



WEBADM AS SUBORDINATE CERTIFICATE AUTHORITY

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs Security.

WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to info@rcdevs.com.

1. Overview

RCDevs' suite offers a public key infrastructure service and that functionality is mandatory for the proper functioning of RCDevs solutions.

The default setup is to make WebADM/Rsignd a standalone CA. In that scenario, you just need to follow the default WebADM setup.

For customers which already have a CA in place and running, you can configure WebADM as a subordinate CA.

This document will present you with how to configure WebADM as a subordinate certificate authority of your enterprise certificate authority. In that document, we demonstrate it with a root CA configured on Windows Server 2019. You need at least WebADM v.2.1.

The steps to achieve this configuration are the following :

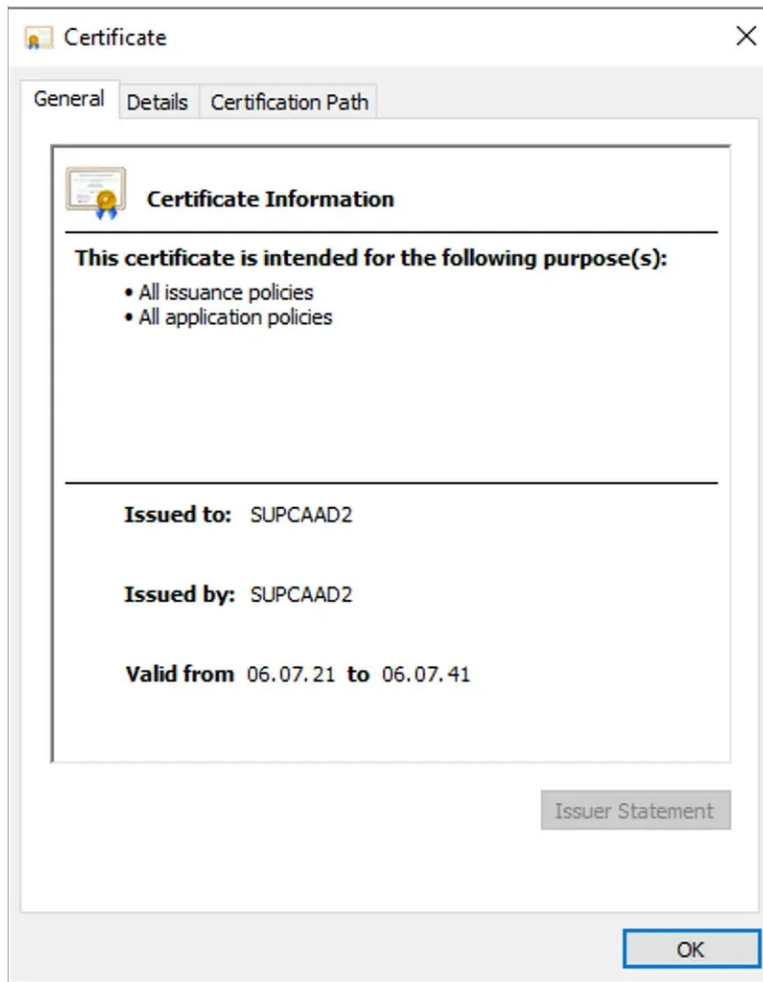
- > Have a root/enterprise or subordinate CA already configured (not shown in that document),
- > Generate from your enterprise CA a subordinate CA certificate and key pair to be used by WebADM (not shown in that document),
- > Import the subordinate certificate and key pair freshly generated in WebADM,
- > Generate the certificate signing request (CSR) and the key which will be used by WebADM SOAP services (webadm.csr and webadm.key),
- > Use OpenSSL to submit the CSR and get back the certificate issued and signed by our new subordinate CA.

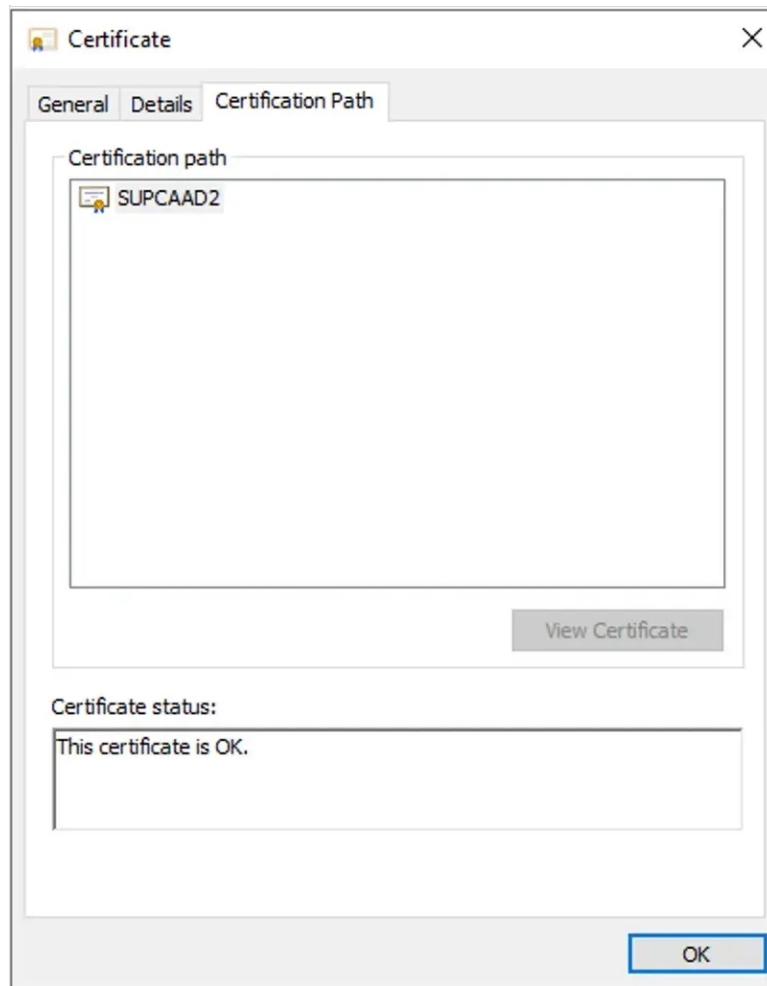
Important

These steps must be done before you implement anything with WebADM and its components, or before you issued any certificate with WebADM internal PKI. Else you will have to renew all certificates used in each deployed component using the SOAP API (port 8443). Actions must be performed just after the `/opt/webadm/bin/setup` script on the master node before configuring WebADM in cluster mode. Once the master is properly configured as a subordinate CA, the slave setup can be normally done as described in the [High Availability documentation](#).

