## Contents

1. DOCUMENT HISTORY  
2. OVERVIEW  
3. SSL PRODUCT TYPE EXPLANATIONS  
   3.1 ALPHASSL  
   3.2 DOMAINSSL  
   3.3 ORGANIZATIONSSL  
   3.4 EXTENDED SSL  
   3.5 SUMMARY OF KEY SSL PRODUCT FEATURES  
   3.6 HIGH RISK CHECKS  
   3.7 WEB SERVICE FUNCTIONS - WORKFLOW OVERVIEW  
   3.8 SSL FUNCTIONS  
   3.9 SERVICE/QUERY FUNCTIONS  
   3.10 ACCOUNT FUNCTIONS  
   3.11 DELIVERY OF ISSUED CERTIFICATES BY EMAIL  
   3.12 WSDL & API URLS  
4. ORDERING DOMAINSSL & ALPHA SSL CERTIFICATES  
   4.1 USING EMAIL VALIDATION  
      4.1.1 DORDER REQUEST  
      4.1.2 DORDER RESPONSE  
   4.2 USING HTTP (URL) VERIFICATION  
      4.2.1 URL VERIFICATION ORDER REQUEST  
      4.2.2 URL VERIFICATION ORDER RESPONSE  
      4.2.3 URL VERIFICATION FOR ISSUE REQUEST  
      4.2.4 URL VERIFICATION FOR ISSUE RESPONSE  
   4.3 USING DNS VERIFICATION  
      4.3.1 DDNSORDER REQUEST  
      4.3.2 DDNS ORDER RESPONSE  
      4.3.3 DDNS ORDER FOR ISSUE REQUEST  
      4.3.4 DDNS ORDER FOR ISSUE RESPONSE  
5. ORDERING ORGANIZATIONSSL CERTIFICATES  
   5.1 DECODECSR  
      5.1.1 DECODECSR REQUEST  
      5.1.2 DECODECSR RESPONSE  
   5.2 ORDERING THE ORGANIZATION SSL CERTIFICATE  
      5.2.1 OORDER REQUEST  
      5.2.2 OORDER RESPONSE  
6. ORDERING EXTENDEDSSL CERTIFICATES  
   6.1 DECODECSR  
      6.1.1 DECODECSR REQUEST  
      6.1.2 DECODECSR RESPONSE  
   6.2 ORDERING THE EXTENDEDSSL CERTIFICATE  
   
GlobalSign API for SSL Certificates v4.5
7. DOMAIN VALIDATION

8. GENERAL SSL FUNCTIONS

8.1 RECEIVE LIST OF APPROVER EMAIL ADDRESSES AND PERFORM PHISHING DB CHECK

8.1.1 GETDVAPPROVERLIST REQUEST
8.1.2 GETDVAPPROVERLIST RESPONSE

8.2 CHANGE THE SUBJECTALTNAME

8.2.1 DOMAINSSL
8.2.2 ORGANIZATIONSSL, EXTENDEDSSL
8.2.3 CHANGESUBJECTALTNAME REQUEST
8.2.4 CHANGESUBJECTALTNAME RESPONSE

8.3 CREATE CERTIFICATE INVITES

8.3.1 CERTINVITEORDER REQUEST
8.3.2 CERTINVITEORDER RESPONSE

8.4 CHANGE APPROVER EMAIL

8.4.1 CHANGEAPPROVEREMAIL REQUEST
8.4.2 CHANGEAPPROVEREMAIL RESPONSE

8.5 RE-SEND APPROVER EMAIL

8.5.1 RESENDEMAIL REQUEST
8.5.2 RESENDEMAIL RESPONSE

8.6 MODIFY EXISTING ORDER

8.6.1 MODIFYORDER REQUEST
8.6.2 MODIFYORDER RESPONSE

9. SERVICE & QUERY APIs

9.1 DECODECSR

9.1.1 DECODECSR REQUEST
9.1.2 DECODECSR RESPONSE

9.2 GET ISSUED CERTIFICATE - SINGLE CERTIFICATE

9.2.1 GETORDERBYORDERID REQUEST
9.2.2 GETORDERBYORDERID RESPONSE

9.3 GET ISSUED CERTIFICATES - MULTIPLE ORDERS

9.3.1 GETORDERBYDATE RANGE REQUEST
9.3.2 GETORDERBYDATERANGE RESPONSE

9.4 GET RECENTLY MODIFIED ORDERS

9.4.1 GETMODIFIEDORDERS REQUEST
9.4.2 GETMODIFIEDORDERS RESPONSE

9.5 DETERMINE UPCOMING RENEWALS

9.5.1 GETORDERBYEXPIRATIONDATE REQUEST
9.5.2 GETORDERBYEXPIRATIONDATE RESPONSE

9.6 GET CERTIFICATE ORDERS

9.6.1 GETCERTIFICATEORDERS REQUEST
9.6.2 GETCERTIFICATEORDERS RESPONSE

9.7 REISSUE CERTIFICATES

9.7.1 REISSUE REQUEST
9.7.2 REISSUE RESPONSE

9.8 VALIDATE ORDER INFORMATION

9.8.1 VALIDATEORDERPARAMETERS REQUEST
9.8.2 VALIDATEORDERPARAMETERS RESPONSE

9.9 TURN ON/OFF RENEWAL NOTICES

19

GlobalSign API for SSL Certificates v4.5
GlobalSign API for SSL Certificates v4.5
1. Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Release Date</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
</table>
| 4.3.4   | 06/02/2014   | Doug Beattie  | • Replaced ValidateOrderParameters with DecodeCSR in recommended workflow  
• Added reference to hash algorithm specific product codes in addendum  
• Added reference to validity period constraints in addendum  
• Removed mention of obsolete TestOV product code  
• Updated BaseOptions to include correct GlobalIP syntax |
| 4.3.5   | 08/26/2016   | Doug Beattie  | • Identified new location for Random Value (meta-tag) for HTTP Domain Validation, per CA/B Forum Ballot 169  
• Reorganized sections for readability and flow, added expanded descriptions of the ordering process and ordering options for each of the products.  
• Identified new test system URLs  
• Removed 4 and 5 year validity period options  
• Removed SHA-1 references as SHA-1 has been deprecated  
• Replaced list of ModificationEventName responses with new values  
• Updated list of error codes and descriptions  
• Clarified the definition of Custom Validity Period.  
• Added new API field to request specific values for KeyUsage and Extended Key Usage |
| 4.3.6   | 12/12/2016   | Doug Beattie  | • Clarified Various minor updates  
• More clearly documented the new location for HTTP Domain Verification in Section 11.2 |
| 4.3.7   | 04/23/2018   | Julie Olson   | • Removed references to AutoCSR.  
• Changed validity period maximums to 825 days where appropriate.  
• Removed non-working APIs: GetAccountSnapshot, ResellerApplication, and AddResellerDeposit  
• Added new Error Code -6035 |
| 4.4     | 06/05/2019   | Julie Olson   | • Added new Document History table  
• Added notes to APIs that have known bugs  
• Removed pictures  
• Changed custom certificate validity period to custom certificate expiration date |
| 4.5     | 02/18/2020   | Julie Olson   | • Added new OV/EV domain validation APIs and process |
2. Overview

GlobalSign offers a Simple Object Access Protocol (SOAP) API for its partners and customers to directly order and manage certificates. Through this API, partners can perform functions such as ordering different products, cancelling and fulfilling orders, and querying for order data. The API supports applications for SSL Certificates placed by partners and by customers using the SSL Managed service platform (MSSL). Partners may place orders for all certificate product types.

If you have any questions about any of the information provided in this document, please contact your Account Manager.

3. SSL Product Type Explanations

3.1 AlphaSSL

AlphaSSL is a low cost domain validated certificate with separate branding from the rest of GlobalSign’s SSL product line. This product can only be purchased in standard or wildcard options with up to 825-day validity period. No value add options, such as SANs, are supported with this product. When placing an AlphaSSL order, the applicant must supply a CSR. Certificates requested by supplying a customer-generated CSR are returned as standard certificate files.

Note: In the API product code specification, AlphaSSL is referenced as **DV_LOW**.

3.2 DomainSSL

DomainSSL is a domain validated certificate from GlobalSign. DomainSSL supports both standard and wildcard configurations, as well as specific SAN options with up to 825-day validity period. SAN options include:

- Free Unified Communications support for www, owa, autodiscover, and mail
- Additional sub-domains

When placing a DomainSSL order, the applicant must supply a CSR. Certificates requested by supplying a customer-generated CSR are returned as standard certificate files.

Note: In the API product code specification, DomainSSL is referenced as **DV**.

3.3 OrganizationSSL

OrganizationSSL is a high assurance organization validated certificate. OrganizationSSL supports both standard and wildcard configurations, as well as specific SAN options with up to 825-day validity period. SAN options include:

- Free Unified Communications support for www, owa, autodiscover, and mail
- Additional domains
- Additional sub-domains
- Public IP Addresses

When placing an OrganizationSSL order, the applicant must supply a CSR. Certificates requested by supplying a customer-generated CSR are returned as standard certificate files.

Note: In the API product code specification, OrganizationSSL is referenced as **OV**.
3.4 Extended SSL

ExtendedSSL, or Extended Validation (EV) SSL, is the highest assurance SSL certificate available. EV SSL supports standard SSL configurations with up to 825-day validity period. No wildcard option is available with this product. The following SAN options are available with ExtendedSSL:

- Free Unified Communications support for www, owa, autodiscover, and mail
- Additional domains.
- Additional sub-domains

Note: In the API product code specification, ExtendedSSL is referenced as EV.

3.5 Summary of Key SSL Product Features

The below table provides a summary of the key features available for the suite of GlobalSign SSL products.

<table>
<thead>
<tr>
<th>Function</th>
<th>EV</th>
<th>OV</th>
<th>DV</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Options</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wildcard</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Global IP</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Private</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Validity Period in Request</td>
<td>825 days</td>
<td>825 days</td>
<td>825 days</td>
<td>825 days</td>
</tr>
<tr>
<td>Maximum Cert Validity Period (months)</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>Extra Renewal Month</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Order Options (OptionName)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAN Option</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Renewal Extension Option</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Validity Period Customize Option</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Order Kind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Renewal</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Transfer</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SAN Types:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unified Communications</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>FQDN</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>SubDomain</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>GlobalIP Address</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Wildcard</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Internal SAN or Reserved IP address</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Extensions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended Key Usage</td>
<td>Future</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Key Usage</td>
<td>Future</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>TLS Feature Extension</td>
<td>Future</td>
<td>Future</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CSR Key Types supported</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSA 2048-8192</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ECC P-256</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ECC P-384</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Post issuance:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change SANs (Add/Delete)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Cancel in 7 days</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Revoke</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Reissue</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Site Seal</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
3.6 High Risk Checks

All domain validated certificates (DomainSSL and AlphaSSL) automatically go through GlobalSign’s anti-phishing checks. These checks involve a series of automated processes to help identify potential phishing risks.

During the vetting process for OV/EV orders, if a certificate is flagged as high risk (“-1” success code), the certificate will not be issued until manually reviewed by a GlobalSign Vetting Agent. This will create some delay in certificate issuance; to check the status of an order users should use the GetOrderByOrderID command (refer to Section 9.2). Any order that is flagged for phishing/high risk will trigger an appropriate alert message, and a vetting agent is assigned to review the order. For more information about Success/Error Codes, refer to Section 12.4.

All other types of orders have the same anti-phishing checks performed during the manual vetting process.

3.7 Web Service Functions - Workflow Overview

Order processing for SSL Certificates and web identity products is asynchronous. For these types of orders, an API client places an order, and then checks the server for the completed order. The functions are organized into a few different categories:

- **SSL Functions**: Calls to place, modify, or cancel orders
- **Service & Query Functions**: Calls that search for complete orders (such as “getting” issued certificates), decode CSRs, validate order parameters
- **Account Functions**: Calls needed to perform account actions, such as checking balance and modified sub-accounts

The general approach for ordering a certificate is to place orders using an SSL function, and then periodically request a list of all orders that have changed status during a specified time interval (for example, the last four hours) using the **Service/Query function GetModifiedOrders**. This returns a list of all orders and detailed information for orders that have changed status in the specified time interval. The status of all returned orders can then be updated locally and used as necessary.

An alternative to querying for a set of modified orders within a time period is to request the status of a specific order. In this case the ordering flow consists of placing an order, and then periodically checking the status of the order by using the **GetOrderByOrderID** function. Once the order has been completed, the fulfillment information is returned with the **GetOrderByOrderID** operation. This approach is less efficient, but might be more appropriate when there is a low volume of certificates being managed.
3.8 SSL Functions
The following table lists API requests that users can use for basic SSL functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>API Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get list of approver email addresses</td>
<td>GetApproverList</td>
</tr>
<tr>
<td>Get list of approver email addresses and OrderID for DVOrder (DomainSSL and AlphaSSL only)</td>
<td>GetDVApproverList</td>
</tr>
<tr>
<td>Order AlphaSSL or DomainSSL Certificate with Approver Email validation</td>
<td>DVOrderDVOrderWithoutCSR</td>
</tr>
<tr>
<td>Order AlphaSSL or DomainSSL Certificate with Metatag validation</td>
<td>URLVerification &amp; URLVerificationForIssue</td>
</tr>
<tr>
<td>Order AlphaSSL or DomainSSL Certificate with DNS validation</td>
<td>DVDNS</td>
</tr>
<tr>
<td>Order OrganizationSSL Certificate</td>
<td>OVOOrderOVOOrderWithoutCSR</td>
</tr>
<tr>
<td>Order ExtendedSSL Certificate</td>
<td>EVOOrder</td>
</tr>
<tr>
<td>Change certificate order status</td>
<td>ModifyOrder</td>
</tr>
<tr>
<td>Resend Approver Emails for AlphaSSL &amp; DomainSSL orders</td>
<td>ResendEmail</td>
</tr>
<tr>
<td>Place an order using the cert invite functionality</td>
<td>CertInviteOrder</td>
</tr>
<tr>
<td>Change the email address that the approval request is sent to for domain validated products</td>
<td>ChangeApproverEmail</td>
</tr>
<tr>
<td>Change the SubjectAltName in certificate.</td>
<td>ChangeSubjectAltName</td>
</tr>
</tbody>
</table>

3.9 Service/Query Functions
The following table lists API requests used for basic Service/Query functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>API Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decode a CSR</td>
<td>DecodeCSR</td>
</tr>
<tr>
<td>Search order information by Order ID</td>
<td>GetOrderByOrderID</td>
</tr>
<tr>
<td>Search modified orders by modified date (from/to)</td>
<td>GetModifiedOrders</td>
</tr>
<tr>
<td>Get order list</td>
<td>GetCertificateOrders</td>
</tr>
<tr>
<td>Search orders by order date (from/to)</td>
<td>GetOrderByDateRange</td>
</tr>
<tr>
<td>Check upcoming expirations</td>
<td>GetOrderByExpirationDate</td>
</tr>
<tr>
<td>Reissue Certificate</td>
<td>ReIssue</td>
</tr>
<tr>
<td>Turn on/off Renewal notice</td>
<td>ToggleRenewalNotice</td>
</tr>
<tr>
<td>Check order parameter validity</td>
<td>ValidateOrderParameters</td>
</tr>
</tbody>
</table>

3.10 Account Functions
There are two API requests used for basic Account functions: AccountSnapshot is used to view account balances and recent usage, and QueryInvoices is used to view outstanding invoices.
3.11 Delivery of Issued Certificates by Email

Issued certificates can be delivered directly to the customer. In the DVOrder/OVOrder/EVOrder Request, specify the end customer and their email address in the <ContactInfo> field.

**Note:** To directly email the end customer, your account must be configured on a specific template group. Contact your Account Manager or Tech Implementation Contact for more information.

3.12 WSDL & API URLs

**WSDL URLs**

<table>
<thead>
<tr>
<th>Feature</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PROD</strong></td>
<td></td>
</tr>
<tr>
<td>SSL Functions WSDL</td>
<td><a href="https://system.globalsign.com/kb/ws/v1/ServerSSLService?wsdl">https://system.globalsign.com/kb/ws/v1/ServerSSLService?wsdl</a></td>
</tr>
<tr>
<td>Service / Query WSDL</td>
<td><a href="https://system.globalsign.com/kb/ws/v1/GASService?wsdl">https://system.globalsign.com/kb/ws/v1/GASService?wsdl</a></td>
</tr>
<tr>
<td>Account WSDL</td>
<td><a href="https://system.globalsign.com/kb/ws/v1/AccountService?wsdl">https://system.globalsign.com/kb/ws/v1/AccountService?wsdl</a></td>
</tr>
<tr>
<td>Subscriber Agreement WSDL</td>
<td><a href="https://system.globalsign.com/qb/ws/GasQuery?wsdl">https://system.globalsign.com/qb/ws/GasQuery?wsdl</a></td>
</tr>
<tr>
<td><strong>TEST</strong></td>
<td></td>
</tr>
<tr>
<td>SSL Functions WSDL</td>
<td><a href="https://test-gcc.globalsign.com/kb/ws/v1/ServerSSLService?wsdl">https://test-gcc.globalsign.com/kb/ws/v1/ServerSSLService?wsdl</a></td>
</tr>
<tr>
<td>Service / Query WSDL</td>
<td><a href="https://test-gcc.globalsign.com/kb/ws/v1/GASService?wsdl">https://test-gcc.globalsign.com/kb/ws/v1/GASService?wsdl</a></td>
</tr>
<tr>
<td>Subscriber Agreement WSDL</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**API URLs**

<table>
<thead>
<tr>
<th>Feature</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PROD</strong></td>
<td></td>
</tr>
<tr>
<td>SSL Functions</td>
<td><a href="https://system.globalsign.com/kb/ws/v1/ServerSSLService">https://system.globalsign.com/kb/ws/v1/ServerSSLService</a></td>
</tr>
<tr>
<td>Service / Query</td>
<td><a href="https://system.globalsign.com/kb/ws/v1/GASService">https://system.globalsign.com/kb/ws/v1/GASService</a></td>
</tr>
<tr>
<td>Account</td>
<td><a href="https://system.globalsign.com/kb/ws/v1/AccountService">https://system.globalsign.com/kb/ws/v1/AccountService</a></td>
</tr>
<tr>
<td>Subscriber Agreement</td>
<td><a href="https://system.globalsign.com/qb/ws/GasQuery">https://system.globalsign.com/qb/ws/GasQuery</a></td>
</tr>
<tr>
<td><strong>TEST</strong></td>
<td></td>
</tr>
<tr>
<td>SSL Functions</td>
<td><a href="https://test-gcc.globalsign.com/kb/ws/v1/ServerSSLService">https://test-gcc.globalsign.com/kb/ws/v1/ServerSSLService</a></td>
</tr>
<tr>
<td>Service / Query</td>
<td><a href="https://test-gcc.globalsign.com/kb/ws/v1/GASService">https://test-gcc.globalsign.com/kb/ws/v1/GASService</a></td>
</tr>
<tr>
<td>Account</td>
<td><a href="https://test-gcc.globalsign.com/kb/ws/v1/AccountService">https://test-gcc.globalsign.com/kb/ws/v1/AccountService</a></td>
</tr>
<tr>
<td>Subscriber Agreement</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*Test system accounts are available to API customers upon request*
4. Ordering DomainSSL & AlphaSSL Certificates

DomainSSL and AlphaSSL certificates can be ordered through three different validation methods:

- **DVOrder** – Approved via email
- **URLVerification** – Approved via HTTP domain validation method, where a random value provided by GlobalSign is posted on the website of the domain being validated
- **DVDNS** – Approved via DNS domain validation method by placing a DNS TXT record containing a random value for the domain being validated

The following subsections describe in greater detail how to use these validation methods.

4.1 Using Email Validation

Use the following steps to place an order for a DV or AlphaSSL certificate using the email verification method.

1. *(Optional) Use the DecodeCSR API to decode the CSR. Refer to Section 9.1 for more information.*
2. Use the GetDVApproverList API to retrieve the list of valid approver email addresses and the OrderID that must be used in DVOrder. This is also where the CommonName is checked against the High Risk database.
   
   Note: You must use this API to place an order, even if you know the email address you want to use. Refer to Section 8.1 for more information.
3. Use the DVOrder API to create the order in the system.
4. Send the certificate approver email to the list of recipients from Step 2 (outside the API).
5. Approve (or deny) the order (outside the API).
6. Use GetOrderByOrderID or a similar API to pick up the issued certificate.

4.1.1 DVOrder Request

When using the DVOrder API, you must use the OrderID and one of the provided ApproverEmail addresses from the GetDVApproverList API call.

```xml
<DVOrder xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>30 String</UserName>
        <Password>30 String</Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderRequestParameter>
      <ProductCode>DV, DV_LOW</ProductCode>
      <OrderKind>new, renewal, transfer</OrderKind>
      <Licenses>1-99</Licenses>
      <Options>
        <Option>
          <OptionName>See section 11.11</OptionName>
          <OptionValue>true, false</OptionValue>
        </Option>
      </Options>
    </OrderRequestParameter>
    <ValidityPeriod>
      <Months>4</Months>
      <NotBefore>25 YYYY-MM-DDTHH:MM:SS.000Z</NotBefore>
      <NotAfter>25 YYYY-MM-DDTHH:MM:SS.000Z</NotAfter>
    </ValidityPeriod>
    <CSR>4000 String</CSR>
    <RenewalTargetOrderID>50 String</RenewalTargetOrderID>
  </Request>
</DVOrder>
```
4.1.2 DVOrder Response

If the response contains a success code of 0, GlobalSign will send an email to the Approval Email contact. After the contact has given permission for the certificate to be issued, the certificate will be issued and sent via email to the reseller for forwarding to the end user.

If the response contains a success code of 1, GlobalSign will manually review the order. Once the manual review has been completed, the Approval Email will be sent.

4.2 Using HTTP (URL) Verification

Use the following steps to place an order for a DV or AlphaSSL certificate using the HTTP (URL) verification method. After the order is placed, the API response will contain a Domain Validation Code (DVC), also known as a metatag, which is used to verify the domain rather
than the approver email (refer to Section 4.1). The metatag must be placed in the <head> of the index of the domain that is being secured.

