Comodo Custom Client Certificate – Integration Guide

1. Allowed “Subject” Attributes....1
2. Prepare Your “Empty” Webpages....2
3. Signup Page....2
    3.1. Add the default CSS definitions....2
    3.2. Add a <SCRIPT> tag for the Comodo certEnroll.js code library....2
    3.3. Add the default JavaScript code....2
    3.4. Add the required HTML forms and <DIV> tag....3
    3.5. Customise the HTML, JavaScript and CSS....3
4. Signup Error Page....4
5. Signup Successful Page....5
6. Collection Email Template....5
    6.1. Open the Email Template....5
    6.2. About the Dynamically Replaced Values....5
    6.3. Customise the Email Template....6
    6.4. Supply your Email Template to Comodo....6
7. Collection Page....6
    7.1. Add the default CSS definitions....6
    7.2. Add a <SCRIPT> tag for the Comodo certInstall.js code library....6
    7.3. Add the default JavaScript code....6
    7.4. Add the required HTML form and <DIV> tag....6
    7.5. Customise the added HTML....7
8. Publish Your Customised Webpages....7
9. Other Considerations....7
    9.1. What if the client tampers with the signup page?....7

Version History

1.00	Original version. Incomplete!	
1.01	Original version completed.	
1.02	Changed:	`&2000000` changed to `&H2000000` in the *generateRequest()* function of the Signup Page VBScript code. *handleParameters()* function added to Collection/Installation Page and <BODY> tag changed, so that the collection code is automatically filled in.
1.03	Added:	‘caCertificateID’ parameter to the Signup Page.
1.04	Added:	Mozilla/Netscape support completed (changes made to the Collection/Installation Page JavaScript and HTML form).
	Changed:	Corrected the URL of the Collection/Installation template webpage.
1.05	Added:	“serialNumber” attribute.
1.06	Added:	Support for Microsoft Windows Vista.
	Changed:	To simplify things, much of the JavaScript code has been put into 2 separate external .js files.
1.07	Added:	Various “Subject” attributes.

1. Allowed “Subject” Attributes

You will need to decide the type and order of the attributes that will appear in the “Subject” field of your Custom Client Certificates. You may have multiple attributes of the same type (e.g. multiple OUs are fairly common). You will also need to consider which attributes are required and which, if any, are optional. Below is a list of the attribute types that are currently recognised by the Comodo Custom Client Certificate system:

Full Name	“Windows Name”	OID	Maximum Length	Defined by	Notes
commonName	CN	2.5.4.3	64	X.520	
surname	SN	2.5.4.4	64	X.520	
serialNumber	2.5.4.5	2.5.4.5	64	X.520	
countryName	C	2.5.4.6	2	X.520	Must be a valid ISO-3166 country code
localityName	L	2.5.4.7	128	X.520	
stateOrProvinceName	S	2.5.4.8	128	X.520	
streetAddress	STREET	2.5.4.9	128	X.520	
organizationName	O	2.5.4.10	64	X.520	
organizationalUnitName	OU	2.5.4.11	64	X.520	
title	T	2.5.4.12	64	X.520	
description	Description	2.5.4.13	1024	X.520	
postalCode	PostalCode	2.5.4.17	40	X.520	
postOfficeBox	POBox	2.5.4.18	40	X.520	
telephoneNumber	Phone	2.5.4.20	32	X.520	Must comply with ITU-T Rec. E.123 (e.g. “+44 582 10101”)
givenName	G	2.5.4.42	64	X.520	
initials	I	2.5.4.43	64	X.520	
emailAddress	E	1.2.840.113549.1.9.1	255	PKCS#9	Must be a valid RFC822 email address subject:emailAddress
DocumentoNacionaldeIdentidad	DNI	1.3.6.1.4.1.5734.1.4			
SIRENE	SIRENE	1.3.2			
rfc822Name	rfc822Name				Must be a valid RFC822 email address subjectAlternateName:rfc822Name
subjectUniqueIdentifier	subjectUniqueIdentifier				Subject:uniqueIdentifier
uniqueIdentifier	uniqueIdentifier				SubjectAlternateName:uniqueIdentifier
sAMAccountName	sAMAccountName	1.2.840.113556.1.4.221			
userId	UID				
userPrincipalName	UPN	1.3.6.1.4.1.311.20.2.3			subjectAlternateName:UPN
unstructuredName	unstructuredName	1.2.840.113549.1.9.2			
domainComponent	DC	0.9.2342.19200300.100.1.25			
extension-certificateTemplate		1.3.6.1.4.1.311.21.7			The certificateTemplate extension will be copied from the request into the issued certificate.
servicePrincipalName	SPN	1.2.840.113556.1.4.771			subjectAlternateName:SPN
dnsName	dnsName				subjectAlternateName:dnsName
applicationPolicies	applicationPolicies				The only valid usage is applicationPolicies=Y When present, an applicationPolicies extension will be added with the same content as the EKU extension.

