft IIS 5 or 6
iis7	Microsoft IIS 7
iis8	Microsoft IIS 8
iis10	Microsoft IIS 10

Server Type	Description
netscape	Netscape Enterprise Server
iplanet	Netscape iPlanet
nginx	nginx
novellichain	Novell iChain
novellnetware	Novell NetWare
oracle	Oracle
qmail	Qmail
sunone	SunOne
tomcat	Tomcat
webstar	WebStar
zeus	Zeus Web Server
other	Other

9.2.2 Sectigo server types

Server Type	Description
apache	Apache
cpanel	cPanel
java	Java Web Server (Javasoft / Sun)
iis56	Microsoft IIS 5 or 6
iis7	Microsoft IIS 7
nginx	nginx
oracle	Oracle
other	Other

9.3 CertificateClasses / Product Matrix

9.3.1 List of available classes

Class	CA	Type	Marketed Name
GEOTRUSTFLEXDV	DigiCert	dv	GeoTrust Flex
GEOTRUSTTRUEBIZIDEV	DigiCert	ev	GeoTrust TrueBusiness ID EV
GEOTRUSTTRUEBIZIDOV	DigiCert	ov	GeoTrust TrueBusiness ID OV
RAPIDSSLDV	DigiCert	dv	RapidSSL
SECURESITEFLEXEV	DigiCert	ev	Secure Site Flex EV SSL
SECURESITEFLEXOV	DigiCert	ov	Secure Site Flex SSL
SECURESITEPROEV	DigiCert	dv	SecureSite Pro EV
SECURESITEPROOV	DigiCert	ov	Secure Site Pro SSL
SSL123DV	DigiCert	dv	SSL123DV
WEBSERVEREV	DigiCert	ev	Webserver EV
WEBSERVEROV	DigiCert	ov	Webserver
INSTANTSSLOV	Sectigo	ov	InstantSSL Certificate

Class	CA	Type	Marketed Name
MULTIDOMAINDV	Sectigo	dv	DV Multi-Domain SSL Certificate
MULTIDOMAINEV	Sectigo	ev	EV Multi-Domain SSL Certificate
MULTIDOMAINOV	Sectigo	ov	OV Multi-Domain SSL Certificate
POSITIVESSLDV	Sectigo	dv	PositiveSSL Certificate
PREMIUMOV	Sectigo	ov	SectigoSSL PremiumOV Certificate
SINGLEDOMAINEV	Sectigo	ev	Sectigo SingleDomain EV Certificate
UNIFIEDCOMMUNICATIONSDV	Sectigo	dv	SSL Unified Communications Certificate
UNIFIEDCOMMUNICATIONSOV	Sectigo	ov	SSL Unified Communications OV Certificate

9.3.2 Supported domain types per class

Class	Wildcard Supported	SAN supported	included SAN	Free alternatives
GEOTRUSTFLEXDV	1	1	1	250
GEOTRUSTTRUEBIZIDEV	0	1	1	250
GEOTRUSTTRUEBIZIDOV	1	1	1	250
RAPIDSSLDV	1	0	1	250
SECURESITEFLEXEV	0	1	1	250
SECURESITEFLEXOV	1	1	1	250
SECURESITEPROEV	0	1	1	250
SECURESITEPROOV	1	1	1	250
SSL123DV	1	1	1	3
WEBSERVEREV	0	1	1	250
WEBSERVEROV	1	1	1	250
INSTANTSSLOV	1	0	1	1
MULTIDOMAINDV	1	1	3	0
MULTIDOMAINEV	0	1	3	0
MULTIDOMAINOV	1	1	3	0
POSITIVESSLDV	1	0	1	1
PREMIUMOV	0	0	1	1
SINGLEDOMAINEV	0	0	1	1
UNIFIEDCOMMUNICATIONSDV	1	1	3	0
UNIFIEDCOMMUNICATIONSOV	1	1	3	0

9.3.3 Supported DCV methods per class

Class	Email	DNS-CNAME	DNS-TXT	HTTP	HTTPS
GEOTRUSTFLEXDV	1	1	1	1	1
GEOTRUSTTRUEBIZIDEV	1	1	1	1	1
GEOTRUSTTRUEBIZIDOV	1	1	1	1	1
RAPIDSSLDV	1	1	1	1	1
SECURESITEFLEXEV	1	1	1	1	1
SECURESITEFLEXOV	1	1	1	1	1
SECURESITEPROEV	1	1	1	1	1
SECURESITEPROOV	1	1	1	1	1
SSL123DV	1	1	1	1	1
WEBSERVEREV	1	1	1	1	1
WEBSERVEROV	1	1	1	1	1
INSTANTSSLOV	1	1	0	1	1
MULTIDOMAINDV	1	1	0	1	1
MULTIDOMAINEV	1	1	0	1	1
MULTIDOMAINOV	1	1	0	1	1
POSITIVESSLDV	1	1	0	1	1
PREMIUMOV	1	1	0	1	1
SINGLEDOMAINEV	1	1	0	1	1
UNIFIEDCOMMUNICATIONSDV	1	1	0	1	1
UNIFIEDCOMMUNICATIONSOV	1	1	0	1	1