4.2.1 URL Verification Order Request

```xml
<URLVerification xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName> 30 String
        <Password> 30 String
      </AuthToken>
    </OrderRequestHeader>
    <OrderRequestParameter>
      <ProductCode> DV_HIGH_URL, DV_LOW_URL</ProductCode>
      <BaseOption>?</BaseOption>
      <OrderKind>new, renewal, transfer</OrderKind>
      <Licenses>1-99</Licenses>
      <Options>
        <Option>
          <OptionName>See section 11.11</OptionName>
          <OptionValue>true, false</OptionValue>
        </Option>+</Options>?
      <ValidityPeriod>
        <Months>4</Months>
        <NotBefore>? 25 YYYY-MM-DDTHH:MM:SS.000Z
        <NotAfter>? 25 YYYY-MM-DDTHH:MM:SS.000Z
      </ValidityPeriod>
      <CSR>4000 String</CSR>
      <RenewalTargetOrderID>? 50 String</RenewalTargetOrderID>
      <TargetCERT>? 4000 String</TargetCERT>
      <SpecialInstructions>? 4000 String</SpecialInstructions>
      <Coupon>? 50 String</Coupon>
      <Campaign>? 50 String</Campaign>
    </OrderRequestParameter>
    <SubID>? 50 String</SubID>
    <ContactInfo>
      <FirstName>100 String</FirstName>
      <LastName>100 String</LastName>
      <Phone>30 String</Phone>
      <Email>255 String</Email>
    </ContactInfo>
    <SecondContactInfo>
      <FirstName>100 String</FirstName>
      <LastName>100 String</LastName>
      <Email>255 String</Email>
    </SecondContactInfo>
    <SANEntries>
      <SANEntry>
        <SANOptionType>1: UC cert option
        <SubjectAltName>4000 String</SubjectAltName>
        <ModifyOperation>? <!-- N/A -->
      </SANEntry>+</SANEntries>
    </SANEntries>
  </Request>
</URLVerification>
```
4.2.2 URL Verification Order Response

The response to this type of order contains both the metatag and a list of allowed domains that GlobalSign can use to verify the FQDN. The metatag must be placed in the <head> of the index of the domain that is being secured.

```xml
<URLVerificationResponse xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Response>
    <OrderResponseHeader>
      <SuccessCode> 2
      <Errors>
        <Error>
          <ErrorCode> 5
          <ErrorField>?1000 String
          <ErrorMessage> 1000 String
        </Error>
        </Errors>
      </SuccessCodeHeader>
    </OrderResponseHeader>
    <OrderID>50 String
    <Metatag>50 String</Metatag>
    <VerificationURLList>
      <VerificationURL>1000 String</VerificationURLList>
    </Response>
  </URLVerificationResponse>
```

4.2.3 URL Verification for Issue Request

After placing the Random Value (metatag) on one of the allowable domains, the following request is used to validate the domain. It is important to specify one of the valid URLs in the ApproverURL, otherwise the domain validation will fail. Refer to Section 11.2 for more information.

```xml
<ns2:URLVerificationForIssue xmlns:ns2="https://system.globalsign.com/bb/ws/">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>30 String
        <Password>30 String
      </AuthToken>
    </OrderRequestHeader>
    <OrderID>ApproverURL
    <ApproverURL>64 String
  </Request>
</ns2:URLVerificationForIssue>
```

4.2.4 URL Verification for Issue Response

```xml
<UrlVerificationForIssue>
  <Response>
    <OrderResponseHeader>
      <SuccessCode> 2
      <Errors>
        <Error>
          <ErrorCode> 5
          <ErrorField>?1000 String
          <ErrorMessage> 1000 String
        </Error>
        </Errors>
      </SuccessCodeHeader>
    </OrderResponseHeader>
    <<CertificateInfo>
      <CertificateStatus>5
      <StartDate>YYYY-MM-DDTHH:MM:SS.000Z
      <EndDate>YYYY-MM-DDTHH:MM:SS.000Z
      <CommonName>64 String
      <SubjectName>3000 String
      (<DNSNames>)?
    </CertificateInfo>
  </Response>
</UrlVerificationForIssue>
```
4.3 Using DNS Verification

Use the following steps to place an order for a DV or AlphaSSL certificate using the DNS verification method. After the order is placed, the API response will contain a DNS TXT record, which is used to verify the domain rather than the approver email (refer to Section 4.1). The DNS TXT record must be placed in the index of the domain that is being secured. The DNS TXT record will resemble:

globalsign-domain-verification=YlhVOpMzmFGzcAzKqDmxqKGYlnMxj_Ioc1Cqq-nTs

Please note that the DNS TXT record check is case sensitive. Some DNS providers may automatically adjust the case of the TXT record. This scenario will cause a failure of the validation.

4.3.1 DVDNSOrder Request

< DVDNSOrder xmlns="http://stub.order.gasapiserver.esp.globalsign.com" >
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName> 30 String</UserName>
        <Password> 30 String</Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderRequestParameter>
      <ProductCode>
        DV_HIGH_DNS, DV_LOW_DNS
      </ProductCode>
      <BaseOption>?
        wildcard
      </BaseOption>
      <OrderKind> new, renewal, transfer
      </OrderKind>
      <Licenses> 1-99
      </Licenses>
      <Options>
        <Option>
          <OptionName> See section 11.11
          <OptionValue> true, false
        </Option>
      </Options>
      <ValidityPeriod>
        <Months> 4
        <NotBefore> 25 YYYY-MM-DDTHH:MM:SS.000Z
        <NotAfter> 25 YYYY-MM-DDTHH:MM:SS.000Z
      </ValidityPeriod>
      <CSR> 4000 String
      <RenewalTargetOrderID> 50 String
      <TargetCERT> 4000 String
      <SpecialInstructions> 4000 String
      <Coupon> 50 String
      <Campaign> 50 String
    </OrderRequestParameter>
    <SubID> 50 String
    <OrderID> 50 String
    <ContactInfo>
      <FirstName> 100 String
      <LastName> 100 String
    </ContactInfo>
  </Request>
</DVDNSOrder>
4.3.2 DVDNS Order Response

The response to this type of order contains both the DNS TXT and a list of allowable domains with which GlobalSign can verify the requested FQDN.

```xml
<DVDNSOrderResponse xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>1000</ErrorField>
          <ErrorMessage>1000</ErrorMessage>
        </Error>
      </Errors>
    </OrderResponseHeader>
    <OrderID>50</OrderID>
    <DNSTXT>50</DNSTXT>
    <VerificationFQDNList>
      <VerificationFQDN>1000</VerificationFQDN>
    </VerificationFQDNList>
  </Response>
</DVDNSOrderResponse>
```

4.3.3 DVDNS Order for Issue Request

After placing the DNS TXT record on one of the allowable domains, the following request is used by a GlobalSign Vetting Agent to verify the DNS TXT record. For more information refer to Section 11.1.

```xml
<ns2:DVDNSVerificationForIssue xmlns:ns2="https://system.globalsign.com/bb/ws/">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>30</UserName>
        <Password>30</Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderID>64</OrderID>
    <ApproverFQDN>1000</ApproverFQDN>
  </Request>
</DVDNSVerificationForIssue>
```
4.3.4 DVDNS Order for Issue Response

```xml
<DVDNSVerificationForIssueResponse>
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>1000</ErrorField>
          <ErrorMessage>1000</ErrorMessage>
        </Error>
      </Errors>
      <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    </OrderResponseHeader>
    <CertificateInfo>
      <CertificateStatus>5</CertificateStatus>
      <StartDate>YYYY-MM-DDTHH:MM:SS.000Z</StartDate>
      <EndDate>YYYY-MM-DDTHH:MM:SS.000Z</EndDate>
      <CommonName>64</CommonName>
      <SerialNumber>64</SerialNumber>
      <SubjectName>3000</SubjectName>
      (<DNSNames>300</DNSNames>)
    </CertificateInfo>
    <Fulfillment>
      <CACertificates>
        <CACertificate>
          <CACertType>Root, Inter</CACertType>
          <CACert>4000</CACert>
        </CACertificate>
      </CACertificates>
      <ServerCertificate>
        <X509Cert>4000</X509Cert>
        <PKCS7Cert>4000</PKCS7Cert>
      </ServerCertificate>
    </Fulfillment>
  </Response>
</DVDNSVerificationForIssueResponse>
```
5. **Ordering Organization SSL Certificates**

The following steps describe how to order an OV certificate.

1. Use the **DecodeCSR** API to decode the CSR. Refer to Section 9.1 for more information.
2. Use the **OVOrder** API to order the certificate in the system.

5.1 **DecodeCSR**

The following API code is used to extract the **CommonName** from the CSR.

5.1.1 **DecodeCSR Request**

```xml
<DecodeCSR>
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName>30 String</UserName>
        <Password>30 String</Password>
      </AuthToken>
    </QueryRequestHeader>
    <CSR>
      <ProductType>OV</ProductType>
    </CSR>
  </Request>
</DecodeCSR>
```

5.1.2 **DecodeCSR Response**

```xml
<DecodeCSR>
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>? 1000 String
          <ErrorMessage>1000 String
        </Error>
      </Errors>
    </OrderResponseHeader>
    <CSRData>
      <CommonName>? 255 String
      <Organization>? 255 String
      <OrganizationUnit>? 255 String
      <Locality>? 255 String
      <State>? 255 String
      <Country>? 30 String
      <EmailAddress>? 255 String
      <KeyLength>? 30 String
    </CSRData>
    <CertificatePreview>
      <CommonName>? 255 String
      <Organization>? 255 String
      <OrganizationUnit>? 255 String
      <Locality>? 255 String
      <State>? 255 String
      <Country>? 30 String
      <EmailAddress>? 255 String
      <KeyLength>? 30 String
    </CertificatePreview>
  </Response>
</DecodeCSR>
```
5.2 Ordering the Organization SSL Certificate

5.2.1 OVOrder Request

```xml
<OVOrder xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>30 String</UserName>
        <Password>30 String</Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderRequestParameter>
      <ProductCode>OV</ProductCode>
      <BaseOption>?</BaseOption>
      <OrderKind>new,renewal,transfer</OrderKind>
      <Licenses>1-99</Licenses>
      <Options>
        <Option>
          <OptionName>See section 11.11</OptionName>
          <OptionValue>true,false</OptionValue>
        </Option>
      </Options>
      <ValidityPeriod>
        <Months>4</Months>
        <NotBefore>?25 YYYY-MM-DDTHH:MM:SS.000Z</NotBefore>
        <NotAfter>?25 YYYY-MM-DDTHH:MM:SS.000Z</NotAfter>
      </ValidityPeriod>
      <CSR>4000 String</CSR>
      <RenewalTargetOrderID>?50 String</RenewalTargetOrderID>
      <TargetCERT>?4000 String</TargetCERT>
      <SpecialInstructions>?4000 String</SpecialInstructions>
      <Coupon>?50 String</Coupon>
      <Campaign>?50 String</Campaign>
    </OrderRequestParameter>
    <SubID>?50 String</SubID>
    <OrganizationInfo>
      <OrganizationName>255 String</OrganizationName>
      <CreditAgency>1:DUNS, 2:TDB</CreditAgency>
      <OrganizationCode>50 String</OrganizationCode>
      <OrganizationAddress>
        <AddressLine1>100 String</AddressLine1>
        <AddressLine2>?100 String</AddressLine2>
        <AddressLine3>?100 String</AddressLine3>
        <City>200 String</City>
        <Region>255 String</Region>
        <PostalCode>20 String</PostalCode>
        <Country>30 String</Country>
        <Phone>30 String</Phone>
        <Fax>?30 String</Fax>
      </OrganizationAddress>
      <OrganizationInfo>
      <ContactInfo>
        <FirstName>100 String</FirstName>
        <LastName>100 String</LastName>
        <Phone>30 String</Phone>
        <Email>255 String</Email>
      </ContactInfo>
      <SANEntries>
        <SANEntry>
          <SANOptionType>1:UC cert option
          2:Subdomain SAN option
          3:GIP SAN option
          4:Internal SAN option
          7:FQDN SAN option</SANOptionType>
          <SubjectAltName>4000 String</SubjectAltName>
          <ModifyOperation>ADDITION, UNCHANGED, DELETE</ModifyOperation>
        </SANEntry>
      </SANEntries>
    </SANEntries>
  </Request>
</OVOrder>
```
5.2.2 OVOrder Response

```
<OVOrder xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName>30</UserName>
        <Password>30</Password>
      </AuthToken>
    </QueryRequestHeader>
    <CSR>50</CSR>
    <ProductType>EV</ProductType>
  </Request>
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>1000</ErrorField>
          <ErrorMessage>1000</ErrorMessage>
        </Error>
      </Errors>
      <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    </OrderResponseHeader>
    <OrderID>50</OrderID>
  </Response>
</OVOrder>
```

6. Ordering ExtendedSSL Certificates

The following steps describe how to order an EV certificate.

1. Use the DecodeCSR API to decode the CSR. Refer to Section 9.1 for more information.
2. Use the EVOrder API to create the Order in the system.

6.1 DecodeCSR

The following API code is used to extract the CommonName from the CSR.

6.1.1 DecodeCSR Request

```
<DecodeCSR>
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName>30</UserName>
        <Password>30</Password>
      </AuthToken>
    </QueryRequestHeader>
    <CSR>50</CSR>
    <ProductType>EV</ProductType>
  </Request>
</DecodeCSR>
```

6.1.2 DecodeCSR Response

```
<DecodeCSR>
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors/>
    </OrderResponseHeader>
  </Response>
</DecodeCSR>
```
6.2 Ordering the ExtendedSSL Certificate

6.2.1 EVOrder Request

```xml
<EVOrder xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>30 String</UserName>
        <Password>30 String</Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderRequestParameter>
      <ProductCode>EV</ProductCode>
      <BaseOption/>
      <OrderKind>
        <Licenses>1-99</Licenses>
        <Options>
          <Option>
            <OptionName>See section 11.11</OptionName>
            <OptionValue>true,false</OptionValue>
          </Option>
        </Options>
        <ValidityPeriod>
          <Months>4</Months>
          <NotBefore>25 YYYY-MM-DDTHH:MM:SS.000Z</NotBefore>
          <NotAfter>25 YYYY-MM-DDTHH:MM:SS.000Z</NotAfter>
        </ValidityPeriod>
        <CSR>
          <RenewalTargetOrderID>50 String</RenewalTargetOrderID>
          <TargetCERT>4000 String</TargetCERT>
          <SpecialInstructions>4000 String</SpecialInstructions>
          <Coupon>50 String</Coupon>
          <Campaign>50 String</Campaign>
        </CSR>
      </OrderRequestParameter>
      <SubID>50 String</SubID>
      <OrganizationInfoEV>
```
{<CreditAgency>}? 1:DUNS, 2:TDB
{<OrganizationCode>}? 50 String
{<BusinessAssumedName>}? 255 String
{<BusinessCategoryCode>} PO:Private Organization
GE:Government Entity
BE:BusinessEntity

{<OrganizationAddress>}
{<AddressLine1>}? 100 String
{<AddressLine2>}? 100 String
{<AddressLine3>}? 100 String
{<City>} 200 String
{<Region>} 255 String
{<PostalCode>} 20 String
{<Country>} 30 String ISO 3166-1
{<Phone>}? 30 String
{<Fax>}? 30 String

{</OrganizationAddress>
{</OrganizationInfoEV>}
{<RequestorInfo>}
{<FirstName>} 100 String
{<LastName>} 100 String
{<Function>}? 255 String
{<OrganizationName>} 255 String
{<OrganizationUnit>}? 100 String
{<Phone>} 30 String
{<Email>} 255 String

{</RequestorInfo>}
{<ApproverInfo>}
{<FirstName>} 100 String
{<LastName>} 100 String
{<Function>}? 255 String
{<OrganizationName>} 255 String
{<OrganizationUnit>}? 100 String
{<Phone>} 30 String
{<Email>} 255 String

{</ApproverInfo>}
{<AuthorizedSignerInfo>}
{<OrganizationName>} 255 String
{<FirstName>} 100 String
{<LastName>} 100 String
{<Function>}? 255 String
{<Phone>} 30 String
{<Email>} 255 String

{</AuthorizedSignerInfo>}
{<JurisdictionInfo>}
{<JurisdictionCountry>} 30 String ISO 3166-1
{<StateOrProvince>} 255 String
{<Locality>} 200 String
{<IncorporatingAgencyRegistrationNumber>} 100 String

{</JurisdictionInfo>}
{<ContactInfo>}
{<FirstName>} 100 String
{<LastName>} 100 String
{<Phone>} 30 String
{<Email>} 255 String

{</ContactInfo>}
{<SANEntries>}
{<SANEntry>}
{<SANOptionType>}
1: UC cert option
2: Subdomain SAN option
3: GIP SAN option
4: Internal SAN option
7: FQDN SAN option
4000 String
{<ModifyOperation>}? <!-- N/A -->

{</SANEntries>}

{</SANEntries>}

{</Request>}
{</EVOrder>
6.2.2 EVOrder Response

```xml
<EVOrder xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Response>
  <OrderResponseHeader>
    <SuccessCode>2</SuccessCode>
    <Errors>
      <Error>
        <ErrorCode>5</ErrorCode>
        <ErrorField>1000</ErrorField>
        <ErrorMessage>1000</ErrorMessage>
      </Error>
    </Errors>
    <Timestamp>25 YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
  </OrderResponseHeader>
  <OrderID>50</OrderID>
</Response>
</EVOrder>
```

7. Domain Validation

Once you place your order, you can validate the domains using GlobalSign APIs. These APIs are for OrganizationSSL and ExtendedSSL products. Below is a high-level process flow that describes how to perform OV and EV domain validation through the GCC APIs:

1. Place the certificate order as normal and receive the OrderID.
2. Use the Get SAN Validation Status API to get a list of SANs that need to be validated for the order and their current status, as well as the domain validation random number.
3. If using the HTTP or DNS validation method, place the random number in your website or DNS TXT record.
4. For each SAN that needs to be validated, use the Validate SANs API and select which validation method you wish to use.
   a. For HTTP and DNS domain validation, the API will verify the presence of the random number that was returned in the GetOrderSANStatus API. The TagLocation element supplied in the VerifySAN API is the location of the random number.
   b. For EMAIL domain validation, the API will send or resend the approver email. You may want to use the GetDVApproverList to get a list of valid approver email addresses.
   c. In both cases, the status of the domain validation will be returned.
5. Continue to use the GetOrderSANStatus API to get the status and confirmation of domain validation for all SANs.
6. Use the GetOrderByOrderID API to get the certificate order status and receive the certificate when it is issued.

8. General SSL Functions

8.1 Receive List of Approver Email Addresses and Perform Phishing DB Check

The details from the DecodeCSR Response can now be used to continue with the request. The next step involves receiving the list of approver email addresses and an OrderID to complete the certificate order. This is only required when the order approval method is set to email validation. This is a required command which must be used even if you know which Approver email you want to use.
8.1.1 GetDVApproverList Request

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="https://system.globalsign.com/kb/ws/v1/"
>
  <soapenv:Header/>
  <soapenv:Body>
    <v1:GetDVApproverList>
      <Request>
        <QueryRequestHeader>
          <AuthToken>
            <UserName></UserName>
            <Password></Password>
          </AuthToken>
        </QueryRequestHeader>
        <FQDN>sslcerts.jp</FQDN>
        <OrderID></OrderID>
        <SANEntries>
          <SANEntry>
            <SANOptionType>2</SANOptionType>
            <SubjectAltName>sub1.sslcerts.jp</SubjectAltName>
          </SANEntry>
          <SANEntry>
            <SANOptionType>2</SANOptionType>
            <SubjectAltName>sub2.sslcerts.jp</SubjectAltName>
          </SANEntry>
        </SANEntries>
      </Request>
    </v1:GetDVApproverList>
  </soapenv:Body>
</soapenv:Envelope>

Note: The FQDN is the **CommonName** from previous response.
8.1.2 GetDVApproverList Response

This response will contain a success code, a list of approver contact details for the end user to choose from, and an OrderID for continuing with the order. If the success code is -1, the request procedure will stop and an error code will display.
8.2 Change the SubjectAltName

This section describes how to replace the current set of SANs in a certificate with a new set of SANs. To use this command, specify the OrderID you want to change and then specify a new set of SANs that will replace the current set of SANs. The Public Key and expiration date of the certificate remain the same, but the SANs are updated.

For most products, if a user requests a certificate with a CommonName like “www.example.com,” GlobalSign automatically includes “example.com” in the order. At this time there is no way to instruct GlobalSign to not include “example.com.”

If a user has “example.com” as the CommonName, that is the only name that will appear in the SAN; however, you may use the free UC SAN option to add “www.example.com” to the certificate.

8.2.1 DomainSSL

1. Retrieve the list of approver emails and OrderIDs by using the GetDVApproverList API.
2. Submit the new SAN(s) using the ChangeSubjectAltName command.
3. Send the approver email to the list of recipients from Step 1 (outside the API).
4. Approve (or deny) the new SAN(s).

8.2.2 OrganizationSSL, ExtendedSSL

Use the ChangeSubjectAltName API to change the SANs of issued certificates. <SANEntries> parameters should be set to what the SubjectAltName(s) will be after the change. GlobalSign will do a manual review of the change for OV/EV certificates. New certificates with updated SANs will be issued and queriable after vetting has been completed.

8.2.3 ChangeSubjectAltName Request

```xml
<ChangeSubjectAltName xmlns="http://stub.order.gasapiserver.esp.globalsign.com ">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>
          <Password>
        </AuthToken>
      </OrderRequestHeader>
      (<OrderID>)?
      (<TargetOrderID>)?
      (<ApproverEmail>)?
      (<SANEntries>)
        (<SANEntryArray>
          <SANOptionType>
            1:UC cert option
            2:Subdomain SAN option
            3:GIP SAN option
            4:Internal SAN option
            7:FQDN SAN option
          </SANOptionType>
          <SubjectAltName>64 String
        </SANEntryArray>)+
        <PIN>?
    </SANEntries>)?
  </Request>
</ChangeSubjectAltName>
```
8.2.4 ChangeSubjectAltName Response

```xml
<ChangeSubjectAltName xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Response>
    <OrderResponseHeader>
      <SuccessCode>
        <Errors>
          <Error>
            <ErrorCode>5</ErrorCode>
            <ErrorField>?1000</ErrorField>
            <ErrorMessage>1000</ErrorMessage>
          </Error>
        </Errors>?
        <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
      </OrderResponseHeader>
      <OrderID>?50</OrderID>
      <TargetOrderID>?50</TargetOrderID>
    </Response>
  </ChangeSubjectAltName>
```

8.3 Create Certificate Invites

The CertInviteOrder feature is used by GlobalSign Partners to order certificates for end customers. The Partner enters the contact details of the customer into the order, and then GlobalSign generates a unique code specific to the order and provides the Partner with a PIN. GlobalSign sends an email to the customer containing a link to the certificate, and the Partner provides the customer with the PIN.

8.3.1 CertInviteOrder Request

```xml
<CertInviteOrder>
  <OrderRequestHeader>
    <AuthToken>
      <UserName>30</UserName>
      <Password>30</Password>
    </AuthToken>
  </OrderRequestHeader>
  <OrderRequestParameter>
    <ProductCode>
      <BaseOption>?</BaseOption>
    </ProductCode>
    <OrderKind>new,renewal,transfer</OrderKind>
    <Options>
      <Option>
        <OptionName>See section 11.11</OptionName>
        <OptionValue>true,false</OptionValue>
      </Option>
    </Options>?
    <ValidityPeriod>
      <Months>4</Months>
      <NotBefore>?25 YYYY-MM-DDTHH:MM:SS.000Z</NotBefore>
      <NotAfter>?25 YYYY-MM-DDTHH:MM:SS.000Z</NotAfter>
    </ValidityPeriod>
    <RenewalTargetOrderID>?50</RenewalTargetOrderID>
    <Coupon>?50</Coupon>
    <Campaign>?50</Campaign>
  </OrderRequestParameter>
  <SANEntries>
    <SANEntry>
      <SANOptionType>1:UC cert option
                       2:Subdomain SAN option
                       3:GIP SAN option
                       4:Internal SAN option
                       7:FQDN SAN option</SANOptionType>
    </SANEntry>
  </SANEntries>
  <CertInviteExpirationDate>25 YYYY-MM-DDTHH:MM:SS.000Z</CertInviteExpirationDate>
  <RecipientDeliveryOption>true,false</RecipientDeliveryOption>
</CertInviteOrder>
```
8.3.2 CertInviteOrder Response

```xml
<CertInviteOrder>
    <Response>
        <OrderResponseHeader>
            <SuccessCode> 2
            <Errors>
                <Error>
                    <ErrorCode> 5
                    <ErrorField> 1000
                    <ErrorMessage> 1000
                </Error>
            </Errors>
            <Timestamp> YYYY-MM-DDTHH:MM:SS.000Z </Timestamp>
        </OrderResponseHeader>
        <OrderID> 50 </OrderID>
        <PIN> 255 </PIN>
    </Response>
</CertInviteOrder>
```

8.4 Change Approver Email

The ChangeApproverEmail API allows users to change the approver email for an order. When the request is submitted, a new approval request will be sent to the new approver email.