If you require any other attributes to be supported, please contact Comodo.

Please note that RFC5280 requires that when email addresses are included they MUST appear in the subjectAlternateName:rfc822Name and that simultaneous inclusion of the emailAddress attribute in the subject emailAddress to support legacy implementations is deprecated but permitted.

2. Prepare Your “Empty” Webpages

It is recommended that you read the associated “System Overview” document before reading the rest of this document.

It is assumed that you will want to “host” your customised Custom Client Certificate signup/collection/installation webpages on your own website. However, if you prefer, it can be arranged for Comodo to “host” these webpages.

You will need to create a separate .html file for each of the following purposes:

- Custom Client Certificate signup.
- Custom Client Certificate signup error (OPTIONAL).
- Custom Client Certificate signup successful (OPTIONAL).
- Custom Client Certificate collection/installation.

Prepare each of these “empty” webpages with your own style-sheets, headers, footers, etc, leaving space on the page for the Custom Client Certificate content.

3. Signup Page

A template webpage is available at https://secure.comodo.com/customClientCert/apply_template.html. It contains all of the required HTML and JavaScript code described in this section.

Note: To ensure compatibility with Windows Vista, we recommend that you serve your Signup Page to your customers over HTTPS.

To construct your Signup Page, create a basic HTML file in your favourite webpage editor, then...

3.1. Add the default CSS definitions

We recommend that you copy the <STYLE> section from the template webpage mentioned above, then modify it later to suit your needs.

3.2. Add a <SCRIPT> tag for the Comodo certEnroll.js code library

Comodo's certEnroll.js code library handles the Private Key Generation and the creation of the Certificate Signing Request (CSR). It provides cross-browser support for Windows (CertEnroll for Vista upwards, XEnroll for old Windows versions) and any browser that supports the <KEYGEN> tag (e.g. Mozilla-based browsers). Add the following line at the end of the <BODY> section of your webpage:

```
<SCRIPT type="text/javascript" language="JavaScript" src="https://secure.comodo.com/signup/certEnroll.js"></SCRIPT>
```

3.3. Add the default JavaScript code

After the certEnroll.js <SCRIPT> tag, immediately before the </BODY> tag, add the following lines of JavaScript code:

```
<SCRIPT type="text/javascript" language="JavaScript">
  <!--
  function checkDetails(
    v_hiddenForm
  )
  {
    with (v_hiddenForm) {
      if (!_firstName.value)
        alert("You must specify a First Name!");
      else if (!_lastName.value)
        alert("You must specify a Last Name!");
      else if (_firstName.value.length + 1 + _lastName.value.length > 64)
        alert("First Name + Last Name is too long!");
      else if (!_emailAddress.value)
        alert("You must specify an Email Address!");
      else if (!_challengePassword.value)
        alert("You must specify a Challenge Password!");
      else
        return true;
    }
    return false;
  }

  function copyDetails(
    v_hiddenForm,
    v_visibleForm
  )
  {
    with (v_hiddenForm) {
      subject_emailAddress_1.value = v_visibleForm._emailAddress.value;
      subject_commonName_2.value = v_visibleForm._firstName.value + " "
                                 + v_visibleForm._lastName.value;
      challengePassword.value = v_visibleForm._challengePassword.value;
      sharedSecret.value = v_visibleForm._sharedSecret.value;
    }
    return true;
  }

  function submitForm(
    v_hiddenForm,
    v_visibleForm
  )
  {
    if (!checkDetails(v_visibleForm))
      return false;
    else if (!copyDetails(v_hiddenForm, v_visibleForm))
      return false;
    else
      return true;
  }
  //-->
```

```
</SCRIPT>
```