2. ROOT/Enterprise Certificate Authority

Please, refer to your PKI provider documentation to set up your root/enterprise certificate authority. We assume in that documentation that the root CA is already configured. My enterprise certificate authority is called **SUPCAAD2**. Found below, the details of my root CA :





Found below, the content of my root CA certificate :

-----BEGIN CERTIFICATE-----

```
MIIDpTCCAo2gAwIBAgIQVz4fdQJ2QZhgNj2/KomwAjANBgkqhkiG9w0BAQsFADBZ
MRMwEQYKCZImiZPyLGBGRYDY29tMRYwFAYKCCZImiZPyLGBGRYGcmNkZXZzMRcw
FQYKCCZImiZPyLGBGRYHc3VwcG9ydDERMA8GA1UEAxMIU1VQQ0FBRDIwHhcNMjEw
NzA2MTYxNTU5WhcNNDUwNzA2MTYxNTU5WjBZMRMwEQYKCCZImiZPyLGBGRYDY29t
MRYwFAYKCCZImiZPyLGBGRYGcmNkZXZzMRcwFQYKCCZImiZPyLGBGRYHc3VwcG9y
dDERMA8GA1UEAxMIU1VQQ0FBRDIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQCDCVs3A/ksabY9Ljz+MhZbbQR4qZB6m/ofHdmzB+Am7Tsv4x9ulbB597jDQ
qNaOoW6QVtUGHZYofYXswiteyibFUIGTEjdPhDxU/SYvmGnmwZ0bEHIS4/xjYW/
MvO3ykEul6Ugo8zAVc5ABQ1CjHi+ZtuaLy+A9eyxAeFL1emxvIbUn2mQvmmbUOtX
Xt3c15iY33fAqgE1+pHm6E7MdUrVpESX5ynaQCgeL9BTgNbfuS00rk62UPoCx0q1
C/K9DqI7f7PQRntk56NKdelb600/xRuUcDx8XK1uyua1HFHZS2q7pjwIzFG8aRV2
ly1S8PxvalPi9RzXEO9AX+Mft/6IAGMBAAGjaTBnMBMGCSsGAQQBggjUAgQGHgQA
QwBBMA4GA1UdDwEB/wQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBbTa
8I08IOjn2sDzuzREii1NThrF3zAQBgkrBgEEAYl3FQEEAwIBADANBgkqhkiG9w0B
AQsFAAOCAQEAODLcWmSutShVG3gPpX5dCxOKGsfBeGJirpCrupSWeDy2oAsyZUfl
www/JYibjBD2RxOOikA8sAtjMmivzvLWaHkTZVpxgTQEUqoHqi0WakjJTvAZIjvt
OqxIso8HZPqZXAuVHW4wOZaLKeZSoE2qjAuYfHVI7ZILqUHeQ7i28nuZvQJtnDw
z292RfmbFE6iL9UAle7i6KdVS6Apx9Pjyt8vjN80hLEc2h94KKCK2S/q6cSTbIIP
JHMSm8SEGSHOAfdR6cVtLqH9FCxNQmyoPI8ISyQa6/HqrdwENoeExQ2XI059vkgY
/RjIEukzs8c3B7cw9C3ViG7AVPwxr0bosw==
```

-----END CERTIFICATE-----

Information contained in my CA certificate read with OpenSSL :

```
openssl x509 -in SUPCAAD2.crt -text -noout
```

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

57:3e:1f:75:02:76:41:98:46:9c:9d:bf:2a:89:b0:02

Signature Algorithm: sha256WithRSAEncryption

Issuer: DC = com, DC = rcdevs, DC = support, CN = SUPCAAD2

Validity

Not Before: Jul 6 16:15:59 2021 GMT

Not After : Jul 6 16:25:58 2041 GMT

Subject: DC = com, DC = rcdevs, DC = support, CN = SUPCAAD2

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public-Key: (2048 bit)

Modulus:

00:c2:56:cd:c0:fe:4b:1a:6d:8f:4b:8f:3f:8c:85:

96:db:41:1e:2a:64:1e:a6:fe:87:c7:76:6c:c1:f8:

09:bb:4e:cb:f8:c7:db:a5:6c:1e:7d:ee:30:d0:a8:

d6:8e:a1:6e:90:56:d5:06:1d:96:28:60:55:ec:c2:
2b:5e:ca:26:c5:50:81:93:12:37:4f:84:3c:54:fd:
26:2f:98:69:e6:c1:9d:1b:10:72:25:4b:8f:f1:25:
85:bf:32:f3:b7:ca:41:2e:97:a5:20:a3:cc:c0:55:
ce:40:05:0d:42:8c:78:be:66:db:9a:2f:2f:80:f5:
ec:b1:01:e1:4b:d5:e9:b1:be:56:d4:9f:69:90:be:
69:9b:50:eb:57:5e:dd:dc:d7:98:98:df:77:c0:aa:
01:35:fa:91:e6:e8:4e:cc:75:4a:d5:a4:44:97:e7:
29:da:40:28:1e:2f:d0:53:80:d6:df:b9:2d:34:ae:
4e:b6:50:fa:02:c7:4a:b5:0b:f2:bd:0e:a9:7b:7f:
b3:d0:46:7b:64:e7:a3:4a:75:e2:1b:eb:4d:3f:c5:
1b:94:70:3c:7c:5c:ad:6e:ca:e6:b5:1c:51:d9:4b:
6a:bb:a6:3c:25:64:51:bc:69:15:76:97:2d:52:f0:
fc:6f:6a:53:e2:f5:1c:d7:10:ef:40:5f:e3:1f:b7:
fe:a5

Exponent: 65537 (0x10001)

X509v3 extensions:

1.3.6.1.4.1.311.20.2:

...C.A

X509v3 Key Usage: critical

Digital Signature, Certificate Sign, CRL Sign

X509v3 Basic Constraints: critical

CA:TRUE

X509v3 Subject Key Identifier:

DA:F0:8D:3C:94:E8:E7:DA:C0:F3:BB:34:44:8A:2D:4D:4E:1A:C5:DF

1.3.6.1.4.1.311.21.1:

...