The user may optionally use the GetApproverList request before submitting the ChangeApproverEmail request, to verify the list of approval emails currently registered to the profile.

8.4.1 ChangeApproverEmail Request

```xml
<ChangeApproverEmail>
    <Request>
        <OrderRequestHeader>
            <AuthToken>
                <UserName> 30 </UserName>
                <Password> 30 </Password>
            </AuthToken>
        </OrderRequestHeader>
        <OrderID> 50 </OrderID>
        <ApproverEmail> 255 </ApproverEmail>
        <FQDN> 255 </FQDN>
    </Request>
</ChangeApproverEmail>
```

8.4.2 ChangeApproverEmail Response

```xml
<ChangeApproverEmail>
    <Response>
        <OrderResponseHeader>
            <SuccessCode> 2
            <Errors>
                <Error>
                    <ErrorCode> 5
                    <ErrorField> 1000
                    <ErrorMessage> 1000
                </Error>
            </Errors>
            <Timestamp> YYYY-MM-DDTHH:MM:SS.000Z </Timestamp>
        </OrderResponseHeader>
        <OrderID> 50 </OrderID>
    </Response>
</ChangeApproverEmail>
```
8.5 Re-Send Approver Email

The ResendEmail function is used if the user did not receive or lost the Approver Email message. Using this API will re-send the approval email to the address listed on the account.

Note: The random value/link in the approver email is valid for 3 days from when the order was placed. If the order is not approved within 30 days (even when the approver email is resent), the order must be cancelled and a new order placed.

8.5.1 ResendEmail Request

```xml
<ResendEmail xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>30</UserName>
        <Password>30</Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderID>50</OrderID>
    <ResendEmailType>20</ResendEmailType> APPROVEREMAIL
  </Request>
</ResendEmail>
```

8.5.2 ResendEmail Response

```xml
<ResendEmail xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>1000</ErrorField>
          <ErrorMessage>1000</ErrorMessage>
        </Error>
      </Errors>
      <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    </OrderResponseHeader>
    <OrderID>50</OrderID>
  </Response>
</ResendEmail>
```

8.6 Modify Existing Order

The ModifyOrder function allows users to Approve, Cancel or Revoke a Certificate or Certificate Request by using the OrderID of the Order. It is not currently possible to change the validation method of an order (email, HTTP, DNS); if this information needs to change for any reason, the original order should be cancelled first and then a new order created with the updated validation method.

8.6.1 ModifyOrder Request

```xml
<ModifyOrder xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>
          <Password>
        </Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderID>50</OrderID>
    <ModifyOrderOperation>APPROVE,CANCEL,REVOKE</ModifyOrderOperation>
  </Request>
</ModifyOrder>
```
8.6.2 ModifyOrder Response

```xml
<ModifyOrder xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>1000</ErrorField>
          <ErrorMessage>String</ErrorMessage>
        </Error>
      </Errors>
    </OrderResponseHeader>
    <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    <OrderID>50</OrderID>
  </Response>
</ModifyOrder>
```

9. Service & Query APIs

9.1 DecodeCSR

The DecodeCSR function is used to extract the **CommonName** from the CSR.

9.1.1 DecodeCSR Request

```xml
<DecodeCSR>
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName>30 String</UserName>
        <Password>30 String</Password>
      </AuthToken>
    </QueryRequestHeader>
    <CSR>50 String</CSR>
    <ProductType>DV_LOW, DV</ProductType>
  </Request>
</DecodeCSR>
```

9.1.2 DecodeCSR Response

```xml
<DecodeCSR>
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>1000</ErrorField>
          <ErrorMessage>String</ErrorMessage>
        </Error>
      </Errors>
    </OrderResponseHeader>
    <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    <CSRData>
      <CommonName>255 String</CommonName>
      <Organization>255 String</Organization>
      <OrganizationUnit>255 String</OrganizationUnit>
      <Locality>255 String</Locality>
      <State>255 String</State>
      <Country>30 String</Country>
      <EmailAddress>255 String</EmailAddress>
      <KeyLength>30 String</KeyLength>
    </CSRData>
    <CertificatePreview>
      <CommonName>255 String</CommonName>
    </CertificatePreview>
  </Response>
</DecodeCSR>
```
9.2 Get Issued Certificate - Single Certificate

The GetOrderByOrderID API allows users to check the status of a specific order using the order’s OrderID.

Note: A few bugs have been identified for this API:

- The Coupon tag does not return a value in this API
- The originalCSR tag is missing from the API response
- When ValidityPeriodCustomizeOption is set to “true,” the response API lists it as “false”
- OrganizationCode tag does not return a value in this API
- SpecialInstructions tag does not return a value in this API or display in GCC
- Jurisdiction Details does not display the value for “Country”

9.2.1 GetOrderByOrderID Request

```xml
<GetOrderByOrderID xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName> 30 String
        <Password> 30 String
      </AuthToken>
    </QueryRequestHeader>
    <OrderID> 50 String
    <OrderQueryOption>
      <<OrderStatus>>? <!-- N/A -->
      <<ReturnOrderOption>>? 5 String true, false
      <<ReturnCertificateInfo>>? 5 String true, false
      <<ReturnFulfillment>>? 5 String true, false
      <<ReturnCACerts>>? 5 String ReturnFulfillment true
      <<ReturnOriginalCSR>>? 5 String true, false
      <<ReturnPKCS12>>? 5 String true, false
      <<ReturnSANEntries>>? 5 String true, false
    </OrderQueryOption>
  </Request>
</GetOrderByOrderID>
```

9.2.2 GetOrderByOrderID Response

```xml
<GetOrderByOrderID xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Response>
    <QueryResponseHeader>
      <SuccessCode> 2
      <Errors>
        <Error>
          <ErrorCode> 5
          <ErrorMessage> 1000 String
        </Error>
      </Errors>
      <Timestamp> 25 YYYY-MM-DDTHH:MM:SS.000Z
      <ReturnCount> 5
    </QueryResponseHeader>
  </Response>
</GetOrderByOrderID>
```
<OrderID>? 50 String
<Pkcs12File>? 4000 String

<OrderDetail>

<OrderInfo>
<OrderID> 50 String
<ProductCode> 20 String
<BaseOption>? 20 String
<OrderKind> 10 String
<License> 3 String
<ExpressOption>? 5 String
<ValidityPeriodCustomizeOption>? 5 String
<InsuranceOption>? 5 String
<GSSupportOption>? 5 String
<RenewalExtentionOption>? 5 String
<DomainName> 255 String
<OrderDate> 25 YYYY-MM-DDTHH:MM:SS.000Z
<OrderCompleteDate>? 25 YYYY-MM-DDTHH:MM:SS.000Z
<OrderCanceledDate>? 25 YYYY-MM-DDTHH:MM:SS.000Z
<OrderDeactivatedDate>? 25 YYYY-MM-DDTHH:MM:SS.000Z
<OrderStatus> 5 String
1: INITIAL
2: Waiting for phishing check
3: Cancelled - Not Issued
4: Issue completed
5: Cancelled - Issued
6: Waiting for revocation
7: Revoked
</OrderInfo>

<Price> 10 String
<Currency> 10 String
</ValidityPeriod>
</ValidityPeriod>
</OrderInfo>

<OrderSubInfo>
<CSRSkipOrderFlag> 5 String true,false
<DNSOrderFlag> 5 String true,false
<TrustedOrderFlag> 5 String true,false
<P12DeleteStatus>? 5 String
<P12DeleteDate>? 25 YYYY-MM-DDTHH:MM:SS.000Z
<VerificationUrl>? 300 String
</OrderSubInfo>

<OrderOption>
</OrderOption>

<OrganizationInfo>
</OrganizationInfo>

<RequestorInfo>
</RequestorInfo>
9.3 Get Issued Certificates - Multiple Orders

The GetOrderByDateRange API allows users to check the status of multiple orders based on a specified date range.

Note: A bug has been identified; the Coupon tag does not return a value in this API.

9.3.1 GetOrderByDateRange Request

```xml
<GetOrderByDateRange xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName>30</UserName>
        <Password>30</Password>
      </AuthToken>
    </QueryRequestHeader>
    <FromDate>YYYY-MM-DDTHH:MM:SS.000Z</FromDate>
    <ToDate>YYYY-MM-DDTHH:MM:SS.000Z</ToDate>
    <OrderQueryOption>
      (<OrderStatus>)? 5 String
      (<ReturnOrderOption>)? 5 String
      (<ReturnCertificateInfo>)? 5 String
      (<ReturnFulfillment>)? 5 String
      (<ReturnCACerts>)? 5 String
    </OrderQueryOption>
  </Request>
</GetOrderByDateRange>
```

9.3.2 GetOrderByDateRange Response

```xml
<GetOrderByDateRange xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Response>
    <QueryResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        (<Error>)
          <ErrorCode>5</ErrorCode>
          <ErrorMessage>1000</ErrorMessage>
        </Error>
      </Errors>
    </QueryResponseHeader>
    <FromDate>YYYY-MM-DDTHH:MM:SS.000Z</FromDate>
    <ToDate>YYYY-MM-DDTHH:MM:SS.000Z</ToDate>
    <OrderDetails>
      (<OrderDetail>)
        <OrderInfo>
          <OrderID>50</OrderID>
          <ProductCode>20</ProductCode>
          (<BaseOption>)? 20 String
          <OrderKind>10</OrderKind>
          <Licenses>3</Licenses>
          (<ExpressOption>)? 5 String
          (<ValidityPeriodCustomizeOption>)? 5 String
          (<InsuranceOption>)? 5 String
          (<GSSupportOption>)? 5 String
          (<RenewalExtentionOption>)? 5 String
          <DomainName>255</DomainName>
          <OrderDate>255</OrderDate>
        </OrderInfo>
      </OrderDetail>
    </OrderDetails>
  </Response>
</GetOrderByDateRange>
```
<OrderStatus> 5
    1: INITIAL
    2: Waiting for phishing check
    3: Cancelled - Not Issued
    4: Issue completed
    5: Cancelled - Issued
    6: Waiting for revocation
    7: Revoked
</OrderStatus>

<Price> 10
    <Currency> 10
        String
    </Currency>
</Price>

<ValidityPeriod>
    <Months> 4
        {<NotBefore>}? 25 YYYY-MM-DDTHH:MM:SS.000Z
        {<NotAfter>}? 25 YYYY-MM-DDTHH:MM:SS.000Z
    </Months>
    {<SpecialInstructions>}? 4000
</ValidityPeriod>

<OrderInfo>
    <OrderSubInfo>
        <CSRSkipOrderFlag> 5
            String true,false
        </CSRSkipOrderFlag>
        <DNSOrderFlag> 5
            String true,false
        </DNSOrderFlag>
        <TrustedOrderFlag> 5
            String true,false
        </TrustedOrderFlag>
        {<P12DeleteStatus>}? 5
        {<P12DeleteDate>}? 25 YYYY-MM-DDTHH:MM:SS.000Z
        {<VerificationUrl>}? 300
        <SubId> 50
            String
        </SubId>
    </OrderSubInfo>
</OrderInfo>

<OrderOption>
    {<ApproverNotifiedDate>?} 725 YYYY-MM-DDTHH:MM:SS.000Z
    {<ApproverConfirmedDate>?} 725 YYYY-MM-DDTHH:MM:SS.000Z
    {<ApproverEmailAddress>}? 255
        <OrganizationInfo>
            <OrganizationName> 255
                String
            </OrganizationName>
            {<CreditAgency>}? 50
                String
            </CreditAgency>
            {<OrganizationCode>}? 50
                String
            </OrganizationCode>
            {<BusinessAssumedName>}? 255
                String
            </BusinessAssumedName>
            {<BusinessCategoryCode>}? 20
                String
            </BusinessCategoryCode>
            {<OrganizationAddress>}
                <AddressLine1>? 100
                    String
                </AddressLine1>
                <AddressLine2>? 100
                    String
                </AddressLine2>
                {<AddressLine3>}? 100
                    String
                </AddressLine3>
                <City> 200
                    String
                </City>
                <Region> 255
                    String
                </Region>
                {<PostalCode>}? 220
                    String
                </PostalCode>
                <Country> 30
                    String
                </Country>
                <Phone> 30
                    String
                </Phone>
                {<Fax>}? 30
                    String
                </Fax>
            </OrganizationAddress>
        </OrganizationInfo>
    </ApproverInfo>
</OrderOption>

<RequestorInfo>
    <RequestorInfo>
        <FirstName> 100
            String
        </FirstName>
        <LastName> 100
            String
        </LastName>
        <Function> 255
            String
        </Function>
        <OrganizationName> 255
            String
        </OrganizationName>
        <OrganizationUnit> 100
            String
        </OrganizationUnit>
        <Phone> 30
            String
        </Phone>
        <Email> 255
            String
        </Email>
    </RequestorInfo>
</RequestorInfo>

<ApproverInfo>
    <ApproverInfo>
        <FirstName> 100
            String
        </FirstName>
        <LastName> 100
            String
        </LastName>
        <Function> 255
            String
        </Function>
        <OrganizationName> 255
            String
        </OrganizationName>
        {<OrganizationUnit>}? 100
        <Phone> 30
            String
        </Phone>
        <Email> 255
            String
        </Email>
    </ApproverInfo>
</ApproverInfo>

<AuthorizedSignerInfo>
    <AuthorizedSignerInfo>
        <FirstName> 100
            String
        </FirstName>
        <LastName> 100
            String
        </LastName>
    </AuthorizedSignerInfo>
</AuthorizedSignerInfo>
<Function> 255 String
<Phone> 30 String
<Email> 255 String
</AuthorizedSignerInfo>

</JurisdictionInfo>

<JurisdictionCountry>30 String
<StateOrProvince> 255 String
<Locality> 200 String
<IncorporatingAgencyRegistrationNumber>100 String
</JurisdictionInfo>

<ContactInfo>
<FirstName> 100 String
<LastName> 100 String
<Phone> 30 String
<Email> 255 String
</ContactInfo>

</OrderOption>

</CertificateInfo>

<Fulfillment>
<CACertificates>
<CACertificate>
<CACertType> 15 String Root, Intermediate
<CACert> 4000 String
</CACertificate>
</CACertificates>

<ServerCertificate>
<X509Cert> 4000 String
<PKCS7Cert> 4000 String
</ServerCertificate>
</Fulfillment>

</ModificationEvents>
</OrderDetail>
</OrderDetails>
</Response>
9.4 Get Recently Modified Orders

The GetModifiedOrders API returns a list of orders that have been modified within a specified time frame.

Note: A bug has been identified; the Coupon tag does not return a value in this API.

9.4.1 GetModifiedOrders Request

```xml
<GetModifiedOrders xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName>
          <Password>
        </AuthToken>
      </QueryRequestHeader>
      <FromDate>YYYY-MM-DDTHH:MM:SS.000Z</FromDate>
      <ToDate>YYYY-MM-DDTHH:MM:SS.000Z</ToDate>
      (<OrderQueryOption>
        (<OrderStatus>)? 5
        1: INITIAL
        2: Waiting for phishing check
        3: Cancelled - Not Issued
        4: Issue completed
        5: Cancelled - Issued
        6: Waiting for revocation
        7: Revoked
      </OrderStatus>)?
      (<ReturnOrderOption>)?
      true,false
      (<ReturnCertificateInfo>)?
      true,false
      (<ReturnFulfillment>)?
      true,false
      (<ReturnCACerts>)?
      true,false
    </OrderQueryOption>
  </Request>
</GetModifiedOrders>
```

9.4.2 GetModifiedOrders Response

```xml
<GetModifiedOrders xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Response>
    <QueryResponseHeader>
      <SuccessCode>2</SuccessCode>
      (<Errors>
        (<Error>
          <ErrorCode>5</ErrorCode>
          <ErrorMessage>1000 String</ErrorMessage>
        </Error>)?
      </Errors>)?
      <Timestamp>25 YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
      <ReturnCount>5</ReturnCount>
    </QueryResponseHeader>
    <FromDate>? 25 YYYY-MM-DDTHH:MM:SS.000Z
    <ToDate>? 25 YYYY-MM-DDTHH:MM:SS.000Z
    (<OrderDetails>
      (<OrderDetail>
        <OrderInfo>
          <OrderID>50 String</OrderID>
          <ProductCode>20 String</ProductCode>
          (<BaseOption>)? 20 String
          <OrderKind>10 String</OrderKind>
          <Licenses>3</Licenses>
          (<ExpressOption>)? 5 String
          (<ValidityPeriodCustomizeOption>)?5 String
          (<InsuranceOption>)? 5 String
          (<GSSupportOption>)? 5 String
          (<RenewalExtensionOption>)?5 String
          <DomainName>255 String</DomainName>
          <OrderDate>25 YYYY-MM-DDTHH:MM:SS.000Z</OrderDate>
          (<OrderCompleteDate>)?25 YYYY-MM-DDTHH:MM:SS.000Z
          (<OrderCanceledDate>)?25 YYYY-MM-DDTHH:MM:SS.000Z
        </OrderInfo>
      </OrderDetail>
      <OrderDetail>
        <OrderInfo>
          <OrderID>50 String</OrderID>
          <ProductCode>20 String</ProductCode>
          (<BaseOption>)? 20 String
          <OrderKind>10 String</OrderKind>
          <Licenses>3</Licenses>
          (<ExpressOption>)? 5 String
          (<ValidityPeriodCustomizeOption>)?5 String
          (<InsuranceOption>)? 5 String
          (<GSSupportOption>)? 5 String
          (<RenewalExtensionOption>)?5 String
          <DomainName>255 String</DomainName>
          <OrderDate>25 YYYY-MM-DDTHH:MM:SS.000Z</OrderDate>
          (<OrderCompleteDate>)?25 YYYY-MM-DDTHH:MM:SS.000Z
          (<OrderCanceledDate>)?25 YYYY-MM-DDTHH:MM:SS.000Z
        </OrderInfo>
      </OrderDetail>
      <OrderDetail>
        <OrderInfo>
          <OrderID>50 String</OrderID>
          <ProductCode>20 String</ProductCode>
          (<BaseOption>)? 20 String
          <OrderKind>10 String</OrderKind>
          <Licenses>3</Licenses>
          (<ExpressOption>)? 5 String
          (<ValidityPeriodCustomizeOption>)?5 String
          (<InsuranceOption>)? 5 String
          (<GSSupportOption>)? 5 String
          (<RenewalExtensionOption>)?5 String
          <DomainName>255 String</DomainName>
          <OrderDate>25 YYYY-MM-DDTHH:MM:SS.000Z</OrderDate>
          (<OrderCompleteDate>)?25 YYYY-MM-DDTHH:MM:SS.000Z
          (<OrderCanceledDate>)?25 YYYY-MM-DDTHH:MM:SS.000Z
        </OrderInfo>
      </OrderDetail>
      <OrderDetail>
        <OrderInfo>
          <OrderID>50 String</OrderID>
          <ProductCode>20 String</ProductCode>
          (<BaseOption>)? 20 String
          <OrderKind>10 String</OrderKind>
          <Licenses>3</Licenses>
          (<ExpressOption>)? 5 String
          (<ValidityPeriodCustomizeOption>)?5 String
          (<InsuranceOption>)? 5 String
          (<GSSupportOption>)? 5 String
          (<RenewalExtensionOption>)?5 String
          <DomainName>255 String</DomainName>
          <OrderDate>25 YYYY-MM-DDTHH:MM:SS.000Z</OrderDate>
          (<OrderCompleteDate>)?25 YYYY-MM-DDTHH:MM:SS.000Z
          (<OrderCanceledDate>)?25 YYYY-MM-DDTHH:MM:SS.000Z
        </OrderInfo>
      </OrderDetail>
    </OrderDetails>
  </Response>
</GetModifiedOrders>
```
<OrderDeactivatedDate>225 YYYY-MM-DDTHH:MM:SS.000Z</OrderDeactivatedDate>
<OrderStatus>5 1: INITIAL
2: Waiting for phishing check
3: Cancelled - Not Issued
4: Issue completed
5: Cancelled - Issued
6: Waiting for revocation
7: Revoked
</OrderStatus>
<Price>10</Price>
<Currency>10</Currency>
<ValidityPeriod>
<Months>4
{(NotBefore)? 25 YYYY-MM-DDTHH:MM:SS.000Z}
{(NotAfter)? 25 YYYY-MM-DDTHH:MM:SS.000Z}
</ValidityPeriod>
</ValidityPeriod>
{SpecialInstructions}? 4000 String
</OrderInfo>
<OrderSubInfo>
<CSRSkipOrderFlag>5 String true,false
<DNSOrderFlag>5 String true,false
<TrustedOrderFlag>5 String true,false
{(P12DeleteStatus)? 5}
{(P12DeleteDate)? 25 YYYY-MM-DDTHH:MM:SS.000Z}
{(VerificationUrl)? 300 String}
<SubId>50 String
</OrderSubInfo>
</OrderOption>
</OrderInfo>

<ApproverNotifiedDate>225 YYYY-MM-DDTHH:MM:SS.000Z</ApproverNotifiedDate>
<ApproverConfirmDate>225 YYYY-MM-DDTHH:MM:SS.000Z</ApproverConfirmDate>
<ApproverEmailAddress>? 255 String
<OrganizationInfo>
<OrganizationName>255 String
{(CreditAgency)? 50 String}
{(OrganizationCode)? 50 String}
{(BusinessAssumedName)? 255 String}
{(BusinessCategoryCode)? 20 String}
<OrganizationAddress>
{(AddressLine1)? 100 String}
{(AddressLine2)? 100 String}
{(AddressLine3)? 100 String}
<City> 200 String
<Region> 255 String
{(PostalCode)? 720 String}
<Country> 30 String
<Phone> 30 String
{(Fax)? 30 String}
</OrganizationAddress>
</OrganizationInfo>
<RequestorInfo>
<FirstName> 100 String
<LastName> 100 String
<Function> 255 String
<OrganizationName> 255 String
<OrganizationUnit> 100 String
<Phone> 30 String
<Email> 255 String
</RequestorInfo>
</RequestorInfo>
</ApproverInfo>
<ApproverInfo>
<FirstName> 100 String
<LastName> 100 String
<Function> 255 String
<OrganizationName> 255 String
{(OrganizationUnit)? 100 String}
<Phone> 30 String
<Email> 255 String
</ApproverInfo>
</ApproverInfo>
</AuthorizedSignerInfo>
<FirstName> 100 String
<OrderOption>)?
</OrderDetail>)?
</Response>

<AuthorizedSignerInfo>)?
</JurisdictionInfo>)?
<ContactInfo>)?
</ContactInfo>)?
</OrderOption>)?
</CertificateInfo>)?
</Fulfillment>)?
</ModificationEvents>?
</OrderDetails>)?
</Response>

<LastName> 100 String
<Function> 255 String
<Phone> 30 String
<Email> 255 String
</AuthorizedSignerInfo>)?
</JurisdictionInfo>)?
<JurisdictionCountry> 30 String
<StateOrProvince> 255 String
<Locality> 200 String
<IncorporatingAgencyRegistrationNumber>100
</JurisdictionInfo>)?
<ContactInfo>)?
<FirstName> 100 String
<LastName> 100 String
<Phone> 30 String
<Email> 255 String
</ContactInfo>)?
</OrderOption>)?
</CertificateInfo>)?
</Fulfillment>)?
</ModificationEvents>?
</OrderDetails>)?
</Response>

<CertificateStatus> 5
1: INITIAL
2: Waiting for phishing check
3: Cancelled - Not Issued
4: Issue completed
6: Waiting for revocation
7: Revoked
</CertificateInfo>)?
</Fulfillment>)?
</ModificationEvents>?
</OrderDetails>)?
</Response>
9.5 Determine Upcoming Renewals

The `GetOrderByExpirationDate` API returns a list of orders with certificates that are expiring within a specified date range.