9.3.4 Supported/required contact types per class

Class	Tech	Organization	EV approver
GEOTRUSTFLEXDV	1	0	0
GEOTRUSTTRUEBIZIDEV	0	1	1
GEOTRUSTTRUEBIZIDOV	0	1	0
RAPIDSSLDV	1	0	0
SECURESITEFLEXEV	0	1	1
SECURESITEFLEXOV	0	1	0
SECURESITEPROEV	1	0	0
SECURESITEPROOV	0	1	0
SSL123DV	1	0	0
WEBSERVEREV	1	0	0
WEBSERVEROV	0	1	1
INSTANTSSLOV	1	0	0
MULTIDOMAINDV	1	0	0
MULTIDOMAINEV	1	0	0
MULTIDOMAINOV	1	0	0
POSITIVESSLDV	1	0	0
PREMIUMOV	1	0	0
SINGLEDOMAINEV	1	0	0
UNIFIEDCOMMUNICATIONSDV	1	0	0
UNIFIEDCOMMUNICATIONSOV	1	0	0

9.3.5 Supported features per class

Class	Available periods	Refund on cancel	Immediate issuance
GEOTRUSTFLEXDV	1y	1	1
GEOTRUSTTRUEBIZIDEV	1y	1	0
GEOTRUSTTRUEBIZIDOV	1y	1	0
RAPIDSSLDV	1y	1	1
SECURESITEFLEXEV	1y	1	0
SECURESITEFLEXOV	1y	1	0
SECURESITEPROEV	1y	1	1
SECURESITEPROOV	1y	1	0
SSL123DV	1y	1	1
WEBSERVEREV	1y	1	1
WEBSERVEROV	1y	1	0
INSTANTSSLOV	1y	0	0
MULTIDOMAINDV	1y	0	0
MULTIDOMAINEV	1y	0	0
MULTIDOMAINOV	1y	0	0
POSITIVESSLDV	1y	0	0
PREMIUMOV	1y	0	0
SINGLEDOMAINEV	1y	0	0
UNIFIEDCOMMUNICATIONSDV	1y	0	0
UNIFIEDCOMMUNICATIONSOV	1y	0	0

9.4 List of possible object statuses

9.4.1 Statuses for Certificates

Status	Description
active	The certificate is valid
processing	The certificate was ordered, but not yet issued (see events)
expired	The certificate is expired and no longer valid
revoked	The certificate was revoked and is no longer valid
failed	The issuance of the certificate failed
canceled	The CertificateOrder was canceled and it was never issued
cancel_processing	A cancellation for the CertificateOrder was requested (see events)
revoke_processing	A revocation for the certificate was requested (see events)

9.4.2 Statuses for CertificateOrders

Status	Description
active	The CertificateOrder has a valid period and had at least one issued certificate
processing	The CertificateOrder was submitted to the CA and a (re-)issuance is currently in progress (see events)
expired	The maximum validity date of the CertificateOrder is in the past
failed	No successful certificate could be issued for this CertificateOrder. Re-Issues are not possible.
canceled	The CertificateOrder and all attached certificates were canceled. No certificate was issued.
cancel_processing	A cancellation for the CertificateOrder and all attached certificates was requested and is currently pending (see events)

9.5 List of events

Class	Subclass	Description
CERTIFICATE_REQUEST	REQUEST_SUCCESSFUL	The CA issued the certificate and it is now available via StatusCertificate
CERTIFICATE_REQUEST	REQUEST_FAILED	The certificate could not be issued
CERTIFICATE_REISSUE	REISSUE_SUCCESSFUL	The reissue was done and the new certificate is available via StatusCertificate
CERTIFICATE_REISSUE	REISSUE_FAILED	The certificate could not be reissued
CERTIFICATE_CLOSE	CLOSE_SUCCESSFUL	The certificate was successfully cancelled or revoked

9.6 Glossary

Term	Definition
CA	A certificate authority or certification authority is a trusted third party company that issues digital certificates and public-private keys as a part of chosen Public Key Infrastructure (PKI). In order to issue these certificates, a CA first consults with a registration authority (RA) such as credit card company to check whether the requester's information is legitimate. Only after the proper verification can the CA issue a certificate claiming that the organization or the individual is the one it claims to be. Having a digital certificate on a website proves the owner's identity, hence developing a trustworthy environment in business.