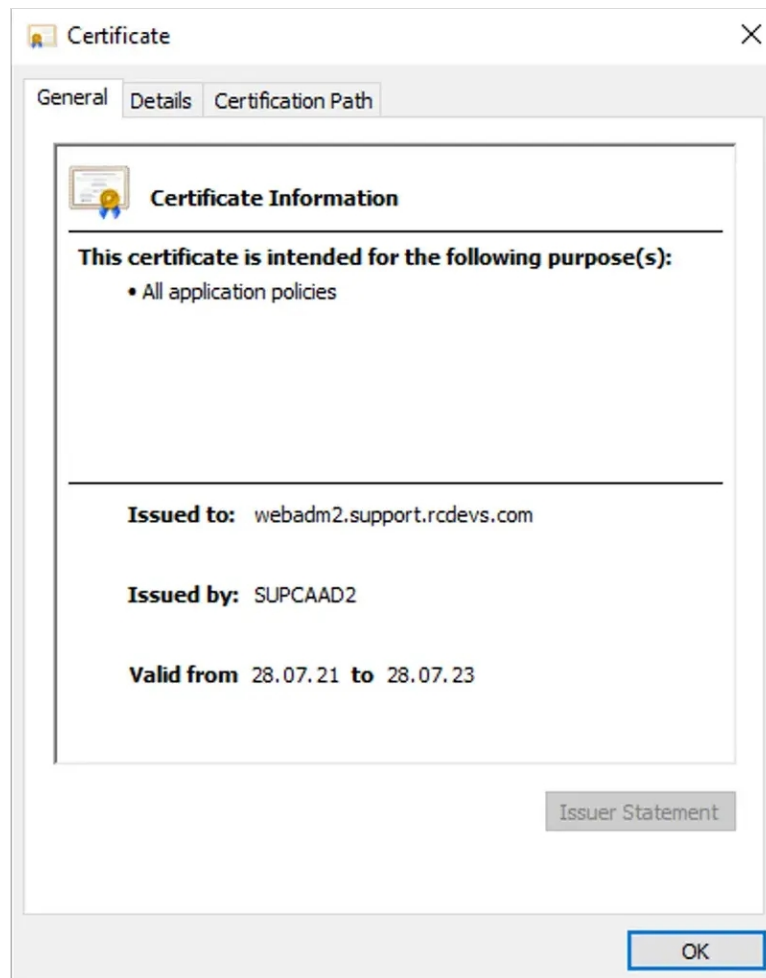
Signature Algorithm: sha256WithRSAEncryption

38:32:dc:5a:64:ae:b5:28:55:1b:78:0f:a5:7e:5d:0b:13:8a:
1a:c7:c1:78:62:62:ae:90:ab:ba:94:96:78:3c:b6:a0:0b:32:
65:47:e5:bf:0c:3f:25:88:9b:8c:10:f6:47:13:8e:8a:40:3c:
b0:0b:63:32:68:af:ce:f2:d6:68:79:13:65:5a:71:81:34:04:
52:aa:07:aa:2d:16:02:42:63:4e:f0:19:96:3b:ed:3a:ac:65:
b2:8f:07:64:fa:99:5c:0b:95:1d:6e:30:39:96:8b:29:e6:52:
a0:4d:aa:26:30:2e:61:f1:d5:23:b6:48:2e:a5:07:79:0e:e2:
db:c9:ee:66:f4:09:b6:70:f0:cf:6f:76:45:f9:9b:14:4e:a2:
2f:d5:00:21:ee:e2:e8:a7:55:4b:a0:29:c7:d3:c9:ca:df:2f:
8c:df:34:84:b1:1c:da:1f:78:28:a0:8a:d9:2f:ea:e9:c4:93:
6e:59:4f:24:73:12:9b:c4:84:19:21:ce:01:f7:6b:e9:c5:6d:
2e:a1:fd:14:2c:4d:42:6c:a8:3c:8f:08:4b:24:1a:eb:f1:ea:
ad:dc:04:36:87:84:c5:0d:97:23:4e:7d:be:48:32:fd:18:e5:
12:e9:33:b3:c7:37:07:b7:30:f4:2d:d5:88:6e:c0:54:fc:31:
af:46:e8:b3

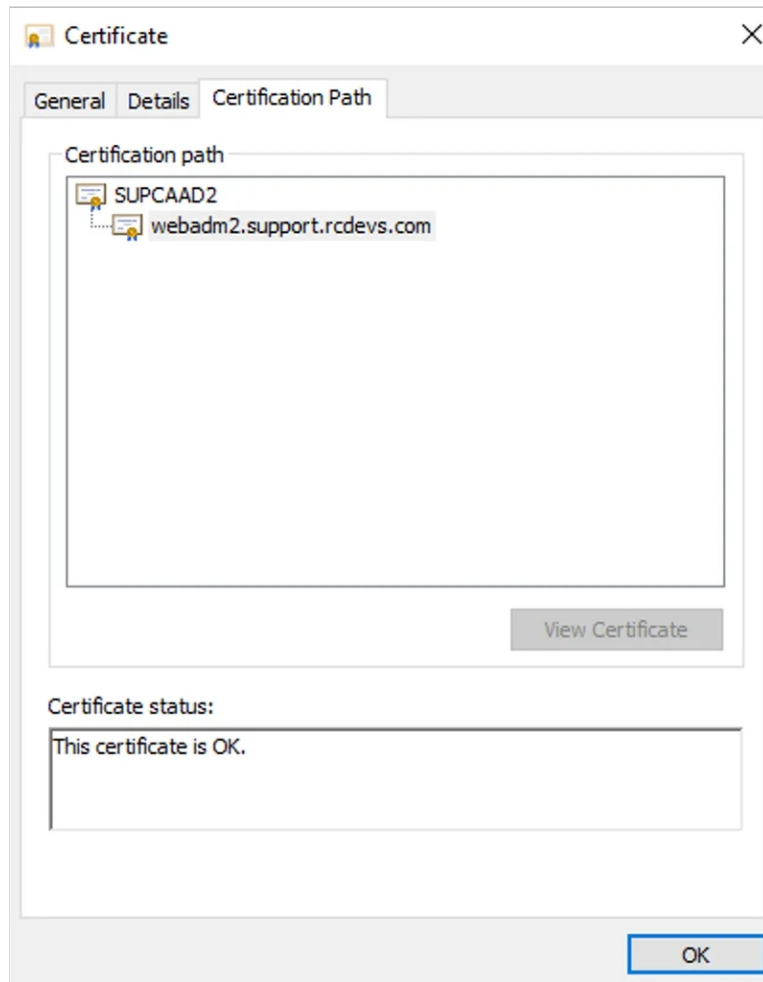
3. Subordinate CA certificate

Please, refer to your PKI provider documentation to generate a Subordinate CA certificate which will be used by WebADM. Found below, the details of my subordinate certificate which will be imported on WebADM to make WebADM a subordinate certificate

authority of your enterprise CA. My WebADM server which will be configured as SubCA is named **webadm2.support.rcdevs.com**. Found below the details:



Below, the proof that is has been issued by my root/enterprise certificate authority:



Found below the content of my subordinate certificate file and his key which will be used by WebADM as ca.crt and ca.key in `/opt/webadm/pki/ca/` folder:

Subordinate CA certificate (ca.crt):