Note: A bug has been identified; the `Coupon` tag does not return a value in this API.

9.5.1 GetOrderByExpirationDate Request

```xml
<GetOrderByExpirationDate>
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName>
        <Password>
      </AuthToken>
    </QueryRequestHeader>
    (ExpirationFromDate)? YYYY-MM-DDTHH:MM:SS.000Z
    (ExpirationToDate)? YYYY-MM-DDTHH:MM:SS.000Z
    (FQDN)? 64 String
    (OrderKind)? 10 String
    (OrderStatus)? 5
      1: INITIAL
      2: Waiting for phishing check
      3: Cancelled - Not Issued
      4: Issue completed
      5: Cancelled - Issued
      6: Waiting for revocation
      7: Revoked
    (SubID)? 50 String
  </Request>
</GetCertificateOrders>
```

9.5.2 GetOrderByExpirationDate Response

```xml
<GetCertificateOrders xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Response>
    <QueryResponseHeader>
      <SuccessCode> 2
      (Errors)
      (Error)
        <ErrorCode> 5
        <ErrorMessage> 1000 String
      </Error>
    </Errors>
    <Timestamp> YYYY-MM-DDTHH:MM:SS.000Z
    <ReturnCount> 5
  </QueryResponseHeader>
  (SearchOrderDetails)
  (SearchOrderDetail>
    <OrderID> 50 String
    (BaseOption)? 20 String
    <OrderKind> 10 String
    <RequestKind> 10 String
    (Licenses)? 3
    (OrderRequestDate) 25 YYYY-MM-DDTHH:MM:SS.0002
    (OrderIssueDate) 25 YYYY-MM-DDTHH:MM:SS.0002
    (OrderCanceledDate)? 25 YYYY-MM-DDTHH:MM:SS.0002
    <OrderStatus> 5
      1: INITIAL
      2: Waiting for phishing check
      3: Cancelled - Not Issued
      4: Issue completed
      5: Cancelled - Issued
      6: Waiting for revocation
      7: Revoked
    <OrganizationName> 255 String
    <Months> 4
    <SubId> 50 String
    <FQDN> 64 String
  </SearchOrderDetail>
</SearchOrderDetails>
```
9.6 Get Certificate Orders

The `GetCertificateOrders` API returns a list of orders based on criteria provided.

Note: A bug has been identified; the Coupon tag does not return a value in this API.

9.6.1 GetCertificateOrders Request

```xml
<GetCertificateOrders xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Request>
    <QueryRequestHeader>
      <AuthToken>
        <UserName>
          <Password></Password>
        </UserName>
      </AuthToken>
    </QueryRequestHeader>
    (<FromDate>)? YYYY-MM-DDTHH:MM:SS.000Z
    (<ToDate>)? YYYY-MM-DDTHH:MM:SS.000Z
    (<FQDN>)? 64 String
    (<OrderKind>)? 10 String
    (<OrderStatus>)? 5
      1: INITIAL
      2: Waiting for phishing check
      3: Cancelled - Not Issued
      4: Issue completed
      5: Cancelled - Issued
      6: Waiting for revocation
      7: Revoked
    (<SubID>)? 50 String
  </Request>
</GetCertificateOrders>
```

9.6.2 GetCertificateOrders Response

```xml
<GetCertificateOrders xmlns="http://stub.query.gasapiserver.esp.globalsign.com">
  <Response>
    <QueryResponseHeader>
      <SuccessCode> 2
      <Errors>
        (<Error>
          <ErrorCode> 5</ErrorCode>
          <ErrorMessage> 1000 String</ErrorMessage>
        </Error>)+
      </Errors>
    </QueryResponseHeader>
    <Timestamp> 25 YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    <ReturnCount> 5</ReturnCount>
    (<SearchOrderDetails>
      <SearchOrderDetail>
        <OrderID> 50 String
        (<BaseOption>)? 20 String
        <OrderKind> 10 String
        <RequestKind> 10 String
        <Licenses> 3
        <OrderRequestDate> 25 YYYY-MM-DDTHH:MM:SS.000Z
        <OrderIssueDate>? 25 YYYY-MM-DDTHH:MM:SS.000Z
        (<OrderCanceledDate>)? 25 YYYY-MM-DDTHH:MM:SS.000Z
        <OrderStatus> 5
          1: INITIAL
          2: Waiting for phishing check
          3: Cancelled - Not Issued
          4: Issue completed
          5: Cancelled - Issued
          6: Waiting for revocation
          7: Revoked
        <OrganizationName> 255 String
```

GlobalSign API for SSL Certificates v4.5
<Months> 4 </Months>
<SubId> 50 String </SubId>
<FQDN> 64 String </FQDN>
</SearchOrderDetails>
</Response>
</GetCertificateOrders>

9.7 Reissue Certificates
The Reissue API function re-issues a certificate.

9.7.1 Reissue Request

<ReIssue xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName> 30 String </UserName>
        <Password> 30 String </Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderParameter>
      <CSR> 4000 String <(DNSNames)> 300 String </DNSNames>
    </OrderParameter>
    <TargetOrderID> 50 String </TargetOrderID>
    <HashAlgorithm> SHA1, SHA256 </HashAlgorithm>
  </Request>
</ReIssue>

9.7.2 Reissue Response

<ReIssue xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Response>
    <OrderResponseHeader>
      <SuccessCode> 2 </SuccessCode>
      <Errors>
        <Error>
          <ErrorCode> 5 </ErrorCode>
          <ErrorField> 1000 String </ErrorField>
          <ErrorMessage> 1000 String </ErrorMessage>
        </Error>
      </Errors>
      <Timestamp> YYYY-MM-DDTHH:MM:SS.000Z </Timestamp>
    </OrderResponseHeader>
    <OrderID> 50 String </OrderID>
    <TargetOrderID> 50 String </TargetOrderID>
  </Response>
</ReIssue>

9.8 Validate Order Information
The ValidateOrderParameters API validates all information in the order without placing the order. It also returns the results of any domain checking performed to detect suspicious domains.

9.8.1 ValidateOrderParameters Request

<ValidateOrderParameters xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName> 30 String </UserName>
        <Password> 30 String </Password>
      </AuthToken>
    </OrderRequestHeader>
  </Request>
</ValidateOrderParameters>
<OrderRequestParameter>
  <ProductCode>
    DV, DV_LOW, OV, EV
  </ProductCode>
  <BaseOption>?
    wildcard, globalip
  </BaseOption>
  <OrderKind>
    new, renewal, transfer
  </OrderKind>
  <Licenses>
    3
  </Licenses>
  <Options>
    <Option>
      <OptionName>
        See section 11.11
      </OptionName>
      <OptionValue>
        true, false
      </OptionValue>
    </Option>
  </Options>
  <ValidityPeriod>
    <Months>
      4
    </Months>
    <NotBefore>?
      25 YYYY-MM-DDTHH:MM:SS.000Z
    </NotBefore>
    <NotAfter>?
      25 YYYY-MM-DDTHH:MM:SS.000Z
    </NotAfter>
  </ValidityPeriod>
  <CSR>? 4000 String
  <RenewalTargetOrderID>?
    50 String
  </RenewalTargetOrderID>
  <TargetCERT>?
    4000 String
  </TargetCERT>
  <SpecialInstructions>? 50 String
  <Campaign>? 50 String
  <Coupon>? 50 String
  <FQDN>
    255 String
  </FQDN>
  <SANEntries>
    <SANEntry>
      <SANOptionType>
        1: UC cert option
        2: Subdomain SAN option
        3: GIP SAN option
        4: Internal SAN option
        7: FQDN SAN option
      </SANOptionType>
      <SubjectAltName>
        4000 String
      </SubjectAltName>
      <ModifyOperation>?
        <!-- N/A -->
      </ModifyOperation>
    </SANEntry>
  </SANEntries>
</OrderRequestParameter>

9.8.2 ValidateOrderParameters Response

<ValidateOrderParameters xmlns="http://stub.order.gasapiserver.esp.globalsign.com">
  <Response>
    <OrderResponseHeader>
      <SuccessCode>
        2
      </SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>
            5
          </ErrorCode>
          <ErrorField>? 1000 String
          <ErrorMessage>1000 String
        </Error>
      </Errors>
    </OrderResponseHeader>
    <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    <Price>?
      10
    </Price>
    <Currency>?
      10
    </Currency>
    <ValidityPeriod>
      <Months>
        4
      </Months>
      <NotBefore>?
        YYYY-MM-DDTHH:MM:SS.000Z
      </NotBefore>
      <NotAfter>?
        YYYY-MM-DDTHH:MM:SS.000Z
      </NotAfter>
    </ValidityPeriod>
    <ParsedCSR>
      <Subject>
        <DomainName>
          <Country>
            <Email>
              <Locality>
                <Organization>
                  <OrganizationUnit>
                    <State>
          
</ValidateOrderParameters>
<IsValidDomainName>
</ParsedCSR>)?
</Response>
</ValidateOrderParameters>

9.9 Turn On/Off Renewal Notices
The ToggleRenewalNotice function turns renewal notices on or off.

9.9.1 ToggleRenewalNotice Request
<ToggleRenewalNotice>
<Request>
  <OrderRequestHeader>
    <AuthToken>
      <UserName> 30 String
      <Password> 30 String
    </AuthToken>
  </OrderRequestHeader>
  <OrderID> 30 String
  <RenewalNotice> true, false
</Request>
</ToggleRenewalNotice>

9.9.2 ToggleRenewalNotice Response
<ToggleRenewalNotice>
<Response>
  <OrderResponseHeader>
    <SuccessCode> 2
    <Errors>
      <Error>
        <ErrorCode> 5
        <ErrorField>? 1000 String
        <ErrorMessage> 1000 String
      </Error>
      </Errors>?
    </SuccessCode>
    <Timestamp> YYYY-MM-DDTHH:MM:SS.000Z
  </OrderResponseHeader>
  <OrderID>? 50 String
</Response>
</ToggleRenewalNotice>

9.10 Get SAN Validation Status
The GetOrderSANStatus API returns a list of SANs that need to be validated for OrganizationSSL or ExtendedSSL certificate orders and their current status, as well as the random number to be used for DNS or HTTP domain validation and a link to the domain verification page.

9.10.1 GetOrderSANStatus Request
<GetOrderSANStatus>
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>
        <Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderID>
  </Request>
9.10.2 GetOrderSANStatus Response

In this API the URL to the Domain Verification Page in GCC is returned. This is for GlobalSign Partners to pass on to their customers to execute domain validation on their own without the APIs.

```xml
<GetOrderSANStatusResponse>
  <Response>
    <OrderResponseHeader>
      <SuccessCode>
      <Errors>
        <Timestamp>
      </OrderResponseHeader>
    <OrderID>
      <DomainVerificationPage>
        <DVC>
          <SANList>
            <SANDetails>
              <SAN>
                <SANStatus>
                  <AuthorizationDomainNameList>
                    <AuthorizationDomainName>
                    <AuthorizationDomainName>
                  </AuthorizationDomainNameList>
                  <ValidationDate>
                  <ExpirationDate>
                </SANDetails>
          </SANList>
        </DVC>
        <DomainVerificationPage>
      </OrderID>
    </Response>
  </GetOrderSANStatusResponse>
  The random number for DNS/HTTP domain validation

9.11 Validate SANs

The VerifySAN API allows users to verify a SAN and select the domain validation method (EMAIL, HTTP, DNS). Before using this API, if using HTTP or DNS, place the domain validation random number received in the GetOrderSANStatus API in your website or DNS TXT record; if using EMAIL, you may want to use the GetDVApproverList API to get a list of valid approver email addresses.

9.11.1 VerifySAN Request

In this API, specify which validation method you wish to use in the <VettingType> field (HTTP, DNS, EMAIL).

- If using HTTP or DNS, place the DVC in the appropriate location to be validated, and then specify that location in the <TagLocation> field. Leave the <ApproverEmail> field blank.
- If using EMAIL, leave the <TagLocation> field blank, and enter the email address to receive the validation email in the <ApproverEmail> field.

```xml
<VerifySAN>
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>
        <Password>
      </AuthToken>
    </OrderRequestHeader>
    <OrderID>
      <TagLocation> Where the DVC is stored (for HTTP/DNS validation methods
    <SAN>
      <VettingType> EMAIL, DNS, HTTP
    <ApproverEmail>
  </Request>
```
9.11.2 VerifySAN Response

Note that it is possible to approve multiple SANs in a single order, and therefore the SANList may be longer than one SAN.

```xml
<VerifySANResponse>
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>1000</ErrorField>
          <ErrorMessage>1000</ErrorMessage>
        </Error>
      </Errors>
      <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    </OrderResponseHeader>
    <OrderID>
      <SANList>
        <SAN>
          <SANStatus>
            <AuthorizationDomainNameList>
              <AuthorizationDomainName>
                <ValidationDate>YYYY-MM-DDTHH:MM:SS.000Z</ValidationDate>
                <ExpirationDate>YYYY-MM-DDTHH:MM:SS.000Z</ExpirationDate>
              </AuthorizationDomainName>
            </AuthorizationDomainNameList>
          </SANStatus>
        </SAN>
        <SANList>
          <SAN>
            <SANStatus>
              <AuthorizationDomainNameList>
                <AuthorizationDomainName>
                  <ValidationDate>YYYY-MM-DDTHH:MM:SS.000Z</ValidationDate>
                  <ExpirationDate>YYYY-MM-DDTHH:MM:SS.000Z</ExpirationDate>
                </AuthorizationDomainName>
              </AuthorizationDomainNameList>
            </SANStatus>
          </SAN>
        </SANList>
      </SANList>
    </OrderID>
  </Response>
</VerifySANResponse>
```

10. Account API Functions

10.1 QueryInvoices

The QueryInvoices API retrieves invoices and determines which ones are outstanding.

10.1.1 QueryInvoices Request

```xml
<QueryInvoices>
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName>30 String</UserName>
        <Password>30 String</Password>
      </AuthToken>
    </OrderRequestHeader>
    <InvoiceNumber>? 30 String
    <OrderID>? 30 String
    <PaymentDueDateFrom>YYYY-MM-DDTHH:MM:SS.000Z</PaymentDueDateFrom>
    <PaymentDueDateTo>YYYY-MM-DDTHH:MM:SS.000Z</PaymentDueDateTo>
  </Request>
</QueryInvoices>
```

10.1.2 QueryInvoices Response

```xml
<QueryInvoices>
  <Response>
    <OrderResponseHeader>
      <SuccessCode>2</SuccessCode>
      <Errors>
        <Error>
          <ErrorCode>5</ErrorCode>
          <ErrorField>1000</ErrorField>
          <ErrorMessage>1000</ErrorMessage>
        </Error>
      </Errors>
      <Timestamp>YYYY-MM-DDTHH:MM:SS.000Z</Timestamp>
    </OrderResponseHeader>
    <OrganizationAddress>
      <AddressLine1>100 String</AddressLine1>
      <AddressLine2>? 100 String
      <AddressLine3>? 100 String
    </OrganizationAddress>
  </Response>
</QueryInvoices>
```
<City> 200 String
<Region> 255 String
<PostalCode> 20 String
<Country> 30 String
<Phone> 30 String
<Fax>? 30 String
</OrganizationAddress>
</Invoices>

<InvoiceInformation> 30 String
<PurchaseOrderNumber> 30 String
<CustomerAccountNo> 30 String
<SupplyDate> YYYY-MM-DDTHH:MM:SS.000Z
<InvoiceDate> YYYY-MM-DDTHH:MM:SS.000Z
<PaymentDueDate> YYYY-MM-DDTHH:MM:SS.000Z
</InvoiceInformation>

<InvoiceInformation> 30 String
<InvoiceNumber> 30 String
<PurchaseOrderNumber>? 30 String
</InvoiceInformation>

<BillTo> 255 String
<CommonName> 255 String
<OrderID> 30 String
Qty> 10 String
<UnitPrice> 30 String
<TaxAmt> 30 String
>TotalLineAmt> 30 String
</InvoiceLineItem>

<Net> 30 String
<Tax> 30 String
<InvoiceTotal> 30 String
</InvoiceInformation>
</Invoices>
</Response>
</QueryInvoices>

10.2 GetAgreement

The GetAgreement API allows users to retrieve the current subscriber agreement version and text.

10.2.1 GetAgreement Request

<GetAgreement>
  <Request>
    <OrderRequestHeader>
      <AuthToken>
        <UserName> 30 String
        <Password> 30 String
      </AuthToken>
    </OrderRequestHeader>
    <ProductCode> 30 String
  </Request>
</GetAgreement>

10.2.2 GetAgreement Response

<GetAgreementResponse>
  <Response>
    <OrderResponseHeader>
      <SuccessCode> 2
      <Errors>
        <Error>
          <ErrorCode> 5
          <ErrorField>? 1000 String
          <ErrorMessage> 1000 String
        </Error>
      </Errors>
      <Timestamp> YYYY-MM-DDTHH:MM:SS.000Z
    </OrderResponseHeader>
    <agreementVersion> String
    <agreementText> String
  </Response>
</GetAgreementResponse>
11. Certificate Order Entry Parameters

The following subsections detail parameters for various certificate fields.

11.1 ApproverFQDN

The ApproverFQDN API is used for DNS-based domain approvals. This type of approval is done using an Authorization Domain Name as defined in the CA/B Forum BRs. The following table provides some examples:

<table>
<thead>
<tr>
<th>Common Name</th>
<th>ApproverFQDN Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.b.c.example.com</td>
<td>a.b.c.example.com</td>
</tr>
<tr>
<td></td>
<td>b.c.example.com</td>
</tr>
<tr>
<td></td>
<td>c.example.com</td>
</tr>
<tr>
<td></td>
<td>example.com</td>
</tr>
<tr>
<td>store.example.com</td>
<td>store.example.com</td>
</tr>
<tr>
<td></td>
<td>example.com</td>
</tr>
<tr>
<td>example.com</td>
<td>example.com</td>
</tr>
</tbody>
</table>

11.2 HTTP Validation

The GlobalSign Vetting Department identifies itself with this User-Agent string:

```
GlobalSign-Approver-URL-Domain-Control-Verification-Agent-www.globalsign.com
```

Based on the SAN being validated, the Vetting Agent will only accept certain locations as valid. The following table provides some examples:

<table>
<thead>
<tr>
<th>Requested Domain</th>
<th>Valid &lt;ApproverURL&gt; Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.example.com or example.com</td>
<td>http(s)://example.com</td>
</tr>
<tr>
<td>*.sub.example.com or sub.example.com</td>
<td>http(s)://example.com</td>
</tr>
<tr>
<td>*.www.example.com or <a href="http://www.example.com">www.example.com</a></td>
<td>http(s)://example.com</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is an example metatag value:

```
<meta name="globalsign-domain-verification" content="8Aetu7b1LEMGdrwZD069ghBGZ-Szq5Md93_DpS44Iq" />
```

**Validity Period of the Random value:** The Domain Validation Code (DVC) is valid for 30 days. After that point the order must be cancelled and a new order placed.

GlobalSign regularly updates validation requirements on its Support webpage. You can review the latest HTTP-based validation requirements at this link: https://support.globalsign.com/customer/en/portal/topics/1015848
11.3 Base Options

The **BaseOptions** command is used to add a certain attribute to certificate orders. Specific types of certificates accept certain **BaseOptions**. Use the below table for the list of **BaseOptions** available by certificate type.

<table>
<thead>
<tr>
<th>BaseOption (API Value)</th>
<th>Supported Product</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>wildcard</td>
<td>AlphaSSL</td>
<td>The CommonName must contain the * character in the first character of the CommonName.</td>
</tr>
<tr>
<td></td>
<td>DomainSSL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OrganizationSSL</td>
<td></td>
</tr>
<tr>
<td>globalip</td>
<td>OrganizationSSL</td>
<td>The CommonName must contain a publicly accessible IP address. If this option is used then the Subdomain and UC SAN options cannot be used. Certificates needing subdomains with IP addresses should not use BaseOptions and then add the IP address via the GIP SAN option.</td>
</tr>
</tbody>
</table>

11.4 Country

The following table lists two-character country codes and if the code is currently supported (**Y** = Supported, **N** = Not Supported).

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>ANDORRA</td>
<td>Y</td>
</tr>
<tr>
<td>AE</td>
<td>UNITED ARAB EMIRATES</td>
<td>Y</td>
</tr>
<tr>
<td>AF</td>
<td>AFGHANISTAN</td>
<td>N</td>
</tr>
<tr>
<td>AG</td>
<td>ANTIGUA AND BARBUDA</td>
<td>Y</td>
</tr>
<tr>
<td>AI</td>
<td>ANGUILLA</td>
<td>Y</td>
</tr>
<tr>
<td>AL</td>
<td>ALBANIA</td>
<td>Y</td>
</tr>
<tr>
<td>AM</td>
<td>ARMENIA</td>
<td>Y</td>
</tr>
<tr>
<td>AN</td>
<td>NL ANTILLES (USE CW or SX)</td>
<td>N</td>
</tr>
<tr>
<td>AO</td>
<td>ANGOLA</td>
<td>N</td>
</tr>
<tr>
<td>AQ</td>
<td>ANTARCTICA</td>
<td>Y</td>
</tr>
<tr>
<td>AR</td>
<td>ARGENTINA</td>
<td>Y</td>
</tr>
<tr>
<td>AS</td>
<td>AMERICAN SAMOA</td>
<td>Y</td>
</tr>
<tr>
<td>AT</td>
<td>AUSTRIA</td>
<td>Y</td>
</tr>
<tr>
<td>AU</td>
<td>AUSTRALIA</td>
<td>Y</td>
</tr>
<tr>
<td>AW</td>
<td>ARUBA</td>
<td>Y</td>
</tr>
<tr>
<td>AX</td>
<td>ALAND ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>AZ</td>
<td>AZERBAIJAN</td>
<td>Y</td>
</tr>
<tr>
<td>BA</td>
<td>BOSNIA AND HERZEGOVINA</td>
<td>Y</td>
</tr>
<tr>
<td>BB</td>
<td>BARBADOS</td>
<td>Y</td>
</tr>
<tr>
<td>BD</td>
<td>BANGLADESH</td>
<td>Y</td>
</tr>
<tr>
<td>BE</td>
<td>BELGIUM</td>
<td>Y</td>
</tr>
<tr>
<td>BF</td>
<td>BURKINA FASO</td>
<td>Y</td>
</tr>
<tr>
<td>BG</td>
<td>BULGARIA</td>
<td>Y</td>
</tr>
<tr>
<td>BH</td>
<td>BAHRAIN</td>
<td>Y</td>
</tr>
<tr>
<td>BI</td>
<td>BURUNDI</td>
<td>Y</td>
</tr>
<tr>
<td>BJ</td>
<td>BENIN</td>
<td>Y</td>
</tr>
<tr>
<td>BM</td>
<td>BERMUDA</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>BRUNEI DARUSSALAM</td>
<td>Y</td>
</tr>
<tr>
<td>BO</td>
<td>BOLIVIA</td>
<td>Y</td>
</tr>
<tr>
<td>BR</td>
<td>BRAZIL</td>
<td>Y</td>
</tr>
<tr>
<td>BS</td>
<td>BAHAMAS</td>
<td>Y</td>
</tr>
<tr>
<td>BT</td>
<td>BHUTAN</td>
<td>Y</td>
</tr>
<tr>
<td>BV</td>
<td>BOUVET ISLAND</td>
<td>Y</td>
</tr>
<tr>
<td>BW</td>
<td>BOTSWANA</td>
<td>Y</td>
</tr>
<tr>
<td>BY</td>
<td>BELARUS</td>
<td>Y</td>
</tr>
<tr>
<td>BZ</td>
<td>BELIZE</td>
<td>Y</td>
</tr>
<tr>
<td>CA</td>
<td>CANADA</td>
<td>Y</td>
</tr>
<tr>
<td>CC</td>
<td>COCOS (KEELING) ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>CD</td>
<td>CONGO, THE DEMOCRATIC REPUBLIC OF THE</td>
<td>Y</td>
</tr>
<tr>
<td>CF</td>
<td>CENTRAL AFRICAN REPUBLIC</td>
<td>Y</td>
</tr>
<tr>
<td>CG</td>
<td>CONGO</td>
<td>Y</td>
</tr>
<tr>
<td>CH</td>
<td>SWITZERLAND</td>
<td>Y</td>
</tr>
<tr>
<td>CI</td>
<td>COTE D'IVOIRE</td>
<td>Y</td>
</tr>
<tr>
<td>CK</td>
<td>COOK ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>CL</td>
<td>CHILE</td>
<td>Y</td>
</tr>
<tr>
<td>CM</td>
<td>CAMEROON</td>
<td>Y</td>
</tr>
<tr>
<td>CN</td>
<td>CHINA</td>
<td>Y</td>
</tr>
<tr>
<td>CO</td>
<td>COLOMBIA</td>
<td>Y</td>
</tr>
<tr>
<td>CR</td>
<td>COSTA RICA</td>
<td>Y</td>
</tr>
<tr>
<td>CU</td>
<td>CUBA</td>
<td>N</td>
</tr>
<tr>
<td>CV</td>
<td>CAPE VERDE</td>
<td>Y</td>
</tr>
<tr>
<td>CW</td>
<td>CURACAO</td>
<td>Y</td>
</tr>
<tr>
<td>CX</td>
<td>CHRISTMAS ISLAND</td>
<td>Y</td>
</tr>
<tr>
<td>CY</td>
<td>CYPRUS</td>
<td>Y</td>
</tr>
<tr>
<td>Code</td>
<td>Name</td>
<td>Status</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>CZ</td>
<td>CZECH REPUBLIC</td>
<td>Y</td>
</tr>
<tr>
<td>DE</td>
<td>GERMANY</td>
<td>Y</td>
</tr>
<tr>
<td>DJ</td>
<td>DJIBOUTI</td>
<td>Y</td>
</tr>
<tr>
<td>DK</td>
<td>DENMARK</td>
<td>Y</td>
</tr>
<tr>
<td>DM</td>
<td>DOMINICA</td>
<td>Y</td>
</tr>
<tr>
<td>DO</td>
<td>DOMINICAN REPUBLIC</td>
<td>Y</td>
</tr>
<tr>
<td>DZ</td>
<td>ALGERIA</td>
<td>Y</td>
</tr>
<tr>
<td>EC</td>
<td>ECUADOR</td>
<td>Y</td>
</tr>
<tr>
<td>EE</td>
<td>ESTONIA</td>
<td>Y</td>
</tr>
<tr>
<td>EG</td>
<td>EGYPT</td>
<td>Y</td>
</tr>
<tr>
<td>EH</td>
<td>WESTERN SAHARA</td>
<td>Y</td>
</tr>
<tr>
<td>ER</td>
<td>ERITREA</td>
<td>Y</td>
</tr>
<tr>
<td>ES</td>
<td>SPAIN</td>
<td>Y</td>
</tr>
<tr>
<td>ET</td>
<td>ETHIOPIA</td>
<td>Y</td>
</tr>
<tr>
<td>FI</td>
<td>FINLAND</td>
<td>Y</td>
</tr>
<tr>
<td>FJ</td>
<td>FIJI</td>
<td>Y</td>
</tr>
<tr>
<td>FK</td>
<td>FALKLAND ISLANDS (MALVINAS)</td>
<td>Y</td>
</tr>
<tr>
<td>FM</td>
<td>MICRONESIA, FEDERATED STATES</td>
<td>Y</td>
</tr>
<tr>
<td>FO</td>
<td>FAROE ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>FR</td>
<td>FRANCE</td>
<td>Y</td>
</tr>
<tr>
<td>GA</td>
<td>GABON</td>
<td>Y</td>
</tr>
<tr>
<td>GB</td>
<td>UNITED KINGDOM</td>
<td>Y</td>
</tr>
<tr>
<td>GD</td>
<td>GRENAADA</td>
<td>Y</td>
</tr>
<tr>
<td>GE</td>
<td>GEORGIA</td>
<td>Y</td>
</tr>
<tr>
<td>GF</td>
<td>FRENCH GUIANA</td>
<td>Y</td>
</tr>
<tr>
<td>GG</td>
<td>GUERNSEY</td>
<td>Y</td>
</tr>
<tr>
<td>GH</td>
<td>GHANA</td>
<td>Y</td>
</tr>
<tr>
<td>GI</td>
<td>GIBRALTAR</td>
<td>Y</td>
</tr>
<tr>
<td>GL</td>
<td>GREENLAND</td>
<td>Y</td>
</tr>
<tr>
<td>GM</td>
<td>GAMBIA</td>
<td>Y</td>
</tr>
<tr>
<td>GN</td>
<td>GUINEA</td>
<td>Y</td>
</tr>
<tr>
<td>GP</td>
<td>GUADALOUPE</td>
<td>Y</td>
</tr>
<tr>
<td>GQ</td>
<td>EQUATORIAL GUINEA</td>
<td>Y</td>
</tr>
<tr>
<td>GR</td>
<td>GREECE</td>
<td>Y</td>
</tr>
<tr>
<td>GS</td>
<td>SOUTH GEORGIA AND THE SOUTH</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>SANDWICH ISLANDS</td>
<td></td>
</tr>
<tr>
<td>GT</td>
<td>GUATEMALA</td>
<td>Y</td>
</tr>
<tr>
<td>GU</td>
<td>GUAM</td>
<td>Y</td>
</tr>
<tr>
<td>GW</td>
<td>GUINEA-BISSAU</td>
<td>Y</td>
</tr>
<tr>
<td>GY</td>
<td>GUYANA</td>
<td>Y</td>
</tr>
<tr>
<td>HK</td>
<td>HONG KONG</td>
<td>Y</td>
</tr>
<tr>
<td>HM</td>
<td>HEARD ISLAND AND MCDONALD</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>ISLANDS</td>
<td></td>
</tr>
<tr>
<td>HN</td>
<td>HONDURAS</td>
<td>Y</td>
</tr>
<tr>
<td>HR</td>
<td>CROATIA</td>
<td>Y</td>
</tr>
<tr>
<td>HT</td>
<td>HAITI</td>
<td>Y</td>
</tr>
<tr>
<td>HU</td>
<td>HUNGARY</td>
<td>Y</td>
</tr>
<tr>
<td>ID</td>
<td>INDONESIA</td>
<td>Y</td>
</tr>
<tr>
<td>IE</td>
<td>IRELAND</td>
<td>Y</td>
</tr>
<tr>
<td>IL</td>
<td>ISRAEL</td>
<td>Y</td>
</tr>
<tr>
<td>IM</td>
<td>ISLE OF MAN</td>
<td>Y</td>
</tr>
<tr>
<td>IN</td>
<td>INDIA</td>
<td>Y</td>
</tr>
<tr>
<td>IO</td>
<td>BRITISH INDIAN OCEAN Territory</td>
<td>Y</td>
</tr>
<tr>
<td>IQ</td>
<td>IRAQ</td>
<td>N</td>
</tr>
<tr>
<td>IR</td>
<td>IRAN, ISLAMIC REPUBLIC OF</td>
<td>N</td>
</tr>
<tr>
<td>IS</td>
<td>ICELAND</td>
<td>Y</td>
</tr>
<tr>
<td>IT</td>
<td>ITALY</td>
<td>Y</td>
</tr>
<tr>
<td>JE</td>
<td>JERSEY</td>
<td>Y</td>
</tr>
<tr>
<td>JM</td>
<td>JAMAICA</td>
<td>Y</td>
</tr>
<tr>
<td>JO</td>
<td>JORDAN</td>
<td>Y</td>
</tr>
<tr>
<td>JP</td>
<td>JAPAN</td>
<td>Y</td>
</tr>
<tr>
<td>KE</td>
<td>KENYA</td>
<td>Y</td>
</tr>
<tr>
<td>KG</td>
<td>KYRGYZSTAN</td>
<td>Y</td>
</tr>
<tr>
<td>KH</td>
<td>CAMBODIA</td>
<td>Y</td>
</tr>
<tr>
<td>KO</td>
<td>KIRIBATI</td>
<td>Y</td>
</tr>
<tr>
<td>KM</td>
<td>COMOROS</td>
<td>Y</td>
</tr>
<tr>
<td>KN</td>
<td>SAINT KITTS AND NEVIS</td>
<td>Y</td>
</tr>
<tr>
<td>KP</td>
<td>NORTH KOREA (DEMOCRATIC PEOPLE’S REPUBLIC OF KOREA)</td>
<td>N</td>
</tr>
<tr>
<td>KR</td>
<td>KOREA, REPUBLIC OF</td>
<td>Y</td>
</tr>
<tr>
<td>KW</td>
<td>KUWAIT</td>
<td>Y</td>
</tr>
<tr>
<td>KY</td>
<td>CAYMAN ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>KZ</td>
<td>KAZAKSTAN</td>
<td>Y</td>
</tr>
<tr>
<td>LA</td>
<td>LAO PEOPLE’S DEMOCRATIC</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>REPUBLIC</td>
<td></td>
</tr>
<tr>
<td>LB</td>
<td>LEBANON</td>
<td>Y</td>
</tr>
<tr>
<td>LC</td>
<td>SAINT LUCIA</td>
<td>Y</td>
</tr>
<tr>
<td>LI</td>
<td>LIECHTENSTEIN</td>
<td>Y</td>
</tr>
<tr>
<td>LK</td>
<td>SRI LANKA</td>
<td>Y</td>
</tr>
<tr>
<td>LR</td>
<td>LIBERIA</td>
<td>N</td>
</tr>
<tr>
<td>LS</td>
<td>LESOTHO</td>
<td>Y</td>
</tr>
<tr>
<td>LT</td>
<td>LITHUANIA</td>
<td>Y</td>
</tr>
<tr>
<td>LU</td>
<td>LUXEMBOURG</td>
<td>Y</td>
</tr>
<tr>
<td>LV</td>
<td>LATVIA</td>
<td>Y</td>
</tr>
<tr>
<td>LY</td>
<td>LIBYAN ARAB JAMAHIRIYA</td>
<td>N</td>
</tr>
<tr>
<td>MA</td>
<td>MOROCCO</td>
<td>Y</td>
</tr>
<tr>
<td>MC</td>
<td>MONACO</td>
<td>Y</td>
</tr>
<tr>
<td>MD</td>
<td>MOLDOVA, REPUBLIC OF</td>
<td>Y</td>
</tr>
<tr>
<td>ME</td>
<td>MONTENEGRO</td>
<td>N</td>
</tr>
<tr>
<td>MG</td>
<td>MADAGASCAR</td>
<td>Y</td>
</tr>
<tr>
<td>MH</td>
<td>MARSHALL ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>Code</td>
<td>Name</td>
<td>Status</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>MK</td>
<td>MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF</td>
<td>Y</td>
</tr>
<tr>
<td>ML</td>
<td>MALI</td>
<td>Y</td>
</tr>
<tr>
<td>MM</td>
<td>MYANMAR</td>
<td>Y</td>
</tr>
<tr>
<td>MN</td>
<td>MONGOLIA</td>
<td>Y</td>
</tr>
<tr>
<td>MO</td>
<td>MACAU</td>
<td>Y</td>
</tr>
<tr>
<td>MP</td>
<td>NORTHERN MARIANA ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>MQ</td>
<td>MARTINIQUE</td>
<td>Y</td>
</tr>
<tr>
<td>MR</td>
<td>MAURITANIA</td>
<td>Y</td>
</tr>
<tr>
<td>MS</td>
<td>MONTSERRAT</td>
<td>Y</td>
</tr>
<tr>
<td>MT</td>
<td>MALTA</td>
<td>Y</td>
</tr>
<tr>
<td>MU</td>
<td>MAURITIUS</td>
<td>Y</td>
</tr>
<tr>
<td>MV</td>
<td>MALDIVES</td>
<td>Y</td>
</tr>
<tr>
<td>MW</td>
<td>MALAWI</td>
<td>Y</td>
</tr>
<tr>
<td>MX</td>
<td>MEXICO</td>
<td>Y</td>
</tr>
<tr>
<td>MY</td>
<td>MALAYSIA</td>
<td>Y</td>
</tr>
<tr>
<td>MZ</td>
<td>MOZAMBIQUE</td>
<td>Y</td>
</tr>
<tr>
<td>NA</td>
<td>NAMIBIA</td>
<td>Y</td>
</tr>
<tr>
<td>NC</td>
<td>NEW CALEDONIA</td>
<td>Y</td>
</tr>
<tr>
<td>NE</td>
<td>NIGER</td>
<td>Y</td>
</tr>
<tr>
<td>NF</td>
<td>NORFOLK ISLAND</td>
<td>Y</td>
</tr>
<tr>
<td>NG</td>
<td>NIGERIA</td>
<td>Y</td>
</tr>
<tr>
<td>NI</td>
<td>NICARAGUA</td>
<td>Y</td>
</tr>
<tr>
<td>NL</td>
<td>NETHERLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>NO</td>
<td>NORWAY</td>
<td>Y</td>
</tr>
<tr>
<td>NP</td>
<td>NEPAL</td>
<td>Y</td>
</tr>
<tr>
<td>NR</td>
<td>NAURU</td>
<td>Y</td>
</tr>
<tr>
<td>NU</td>
<td>NIUE</td>
<td>Y</td>
</tr>
<tr>
<td>NZ</td>
<td>NEW ZEALAND</td>
<td>Y</td>
</tr>
<tr>
<td>OM</td>
<td>OMAN</td>
<td>Y</td>
</tr>
<tr>
<td>PA</td>
<td>PANAMA</td>
<td>Y</td>
</tr>
<tr>
<td>PE</td>
<td>PERU</td>
<td>Y</td>
</tr>
<tr>
<td>PF</td>
<td>FRENCH POLYNESIA</td>
<td>Y</td>
</tr>
<tr>
<td>PG</td>
<td>PAPUA NEW GUINEA</td>
<td>Y</td>
</tr>
<tr>
<td>PH</td>
<td>PHILIPPINES</td>
<td>Y</td>
</tr>
<tr>
<td>PK</td>
<td>PAKISTAN</td>
<td>Y</td>
</tr>
<tr>
<td>PL</td>
<td>POLAND</td>
<td>Y</td>
</tr>
<tr>
<td>PM</td>
<td>SAINT PIERRE AND MIQUELON</td>
<td>Y</td>
</tr>
<tr>
<td>PN</td>
<td>PITCAIRN</td>
<td>Y</td>
</tr>
<tr>
<td>PR</td>
<td>PUERTO RICO</td>
<td>Y</td>
</tr>
<tr>
<td>PS</td>
<td>PALESTINIAN TERRITORY, OCCUPIED</td>
<td>Y</td>
</tr>
<tr>
<td>PT</td>
<td>PORTUGAL</td>
<td>Y</td>
</tr>
<tr>
<td>PW</td>
<td>PALAU</td>
<td>Y</td>
</tr>
<tr>
<td>PY</td>
<td>PARAGUAY</td>
<td>Y</td>
</tr>
<tr>
<td>QA</td>
<td>QATAR</td>
<td>Y</td>
</tr>
<tr>
<td>RE</td>
<td>REUNION</td>
<td>Y</td>
</tr>
<tr>
<td>RO</td>
<td>ROMANIA</td>
<td>Y</td>
</tr>
<tr>
<td>RS</td>
<td>SERBIA</td>
<td>N</td>
</tr>
<tr>
<td>RU</td>
<td>RUSSIAN FEDERATION</td>
<td>Y</td>
</tr>
<tr>
<td>RW</td>
<td>RWANDA</td>
<td>N</td>
</tr>
<tr>
<td>SA</td>
<td>SAUDI ARABIA</td>
<td>Y</td>
</tr>
<tr>
<td>SB</td>
<td>SOLOMON ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>SC</td>
<td>SEYCHELLES</td>
<td>Y</td>
</tr>
<tr>
<td>SD</td>
<td>SUDAN</td>
<td>N</td>
</tr>
<tr>
<td>SE</td>
<td>SWEDEN</td>
<td>Y</td>
</tr>
<tr>
<td>SG</td>
<td>SINGAPORE</td>
<td>Y</td>
</tr>
<tr>
<td>SH</td>
<td>SAINT HELENA</td>
<td>Y</td>
</tr>
<tr>
<td>SI</td>
<td>SLOVENIA</td>
<td>Y</td>
</tr>
<tr>
<td>SJ</td>
<td>SVALBARD AND JAN MAYEN</td>
<td>Y</td>
</tr>
<tr>
<td>SK</td>
<td>SLOVAKIA</td>
<td>Y</td>
</tr>
<tr>
<td>SL</td>
<td>SIERRA LEONE</td>
<td>N</td>
</tr>
<tr>
<td>SM</td>
<td>SAN MARINO</td>
<td>Y</td>
</tr>
<tr>
<td>SN</td>
<td>SENEGAL</td>
<td>Y</td>
</tr>
<tr>
<td>SO</td>
<td>SOMALIA</td>
<td>N</td>
</tr>
<tr>
<td>SR</td>
<td>SURiname</td>
<td>Y</td>
</tr>
<tr>
<td>ST</td>
<td>SAO TOME AND PRINCIPE</td>
<td>Y</td>
</tr>
<tr>
<td>SV</td>
<td>EL SALVADOR</td>
<td>Y</td>
</tr>
<tr>
<td>SX</td>
<td>SINT MAARTEN</td>
<td>Y</td>
</tr>
<tr>
<td>SY</td>
<td>SYRIAN ARAB REPUBLIC</td>
<td>N</td>
</tr>
<tr>
<td>SZ</td>
<td>SWAZILAND</td>
<td>Y</td>
</tr>
<tr>
<td>TC</td>
<td>TURKS AND CAICOS ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>TD</td>
<td>CHAD</td>
<td>Y</td>
</tr>
<tr>
<td>TF</td>
<td>FRENCH SOUTHERN TERRITORIES</td>
<td>Y</td>
</tr>
<tr>
<td>TG</td>
<td>TOGO</td>
<td>Y</td>
</tr>
<tr>
<td>TH</td>
<td>THAILAND</td>
<td>Y</td>
</tr>
<tr>
<td>TJ</td>
<td>TAJIKISTAN</td>
<td>Y</td>
</tr>
<tr>
<td>TK</td>
<td>TOKELAU</td>
<td>Y</td>
</tr>
<tr>
<td>TL</td>
<td>TIMOR-LESTE</td>
<td>Y</td>
</tr>
<tr>
<td>TM</td>
<td>TURKMENISTAN</td>
<td>Y</td>
</tr>
<tr>
<td>TN</td>
<td>TUNISIA</td>
<td>Y</td>
</tr>
<tr>
<td>TO</td>
<td>TONGA</td>
<td>Y</td>
</tr>
<tr>
<td>TR</td>
<td>TURKEY</td>
<td>Y</td>
</tr>
<tr>
<td>TT</td>
<td>TRINIDAD AND TOBAGO</td>
<td>Y</td>
</tr>
<tr>
<td>TV</td>
<td>TUVALU</td>
<td>Y</td>
</tr>
<tr>
<td>TW</td>
<td>TAIWAN, PROVINCE OF CHINA</td>
<td>Y</td>
</tr>
<tr>
<td>TZ</td>
<td>TANZANIA, UNITED REPUBLIC OF</td>
<td>Y</td>
</tr>
<tr>
<td>UA</td>
<td>UKRAINE</td>
<td>Y</td>
</tr>
<tr>
<td>UG</td>
<td>UGANDA</td>
<td>Y</td>
</tr>
<tr>
<td>UM</td>
<td>UNITED STATES MINOR OUTLYING ISLANDS</td>
<td>Y</td>
</tr>
<tr>
<td>US</td>
<td>UNITED STATES</td>
<td>Y</td>
</tr>
<tr>
<td>Code</td>
<td>Name</td>
<td>Status</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>UY</td>
<td>URUGUAY</td>
<td>Y</td>
</tr>
<tr>
<td>UZ</td>
<td>UZBEKISTAN</td>
<td>Y</td>
</tr>
<tr>
<td>VA</td>
<td>HOLY SEE (VATICAN CITY STATE)</td>
<td>Y</td>
</tr>
<tr>
<td>VC</td>
<td>SAINT VINCENT AND THE GRENADINES</td>
<td>Y</td>
</tr>
<tr>
<td>VE</td>
<td>VENEZUELA</td>
<td>Y</td>
</tr>
<tr>
<td>VG</td>
<td>VIRGIN ISLANDS, BRITISH</td>
<td>Y</td>
</tr>
<tr>
<td>VI</td>
<td>VIRGIN ISLANDS, U.S.</td>
<td>Y</td>
</tr>
<tr>
<td>VN</td>
<td>VIET NAM</td>
<td>Y</td>
</tr>
<tr>
<td>VU</td>
<td>VANUATU</td>
<td>Y</td>
</tr>
<tr>
<td>WF</td>
<td>WALLIS AND FUTUNA</td>
<td>Y</td>
</tr>
<tr>
<td>WS</td>
<td>SAMOA</td>
<td>Y</td>
</tr>
<tr>
<td>YE</td>
<td>YEMEN</td>
<td>Y</td>
</tr>
<tr>
<td>YT</td>
<td>MAYOTTE</td>
<td>Y</td>
</tr>
<tr>
<td>ZA</td>
<td>SOUTH AFRICA</td>
<td>Y</td>
</tr>
<tr>
<td>ZM</td>
<td>ZAMBIA</td>
<td>Y</td>
</tr>
<tr>
<td>ZW</td>
<td>ZIMBABWE</td>
<td>Y</td>
</tr>
</tbody>
</table>

### 11.5 CreditAgency/OrganizationCode

**CreditAgency/OrganizationCode** is added to help the GlobalSign Vetting Department validate the customer’s organization. If the customer has one of these numbers it should be flagged as available; the actual code is not to be entered.

<table>
<thead>
<tr>
<th>Value</th>
<th>Credit Agency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dunn and Bradstreet number</td>
</tr>
<tr>
<td>2</td>
<td>Teikoku Databank code</td>
</tr>
</tbody>
</table>

### 11.6 Custom Expiration Date

Users can control when a certificate expires by setting the notAfter date. The notAfter date cannot be any further in the future than the current date plus the requested validity period, plus any added validity because of renewal, trade-in, etc. If the notAfter date is set incorrectly, the order will be rejected.
not After dates should be set according to the following table:

<table>
<thead>
<tr>
<th>Number of Months</th>
<th>Restrictions for Not After Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>• DomainSSL: 4 days after the ordering date and within 184 days</td>
</tr>
<tr>
<td></td>
<td>• OrganizationSSL: 8 days after the ordering date and within 184 days</td>
</tr>
<tr>
<td>12</td>
<td>185 days after the ordering date and within 366 days</td>
</tr>
<tr>
<td>24</td>
<td>367 days after the ordering date and within 731 days.</td>
</tr>
<tr>
<td></td>
<td>All products have a maximum validity period of 825 days.</td>
</tr>
</tbody>
</table>

11.7 Date/Time Formatting
The Date/Time of a certificate is based on UTC and includes milliseconds (e.g., 2006-12-07T18:33.594Z). This format is defined as “xsd:dateTime XML Simple Type.” Refer to [http://www.w3.org/TR/xmlschema-2/#dateTime](http://www.w3.org/TR/xmlschema-2/#dateTime) for more information.

11.8 Extensions
Extensions allow users to assign OIDs to a certificate. Some of the most common extensions assigned to certificates include:

- ExtendedKeyUsage
- KeyUsage
- TLSFeature (future)

These options use name value pairs, where the “Name” is the name of the extension (ExtendedKeyUsage, KeyUsage or TLSFeature) and the value is the value to be placed in the certificate. Multiple name value pairs will be used to specify multiple values for an extension (e.g., to request Server Auth and Client Auth in the EKU extension you would need to supply two name value pairs in the API).

11.8.1 Extension to Product Map
The values permitted for your account will depend on how each of the products are configured within your account (Default, Basic or Advanced). If you need to specify certain values in any of the supported extensions listed in the below table, please contact your account manager or GlobalSign Support and ask that the configuration be updated for the product(s) needing this flexibility.

- The Default behavior for all SSL products is listed in the Default column
- OV (and EV in the future) have the Basic and Advanced options as listed below

<table>
<thead>
<tr>
<th>Name of OID/Bit Position of EKU</th>
<th>Name of OID/Bit Position of EKU</th>
<th>Default (all)</th>
<th>Basic (OV-only)</th>
<th>Advanced (OV-only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.6.1.5.5.7.3.1</td>
<td>Server Authentication</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1.3.6.1.5.5.7.3.2</td>
<td>Client Authentication</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1.3.6.1.5.5.7.3.4</td>
<td>Email Protection (S/MIME)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
### 1.3.6.1.5.5.7.3.5 IPSec End System

<table>
<thead>
<tr>
<th>KeyUsage (RSA keys)</th>
<th>digitalSignature</th>
<th>keyEncipherment</th>
<th>dataEncipherment</th>
<th>keyAgreement</th>
<th>encipherOnly</th>
<th>decipherOnly</th>
<th>Any Extended Key Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>digitalSignature</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>keyEncipherment</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dataEncipherment</td>
<td>3</td>
<td>X</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>keyAgreement</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>encipherOnly</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>decipherOnly</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 1.3.6.1.5.5.7.3.6 IPSec Tunnel

<table>
<thead>
<tr>
<th>KeyUsage (ECC keys)</th>
<th>digitalSignature</th>
<th>keyEncipherment</th>
<th>dataEncipherment</th>
<th>keyAgreement</th>
<th>encipherOnly</th>
<th>decipherOnly</th>
<th>Any Extended Key Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>digitalSignature</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>keyEncipherment</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dataEncipherment</td>
<td>3</td>
<td>X</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>keyAgreement</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>encipherOnly</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>decipherOnly</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 1.3.6.1.5.5.7.3.7 IPSec User

### 1.3.6.1.5.5.7.3.17 id-tp-ipseclKE

### 1.3.6.1.4.1.311.20.2.2 Smart Card Logon

### 1.3.6.1.4.1.311.21.5 Key Archival

### 1.3.6.1.5.5.8.2.2 IKE Intermediate

### 2.5.29.37.0 Any Extended Key Usage

#### 11.9 KeyLength

The **KeyLength** reflects the Key Length to be used if the certificate is being created on GlobalSign servers. Valid values are 2048 or 4096.

#### 11.10 Licenses

GlobalSign uses an unlimited server licensing model for its SSL certificates. The values passed through the API in the **License** field are thus irrelevant. For new implementations, the request should be configured to submit a value of 1 licenses.

#### 11.11 OptionName

If a certificate has extended options, the certificate should specify an **OptionName**. The below table lists available OptionNames; set the value to "True" to activate.

<table>
<thead>
<tr>
<th>OptionName (API Values)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAN</td>
<td>The SAN option activates the Subject Alternative Name (SAN) option – see Section 8.2.</td>
</tr>
<tr>
<td>VPC</td>
<td>The Validity Period Customize (VPC) option allows the start date and end date of the Certificate to be customized – see Section 11.6.</td>
</tr>
<tr>
<td>REX</td>
<td>The Renewal Extension (REX) option will add 30 days to renewal orders.</td>
</tr>
</tbody>
</table>
11.12 OrderType/OrderKind
Customers can place different types of orders through the API. To set the type of order being ordered through the API, use one of the following OrderTypes. These types of orders apply for all products (EV, OV, DV, Alpha).

<table>
<thead>
<tr>
<th>No.</th>
<th>OrderKind (API Value)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>new</td>
<td>A new order</td>
</tr>
<tr>
<td>2</td>
<td>renewal</td>
<td>A renewal order for replacing an expiring certificate</td>
</tr>
<tr>
<td>3</td>
<td>transfer</td>
<td>A competitive switch – a certificate is being traded in from another SSL provider</td>
</tr>
</tbody>
</table>

11.13 Product Codes
All products offered by GlobalSign have a product code. The product code is a necessary item in most API calls, and specifies the type of product the customer is placing. The following table lists all GlobalSign Product Codes by product type.

<table>
<thead>
<tr>
<th>ProductCode</th>
<th>Certificate Type</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>EV*</td>
<td>ExtendedSSL</td>
<td>Used in EVOrder</td>
</tr>
<tr>
<td>EV_SHA2</td>
<td>ExtendedSSL</td>
<td>Used in EVOrder</td>
</tr>
<tr>
<td>OV*</td>
<td>OrganizationSSL</td>
<td>Used in OVOrder</td>
</tr>
<tr>
<td>OV_SHA2</td>
<td>OrganizationSSL</td>
<td>Used in OVOrder</td>
</tr>
<tr>
<td>DV*</td>
<td>DomainSSL</td>
<td>Used in DVOrder (email)</td>
</tr>
<tr>
<td>DV_SHA2</td>
<td>DomainSSL</td>
<td>Used in DVOrder (email)</td>
</tr>
<tr>
<td>DV_HIGH_URL*</td>
<td>DomainSSL</td>
<td>Used in URLVerification (HTTP/metatag)</td>
</tr>
<tr>
<td>DV_HIGH_URL_SHA2</td>
<td>DomainSSL</td>
<td>Used in URLVerification (HTTP/metatag)</td>
</tr>
<tr>
<td>DV_HIGH_DNS*</td>
<td>DomainSSL</td>
<td>Used in URLVerification (DNS TXT)</td>
</tr>
<tr>
<td>DV_HIGH_DNS_SHA2</td>
<td>DomainSSL</td>
<td>Used in URLVerification (DNS TXT)</td>
</tr>
<tr>
<td>DV_LOW*</td>
<td>AlphaSSL</td>
<td>Used in DVOrder</td>
</tr>
<tr>
<td>DV_LOW_SHA2</td>
<td>AlphaSSL</td>
<td></td>
</tr>
<tr>
<td>DV_LOW_URL*</td>
<td>AlphaSSL</td>
<td>Used in URLVerification (metatag)</td>
</tr>
<tr>
<td>DV_LOW_URL_SHA2</td>
<td>AlphaSSL</td>
<td></td>
</tr>
<tr>
<td>DV_LOW_DNS*</td>
<td>AlphaSSL</td>
<td>Used in URLVerification (DNS txt)</td>
</tr>
<tr>
<td>DV_LOW_DNS_SHA2</td>
<td>AlphaSSL</td>
<td></td>
</tr>
</tbody>
</table>

Items marked with a '*' identify legacy product codes that were used for SHA-1 issuance. Do not use these product codes.

11.14 Subject Alternative Names (SANs) Entry
SANs specify additional host names (sites, IP addresses, common names, etc.) to be protected by a single SSL Certificate. When SANs are added to a certificate, set OptionName to “SAN” and SANOptionType to one of the following, along with the full SAN in SubjectAltName.

<table>
<thead>
<tr>
<th>API Value</th>
<th>Explanation</th>
<th>EV</th>
<th>OV</th>
<th>DV</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Unified Communication Cert Option – allows www, owa, mail or autodiscover subdomains to be added for free</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Subdomain – The value is a subdomain of the domain in the CommonName</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Public IP Option – Allows a public IP address to be included in the SAN</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Below are some SAN examples based on the certificate **CommonName** value "shop.example.com":

- A Subdomain SAN must be a Subdomain of the **DomainName** in the **CommonName**. The following are valid Subdomain SAN values:
  - www.example.com
  - www.store.example.com
  - example.com
- Unified Communications (UC) SANs are Subdomains that start with www, owa, mail, or autodiscover. The following are valid UC SAN values:
  - autodiscover.example.com
  - mail.shop.example.com
  - owa.mystore.example.com
- FQDN SANs may or may not have any relationship to the **CommonName**. The following are valid FQDN SAN values:
  - shop.example1.com
  - mystore.example.com
  - www.example.com

For more information about changing SANs, refer to Section 8.2.

### 11.15 Validity Period

Users can control the validity period of ordered certificates by setting the number of months the certificate may be valid.

**Note:** Not all validity periods can be used for all products. The specific validity period constraints for each product are described in Section 11.6.

<table>
<thead>
<tr>
<th>Number Of Months</th>
<th>Certificate Validity Period (days)</th>
<th>Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>184</td>
<td>EV, OV, DV, Alpha</td>
</tr>
<tr>
<td>12</td>
<td>366</td>
<td>EV, OV, DV, Alpha</td>
</tr>
<tr>
<td>24</td>
<td>731</td>
<td>EV, OV, DV, Alpha</td>
</tr>
<tr>
<td>36</td>
<td>1096</td>
<td>OV, DV, Alpha</td>
</tr>
</tbody>
</table>

### 12. Status Explanations

#### 12.1 Order/Certificate Status

Use the **GetModifiedOrders** command to retrieve the Order/Certificate status of a certificate request. The order status will be indicated by a specific value, as described in the below table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Order Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INITIAL</td>
</tr>
<tr>
<td>2</td>
<td>Waiting for phishing check</td>
</tr>
<tr>
<td>3</td>
<td>Cancelled - Not Issued</td>
</tr>
<tr>
<td>4</td>
<td>Issue completed</td>
</tr>
<tr>
<td>5</td>
<td>Cancelled - Issued</td>
</tr>
<tr>
<td>6</td>
<td>Waiting for revocation</td>
</tr>
<tr>
<td>7</td>
<td>Revoked</td>
</tr>
</tbody>
</table>
12.2 ModificationEventName

To understand what type of modification was made to an order, use the GetModifiedOrders command to retrieve the ModificationEventName. The type of modification will be indicated by a specific code, as described in the below table.

<table>
<thead>
<tr>
<th>Code</th>
<th>ModificationEventName</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ORDER_REQUEST</td>
<td>Certificate application accepted</td>
</tr>
<tr>
<td>1</td>
<td>ORDER_CONSENT</td>
<td>Certificate application permitted</td>
</tr>
<tr>
<td>2</td>
<td>ORDER_NOT_CONSENT</td>
<td>Certificate application refused</td>
</tr>
<tr>
<td>3</td>
<td>ORDER_VALIDATE_REGISTER</td>
<td>Vetting requested to RA</td>
</tr>
<tr>
<td>6</td>
<td>ORDER_APPROVE_DENIAL</td>
<td>Order rejected by RA</td>
</tr>
<tr>
<td>7</td>
<td>CERT_ISSUE</td>
<td>Issue certificate</td>
</tr>
<tr>
<td>8</td>
<td>ORDER_ISSUE_BEFORE_CANCEL</td>
<td>Order cancelled before issue</td>
</tr>
<tr>
<td>9</td>
<td>ORDER_ISSUE_AFTER_CANCEL</td>
<td>Order cancelled after issue</td>
</tr>
<tr>
<td>10</td>
<td>ORDER_CANCEL_REQUEST</td>
<td>Request to cancel order</td>
</tr>
<tr>
<td>11</td>
<td>CERT_REVOKE_REQUEST</td>
<td>Request to revoke certificate</td>
</tr>
<tr>
<td>12</td>
<td>CERT_REVOKE</td>
<td>Certificate revoked</td>
</tr>
<tr>
<td>13</td>
<td>CERT_REVOKE_DENIAL</td>
<td>Certificate revocation refused</td>
</tr>
<tr>
<td>14</td>
<td>CERT_CA_REVOKE</td>
<td>Certificate revoked by CA</td>
</tr>
<tr>
<td>15</td>
<td>CERT_TRANSFER</td>
<td>Certificate transferred to other corporations</td>
</tr>
<tr>
<td>16</td>
<td>CERT_REISSUE</td>
<td>Certificate reissue</td>
</tr>
<tr>
<td>17</td>
<td>ORDER_ERROR_RECOVERY</td>
<td>Error recovered</td>
</tr>
<tr>
<td>23</td>
<td>CERT_REVOKE_CANCEL</td>
<td>Certificate revocation cancelled</td>
</tr>
<tr>
<td>24</td>
<td>ORDER_REISSUE_REQUEST</td>
<td>Application for certificate reissue</td>
</tr>
<tr>
<td>25</td>
<td>REORDER_CANCEL_REQUEST</td>
<td>Cancelled certificate reorder</td>
</tr>
<tr>
<td>27</td>
<td>CERT_RENEWAL_INFORMATION</td>
<td>Certificate renewal notice</td>
</tr>
<tr>
<td>28</td>
<td>CERT_REVOKE_REGISTER</td>
<td>Request vetting for revoke to RA</td>
</tr>
<tr>
<td>29</td>
<td>ORDER_RESEND_APPROVAL_MAIL</td>
<td>Resend approval e-mail</td>
</tr>
<tr>
<td>30</td>
<td>ORDERCHANGE_APPROVAL_MAIL</td>
<td>Change approval e-mail address</td>
</tr>
<tr>
<td>31</td>
<td>ORDER_CHANGE_PAY_AFTER</td>
<td>Change the payment method to after payment</td>
</tr>
<tr>
<td>32</td>
<td>ORDER_CHANGE_CONTRACTOR</td>
<td>Change the contractor</td>
</tr>
<tr>
<td>33</td>
<td>ORDER_CHANGE_SALES</td>
<td>Change the sales group, sales staff</td>
</tr>
<tr>
<td>35</td>
<td>SEAL_REGISTER</td>
<td>Request for seal register</td>
</tr>
<tr>
<td>36</td>
<td>SEAL_REVOKE</td>
<td>Request to delete seal</td>
</tr>
<tr>
<td>37</td>
<td>SEAL_CHANGE</td>
<td>Request to change seal</td>
</tr>
<tr>
<td>38</td>
<td>DELETE_PKCS12</td>
<td>Delete PKCS12</td>
</tr>
<tr>
<td>39</td>
<td>DOWNLOAD_PKCS12</td>
<td>Download PKCS12</td>
</tr>
<tr>
<td>40</td>
<td>VALIDATE_PHISHING</td>
<td>Caught in Phishing, vetting</td>
</tr>
<tr>
<td>41</td>
<td>EDIT_CONTACT</td>
<td>Change the contact information</td>
</tr>
<tr>
<td>42</td>
<td>AGENCY_AUTHENTIC</td>
<td>Approve order application from agency</td>
</tr>
<tr>
<td>43</td>
<td>CHANGE_AUTH</td>
<td>Change authenticate information</td>
</tr>
<tr>
<td>44</td>
<td>ORDER_REISSUE_REGISTER</td>
<td>Request reissue to RA</td>
</tr>
<tr>
<td>45</td>
<td>ORDER_REISSUED_REQUEST</td>
<td>Reissue request from GAS</td>
</tr>
<tr>
<td>46</td>
<td>ORDER_CHANGE_SAN_REQUEST</td>
<td>Request to change the SAN</td>
</tr>
<tr>
<td>47</td>
<td>ORDER_CHANGED_SAN_REQUEST</td>
<td>Received request to change SAN</td>
</tr>
<tr>
<td>48</td>
<td>ORDER_CHANGE_SAN_REGISTER</td>
<td>Send request to change SAN to RA</td>
</tr>
<tr>
<td>49</td>
<td>EV_AUTHENTIC</td>
<td>Primary approval of EV</td>
</tr>
<tr>
<td>51</td>
<td>CERT_ISSUE_PAID</td>
<td>Card payment done when the certificate is issued</td>
</tr>
<tr>
<td>52</td>
<td>ORDER_CANCEL_REQUEST_4_RA_OPERATOR</td>
<td>Order cancel request by RA operator</td>
</tr>
<tr>
<td>53</td>
<td>ORDER_CANCEL_REQUEST_4_RA_APP_ROVAL_EMAIL</td>
<td>Request order cancel by approval e-mail</td>
</tr>
<tr>
<td>54</td>
<td>AUTHENTICATE_PHISHING</td>
<td>Approve phishing</td>
</tr>
<tr>
<td>55</td>
<td>READY_VARIFICATION_URL</td>
<td>Ready for URL approval</td>
</tr>
<tr>
<td>56</td>
<td>VARIFICATION_URL</td>
<td>URL approval completed</td>
</tr>
</tbody>
</table>
12.3 ResendEmailType

The ResendEmailType function resends the approver email for DVOrder and DVOrderWithoutCSR. This function is useful if a user has misplaced or lost the original email.

12.4 Success / Error Codes

A SuccessCode is always returned from the API. If the SuccessCode is 0 or 1, the order will normally be able to continue. A SuccessCode of -1 indicates the order failed, and the API will provide one or more ErrorCodes. ErrorCodes provide more information on the Error created with the order, including details regarding the specific fields that may be causing problems in the XML response.