3.4. Add the required HTML forms and <DIV> tag

At the appropriate place in the <BODY> section of the webpage, add the following HTML:

```
<FORM name="visibleForm" id="certEnroll_form">
  <TABLE>
    <TR id="certEnroll_field">
      <TD id="certEnroll_header" colspan="2">
        <B>Apply for Comodo Custom Client Certificate</B>: Template Page
      </TD>
    </TR>
    <TR id="certEnroll_field"><TD colspan="2"> </TD></TR>
    <TR id="certEnroll_field">
      <TD id="certEnroll_header" colspan="2">"Subject" Details</TD>
    </TR>
    <TR id="certEnroll_field">
      <TD id="certEnroll_fieldName">First Name</TD>
      <TD><INPUT type="text" name="_firstName" maxLength="62" size="32"></TD>
    </TR>
    <TR id="certEnroll_field">
      <TD id="certEnroll_fieldName">Last Name</TD>
      <TD><INPUT type="text" name="_lastName" maxLength="62" size="32"></TD>
    </TR>
    <TR id="certEnroll_field">
      <TD id="certEnroll_fieldName">Email Address</TD>
      <TD><INPUT type="text" name="_emailAddress" maxLength="255" size="32"></TD>
    </TR>
    <TR id="certEnroll_field">
      <TD id="certEnroll_fieldName">Challenge Password</TD>
      <TD><INPUT type="text" name="_challengePassword" maxLength="255" size="32"></TD>
    </TR>
    <TR id="certEnroll_field">
      <TD id="certEnroll_fieldName">Shared Secret</TD>
      <TD><INPUT type="text" name="_sharedSecret" maxLength="255" size="32"></TD>
    </TR>
  </TABLE>
</FORM>
<FORM name="hiddenForm" method="post" id="certEnroll_form"
      action="https://secure.comodo.com/products/!applyCustomClientCert"
      onSubmit="return submitForm(this, document.visibleForm)">
  <INPUT type="hidden" name="ap" value="xxxxx">
  <INPUT type="hidden" name="days" value="365">
  <INPUT type="hidden" name="errorURL" value="">
  <INPUT type="hidden" name="successURL" value="">
  <INPUT type="hidden" name="subject_emailAddress_1" value="" maxLength="255">
  <INPUT type="hidden" name="subject_commonName_2" value="" maxLength="64">
  <INPUT type="hidden" name="subject_organizationalUnitName_3" value="Test Department" maxLength="64">
  <INPUT type="hidden" name="challengePassword" value="" maxLength="255">
  <INPUT type="hidden" name="sharedSecret" value="" maxLength="255">
  <TABLE>
    <TR id="certEnroll_field">
      <TD colspan="2"> </TD></TR>
    <TR id="certEnroll_field">
      <TD colspan=2>
        <DIV id="certEnroll_advancedOptions" align="left"></DIV>
      </TD>
    </TR>
    <TR id="certEnroll_field">
      <TD colspan="2"> </TD></TR>
    <TR id="certEnroll_field">
      <TD colspan="2">
        <INPUT type="submit" value="Apply for Comodo Custom Client Certificate">
      </TD>
    </TR>
  </TABLE>
</FORM>
```

Note: The “certEnroll_advancedOptions” <DIV> element must be present. It is used by certEnroll.js to display browser-specific information on the webpage.

3.5. Customise the HTML, JavaScript and CSS

3.5.1. Specify the correct “ap=xxxxx” value

In the `<INPUT type="hidden" name="ap" value="xxxxx">` line, change `xxxxx` to the correct value, which you can find by logging into your Reseller account and examining your Ordering URLs.

3.5.2. Alternative values for “days”

The HTML “hiddenForm” form has the certificate validity hard-coded as 365 days. If you require 2 or 3 year certificates, you could either hard-code 730 or 1095 days, or offer a <SELECT> box for the client to choose between any of these values.