-----BEGIN CERTIFICATE-----

MIIFWDCCBECgAwIBAgIHTAAAAAqFCxCd4ea1rgAAAAACjANBgkqhkiG9w0BAQsF
ADBZMRMwEQYKCZImiZPyLGBGRYDY29tMRYwFAYKCZImiZPyLGBGRYGcmNkZXZz
MRcwFQYKCZImiZPyLGBGRYHc3VwcG9ydDERMA8GA1UEAxMIU1VQQ0FBRDIwHhcN
MjEwNzI4MTU1NDMxWhcNMjMwNzI4MTYwNDMxWjCBpzELMAkGA1UEBhMCTFUEzAR
BgNVBAGTCkx1eGVtYm91cmcxZzANBgNVBACTBKJlbHZhbDEXMBUGA1UEChMOU3Vw
cG9ydCBSQ0RldnMxCzAJBgNVBAsTAKIUMSMwIQYDVQQDExp3ZWJhZG0yLnN1cHBv
cnQucmNkZXZzLmNvbTEnMCUGCSqGS1b3DQEJARYYeW9hbm5Ac3VwcG9ydC5yY2Rl
dnMuY29tMIGfMA0GCSqGS1b3DQEBAAQAA4GNADCBiQKBgQCXQsn7nClgZmS4XbFy
CkNPmiT7c9rgCdZeR9jo1uKQkXzzJ8p8bmSPd+Xd39IupHJbSRybNyDtN425Q/wW
FYBTmwdBCWMQkGF5itke6sim6pOG8a8EVbMKH/WLEzfBoeR2zlzEFTueBp+NGwXy
1VVgdYman26wNsXPvQEI2YkxbQIDAQABo4ICTDCCAgwDgYDVR0PAQH/BAQDAgGG
MBkGCSsGAQQBgjcUAQMHgoAUwB1AGIAQwBBMB0GA1UdDgQWBbBQtoARJo1wNiAcp
L4Gsyoll7aoGzAFBgNVHSMEGDAWgBTa8I0I0jn2sDzuzREii1NThrF3zCBzwYD
VR0fBIHHMIHEMIHBoIG+olG7hoG4bGRhcDovLy9DTj1TVVBDQUFEMixDTj1BRDE5
LTIsQ049Q0RQLENOPVB1YmXpYyUyMETleSUyMFNlcnZpY2VzLENOPVNIcnZpY2Vz
LENOPUNvbmZpZ3VyYXRpb24sREM9c3VwcG9ydCxEQz1yY2RldnMsREM9Y29tP2NI
cnRpZmljYXRlUmV2b2NhdGlvbXpc3Q/YmFzZT9vYmpIY3RDdbGFzcz1jUkxEaXN0
cmliidXRpb25Qb2ludDCB9wYIKwYBBQUHAQEeEgeowgecwgbEGCCsGAQUFBzAChoGk
bGRhcDovLy9DTj1TVVBDQUFEMixDTj1BSUESQ049UHVibGljTlws2V5JTlWU2Vy
dmVjZXMzQ049U2VydmljZXMzQ049Q29uZmlndXJhdGlvbXpc3Qz1zdXBwb3J0LERN
PXJjZGV2cyxEQz1jb20/Y0FDZXJ0aWZpY2F0ZT9iYXNIP29iamVjdENsYXNzPWNI
cnRpZmljYXRpb25BdXRpb25BdXRpb25BdXRpb25BdXRpb25BdXRpb25BdXRpb25B
c3VwcG9ydC5yY2RldnMuY29tL29jc3AwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG
9w0BAQsFAAOCAQEAAoR5d7dlmjfXfbnqLS+kvlp59X7HQGT8MnXj74iesoxRQES
i0LmizyroD4UGS0zzTXjQUVfRQqoSesdSKetekXJSPue3KS0i+HCfCPhK4YxTojV
rdmYrpCjf0SoYaOaQoNmQdt28Wvj+WlQU91gGRdiyN5zVuzyz8cR2NLlhSdKcVba
zAZeUdkKIX16KIVhthdvmtJfhEjePef8/BOpzHVVAnNkhqRuPZOEk2bbH05wBJKa
fHFnLy1o/BI08b0ye7fnTLYiWxojahD5PYF8QlcMMgl6glBD67VSWvyCH7NVv46Y
jLlj9pUdR7bis7OJKtkXoJolHRrlrARAJYx8rA==

-----END CERTIFICATE-----

Subordinate CA key (ca.key) :

-----BEGIN PRIVATE KEY-----

```
MIIcDgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAJdCyfucKWBmZLhd
sXIKQ0+aJptz2uAJ1I5H2OjW4pCRfPMnynxuZ1935d3f0i6kcltjFhs3IO03jbID
/BYVgF0bB0EJYxCQYXmK2R7qyKbqk4bxrwRVswof9YsTN8Gh5HbMjMQVO54Gn40b
BfLVVWB1iZqfbrA2xc+9AQjZiFtAgMBAAECgYB1jozYJbfqmemxd+/Ox7ckijlx
cwsqj7qxL3mfDFKtNMU9XTF7n3g8IFzgIXGCmn/d/mFV6XSHqGNXF0U2DKPcJHgX
VFhooj52v6yh3a18L0ngAFd84jyMvNM1PsG9iHPvn2sGDy05qCdmnLhABWksdhHY
00Bf6FuHKgLL1MTxaQJBAMgV4UCIIBYlVwuQKSHDJYrU5UjTSUIUwUQHEAWJxQ7
rHnZszHDuAYWXOUjvNOuJ4a6UXi3qiqiB7rytDe5Bz8CQQDBh/9pZo9LR35WmO1m
x0m/SDAX6/rFyM6FFAEntz1bs9PJEjumUwM8sXGR4NhXbtHr5JywKz3vIcNgU3DA
FihTAKADsCOOXPOSk5mTW+blIXgh7HqF7Timzhh5p2pd5AqkXNU5Ccl70Je7xP3B
WwSYAkXIPfbyerAwSPxLfd3EiSyRAkBrNv+Nkc7iwonASdqDbPZZLPfP2ODFv9iB
qzJ0oTQx4G783sgC/cw2TluBajIR5ggP6kfQhtjZ71eAvtkg4WWXAkEAvpelsk/5
2Mv0m+VG+nWWCtA9ncwqfwfIz9HGHOs4KZ2uhglac17XQWvkKbplB8sYKvG9qNey
HIDEqQ7fouU/+w==
-----END PRIVATE KEY-----
```

Information contained in the certificate read with OpenSSL :

```
openssl x509 -in ca.crt -text -noout
```

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

1c:00:00:00:0a:85:0b:10:9d:e1:e6:b5:ae:00:00:00:00:00:0a

Signature Algorithm: sha256WithRSAEncryption

Issuer: DC = com, DC = rcdevs, DC = support, CN = SUPCAAD2

Validity

Not Before: Jul 28 15:54:31 2021 GMT

Not After : Jul 28 16:04:31 2023 GMT

Subject: C = LU, ST = Luxembourg, L = Belval, O = Support RCDevs, OU = IT, CN = webadm2.support.rcdevs.com, emailAddress = yoann@support.rcdevs.com