Term	Definition
CA/B	The Certificate Authority/Browser (CA/B) Forum is the standards-setting body that collaborates on aspects of website security. Composed of about 50 Certificate Authority (CA) and nine browser members. The Forum produces standards, called Baseline Requirements, which all public CAs, whether members of the Forum or not, must adhere to.
CRL	A method to inform user agents about the revocation status of a certificate. This is a list of the serial numbers of all revoked certificates from a given CA, signed by that CA.
CRT	The .crt extension is a security certificate file that is used by secure websites to establish secure connections from web server to a browser. Secure websites make it possible to secure data transfers, logins, payment card transactions, and provide protected browsing to the site. CRT files are in ASCII format and can be opened in any text editor to view the contents of the certificate file. It follows the X.509 certification standard that defines the structure of the certificate. It defines the data fields that should be included in the SSL certificate. CRT belongs to the PEM format of certificates that are Base64 ASCII encoded files.
CSR	The Certificate Signing Request (CSR) is needed to order a TLS/SSL certificate. It must to be created on and encrypted by the web server where the certificate shall be installed. The CSR provides information about the domains that will be covered.
DNS	A Domain Name System server contains a registry that maps each human-readable domain name (e.g. www.example.com) to its IP address (e.g. 123.123.123.123), thus enabling visitors to access the relevant site from anywhere in the world.
FQDN	A fully-qualified domain name (FQDN) is that portion of an Internet Uniform Resource Locator (URL) that fully identifies the server program that an Internet request is addressed to. The FQDN consists of two parts: the hostname and the domain name. For example, an FQDN for a hypothetical mail server might be mymail.somecollege.edu . The hostname is mymail , and the host is located within the domain somecollege.edu.
PEM	The .pem file format is mostly used to store cryptographic keys. This file can be used for different purposes. The .pem file defines the structure and encoding file type that is used to store the data. The pem file contains the standard dictated format to start and end a file. Using the pem files, you can store the SSL certificates with their associated private keys.

Term	Definition
Private Key	The key that a user keeps secret in asymmetric encryption. It can encrypt or decrypt data for a single transaction but cannot do both.
Public Key	The key that a user allows the world to know in asymmetric encryption. It can encrypt or decrypt data for a single transaction but cannot do both.
OCSP	Online Certificate Status Protocol is a method to check the revocation status of a certificate. In other words, a way to check whether a Certificate Authority indicates that the certificate should no longer be considered valid, even though its expiration date has not yet been reached. This request can create privacy problems because it allows the certificate authority, and Internet service providers, to directly observe who is visiting which sites.
SAN	When ordering certificates you can choose if you want to include subject alternative names (multiple domains) or not. The functionality offers TLS-secured communications for servers using multiple domain names and host names within a single certificate. Certificates with SAN support provide a Subject Alternative Name (SAN) field that allows additional domain names to be protected with just one certificate. If it is supported by the certificate type you choose, your certificate can cover example.org as well as the fully qualified domain name www.example.org for free. It will be added automatically for supporting classes, but you are free to specify it as well.
TTL	DNS TTL (time to live) is a setting that tells the DNS resolver how long to cache a query before requesting a new one. The information gathered is then stored in the cache of the recursive or local resolver for the TTL before it reaches back out to collect new, updated details.
URI	A Uniform Resource Identifier is a unique sequence of characters that identifies a logical or physical resource used by web technologies. URIs may be used to identify anything, including real-world objects, such as people and places, concepts, or information resources such as web pages and books.

Term	Definition
X.509	An X.509 certificate is a digital certificate that uses the widely accepted international X. 509 public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate. Difference between SSL and x509 certificate: 509 certificates contain a public key and the identity of a hostname, organization, or individual. The SSL/TLS certificate fulfills two functions as machine identities: Authentication and Data Encryption.

9.7 Reference

This section of the documentation contains references.

- **Cheap SSL Security**. 2022. What is SSL Certificate Chain. [Online] Available at: https://cheapsslsecurity.com/p/what-is-ssl-certificate-chain
- **IBM**. 2022. IBM Documentation. [Online] Available at: https://www.ibm.com/docs/en/ibm-mq/8.0?topic=certificates-how-certificate-chains-work
- **Digicert**. 2022. Certificate Transparency. [Online] Available at: https://www.digicert.com/faq/certificate-transparency/overview.htm
- **Comodo SSL Resources**. 2022. Glossary. [Online] Available at: https://comodosslstore.com/uk/support/glossary.aspx
- **Comodo SSL Resources**. 2022. SSL Basics. [Online] Available at: https://comodosslstore.com/resources/how-to-fix-neterr_certificate_transparency_required-error-in-google-chrome/
- **ICANN Wiki**. 2022. CA. [Online] Available at : https://icannwiki.org/Certificate_authority
- **Wikipedia**. 2022. CRL. [Online] Available at https://en.wikipedia.org/wiki/Certificate_revocation_list
- **Wikipedia**. 2022. PEM. [Online] Available at https://en.wikipedia.org/wiki/Privacy-Enhanced_Mail
- **Wikipedia**. 2022. OCSP. [Online] Available at https://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol
- **Wikipedia**. 2022. SAN. [Online] Available at https://en.wikipedia.org/wiki/Subject_Alternative_Name
- **Wikipedia**. 2022. URI. [Online] Available at https://en.wikipedia.org/wiki/Uniform_Resource_Identifier
- **Wikipedia**. 2022. X.509. [Online] Available at: https://en.wikipedia.org/wiki/X.509