3.5.3. Specify an “Error URL”

The default “signup error” page is located at https://secure.comodo.com/customClientCert/applyError.html?errorCode=xx&errorMessage=yy&errorItem=zz.
As mentioned before, you may choose to customise this page. If so, change the value of the `errorURL` form field from `""` to the absolute URL of your customised “signup error” page. Details on how to create a customised “signup error” page are provided in a later section.

3.5.4. Specify a “Success URL”

The default “signup successful” page is located at https://secure.comodo.com/customClientCert/applySuccess.html.
As mentioned before, you may choose to customise this page. If so, change the value of the `successURL` form field from `""` to the absolute URL of your customised “signup successful” page. Details on how to create a customised “signup successful” page are provided in a later section.

3.5.5. Configure your “Subject” fields

The HTML “hiddenForm” form is preconfigured with a number of demonstration “Subject” fields. The names of these fields are of the form `subject_xxxx_nn`, where *xxxx* is the Full Name (see section 1) of the required attribute and *nn* controls the order of the attributes (1 means the first attribute in the certificate “Subject”, then 2, then 3, etc). The attributes contained in the “Subject” of each Custom Client Certificate are controlled solely by the names and values of these form fields.

It is strongly recommended that you keep the `subject_emailAddress_1` demonstration field (of course, you may change the *nn* value), because without an email address, no email will be sent to the client regarding collection/installation of the certificate, and the certificate will be unusable for Secure Email.

The `subject_commonName_2` demonstration field shows how two visible form fields (first name and last name in the “visibleForm”) can be combined into one field within the certificate “Subject”.

The `subject_organizationalUnitName_3` demonstration field shows how to incorporate a “Subject” attribute with a constant value that is not configurable by the client.

You will need to add whatever additional subject_xxxx_nn <INPUT> fields you require to the “hiddenForm”. You will also need to add appropriate <INPUT> fields to the “visibleForm”. It is recommended that you use the `maxLength=xx` attribute of the <INPUT> tag wherever possible (please consult section 1 for the minimum and maximum lengths of different “Subject” attribute types).

3.5.6. About the Challenge Password field

The client must specify a value for the `challengePassword` field. If at any time later the client wishes to perform certificate management functions such as revocation (not yet implemented), they will be required to enter this challenge password.

3.5.7. About the Shared Secret field

The `sharedSecret` field is optional. If you do not require it, you may remove it from the “visibleForm”, the “hiddenForm” and the copyDetails() function.
You may find the shared secret facility useful for your own validation procedures. It enables you the Reseller to send to the client a “shared secret” string of some kind through whatever communications channel you choose, which the client must then enter into the signup form. Whatever the client enters will be visible to you, the Reseller, as you are deciding whether or not to authorize the client’s order.

3.5.8. Specifying an alternative CA

If you have an agreement with Comodo for an alternative CA to be used to issue Custom Client Certificates, you will need to add the following additional parameter to the HTML “hiddenForm”:

```
<INPUT type="hidden" name="caCertificateID" value="nn">
```

Comodo will inform you of the value(s) that you may use for *nn*.

3.5.9. Configure the copyDetails() function

Having configured your “Subject” fields, you will need to change the copyDetails() function appropriately to copy all of the details from the “visibleForm” to the “hiddenForm”.

3.5.10. Configure the checkDetails() function

Having configured your “Subject” fields, you will need to change the checkDetails() function appropriately to control optional/required fields and to apply any other checks you may need to.

3.5.11. Apply cosmetic changes

You are welcome to make whatever changes you require to customise the look of the webpage. Be careful when modifying the default CSS though, since some of the styles it defines are used in HTML snippets that are dynamically generated by certEnroll.js (see section 3.2).