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public-Key: (1024 bit)

Modulus:

```
00:97:42:c9:fb:9c:29:60:66:64:b8:5d:b1:72:0a:
43:4f:9a:24:fb:73:da:e0:09:d6:5e:47:d8:e8:d6:
e2:90:91:7c:f3:27:ca:7c:6e:64:8f:77:e5:dd:df:
d2:2e:a4:72:5b:49:16:1b:37:20:ed:37:8d:b9:43:
fc:16:15:80:53:9b:07:41:09:63:10:90:61:79:8a:
d9:1e:ea:c8:a6:ea:93:86:f1:af:04:55:b3:0a:1f:
f5:8b:13:37:c1:a1:e4:76:cc:8c:c4:15:3b:9e:06:
9f:8d:1b:05:f2:d5:55:60:75:89:9a:9f:6e:b0:36:
```

```
c5:cf:bd:01:08:d9:89:31:6d
Exponent: 65537 (0x10001)
X509v3 extensions:
  X509v3 Key Usage: critical
    Digital Signature, Certificate Sign, CRL Sign
  1.3.6.1.4.1.311.20.2:
    .
.S.u.b.C.A
  X509v3 Subject Key Identifier:
    2D:38:04:49:A3:5C:0D:88:07:29:2F:81:AC:CA:82:25:2F:B6:A8:1B
  X509v3 Authority Key Identifier:
    keyid:DA:F0:8D:3C:94:E8:E7:DA:C0:F3:BB:34:44:8A:2D:4D:4E:1A:C5:DF

  X509v3 CRL Distribution Points:

    Full Name:
      URI:ldap:///CN=SUPCAAD2,CN=AD19-
2,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=support,DC=rcdevs,DC=com
certificateRevocationList?base?objectClass=cRLDistributionPoint

    Authority Information Access:
      CA Issuers -
URI:ldap:///CN=SUPCAAD2,CN=AIA,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=supp
cACertificate?base?objectClass=certificationAuthority
      OCSP - URI:http://AD19-2.support.rcdevs.com/ocsp

  X509v3 Basic Constraints: critical
    CA:TRUE
Signature Algorithm: sha256WithRSAEncryption
a1:1e:5d:ed:d2:26:25:f9:57:7d:b9:ea:2d:2f:a4:be:92:29:
e7:d5:fb:1d:01:ad:f0:c9:d7:8f:be:22:7a:ca:31:45:01:12:
8b:42:e6:8b:3c:ab:a0:3e:14:19:2d:33:cd:35:e3:41:45:5f:
45:0a:a8:49:eb:1d:48:a1:2d:7a:45:c9:48:fb:9e:dc:a4:b4:
8b:e1:c2:7c:23:e1:2b:86:31:4e:88:d5:ad:d9:98:ae:90:a3:
7f:44:a8:61:a3:9a:42:83:66:41:db:76:f1:6b:c9:f9:69:50:
53:dd:60:19:17:62:c8:de:73:56:ec:f2:cf:c7:11:d8:d2:c8:
85:27:4a:0a:f6:da:cc:06:5e:51:d9:0a:95:7d:7a:2a:55:61:
b6:17:6f:9a:d2:5f:84:48:de:3d:e7:fc:fc:13:a9:cc:75:55:
02:73:64:86:a4:6e:3d:93:84:93:66:db:1f:4e:70:04:92:9a:
7c:71:67:2f:2d:68:fc:12:34:f1:bd:32:7b:b7:e7:4c:b6:22:
5b:1a:09:6a:10:f9:3d:81:7c:42:57:0c:32:09:7a:82:50:43:
eb:b5:52:5a:fc:82:1f:b3:55:bf:8e:98:8c:b9:63:f6:95:1d:
47:b6:e2:b3:b3:89:2a:d9:17:a0:9a:08:1d:1a:e5:ac:04:40:
8d:8c:7c:ac
```

This certificate and the key will replace the default `ca.crt` and `ca.key` generated after the WebADM setup script in `/opt/webadm/pki/ca/` folder. Remove the existing `ca.crt` and `ca.key` files available in that folder and copy the new ones. The new `ca.crt` file must also be copied in `/opt/webadm/pki/trusted/` folder. All CA certificates chain between

your root CA and the WebADM CA must also be copied in the `trusted/` folder. Once the 2 (or more) CA certificates are copied in the `trusted/` folder do the make command inside the folder:

```
[root@webadm2 tmp]# cd /opt/webadm/pki/trusted/
[root@webadm2 trusted]# make
ADROOT.crt    ... 148c96ae.0
ca.crt        ... faff8618.0

[root@webadm2 trusted]# ls -al
lrwxrwxrwx. 1 root root  10 Dec  8 11:54 148c96ae.0 -> ADROOT.crt
-rw-r--r--. 1 root root 1326 Aug 16 11:37 ADROOT.crt
-rw-r--r--. 1 root root 1071 Dec  8 10:46 Makefile
-rw-r--r--. 1 root root 1915 Aug 16 11:38 ca.crt
lrwxrwxrwx. 1 root root   6 Dec  8 11:54 faff8618.0 -> ca.crt
```

My WebADM server is now configured as a subordinate CA of my root/enterprise CA. The next step is to generate the certificate for WebADM services usage.

3. Regenerate WebADM certificate and key based on CSR

WebADM/Rsignd is now configured as a subordinate CA of our enterprise CA, and we need now to issue a certificate and its associate key which will be used by WebADM SOAP services and signed by our new subordinate CA.

On your WebADM server, navigate in `/opt/webadm/pki/` and you should find the default `webadm.crt`, `webadm.key` and `webadm.csr` generated during the first WebADM setup script.

```
[root@webadm2 trusted]# cd /opt/webadm/pki/
[root@webadm2 pki]# ls -al
total 12
drwx-----. 2 root root   69 Dec  8 10:46 ca
drwxr-xr-x.  2 root root   90 Dec  8 11:54 trusted
-rw-r-----. 1 root webadm 1164 Aug 26 10:44 webadm.crt
-rw-r--r--.  1 root root  1094 Aug 16 11:28 webadm.csr
-rw-r-----. 1 root webadm 1708 Aug 16 11:27 webadm.key
```

You can remove `webadm.crt` and keep the existing `webadm.key` :

```
[root@webadm2 pki]# rm -f webadm.crt
```

The following command will generate a new certificate based on the existing CSR (`webadm.csr`) and generate the associate certificate signed by our subordinate CA (WebADM):

```
[root@webadm2 pki]# openssl x509 -req -days 365 -in /opt/webadm/pki/webadm.csr -out /opt/webadm/pki/webadm.crt -CA /opt/webadm/pki/ca/ca.crt -CAkey /opt/webadm/pki/ca/ca.key -CAserial /opt/webadm/pki/ca/serial
```

Note

In the previous command, I configure the validity to 365 days. You can extend the validity if needed. WebADM should auto-renew its certificate when the certificate is near expiration after a restart of WebADM services.