There are two types of errors: Client Error and Server Error. Client error codes suggest that the error was caused by something on the client end. These issues are often due to malformed XML requests, incorrect or missing data, or other API implementation issues. A client error code indicates that the request has not been accepted and the user must make changes and resubmit.

Server error codes suggest a server-side issue caused the error. This type of error could indicate the order request was received but not processed immediately, or the request cannot be received by GCC. If a server error code is received, compare the error code with the table in Section 12.4.3. Report any server error codes to api@globalsign.com.

12.4.1 SuccessCodes

<table>
<thead>
<tr>
<th>Code</th>
<th>Code Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
<td>The order/request has failed; please consult the Error Code list, as well as the error message in the XML response for remedial actions.</td>
</tr>
<tr>
<td>-1</td>
<td>Failure</td>
<td>Indicates order has been flagged for High Risk/Phishing. The order is valid, but will experience a delay in processing until the GlobalSign Vetting Team manually reviews and clears the order’s phishing flag.</td>
</tr>
</tbody>
</table>

12.4.2 Client Error Codes

<table>
<thead>
<tr>
<th>Success Code</th>
<th>Error Code</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>Success</td>
<td>The Common Name has expressions that match our Phishing/Keyword warning list. This order will be delayed until the vetting team can manually review the requested domain. You should receive an e-mail from the system to confirm this issue and another one when it has been cleared. Please have the domain name and OrderID available for our support team if you need immediate resolution.</td>
</tr>
<tr>
<td>1</td>
<td>-5001</td>
<td>Phishing warning CN matched in phishing database</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-101</td>
<td>Invalid Parameter</td>
<td>Invalid parameter entered. Please check that the parameters match the API specification. Please review the specific ErrorMessage returned in the XML response for parameter details and consult the XML Field definitions section of the applicable API document.</td>
</tr>
<tr>
<td>-1</td>
<td>-102</td>
<td>An essential parameter is missing</td>
<td>Mandatory parameter missing. Please check that the parameters match the API specification. Please review the specific ErrorMessage returned in the XML response for parameter details and consult the XML Field definitions section of the applicable API document.</td>
</tr>
<tr>
<td>Success Code</td>
<td>Error Code</td>
<td>Title</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>-1</td>
<td>-103</td>
<td>A parameter is too long</td>
<td>Parameter length check error. Please check that the parameters match the API specification. Please review the specific ErrorMessage returned in the XML response for parameter details and consult the XML Field definitions section of the applicable API document.</td>
</tr>
<tr>
<td>-1</td>
<td>-104</td>
<td>The format for a parameter is incorrect</td>
<td>Parameter format check error. Please check that the parameters match the API specification. Please review the specific ErrorMessage returned in the XML response for parameter details and consult the XML Field definitions section of the applicable API document.</td>
</tr>
<tr>
<td>-1</td>
<td>-105</td>
<td>Invalid Parameter</td>
<td>Invalid parameter combination. Please check that the parameters match the API specification.</td>
</tr>
<tr>
<td>-1</td>
<td>-3002</td>
<td>Domain not found in WHOIS database</td>
<td>Unable to find approver emails for the domain name specified. Please make sure that your Domain Name is found in WHOIS database.</td>
</tr>
<tr>
<td>-1</td>
<td>-3012</td>
<td>We have been unable to validate your domain through a Domain Name Search</td>
<td>We have been unable to validate your domain through a Domain Name Search. If you have recently updated your DNS records, please try again in 24 hours to allow for propagation of the details and the issue to clear. To ensure there is no other issue you should check that your domain is registered via a PING request or alternative check.</td>
</tr>
<tr>
<td>-1</td>
<td>-3013</td>
<td>We are currently experiencing issues in connecting to a DNS service to resolve your domain</td>
<td>We are currently experiencing issues in connecting to a DNS service to resolve your domain. If you have recently updated your DNS records, please try again in 24 hours to allow for propagation of the details and the issue to clear. To ensure there is no other issue you should check that your domain is registered via a PING request or alternative check.</td>
</tr>
<tr>
<td>-1</td>
<td>-3019</td>
<td>We are currently experiencing issues in searching the DNS service to resolve your domain</td>
<td>We are currently experiencing issues in searching the DNS service to resolve your domain. If you have recently updated your DNS records, please try again in 24 hours to allow for propagation of the details and the issue to clear. To ensure there is no other issue you should check that your domain is registered via a PING request or alternative check.</td>
</tr>
<tr>
<td>-1</td>
<td>-3022</td>
<td>We were unable to find the SOA Record of the DNS</td>
<td>We were unable to find the SOA Record of the DNS for (0). Please make sure that the DNS has an SOA Record.</td>
</tr>
<tr>
<td>-1</td>
<td>-3025</td>
<td>We were unable to get the DNS txt record for [0]</td>
<td>We were unable to get the DNS txt record for [0]. Check that you have edited the correct domain, and note that your changes may take several hours to propagate.</td>
</tr>
<tr>
<td>-1</td>
<td>-3026</td>
<td>We were unable to find the DNS txt record for [0]</td>
<td>We were unable to find the DNS txt record for [0]. Check that you have edited the correct domain, and note that your changes may take several hours to propagate.</td>
</tr>
<tr>
<td>-1</td>
<td>-4001</td>
<td>Login failure invalid user ID or password</td>
<td>Login failure. UserName or Password is incorrect. Please make sure that you have specified the correct UserName and Password.</td>
</tr>
<tr>
<td>-1</td>
<td>-4003</td>
<td>Specified OrderID/Voucher Number does not exist</td>
<td>The specified OrderID does not exist. Please make sure you supplied a valid OrderID.</td>
</tr>
<tr>
<td>Success Code</td>
<td>Error Code</td>
<td>Title</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>--------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-1</td>
<td>-4004</td>
<td>OrderId has already been used</td>
<td>The specified OrderID already exists. Please make sure you supplied a valid OrderID.</td>
</tr>
<tr>
<td>-1</td>
<td>-4005</td>
<td>Your request has not been accepted due to a logical limitation. The notes describe the limitations that may be encountered for the request that was submitted.</td>
<td>Your request has not been accepted because the status of the OrderID specified has already been changed. Please recheck the status of the OrderID specified.</td>
</tr>
<tr>
<td>-1</td>
<td>-4006</td>
<td>Specified ProductCode is invalid.</td>
<td>Specified ProductCode is not allowed for this API request. Please check that the ProductCode specified matches the API specifications for this request.</td>
</tr>
<tr>
<td>-1</td>
<td>-4007</td>
<td>Specified CSR is invalid.</td>
<td>Unable to process this request. It could be that the Common Name in your CSR does not match the Common Name in the original order, or the format of the Common Name is incorrect, or the Encryption type of your CSR is not RSA nor ECC, or the length of the Key of the CSR is insufficient. Please make sure that the CSR provided coincides with these conditions.</td>
</tr>
<tr>
<td>-1</td>
<td>-4008</td>
<td>TargetCERT is expired or inaccessible</td>
<td>Unable to process this request. It could be that the Common Name in the TargetCERT specified does not match the Common Name specified for this request or the TargetCERT is inaccessible on the Common Name by the GlobalSign system. Please review the contents and accessibility of the Common Name in the TargetCERT before proceeding with this request.</td>
</tr>
<tr>
<td>-1</td>
<td>-4009</td>
<td>Specified ApproverEmail is invalid.</td>
<td>Failed to confirm ApproverEmail. Please make sure that the ApproverEmail specified is within the ApproverEmail list that we provided when GetDVApproverList was used.</td>
</tr>
<tr>
<td>-1</td>
<td>-4016</td>
<td>The certificate has been already reissued</td>
<td>The certificate you are trying to reissue has already been reissued. You can reissue a certificate only once. Please specify a certificate that is valid for a reissue.</td>
</tr>
<tr>
<td>-1</td>
<td>-4147</td>
<td></td>
<td>We were unable to verify the domain {0}. Please ensure: 1. The domain is publicly accessible 2. The verification meta tag is located in an approved location (See API specification for details) 3. The meta tag exactly matches the one we have provided.</td>
</tr>
<tr>
<td>-1</td>
<td>-4148</td>
<td></td>
<td>We were unable to connect to the authentication URL during Onpage Authentication. Please make sure that the authentication URL is accessible publicly.</td>
</tr>
<tr>
<td>-1</td>
<td>-4149</td>
<td></td>
<td>We were unable to find our verification meta tag on {0}. Please make sure that the meta tag we provided is located in an approved location (See API specification for details)</td>
</tr>
<tr>
<td>-1</td>
<td>-4150</td>
<td></td>
<td>We found the verification meta tag on {0}, but it does not match the tag associated with this order. Please make sure that the meta tag we provided is located in an approved location (See API specification for details)</td>
</tr>
<tr>
<td>-1</td>
<td>-4151</td>
<td></td>
<td>We found the verification meta tag on {0}, but it does not match the tag associated with this order. Please make sure that the meta tag we provided is located in an approved location (See API specification for details)</td>
</tr>
<tr>
<td>-1</td>
<td>-4169</td>
<td></td>
<td>We were unable to get the DNS Txt record for {0}. Please make sure that the DNS Txt record we provided is</td>
</tr>
<tr>
<td>Success Code</td>
<td>Error Code</td>
<td>Title</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>-1</td>
<td>-4171</td>
<td>Success Code</td>
<td>added into your DNS, and note that your changes may take several hours to propagate.</td>
</tr>
<tr>
<td>-1</td>
<td>-4201</td>
<td>Error Code</td>
<td>We found the DNS for [0], but it does not match the one associated with this order. Please make sure that the DNS Txt we provided is added into your DNS, and note that your changes may take several hours to propagate.</td>
</tr>
<tr>
<td>-1</td>
<td>-6002</td>
<td>Title</td>
<td>This IP Address is not registered for API access</td>
</tr>
<tr>
<td>-1</td>
<td>-6007</td>
<td>Description</td>
<td>Specified certificate is invalid</td>
</tr>
<tr>
<td>-1</td>
<td>-6017</td>
<td>Maximum number of SANs options have been exceeded</td>
<td>The number of SANEntry has exceeded the maximum allowed number of SANEntry. Please do not exceed the maximum allowed number of SANEntry. See the API specification for further information.</td>
</tr>
<tr>
<td>-1</td>
<td>-6018</td>
<td>Specified SubjectAltName is invalid</td>
<td>There are duplicates in the SANs that you specified. Please make sure that each SubjectAltName is unique.</td>
</tr>
<tr>
<td>-1</td>
<td>-6021</td>
<td>CN in CSR and FQDN are not same</td>
<td>Common Name in CSR and FQDN for check do not match. Please make sure that the CSR has been entered correctly.</td>
</tr>
<tr>
<td>-1</td>
<td>-6035</td>
<td>Validity Period Warning</td>
<td>Due to recent CA/B Forum changes, reissue past 27 months has been disabled. Please contact your account manager for more information.</td>
</tr>
<tr>
<td>-1</td>
<td>-6101</td>
<td>Account Balance Error</td>
<td>Your account does not have enough remaining balance to process this request. Please make sure you have enough remaining balance in your account before proceeding with this request.</td>
</tr>
<tr>
<td>-1</td>
<td>-6102</td>
<td>Specified order cannot be a renewal</td>
<td>The renewal of the certificate failed. Please note that when renewing a certificate, the Common Name of the original certificate and this request must be the same. Please also check that the status of the original order is ISSUED and that the order has not been previously renewed.</td>
</tr>
<tr>
<td>-1</td>
<td>-9025</td>
<td>Could not renew because you have no renewal authority</td>
<td>Please make sure that the RenewalTargetOrderID specified is associated with this user.</td>
</tr>
<tr>
<td>-1</td>
<td>-9100</td>
<td>Unable to process this request</td>
<td>The following rules may have been violated by this ChangeSAN request: 1. TargetOrderID should be owned by this user. 2. TargetOrderID should not have been requested for a Cancel or a ChangeSAN previously. 3. TargetOrderID should already be ISSUED. 4. TargetOrderID should not be expired. 5. TargetOrderID should not be an MSSL order. Please make sure that the rules above are followed before proceeding with this request.</td>
</tr>
<tr>
<td>-1</td>
<td>-9101</td>
<td>Illegal SAN Option</td>
<td>Unable to process this request because of an incorrect SANEntry. The SANEntry you specified may have</td>
</tr>
<tr>
<td>Success Code</td>
<td>Error Code</td>
<td>Title</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>-9110</td>
<td></td>
<td>We found no difference in the SANs you specified in this request from SANs of the original order. Please make sure that the SANs are different from your current certificate.</td>
</tr>
<tr>
<td></td>
<td>-9151</td>
<td></td>
<td>Unable to use ReissueWithoutCSR on a certificate that does not support AutoCSR. Please make sure that the certificate you are trying to ReissueWithoutCSR supports AutoCSR. See the API specification for the list of features for each product, including AutoCST.</td>
</tr>
<tr>
<td></td>
<td>-9152</td>
<td></td>
<td>The format of the ApproverURL specified is incorrect. Please make sure that the format of the ApproverURL is correct and the same with the one we have provided.</td>
</tr>
<tr>
<td></td>
<td>-9200</td>
<td></td>
<td>The type of your user is not allowed to use this API. Please check your permission and retry.</td>
</tr>
<tr>
<td>-1</td>
<td>-9901</td>
<td></td>
<td>The Product Group of this user does not allow ordering of the specified ProductCode. Please contact Globalsign Support if you wish to order using this ProductCode.</td>
</tr>
<tr>
<td>-1</td>
<td>-9902</td>
<td></td>
<td>You do not have permission to access the OrderID specified. Please make sure that you have the proper permission to the order before continuing with this request.</td>
</tr>
<tr>
<td>-1</td>
<td>-9911</td>
<td></td>
<td>There are insufficient funds in the account to complete the order process. Please verify that the account has sufficient funds and try again.</td>
</tr>
<tr>
<td>-1</td>
<td>-9912</td>
<td></td>
<td>There is insufficient funds in the account to complete the order process. Please verify that the account has sufficient funds and try again.</td>
</tr>
<tr>
<td>-1</td>
<td>-9913</td>
<td></td>
<td>We were unable to find the Coupon specified. Please make sure that it is correctly entered.</td>
</tr>
<tr>
<td>-1</td>
<td>-9914</td>
<td></td>
<td>We were unable to find the Campaign specified. Please make sure that it is correctly entered.</td>
</tr>
<tr>
<td>-1</td>
<td>-9915</td>
<td></td>
<td>The OrderID you are trying to modify has been cancelled previously. Please make sure that the OrderID is correctly entered.</td>
</tr>
<tr>
<td>-1</td>
<td>-9916</td>
<td></td>
<td>We were not able to find the OrderID specified. Please make sure that the OrderID is correctly entered.</td>
</tr>
<tr>
<td>-1</td>
<td>-9918</td>
<td></td>
<td>The coupon or campaign you specified has expired. Please make sure that the coupon or campaign is correctly entered.</td>
</tr>
<tr>
<td>-1</td>
<td>-9919</td>
<td></td>
<td>The coupon you specified has been previously used. Please make sure that the coupon is correctly entered.</td>
</tr>
<tr>
<td>-1</td>
<td>-9920</td>
<td></td>
<td>The coupon or campaign you specified is not yet activated. Please make sure that the coupon or campaign is correctly entered.</td>
</tr>
<tr>
<td>-1</td>
<td>-9922</td>
<td></td>
<td>The currency of the specified Coupon or Campaign is not the same with the currency of your user. Please make sure that the coupon or campaign is correctly entered.</td>
</tr>
<tr>
<td>-1</td>
<td>-9925</td>
<td></td>
<td>The Common Name of your CSR or the FQDN you have entered does not match the type of BaseOption</td>
</tr>
</tbody>
</table>

GlobalSign API for SSL Certificates v4.5

Page 58
<table>
<thead>
<tr>
<th>Success Code</th>
<th>Error Code</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-9934</td>
<td>The Top Level Domain used belongs to Globalsign’s Banned List, therefore, a certificate cannot</td>
<td>You provided. Please check your BaseOption matches the Common Name (e.g., Standard, Wildcard, PublicIP). Contact support for more information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>be issued. Please make sure that Common Name is correctly entered. Contact support for more</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>-9936</td>
<td>The key you used in your CSR is either too short (RSA minimum 2048, ECC minimum 256), or the</td>
<td>The key you used in your CSR is either too short (RSA minimum 2048, ECC minimum 256), or the key failed the Debian weak key check as well as key length. Please generate a new keypair and try again.</td>
</tr>
<tr>
<td></td>
<td>-9938</td>
<td>The certificate you are trying to modify has already been modified. Please make sure that the</td>
<td>The certificate you are trying to modify has already been modified. Please make sure that the OrderID is correctly entered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OrderID is correctly entered. Contact customer support for assistance.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9940</td>
<td>The specified NotBefore or NotAfter should not be before the current date/time. Please recheck</td>
<td>The specified NotBefore or NotAfter should not be before the current date/time. Please recheck these parameters before continuing with this request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>these parameters before continuing with this request.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9941</td>
<td>We are unable to find the associated partner account from the specified CorporateID. Please</td>
<td>We are unable to find the associated partner account from the specified CorporateID. Please make sure that the CorporateID is correctly entered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>make sure that the CorporateID is correctly entered.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9945</td>
<td>Cannot find the balance information that is associated with the CorporateID you have supplied.</td>
<td>Cannot find the balance information that is associated with the CorporateID you have supplied. Please recheck the CorporateID specified then try again.</td>
</tr>
<tr>
<td></td>
<td>-9946</td>
<td>Your user does not have access to the CorporateID specified. Please make sure that the</td>
<td>Your user does not have access to the CorporateID specified. Please make sure that the CorporateID is correctly inputted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CorporateID is correctly inputted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9947</td>
<td>The ProductCode of the OrderID specified is not supported for URLVerificationForIssue. Please</td>
<td>The ProductCode of the OrderID specified is not supported for URLVerificationForIssue. Please make sure that the OrderID is correctly inputted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>make sure that the OrderID is correctly inputted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9948</td>
<td>AutoCSR method has been discontinued for Partner users. Please make sure that you are not</td>
<td>AutoCSR method has been discontinued for Partner users. Please make sure that you are not using a Partner account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>using a Partner account.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9949</td>
<td>The NotAfter specified is after the calculated BaseLine Validity Limit. Please take note that</td>
<td>The NotAfter specified is after the calculated BaseLine Validity Limit. Please take note that validity should not exceed 825 days.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>validity should not exceed 825 days.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9952</td>
<td>The Top Level Domain you specified is not supported. Please make sure that Common Name is</td>
<td>The Top Level Domain you specified is not supported. Please make sure that Common Name is correctly entered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>correctly entered.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9953</td>
<td>Cannot complete this request because the region or country of your Domain is not permitted.</td>
<td>Cannot complete this request because the region or country of your Domain is not permitted. Please make sure that Common Name is correctly entered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Please make sure that Common Name is correctly entered.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9961</td>
<td>The ECC CSR key type you used is not allowed. Please enter an ECC CSR using either P-256 or</td>
<td>The ECC CSR key type you used is not allowed. Please ensure that you are using an ECC CSR using either P-256 or P-384 curves.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P-384 curves.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9962</td>
<td>Key Compression is not allowed. Please make sure that CSR is correctly entered.</td>
<td>Key Compression is not allowed. Please make sure that CSR is correctly entered.</td>
</tr>
<tr>
<td></td>
<td>-9969</td>
<td>Due to industry requirements, you can no longer reissue certificates with internal server names.</td>
<td>Due to industry requirements, you can no longer reissue certificates with internal server names. Please delete internal server name SANs and try again.</td>
</tr>
<tr>
<td></td>
<td>-9971</td>
<td>Due to industry requirements, you can no longer issue certificates with internal server names in</td>
<td>Due to industry requirements, you can no longer issue certificates with internal server names in Common Name. Please update the Common Name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Common Name.</td>
<td></td>
</tr>
</tbody>
</table>
12.4.3 Server Error Codes

<table>
<thead>
<tr>
<th>Success Code</th>
<th>Error Code</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>Internal system error</td>
<td>The system has experienced an internal error. Cross-reference the appropriate API request action with the API Spec. Please try to do what you were doing again, and if the problem persists, please report this error code to GlobalSign Support.</td>
</tr>
<tr>
<td>-1</td>
<td>-201</td>
<td>Internal system error - Failed database operation</td>
<td>System Error. (Database error - database operation). Please retry and if the issue persists contact support with detailed information concerning the issue.</td>
</tr>
<tr>
<td>-1</td>
<td>-301</td>
<td>Internal system error - Failed database operation</td>
<td>System Error. (Database error - database operation). Please retry and if the issue persists contact support with detailed information concerning the issue.</td>
</tr>
</tbody>
</table>

13. XML Field Definitions

13.1 Data Types

<table>
<thead>
<tr>
<th>DataType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Fixed-length character string</td>
</tr>
<tr>
<td>Boolean</td>
<td>Logical Boolean (true/false)</td>
</tr>
<tr>
<td>Int</td>
<td>Signed four-byte integer</td>
</tr>
<tr>
<td>DateTime</td>
<td>YYYY-MM-DDTHH:MM:SS.000Z</td>
</tr>
</tbody>
</table>

13.2 Data Definitions
The following table lists all of the data types used in the API specification.

<table>
<thead>
<tr>
<th>XML Structure</th>
<th>Description</th>
<th>Type/Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;AddressLine1&gt;</td>
<td>Part of the Address structure. Contains the first line of the address.</td>
<td>String/100</td>
</tr>
<tr>
<td>&lt;AddressLine2&gt;</td>
<td>Part of the Address structure. Contains the second line of the address.</td>
<td>String/100</td>
</tr>
<tr>
<td>&lt;AddressLine3&gt;</td>
<td>Part of the Address structure. Contains the third line of the address.</td>
<td>String/100</td>
</tr>
<tr>
<td>&lt;Approver&gt; &lt;ApproverType&gt; &lt;ApproverEmail&gt; &lt;/Approver&gt;</td>
<td>This is the &lt;Approver&gt; information for each Approver in the &lt;ApproverList&gt;. Today only the email address is returned, but there could be other fields returned in the future.</td>
<td></td>
</tr>
<tr>
<td>&lt;ApproverEmail&gt;</td>
<td>This is the email of the Approver – For DomainSSL and DV_LOW products the person responsible for approving the certificate order.</td>
<td>String/255</td>
</tr>
<tr>
<td>&lt;ApproverEmailAddress&gt;</td>
<td>This is the email of the Approver – For DomainSSL and DV_LOW products the person responsible for approving the certificate order.</td>
<td>String/255</td>
</tr>
<tr>
<td>&lt;ApproverFQDN&gt;</td>
<td>The ApproverFQDN function is used for DNS-based domain approvals. For more info refer to Section 9.1.</td>
<td>String/64</td>
</tr>
<tr>
<td>&lt;ApproverType&gt;</td>
<td>The type of Approver email address. One of the following: Domain – From WHOIS data Generic – From the computed list</td>
<td>String/10</td>
</tr>
<tr>
<td>&lt;ApproverURL&gt;</td>
<td>A list of approved URLs</td>
<td>String</td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td><code>&lt;Approvers&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(<code>&lt;Approver&gt;</code>)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/Approvers&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ApproverInfo&gt;</code></td>
<td>A Certificate Approver is a natural person who is Applicant, employed by Applicant, or an authorized agent who has express authority to represent Applicant to (i) act as a Certificate Requester and to authorize other employees or third parties to act as a Certificate Requester, and (ii) to approve Certificate Requests submitted by other Certificate Requesters.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;AuthorizedSignerInfo&gt;</code></td>
<td>A Contract Signer is a natural person who is Applicant, employed by Applicant, or an authorized agent who has express authority to represent Applicant who has authority on behalf of Applicant to sign Subscriber Agreements on behalf of Applicant.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;AuthToken&gt;</code></td>
<td>Used for partner authentication on each message posted to GlobalSign. This partner has to be set up by GlobalSign for API access.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;BaseOption&gt;</code></td>
<td>Options for the certificate. Currently allowed fields are:</td>
<td>String/20</td>
</tr>
<tr>
<td></td>
<td>• wildcard – certificate with *</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• globalip – certificate with global ip address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• subaltname – certificate with alternative subject names</td>
<td></td>
</tr>
<tr>
<td><code>&lt;BusinessAssumedName&gt;</code></td>
<td>Business Type</td>
<td>String/255</td>
</tr>
<tr>
<td><code>&lt;BusinessCategoryCode&gt;</code></td>
<td>Business Type</td>
<td>String/20</td>
</tr>
<tr>
<td><code>&lt;CACert&gt;</code></td>
<td>This is the content of a CA certificate in the certificate chain for the server certificate in Base64 encoded format.</td>
<td>String/4000</td>
</tr>
<tr>
<td><code>&lt;CACertificate&gt;</code></td>
<td>This identifies the type of certificate for each CA certificate in the chain, and also contains the actual certificate.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;CACertType&gt;</code></td>
<td>The Type of CA certificate: ROOT or INTER</td>
<td>String/15</td>
</tr>
<tr>
<td><code>&lt;Campaign&gt;</code></td>
<td>Campaign can be used for payment</td>
<td>String/50</td>
</tr>
<tr>
<td><code>&lt;CertificateInfo&gt;</code></td>
<td>This structure contains information stored related to the certificate in various Query operations.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;DNSNames&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CertificateStatus&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CommonName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;EndDate&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;SerialNumber&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;StartDate&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;SubjectName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/CertificateInfo&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>&lt;CertificatePreview&gt;</code></td>
<td>Provides a preview of the certificate.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;CommonName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Organization&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrganizationUnit&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Locality&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;State&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Country&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CertificatePreview&gt;</code></td>
<td>The current status of a certificate.</td>
<td>Int</td>
</tr>
<tr>
<td><code>&lt;CertificateStatus&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CommonName&gt;</code></td>
<td>The common name in certificate.</td>
<td>String/255</td>
</tr>
<tr>
<td><code>&lt;CSRData&gt;</code></td>
<td>Basic information on a certificate’s CSR</td>
<td></td>
</tr>
<tr>
<td><code>&lt;CommonName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Organization&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrganizationUnit&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Locality&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;State&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Country&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;EmailAddress&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CSRData&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CSRSkipOrderFlag&gt;</code></td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td><code>&lt;City&gt;</code></td>
<td>Part of the Address structure.</td>
<td>String/200</td>
</tr>
<tr>
<td><code>&lt;ContactInfo&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;FirstName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;LastName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Phone&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Email&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ContactInfo&gt;</code></td>
<td>Primary Contact Information of for a certificate request</td>
<td></td>
</tr>
<tr>
<td><code>&lt;Country&gt;</code></td>
<td></td>
<td>String/2</td>
</tr>
<tr>
<td><code>&lt;Coupon&gt;</code></td>
<td>Coupons can be used for payment.</td>
<td>String/50</td>
</tr>
<tr>
<td><code>&lt;CreditAgency&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Currency&gt;</code></td>
<td></td>
<td>String/10</td>
</tr>
<tr>
<td><code>&lt;CSR&gt;</code></td>
<td>Certificate Signing Request. This is the Base64 encoded X.509 digital certificate signing request typically generated by the end user on their target web server. This is a critical element for all SSL orders.</td>
<td>String/4000</td>
</tr>
<tr>
<td><code>&lt;DNSOrderFlag&gt;</code></td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td><code>&lt;DNSNames&gt;</code></td>
<td></td>
<td>String/300</td>
</tr>
<tr>
<td><code>&lt;DomainName&gt;</code></td>
<td>The domain name for an Order. For an SSL Order this can be a fully qualified Domain (e.g., <a href="http://www.globasign.com">www.globasign.com</a>) or possibly a wildcard domain (e.g., *.globasign.com).</td>
<td>String/255</td>
</tr>
<tr>
<td><code>&lt;EndDate&gt;</code></td>
<td></td>
<td>DateTime</td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>&lt;Email&gt;</code></td>
<td>From the ContactInfo structure. The Email Address of the contact.</td>
<td>String/255</td>
</tr>
<tr>
<td><code>&lt;Error&gt;</code></td>
<td>A structure that contains an ErrorCode and an ErrorMessage. Error is part of the Errors structure.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;ErrorCode&gt;</code></td>
<td>A unique code identifying the error.</td>
<td>Int</td>
</tr>
<tr>
<td><code>&lt;ErrorField&gt;</code></td>