4. Signup Error Page

If you have chosen to customise the “signup error” page by specifying a value for the `errorURL` (see section 3.5.3), you will need to be aware that when Comodo redirects the client’s browser to this page, the following parameters are passed in the query string:

errorCode	A negative integer (see table below for possible values)
errorMessage	A string (see table below for possible values)
errorItem (OPTIONAL)	The name of the problematic “hiddenForm” field (if applicable for the error)

Here is a list of the possible values for *errorCode* and *errorMessage*:

errorCode	**errorMessage**
-1	Request was not made over https!
-2	*errorItem* is an unrecognised argument!
-3	The *errorItem* argument is missing!
-4	The value of the *errorItem* argument is invalid!
-7	*xx* is not a valid ISO-3166 country! NOTE: *xx* will be the value of any `subject_countryName_nn` field
-9	The CSR is not valid Base-64 data!
-10	The CSR cannot be decoded!
-11	The CSR uses an unsupported algorithm!
-12	The CSR has an invalid signature!
-13	The CSR uses an unsupported key size!
-14	An unknown error occurred!
-16	Permission denied!
-17	Request used GET rather than POST!

The JavaScript code in the default “signup error” page demonstrates how to parse these parameters. You are welcome to copy-and-paste this JavaScript, changing it to suit your requirements.

5. Signup Successful Page

If you have chosen to customise the “signup successful” page by specifying a value for the `successURL` (see section 3.5.4), note that there are no special instructions required to create this customised page.

6. Collection Email Template

When you authorise an order for a Custom Client Certificate, Comodo will (optionally) send an email to the client with collection/installation instructions.
If you require these emails to be sent, you first need to customise the email template to suit your requirements, then provide it to your Account Manager. Your Account Manager will then ensure that the template is added to Comodo's systems. After this has been done, Comodo will send collection emails to your clients on your behalf.

6.1. Open the Email Template

The email template (shown below) is available at https://secure.comodo.com/customClientCert/collection_email_template.txt.
Open (or copy-and-paste) it in a text editor.

```
Subject: Your certificate is ready for collection!
From: "Comodo Security Services" <secureemail@comodo.com>
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="(AlternativeBoundary)"
Content-Transfer-Encoding: 8bit
To: "__customerName__" <__emailAddress__>
__date__

--(AlternativeBoundary)
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit

Dear __customerName__,

Your certificate is now ready!

Your Collection Code is: __collectionCode__

1. Open this webpage:
     https://secure.comodo.com/customClientCert/collect_template.html
2. Copy and paste your Collection Code where requested on the webpage.
3. Click the "Collect and Install Certificate" button.


Comodo Security Services
secureemail@comodo.com

--(AlternativeBoundary)
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: 8bit

<html>
<body>
  <div><font face="Arial" size="2">Dear __customerName__,</font></div>
  <div> </div>
  <div><font face="Arial" size="2">Your certificate is now ready!</font></div>
  <div> </div>
  <div>
    <font face="Arial" size="2">Please click
      <a href="https://secure.comodo.com/customClientCert/collect_template.html?collectionCode=__collectionCode__"
        >here</a> to collect and install your certificate.
    </font>
  </div>
  <div> </div>
  <div>
    <font face="Arial" size="2">Comodo Security Services
      <br><a href="mailto:secureemail@comodo.com">
        secureemail@comodo.com
      </a>
    </font>
  </div>
</body>
</html>

--(AlternativeBoundary)--
```

6.2. About the Dynamically Replaced Values

In each collection email sent by Comodo, every occurrence of `__xxxx__` will be dynamically replaced, where `xxxx` is one of the following values:

Value of *xxxx*	**Behaviour**
customerName	This will be replaced with the value of the first `subject_commonName_nn` field. If there are no common name fields, the value of the first `subject_emailAddress_nn` field will be used instead. If there is no email address, then no collection email will be sent at all!
emailAddress	This will be replaced with the value of the first `subject_emailAddress_nn` field.
collectionCode	This is a random 16-character string generated by Comodo, which the client is required to enter on the collection/installation page.
date	This will be replaced with a MIME “Date:” header for the current date/time.

When customising your email template, you will need to:

- retain the occurrences of `__emailAddress__` that enable the email to be sent to the client.
- make sure that `__collectionCode__` is included somewhere, to enable the client to collect and install their certificate.

6.3. Customise the Email Template

You will need to change the contents of the email template to suit your requirements. This will include:

- changing the value of the “Subject:” line.
- changing the value of the “From:” line.
- specifying the correct “collection/installation” URL in the appropriate places.
- adding your own content, such as your contact details.
- any other cosmetic changes.