You should have the following output if everything goes fine :

```
Signature ok
subject=C = LU, ST = Luxembourg, L = LU, O = RCDevs Support, OU = ITSEC, CN = webadm2,
emailAddress = yoann@support.rcdevs.com
Getting CA Private Key
```

And the certificate should now be there :

```
[root@webadm2 pki]# ls -al
total 16
drwx-----. 2 root root  69 Dec  8 10:46 ca
drwxr-xr-x.  2 root root  90 Dec  8 11:54 trusted
-rw-r--r--.  1 root root 1164 Dec  8 12:23 webadm.crt
-rw-r--r--.  1 root root 1094 Aug 16 11:28 webadm.csr
-rw-r-----.  1 root webadm 1708 Aug 16 11:27 webadm.key
```

Below, I execute the OpenSSL verification command to check the newly generated certificate match with the old key:

```
[root@webadm2 pki]# openssl rsa -modulus -noout -in webadm.key | openssl md5
(stdin)= 67ded38c0ad5bf0ab68dac8e87e0f5fb

[root@webadm2 pki]# openssl x509 -modulus -noout -in webadm.crt | openssl md5
(stdin)= 67ded38c0ad5bf0ab68dac8e87e0f5fb
```

It matches!

Now, I verify if the issuer is correct for my new WebADM certificate. It must be issued by my WebADM Subordinate CA :

```
[root@webadm2 pki]# openssl x509 -in webadm.crt -noout -text
```

Certificate:

Data:

Version: 1 (0x0)

Serial Number: 58 (0x3a)

Signature Algorithm: sha256WithRSAEncryption

Issuer: C = LU, ST = Luxembourg, L = Belval, O = Support RCDevs, OU = IT, CN = webadm2.support.rcdevs.com, emailAddress = yoann@support.rcdevs.com

Validity

Not Before: Dec 8 11:23:33 2021 GMT

Not After : Dec 8 11:23:33 2022 GMT

Subject: C = LU, ST = Luxembourg, L = LU, O = RCDevs Support, OU = ITSEC, CN = webadm2, emailAddress = yoann@support.rcdevs.com

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public-Key: (2048 bit)

Modulus:

```
00:cc:39:17:43:8d:a9:be:16:1c:79:25:1f:e6:9f:
23:ed:64:6f:c6:53:87:9a:65:d1:a5:52:04:46:91:
2e:01:3d:39:24:82:33:48:e1:cc:08:32:86:fa:3a:
a5:67:9d:79:43:07:07:a2:43:50:b0:b3:fe:e9:41:
d1:af:3a:ae:9c:8c:cc:9f:fb:66:5e:af:53:8b:a5:
d6:5e:4f:83:a4:58:ce:b8:8a:de:ff:46:cd:02:90:
8d:75:16:0b:87:cf:eb:f4:bd:91:6f:d2:fe:06:5c:
3a:e5:fd:1d:73:25:20:80:8e:73:99:eb:ef:8b:41:
ce:1c:f5:f8:27:aa:85:07:e0:76:8d:4a:97:e3:98:
83:ba:c8:20:87:08:60:e6:7d:19:a8:17:55:a1:c1:
26:6d:5a:6b:c1:a0:3c:70:b2:92:b3:80:92:e7:f2:
3a:61:0c:ec:15:cd:c6:d8:ff:ed:f9:8c:c3:e6:11:
2e:5a:4e:7c:c6:2c:cc:c6:73:2d:6b:9d:63:26:92:
c0:6d:b9:5b:dd:27:50:3d:cc:3c:ee:de:5a:e8:6a:
a0:b8:21:8e:47:72:b0:a3:67:58:aa:17:55:0c:44:
eb:89:b1:6f:e0:74:b8:c7:70:30:82:9b:96:ab:a3:
43:9a:4f:a4:9c:56:3e:7f:a3:8e:63:5b:3d:d1:15:
89:2f
```

Exponent: 65537 (0x10001)

Signature Algorithm: sha256WithRSAEncryption

```
32:50:a6:75:f5:6f:1a:c4:a1:ea:77:51:fb:85:a4:e6:99:e9:
57:ed:4d:e8:38:4a:72:b5:49:8a:04:70:23:64:94:40:cb:b5:
a5:ab:26:9d:08:41:23:1e:6f:e3:6e:0b:65:a1:45:a9:70:51:
91:49:fd:3c:9a:bf:fd:88:84:e4:93:a6:b8:57:af:28:2e:9e:
41:46:d5:4d:eb:8c:90:7f:29:03:98:53:bf:f8:46:8c:db:3b:
ac:dd:f5:02:cb:c8:81:7f:45:ca:1b:25:d9:31:db:8a:ad:17:
64:c3:3f:63:c2:4b:60:f1:17:f6:78:a6:af:50:e6:a7:ff:f2:
33:af
```

Everything looks good.

You can now start/restart WebADM services and check the Rsignd service (PKI) is working correctly :

```
[root@webadm2 pki]# /opt/webadm/bin/webadm start
Checking libudev dependency... Ok
Checking system architecture... Ok
Checking server configurations... Ok

Found Trial license (RCDEVSSUPPORT)
Licensed by RCDevs Security SA to RCDevs Support
Licensed product(s): OpenOTP,SpanKey

Starting WebADM PKI service... Ok
Starting WebADM Session service... Ok
Starting WebADM Watchd service... Ok
Starting WebADM HTTP service... Ok

Checking server connections...
Connected LDAP server: AD1 (192.168.4.2)
Connected SQL server: SQL19 (192.168.4.4)
Connected PKI server: PKI Server (localhost)
Connected Mail server: SMTP Server (192.168.4.1)
Connected Session server: Session Server 1 (192.168.4.20)

Checking LDAP proxy user access... Ok
Checking SQL database access... Ok
Checking PKI service access... Ok
Checking Mail service access... Ok
Checking Cloud service access... Ok

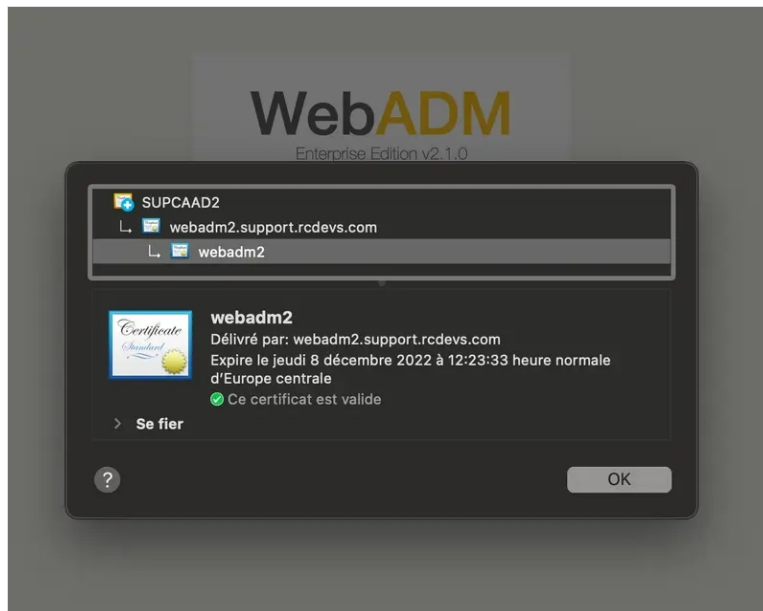
Cluster mode enabled with 2 nodes (I'm master)
```