<td>When there is a specific field that has caused the error, the XML tag for that field is placed in this structure. Where the tag is not unique in the entire message, one or more tags precede this so this field can be uniquely identified. For example, if the <code>&lt;Phone&gt;</code> field was invalid in the <code>&lt;AdminContact&gt;</code> structure, the return code would have <code>&lt;AdminContact&gt;&lt;Phone&gt;</code>.</td>
<td>String/1000</td>
</tr>
<tr>
<td><code>&lt;ErrorMessage&gt;</code></td>
<td>A message describing an error in more detail. ErrorMessage is a part of the Error Structure</td>
<td>String/1000</td>
</tr>
<tr>
<td><code>&lt;Errors&gt;</code></td>
<td>A list of the errors returned from a request. An Errors structure can have multiple Error elements. Errors is a part of the OrderResponseHeader structure. If present, this structure contains one or more errors.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;ErrorCode&gt;</code></td>
<td></td>
<td>Int</td>
</tr>
<tr>
<td><code>&lt;ErrorField&gt;</code></td>
<td></td>
<td>String/1000</td>
</tr>
<tr>
<td><code>&lt;ErrorMessage&gt;</code></td>
<td></td>
<td>String/1000</td>
</tr>
<tr>
<td>(<code>&lt;ExpressOption&gt;</code>)?</td>
<td>To add Express Options set to true. If not false.</td>
<td>Boolean</td>
</tr>
<tr>
<td><code>&lt;Fax&gt;</code></td>
<td>From the OrganizationAddress structure. The Fax number for the organization.</td>
<td>String/30</td>
</tr>
<tr>
<td><code>&lt;FirstName&gt;</code></td>
<td>From one of the Contact structures. The First Name of the contact.</td>
<td>String/100</td>
</tr>
<tr>
<td><code>&lt;FQDN&gt;</code></td>
<td>Fully Qualified Domain Name</td>
<td>String/255</td>
</tr>
<tr>
<td><code>&lt;FromDate&gt;</code></td>
<td>The starting date used in various queries.</td>
<td>DateTime</td>
</tr>
<tr>
<td><code>&lt;Fulfillment&gt;</code></td>
<td>Contains the CA certificate(s) and/or the ServerCertificate (in x509 and/or PKCS7 formats).</td>
<td></td>
</tr>
<tr>
<td><code>&lt;CACertificates&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CACertificate&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CACertType&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;CACert&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;/CACertificate&gt;</code>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Certificate&gt;</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ServerCertificate&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;x509Cert&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;PKCS7Cert&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;/ServerCertificate&gt;</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;/Fulfillment&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Function&gt;</code></td>
<td>Requestor job function</td>
<td>String/255</td>
</tr>
<tr>
<td><code>&lt;GSSupportOption&gt;</code></td>
<td>To add GS Support set to true. If not false.</td>
<td>Boolean</td>
</tr>
<tr>
<td><code>&lt;IncorporatingAgencyRegistrationNumber&gt;</code></td>
<td></td>
<td>String/100</td>
</tr>
<tr>
<td><code>&lt;InsuranceOption&gt;</code></td>
<td>To add Insurance Options set to true. If not false.</td>
<td>Boolean</td>
</tr>
<tr>
<td><code>&lt;IsValidDomainName&gt;</code></td>
<td>Returns true if the domain name is valid for a certificate orders</td>
<td>Boolean</td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&lt;JurisdictionInfo&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;JurisdictionCountry&gt;</td>
<td>Jurisdiction of Incorporation Details</td>
<td></td>
</tr>
<tr>
<td>(&lt;StateOrProvince&gt;)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;Locality&gt;)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;IncorporatingAgencyRegistrationNumber&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/JurisdictionInfo&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;KeyLength&gt;</td>
<td>From one of the Contact structures. The Last Name of the contact.</td>
<td>String/4</td>
</tr>
<tr>
<td>&lt;LastName&gt;</td>
<td>This is the Number of Licenses.</td>
<td>Int 1-99 Only</td>
</tr>
<tr>
<td>&lt;Locality&gt;</td>
<td>The Locality field from the CSR or Certificate</td>
<td>String/255</td>
</tr>
<tr>
<td>&lt;ModificationEvent&gt;</td>
<td>One event in the set of ModificationEvents</td>
<td></td>
</tr>
<tr>
<td>&lt;ModificationEventName&gt;</td>
<td>The name of the event</td>
<td>String/50</td>
</tr>
<tr>
<td>&lt;ModificationEvents&gt;</td>
<td>The set of events for the order that caused the status to be changed within the specified time period. This is contained in OrderDetail. Used only in GetModifiedOrders.</td>
<td></td>
</tr>
<tr>
<td>(&lt;ModificationEvent&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;ModificationEventName&gt;</td>
<td>)+</td>
<td></td>
</tr>
<tr>
<td>&lt;ModificationEventTimestamp&gt;</td>
<td>The time of the event</td>
<td>DateTime</td>
</tr>
<tr>
<td>&lt;/ModificationEvent&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/ModificationEvents&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;ModifyOrderOperation&gt;</td>
<td>Specifies the operation to be performed on the order or certificate.</td>
<td>String/20</td>
</tr>
<tr>
<td>APPROVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANCEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REVOKE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Months&gt;</td>
<td>The number of months that a certificate will be valid for.</td>
<td>Int/4</td>
</tr>
<tr>
<td>&lt;NotAfter&gt;</td>
<td>The date the order was created.</td>
<td>DateTime</td>
</tr>
<tr>
<td>&lt;NotBefore&gt;</td>
<td></td>
<td>DateTime</td>
</tr>
<tr>
<td>&lt;OrderDate&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrderDetail&gt;</td>
<td>OrderDetail is returned in many Order Query operations. The specific content is dependent on the values in the request. ModificationEvents is only returned in GetModifiedOrders.</td>
<td></td>
</tr>
<tr>
<td>(&lt;OrderOption&gt;)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;CertificateInfo&gt;)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;Fulfillment&gt;</td>
<td>)?</td>
<td></td>
</tr>
<tr>
<td>(&lt;ModificationEvents&gt;)</td>
<td>)</td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>&lt;/OrderDetail&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;OrderKind&gt;</td>
<td>Type of order: new: a new request renewal: renewal of current certificate transfer: a commercial upgrade of a current valid certificate</td>
<td>String/10</td>
</tr>
<tr>
<td>&lt;OrderID&gt;</td>
<td>This is the OrderID assigned by GlobalSign to the order and provided to the person requesting the certificate.</td>
<td>String/50</td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>&lt;OrderInfo&gt;</code></td>
<td>This structure contains basic information that apply to most orders and is profiled within each order response structure.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrderID&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ProductCode&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;BaseOption&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrderKind&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Licenses&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;ExpressOption&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;ValidityPeriodCustomizeOption&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;InsuranceOption&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;GSSupportOption&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;RenewalExtensionOption&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;DomainName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrderDate&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;OrderCompleteDate&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;OrderCanceledDate&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;OrderDeactivatedDate&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrderStatus&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Price&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Currency&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ValidityPeriod&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Months&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;NotBefore&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;NotAfter&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ValidityPeriod&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{&lt;SpecialInstructions&gt;}</code>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrderInfo&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>&lt;OrderOption&gt;</code></td>
<td>This structure is in many order request messages and contains basic order information common to all types of orders.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;ApproverNotifiedDate&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ApproverConfirmDate&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ApproverEmailAddress&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrganizationInfo&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrganizationName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( <code>&lt;CreditAgency&gt;</code> )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( <code>&lt;OrganizationCode&gt;</code> )</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;OrganizationAddress&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;AddressLine1&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( <code>&lt;AddressLine2&gt;</code> )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( <code>&lt;AddressLine3&gt;</code> )</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;City&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Region&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;PostalCode&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Country&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Phone&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( <code>&lt;Fax&gt;</code> )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/OrganizationAddress&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/OrganizationInfo&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ContactInfo&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;FirstName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;LastName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Phone&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Email&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/ContactInfo&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/OrderOption&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>&lt;OrderParameter&gt;</code></td>
<td>This structure is part of the order validation and order processes. It includes all details relating to the order and also the CSR for parsing.</td>
<td></td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>&lt;OrderParameterWithoutCSR&gt;</code></td>
<td>This structure is part of the order validation and order processes. It includes all details relating to the order without a CSR.</td>
<td></td>
</tr>
</tbody>
</table>
| `{<ProductCode>}
{<OrderKind>}
{<License>}
{<ExpressOption>}`?                                                         |                                                                             |             |
| `{<ValidityPeriodCustomizeOption>}`?                                         |                                                                             |             |
| `{<InsuranceOption>}`?                                                        |                                                                             |             |
| `{<GSSupportOption>}`?                                                        |                                                                             |             |
| `{<RenewalExtensionOption>}`?                                                |                                                                             |             |
| `<ValidityPeriod>`                                                           |                                                                             |             |
| `{<Months>}
{<NotBefore>}`?                                                    |                                                                             |             |
| `{<NotAfter>}`?                                                              |                                                                             |             |
| `{<ValidityPeriod>`                                                        |                                                                             |             |
| `<PIN>`                                                                      |                                                                             |             |
| `<KeyLength>`                                                                |                                                                             |             |
| `{<RenewalTargetOrderID>}`?                                                  |                                                                             |             |
| `{<TargetCERT>}`?                                                            |                                                                             |             |
| `{<DNSNames>}`?                                                              |                                                                             |             |
| `{<SpecialInstructions>}`?                                                   |                                                                             |             |
| `{<Coupon>}`?                                                                |                                                                             |             |
| `{<Campaign>}`?                                                              |                                                                             |             |
| `</OrderParameterWithoutCSR>`                                                |                                                                             |             |
| `<OrderQueryOption>`                                                         | Specifies what is returned in the response message. All values default to false if not supplied so the corresponding data structure will not appear in the response. |             |
| `{<OrderStatus>}`}?                                                          |                                                                             |             |
| `{<ReturnOrderOption>}`?                                                     |                                                                             |             |
| `{<ReturnCertificateInfo>}`?                                                 |                                                                             |             |
| `{<ReturnFulfillment>}`?                                                     |                                                                             |             |
| `{<ReturnCACerts>}`?                                                         |                                                                             |             |
| `</OrderQueryOption>`                                                        |                                                                             |             |
| `<OrderRequestHeader>`                                                        | The OrderRequestHeader is used in all of the order operations.             |             |
| `<AuthToken>`                                                                |                                                                             |             |
| `<UserName>`                                                                 |                                                                             |             |
| `<Password>`                                                                 |                                                                             |             |
| `</AuthToken>`                                                               |                                                                             |             |
| `</OrderRequestHeader>`                                                       |                                                                             |             |
| `<OrderResponseHeader>`                                                      | This is the header returned in all Order operations.                        |             |
| `<SuccessCode>`                                                              |                                                                             |             |
| `{<Errors>}
{<Error>`                                                       |                                                                             |             |
<p>| <code>{ErrorCode&gt;</code>                                                                |                                                                             |             |
| <code>{ErrorField&gt;}</code>?                                                            |                                                                             |             |
| <code>{ErrorMessage&gt;</code>                                                             |                                                                             |             |
| <code>&lt;/Error&gt;</code>                                                                  |                                                                             |             |
| <code>{Errors&gt;</code>*                                                                 |                                                                             |             |
| <code>&lt;Timestamp&gt;</code>                                                                |                                                                             |             |
| <code>&lt;/OrderResponseHeader&gt;</code>                                                     |                                                                             |             |</p>
<table>
<thead>
<tr>
<th>XML Structure</th>
<th>Description</th>
<th>Type/Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;OrderStatus&gt;</td>
<td>The current status of an Order.</td>
<td>Int</td>
</tr>
<tr>
<td></td>
<td>1 INITIAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 Waiting for phishing check</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 Cancelled - Not Issued</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 Issue completed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 Cancelled - Issued</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 Waiting for revocation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 Revoked</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;OrderSubInfo&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;CSRSkipOrderFlag&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;DNSOrderFlag&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;TrustedOrderFlag&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;P12DeleteStatus&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;P12DeleteDate&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;VerificationUrl&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;SubId&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;/OrderSubInfo&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;Organization&gt;</td>
<td>The Organization field from the certificate</td>
<td>String/255</td>
</tr>
<tr>
<td>&lt;OrganizationCode&gt;</td>
<td>Can be used to indicate company numbers lookup eg. For DUNS enter 1 in this field.</td>
<td>String/50</td>
</tr>
<tr>
<td>&lt;OrganizationInfo&gt;</td>
<td>Organization Info sent with Certificate request.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(CreditAgency)&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(OrganizationCode)&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(OrganizationAddress)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;AddressLine1&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(AddressLine2)&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(AddressLine3)&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;City&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Region&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;PostalCode&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Country&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Phone&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Fax&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;/OrganizationAddress&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;OrganizationInfoEV&gt;</td>
<td>Organization Info sent with Certificate request.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(CreditAgency)&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(OrganizationCode)&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(BusinessAssumedName)&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(BusinessCategoryCode)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(OrganizationAddress)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;AddressLine1&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;AddressLine2&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;AddressLine3&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;City&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Region&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;PostalCode&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Country&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Phone&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Fax)&gt;&gt;?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;/OrganizationAddress&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;OrganizationName&gt;</td>
<td>The name of the Organization applying for a certificate.</td>
<td>String/255</td>
</tr>
<tr>
<td>&lt;OrganizationUnit&gt;</td>
<td>The OrganizationalUnit name from the CSR.</td>
<td>String/255</td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td><code>&lt;OVCSRInfo&gt;</code></td>
<td>Info to be used in the creation of the Certificate</td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;OrganizationName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>(Option: </code>&lt;OrganizationUnit&gt;<code>)</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Locality&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;StateOrProvince&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Country&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;ParsedCSR&gt;</code></td>
<td>Details from the CSR</td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;DomainName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Country&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Email&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Locality&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Organization&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;OrganizationUnit&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;State&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;IsValidDomainName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Password&gt;</code></td>
<td>Required for user authentication over the API</td>
<td>String/30</td>
</tr>
<tr>
<td><code>&lt;Phone&gt;</code></td>
<td>From one of the Contact or OrganizationAddress structures.</td>
<td>String/30</td>
</tr>
<tr>
<td><code>&lt;P12DeleteDate&gt;</code></td>
<td></td>
<td>DateTime</td>
</tr>
<tr>
<td><code>&lt;P12DeleteStatus&gt;</code></td>
<td></td>
<td>Int</td>
</tr>
<tr>
<td><code>&lt;PKCS12File&gt;</code></td>
<td>A base64-encoded PKCS#12</td>
<td>String/4000</td>
</tr>
<tr>
<td><code>&lt;PKCS7Cert&gt;</code></td>
<td>A Base64-encoded PKCS#7</td>
<td>String/20000</td>
</tr>
<tr>
<td><code>&lt;PostalCode&gt;</code></td>
<td>From the Address structure. The Postal Code (e.g., Zip Code in the U.S.) for the Address</td>
<td>String/20</td>
</tr>
<tr>
<td><code>&lt;ProductCode&gt;</code></td>
<td>A code for the product that a particular request relates to. Note that a partner must have a valid contract for a product code for it to be valid in a request. Also, a product code must be valid for the context of the request.</td>
<td>String/20</td>
</tr>
<tr>
<td><code>&lt;QueryRequestHeader&gt;</code></td>
<td>The header on all Query Request operations.</td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;AuthToken&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;UserName&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Password&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>/AuthToken&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;QueryResponseHeader&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Errors&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;ReturnCount&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;SuccessCode&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;Timestamp&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>/OrderResponseHeader&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;Region&gt;</code></td>
<td>Region, state/prov</td>
<td>String/255</td>
</tr>
<tr>
<td>- From the Address structure. This is the region of the address such as state or province. If this is a U.S. state it must have a valid 2 character abbreviation</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;RenewalExtensionOption&gt;</code></td>
<td>To add bonus to validity period set to true. If not false.</td>
<td>Boolean</td>
</tr>
<tr>
<td><code>&lt;ReOrderParameter&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;CSR&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>&lt;DNSNames&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- <code>/ReOrderParameter</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML Structure</td>
<td>Description</td>
<td>Type/Length</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&lt;ReOrderParameterWithoutCSR&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;DNSNames&gt;?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;PIN&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;KeyLength&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/ReOrderParameterWithoutCSR&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;RenewalTargetOrderID&gt;</td>
<td>Original OrderID for renewal orders.</td>
<td>String/50</td>
</tr>
<tr>
<td>&lt;RequestorInfo&gt;</td>
<td>A Certificate Requester is a natural person who is Applicant, employed by Applicant, or an authorized agent who has express authority to represent Applicant or a third party (such as an ISP or hosting company) that completes and submits a Certificate Request on behalf of Applicant.</td>
<td></td>
</tr>
<tr>
<td>&lt;FirstName&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;LastName&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;Function&gt;)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrganizationName&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{&lt;OrganizationUnit&gt;}?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Phone&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Email&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/RequestorInfo&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;ResendEmailType&gt;</td>
<td>Current values are: ApproverEmail – resend the approver email for any QuickSSL order.</td>
<td>String/20</td>
</tr>
<tr>
<td>&lt;ReturnCACerts&gt;</td>
<td>If set to true in the request message, the CACerts structure is populated in the Fulfillment structure of the response message.</td>
<td>Boolean</td>
</tr>
<tr>
<td>&lt;ReturnCertificateInfo&gt;</td>
<td>If set to true in the request message, the CertificateInfo structure appears in the response message.</td>
<td>Boolean</td>
</tr>
<tr>
<td>&lt;ReturnCount&gt;</td>
<td>The number of items returned in the message</td>
<td>Int</td>
</tr>
<tr>
<td>&lt;ReturnFulfillment&gt;</td>
<td>If set to true in the request message, the Fulfillment structure appears in the response message.</td>
<td>Boolean</td>
</tr>
<tr>
<td>&lt;SearchOrderDetail&gt;</td>
<td>In the response, product information will be in details if set to true.</td>
<td>Boolean</td>
</tr>
<tr>
<td>&lt;RequestKind&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrderKind&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrderID&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;BaseOption&gt;?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrderRequestDate&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrderIssueDate&gt;?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrderCanceledDate&gt;?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrderStatus&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;OrganizationName&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Months&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SubId&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;FQDN&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/SearchOrderDetail&gt;</td>
<td>Secondary contact for this order</td>
<td></td>
</tr>
<tr>
<td>&lt;SecondContactInfo&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SerialNumber&gt;</td>
<td>The serial number of a certificate specified as a hex string.</td>
<td>String/64</td>
</tr>
<tr>
<td>&lt;ServerCertificate&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;X509Cert&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;PKCS7Cert&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;/ServerCertificate&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SpecialInstructions&gt;</td>
<td>Special Instructions for the order</td>
<td>String/4000</td>
</tr>
<tr>
<td>&lt;StartDate&gt;</td>
<td>Start date of certificate.</td>
<td>DateTime</td>
</tr>
<tr>
<td>&lt;State&gt;</td>
<td>The value of the State in the ParseCSRResponse.</td>
<td>String/255</td>
</tr>
<tr>
<td>&lt;StateOrProvince&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GlobalSign API for SSL Certificates v4.5
<table>
<thead>
<tr>
<th>XML Structure</th>
<th>Description</th>
<th>Type/Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;SubID&gt;)?</td>
<td>This field is depreciated and no longer used in any API. Please ignore.</td>
<td>String/50</td>
</tr>
<tr>
<td>&lt;SubjectName&gt;</td>
<td>The SubjectName in certificate.</td>
<td>String/255</td>
</tr>
<tr>
<td>&lt;SuccessCode&gt;</td>
<td>Code in the Order and Query Response Headers which indicates the success or failure of the request. A zero SuccessCode indicates a success with no warnings. A positive SuccessCode indicates a success with warnings. A negative SuccessCode indicates a failure. Note that if the Success in non-zero an accompanying Errors structure will be present.</td>
<td>Int</td>
</tr>
<tr>
<td>&lt;TargetCERT&gt;</td>
<td>The base64-encoded certificate you are transferring from</td>
<td>String/4000</td>
</tr>
<tr>
<td>&lt;TargetOrderID&gt;</td>
<td></td>
<td>String/50</td>
</tr>
<tr>
<td>&lt;Timestamp&gt;</td>
<td>A date timestamp used in a variety of contexts. Note that the XML format is: YYYY-MM-DDTHH:MM:SS.000Z (for example, 2001-01-01T24:00:00:000Z is for Jan 1, 2001 at midnight).</td>
<td>DateTime</td>
</tr>
<tr>
<td>&lt;TrustedOrderFlag&gt;</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>&lt;UserName&gt;</td>
<td>Required for user authentication</td>
<td>String/30</td>
</tr>
<tr>
<td>&lt;ValidityPeriod&gt;</td>
<td>The number of months that a certificate or site seal will be valid for. Defaults to 12 if not present.</td>
<td>String/30</td>
</tr>
<tr>
<td>&lt;Months&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;NotBefore&gt;)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&lt;NotAfter&gt;)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;ValidityPeriodOpt&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;VerificationUrl&gt;</td>
<td>An allowable URL which the metatag for URLVerification can occur on</td>
<td>String/300</td>
</tr>
<tr>
<td>&lt;VerificationFQDN&gt;</td>
<td>An allowable FQDN which the DNS txt record for DNS Verification can occur on</td>
<td>String/300</td>
</tr>
<tr>
<td>&lt;X509Cert&gt;</td>
<td>A base64-encoded certificate.</td>
<td>String/4000</td>
</tr>
</tbody>
</table>
14. Code Example

The following text is an example of an OrganizationSSL Certificate order in PHP 5. This code is for OVOrder only, and would follow after executing DecodeCSR.