6.4. Supply your Email Template to Comodo

Once completed, you will need to send your customised Collection Email Template to your Comodo Account Manager, who will ensure that it is incorporated into Comodo's email generation system.

7. Collection Page

A template webpage is available at https://secure.comodo.com/customClientCert/collect_template.html. It contains all of the required HTML and JavaScript code described in this section.

Note: To ensure compatibility with Windows Vista, we recommend that you serve your Collection Page to your customers over HTTPS.

To construct your Collection Page, create a basic HTML file in your favourite webpage editor, then...

7.1. Add the default CSS definitions

We recommend that you copy the <STYLE> section from the template webpage mentioned above, then modify it later to suit your needs.

7.2. Add a <SCRIPT> tag for the Comodo certInstall.js code library

Comodo's certInstall.js code library handles the Certificate Installation. Like certEnroll.js, it provides cross-browser support.
Add the following line at the end of the <BODY> section of your webpage:

```
<SCRIPT type="text/javascript" language="Javascript" src="https://secure.comodo.com/signup/certInstall.js"></SCRIPT>
```

7.3. Add the default JavaScript code

After the certInstall.js <SCRIPT> tag, immediately before the </BODY> tag, add the following lines of JavaScript code:

```
<SCRIPT type="text/javascript" language="JavaScript">
  <!--
  function submitForm(
    v_form
  )
  {
    g_collectionCode = v_form.collectionCode.value;
    if (g_collectionCode == "") {
      alert("You must enter your Collection Code!");
      return false;
    }
    return true;
  }

  certInstall_tryToInstallCertificate(document.collect_form.collectionCode);
  //-->
</SCRIPT>
```

7.4. Add the required HTML form and <DIV> tag

At the appropriate place in the <BODY> of the webpage, add the following HTML:

```
<FORM name="collect_form" onSubmit="return submitForm(this)" id="certEnroll_form">
  <TABLE>
    <TR id="certEnroll_field">
      <TD id="certEnroll_header" colspan="2">
        <B>Collect Comodo Custom Client Certificate</B>: Template Page
      </TD>
    </TR>
    <TR id="certEnroll_field"><TD colspan="2"> </TD></TR>
    <TR id="certEnroll_field">
      <TD id="certEnroll_header" colspan="2">Collection Details</TD>
    </TR>
    <TR id="certEnroll_field">
      <TD id="certEnroll_fieldName">Collection Code</TD>
      <TD><INPUT type="text" name="collectionCode" maxLength="16" size="16"></TD>
    </TR>
    <TR id="certEnroll_field">&lt;TD colspan="2"&gt;&amp;nbsp;&lt;/TD&gt;&lt;/TR&gt;
    <TR id="certEnroll_field">
      <TD colspan="2">
        <INPUT type="submit" value="Collect Comodo Custom Client Certificate">
      </TD>
    </TR>
  </TABLE>
</FORM>
<DIV id="certInstall_status" align=left></DIV>
```

Note: The “certInstall_status” <DIV> element must be present. It is used by certInstall.js to display feedback on whether or not the certificate has been installed correctly.

7.5. Customise the added HTML

The only customisations that are required are any cosmetic changes that you wish to apply to the HTML form.

8. Publish Your Customised Webpages

When you have completed the above steps, you will need to publish your customised webpages on your website. You will also need to provide the URL of the signup page to your customers. Also, remember to supply your collection email template to Comodo, if you require Comodo to send collection emails to your clients.

9. Other Considerations

9.1. What if the client tampers with the signup page?

You may have noticed that there is nothing to stop the client tampering with the signup page, so that different `subject_xxxx_`*nn* fields are submitted to Comodo. Obviously, all of the submitted details will be displayed to you, the Reseller, before you authorise the order. However, if this possibility of tampering is a problem for you, you may wish to consider implementing a middle-layer between the client and Comodo. You could change the signup page to submit the form details to your own server, perform your own checks on the supplied values and then call the Comodo server from your own server.

The call to the Comodo server is documented separately in the “ApplyCustomClientCert” document.