In the previous output, you can see the 3 following lines which indicate everything is fine :

```
Starting WebADM PKI service... Ok
...
Connected PKI server: PKI Server (localhost)
...
Checking PKI service access... Ok
```

Everything is now configured correctly.

You can double-check the certificate used on your WebADM through your web browser :

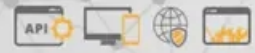


4. Issue clients, servers and users certificates

Now our WebADM is configured as a subordinate CA, we will check that WebADM can generate the different kinds of certificates which can be issued by WebADM/Rsignd.

4.1 Issue an OpenOTP or Spankey client certificate

Login on WebADM administrator portal, click **Admin** tab and then click on **Issue Server or Client SSL Certificate**.



WebADM Server Administration

WebADM v2.1.0 (64bit) running on server webadm2.support.rcdevs.com (192.168.4.21) in cluster mode (2 servers). Currently handling 1 connection(s).

Server Version Details: Apache/2.4.51 PHP/7.4.26 OpenSSL/1.1.1l
Internal Server Time: 2021-12-08 12:49:59 Europe/Berlin (NTP check Ok)
Hardware Modules: No HSM Connected
WebADM Features: WebApps (Enabled), WebSrvs (Enabled), Manager (Enabled)
RCDevs Cloud Services: BASE, LICENSE, PUSH, SMS, PROOF (Connected)
Active LDAP Server: AD1 (192.168.4.2) Active SQL Server: SQL19 (192.168.4.4)
Active Session Server: Session Server 1 (192.168.4.20) Active PKI Server: PKI Server (192.168.4.21)
Active Mail Server: SMTP Server (146.59.204.189)

- User Domains (3)**
Associate domain names with LDAP user search bases.
- Client Policies (22)**
Define custom policy settings for consumer applications.
- Access Devices (0)**
Hardware devices for badging and physical access control.
- LDAP Mount Points (3)**
Connect secondary LDAP servers to the tree view.
- LDAP Option Sets (2)**
Define LDAP tree constraints for your 'other' administrators.
- Administrator Roles (3)**
Create admin role templates for your 'other' administrators.

Licensing and Configurations

Runtime Actions

- Software License Details
- LDAP Server Details
- LDAP Server Schema
- Memory Usage Details
- Hardware Modules Details
- Remote Manager Interface
- Config Object Statuses
- Network Service Statuses
- WebADM Base Settings
- Download WebADM CA Certificate
- Download WebADM SSL Certificate
- Issue Server or Client SSL Certificate
- Clear Admin Session Cache (4 KB)
- Clear WebADM License Cache
- Clear WebADM System Caches (813 KB)
- Flush WebADM Session Data (2013 KB)
- Reload WebADM Configurations
- Send Test Alert Email

You arrive at the following page :

Create Third-party SSL Server Certificate

You can use this form to issue a X.509 SSL certificate and private key for a third-party server or component. The certificate is generated with the provided information and signed by WebADM certificate authority.

Auto Confirm Mode	
Enable Auto Confirm:	<input type="radio"/> Yes <input checked="" type="radio"/> No i
Auto Confirm Time:	5 Minutes v
Auto Confirm App:	[All] v
Auto Confirm IPs:	<input type="text"/> i
Main information	
Server Hostname (FQDN):	<input type="text"/>
Certificate Type:	Server v i
Certificate validity (in days):	365 i
Private Key Password (optional):	<input type="password"/> i
Additional information	
Alternative Name(s):	<input type="text"/> i
Organization Name:	<input type="text"/>
Organizational Unit:	<input type="text"/>
Country Name:	<input type="text"/> i
Locality Name:	<input type="text"/>
State or Province:	<input type="text"/>
Street Address:	<input type="text"/>
Email Address:	<input type="text"/>
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	

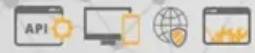
To generate a client certificate, change the `certificate type` to a `client` and provide the information of your client. Here, I restricted the certificate usage to OpenOTP service only :

Create Third-party SSL Server Certificate

You can use this form to issue a X.509 SSL certificate and private key for a third-party server or component. The certificate is generated with the provided information and signed by WebADM certificate authority.

Auto Confirm Mode	
Enable Auto Confirm:	<input type="radio"/> Yes <input checked="" type="radio"/> No ⓘ
Auto Confirm Time:	5 Minutes ▾
Auto Confirm App:	[All] ▾
Auto Confirm IPs:	<input type="text"/> ⓘ
Main information	
Client Name or Description:	<input type="text" value="OpenOTP_Client.support.rcdevs.com"/>
Certificate Type:	Client ▾ ⓘ
Restricted Application:	OpenOTP ▾ ⓘ
Certificate validity (in days):	<input type="text" value="365"/> ⓘ
Private Key Password (optional):	<input type="password"/> ⓘ
Additional information	
Organization Name:	<input type="text" value="RCDevs Support"/>
Organizational Unit:	<input type="text" value="IT"/>
Country Name:	LU ⓘ
Locality Name:	<input type="text" value="Belval"/>
State or Province:	<input type="text" value="Luxembourg"/>
Street Address:	<input type="text"/>
Email Address:	<input type="text"/>
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	

Once you completed all needed information, press **Ok** button to submit the request to Rsignd service and then the certificate and key for your OpenOTP client are generated:



WebADM Server Administration

WebADM v2.1.0 (64bit) running on server webadm2.support.rcdevs.com (192.168.4.21) in cluster mode (2 servers). Currently handling 1 connection(s).