```php
$url = 'http://testsystem.globalsign.com/wsdls/gasorder.wsdl';
$nameSpace = 'http://stub.order.gasapiserver.esp.globalsign.com';
$client = new SoapClient($url);
$params = array('OVOrder' => array(
    'Request' => array(
        'OrderRequestHeader' => array(
            'AuthToken' => array(
                'UserName' => 'PAR000000',
                'Password' => 'testpassword'
            ),
        ),
        'OrderRequestParameter' => array(
            'ProductCode' => 'OV',
            'OrderKind' => 'new',
            'Licenses' => '1',
            'ValidityPeriod' => array(
                'Months' => '12',
            ),
            'CSR' => '-----BEGIN CERTIFICATE REQUEST-----
MIIBvzCCASgCAQAwbDELMAkGA1UEBhMCQkUxEDAOBgNVBAgTB0JyYWJhbnQxDzAN
BgNVBAcTBkxldXZlbjETMBEGA1UEChMKR2xvYmFsU2lnbjEMMAoGA1UECxMDd2Vi
MRcwFQYDVQQDEw53d3cucGpiMTU0LmNvbTCBnzANBgkqhkiG9w0BAQEFAAAO8jBy
ykCgYEAsq7c9pSA56FgV5GIR/Ih1edMMyj9q9fNraAl1b3c7QQUykhPBeRVx
6gFz/1z2gH6l/9Af6q2Fp3AT7+jxtXxKR5Uw2ahlTw3WyV+0gOUIYT/3WcbDQ3kP
D2X1me/50sAx59jaedOl1qTsGFd+GHP54cVERfffF3JXTD4a+AMCAwEAAaATMBEG
CSq5S1b3DQE3kJEFPAJus5JANBqphkHJ9w0BAQUPAAObgQByVUN0PUxnsk3hYyg
C1i8F5sz/hB0lyx611xz05cMBw=nnN+61IfyfyfP8haO91lACVCDBwYQYRMb7
LqUex4xr2t2C80tIgp3iJMI8Hs+9xHvlB4WjdinBnnkfgYH890yQUPVbKwXQJ12F
k+h2sEZYU7k222WYoVnS212G1w=-----END CERTIFICATE REQUEST-----',
            'Options' => array(
                'Option' => array(
                    'OptionName' => 'SAN',
                    'OptionValue' => 'true',
                ),
            ),
            'ContactInfo' => array(
                'FirstName' => 'First',
                'LastName' => 'Last',
                'Phone' => '123456',
                'Email' => 'admin@test12345.com',
            ),
            'OrganizationInfo' => array(
                'OrganizationName' => 'test12345',
                'OrganizationAddress' => array(
                    'AddressLine1' => 'Philipssite 5',
                    'City' => 'Leuven',
                    'Region' => 'Brabant',
                    'PostalCode' => '3001',
                    'Country' => 'BE',
                    'Phone' => '02123456',
                ),
                'SANEntries' => array(
                    'SANEntry' => array(
                        'SANOptionType' => '1',
                        'SubjectAltName' => 'mail.test12345.com',
                    ),
                    array(
                        'SANOptionType' => '3',
                        'SubjectAltName' => 'tester.test12345.com',
                    ),
                ),
            ),
        ),
    ),
),

```
},
}$params);
$res = $client->__soapCall("OVOrder",$params);
print_r($res);
?>