Server Version Details: Apache/2.4.51 PHP/7.4.26 OpenSSL/1.1.1l
Internal Server Time: 2021-12-08 12:49:59 Europe/Berlin (NTP check Ok)
Hardware Modules: No HSM Connected
WebADM Features: WebApps (Enabled), WebSrvs (Enabled), Manager (Enabled)
RCDevs Cloud Services: BASE, LICENSE, PUSH, SMS, PROOF (Connected)
Active LDAP Server: AD1 (192.168.4.2) Active SQL Server: SQL19 (192.168.4.4)
Active Session Server: Session Server 1 (192.168.4.20) Active PKI Server: PKI Server (192.168.4.21)
Active Mail Server: SMTP Server (146.59.204.189)

- User Domains (3)**
Associate domain names with LDAP user search bases.
- Client Policies (22)**
Define custom policy settings for consumer applications.
- Access Devices (0)**
Hardware devices for badging and physical access control.
- LDAP Mount Points (3)**
Connect secondary LDAP servers to the tree view.
- LDAP Option Sets (2)**
Define LDAP tree constraints for your 'other' administrators.
- Administrator Roles (3)**
Create admin role templates for your 'other' administrators.

Licensing and Configurations

Runtime Actions

- Software License Details
- LDAP Server Details
- LDAP Server Schema
- Memory Usage Details
- Hardware Modules Details
- Remote Manager Interface
- Config Object Statuses
- Network Service Statuses
- WebADM Base Settings
- Download WebADM CA Certificate
- Download WebADM SSL Certificate
- Issue Server or Client SSL Certificate
- Clear Admin Session Cache (4 KB)
- Clear WebADM License Cache
- Clear WebADM System Caches (813 KB)
- Flush WebADM Session Data (2013 KB)
- Reload WebADM Configurations
- Send Test Alert Email

You arrive at the following page :

Create Third-party SSL Server Certificate

You can use this form to issue a X.509 SSL certificate and private key for a third-party server or component. The certificate is generated with the provided information and signed by WebADM certificate authority.

Auto Confirm Mode	
Enable Auto Confirm:	<input type="radio"/> Yes <input checked="" type="radio"/> No ?
Auto Confirm Time:	5 Minutes ▼
Auto Confirm App:	[All] ▼
Auto Confirm IPs:	<input type="text"/> ?
Main information	
Server Hostname (FQDN):	<input type="text"/>
Certificate Type:	Server ▼ ?
Certificate validity (in days):	365 ?
Private Key Password (optional):	<input type="password"/> ?
Additional information	
Alternative Name(s):	<input type="text"/> ?
Organization Name:	<input type="text"/>
Organizational Unit:	<input type="text"/>
Country Name:	<input type="text"/> ?
Locality Name:	<input type="text"/>
State or Province:	<input type="text"/>
Street Address:	<input type="text"/>
Email Address:	<input type="text"/>
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>	

To generate a server certificate, choose the `certificate type` to `Server` and provide the information of your server.

Create Third-party SSL Server Certificate

You can use this form to issue a X.509 SSL certificate and private key for a third-party server or component. The certificate is generated with the provided information and signed by WebADM certificate authority.

Auto Confirm Mode	
Enable Auto Confirm:	<input type="radio"/> Yes <input checked="" type="radio"/> No ?
Auto Confirm Time:	5 Minutes ▼
Auto Confirm App:	[All] ▼
Auto Confirm IPs:	<input type="text"/> ?
Main information	
Server Hostname (FQDN):	<input type="text" value="waproxy.support.rcdevs.com"/>
Certificate Type:	Server ▼ ?
Certificate validity (in days):	<input type="text" value="365"/> ?
Private Key Password (optional):	<input type="password"/> ?
Additional information	
Alternative Name(s):	<input type="text" value="waproxy"/> ?
Organization Name:	<input type="text" value="RCDevs Support"/>
Organizational Unit:	<input type="text" value="IT"/>
Country Name:	<input type="text" value="LU"/> ?
Locality Name:	<input type="text" value="Belval"/>
State or Province:	<input type="text" value="Luxembourg"/>
Street Address:	<input type="text"/>
Email Address:	<input type="text"/>

Ok

Cancel

Once you have completed all needed information, press the **Ok** button to submit the request to the Rsignd service and then the certificate and key for your OpenOTP client are generated:

Object CN=valery,OU=SUPAdmins,DC=support,DC=rcdevs,DC=com (Other Administrator) ⓘ

WARNING: User password will expire in 9 days!

LDAP Actions

- Delete this object
- Copy this object
- Move this object
- Export to LDIF
- Change password
- Create certificate
- Unlock WebApp access
- Advanced edit mode

Object Details

Object class(es): [webadmAccount, person, user](#)

Account is unique: **Yes** (in dc=support,dc=rcdevs,dc=c...)

WebADM settings: **1 settings** [CONFIGURE]

WebADM data: **10 data** [EDIT]

User activated: **Yes Deactivate** ⓘ

Logs and inventory: [WebApp](#), [WebSrv](#), [Inventory](#), [Record](#)

Application Actions

- [Secure Password Reset](#) (1 actions)
- [User Self-Registration](#) (1 actions)
- [MFA Authentication Server](#) (15 actions)
- [SMS Hub Server](#) (1 actions)
- [SSH Public Key Server](#) (3 actions)

Object Name: Rename

Add Attribute (16): Add

Add Extension (1): Add

Account Created [delete attribute]	01-07-2021
Account Modified [delete attribute]	08-12-2021
Object GUID [delete attribute]	30653132-6130-3533-3565-396665353463
Account Flags [delete attribute]	Normal Account Flags: <input type="text" value="Keep Unchanged"/>
Last Bad Logon Password [delete attribute]	2021-11-05 09:55:26
Last Logoff [delete attribute]	Never
Last Logon [delete attribute]	Never
Last Password Set [delete attribute]	2021-11-05 09:55:13
Object SID [delete attribute]	S-48-49-53004999864368-808464432-808464437-825569328-8084644...
Account Expires [delete attribute]	Never Reset
Logon Count [delete attribute]	<input type="text" value="0"/>
Login Name	<input type="text" value="valery"/> ⓘ
Account Type [delete attribute]	SAM User Account
Principal Name [delete attribute]	<input type="text" value="valery@support.rcdevs.com"/>

Once you are on the user account, in **LDAP Actions** box, click on **Create certificate** button, and you are then prompted for the following :

WebADM Enterprise Edition v2.1.0
 Copyright © 2010-2021 RCDevs Security, All Rights Reserved

Home | Admin | Cluster | Create | Search | Import | Databases | Statistics | Applications | About | Logout

New User Certificate Value(s) for CN=valery,OU=SUPAdmins,DC=support,DC=rcdevs,DC=com

Certificate Validity (in days):

Certificate Export Format: PKCS12

Email Address: yoann@rcdevs.com

Send Email Notification: Yes No

*Admin certificates are used to enter Admin Portal with PKI mode.
 User certificates are used to enter WebApps requiring certificates.*

Certificate Usage: Admin User

Alternative Name(s):

Login Name: valery

User Domain: SUPPORT

Create Cert Import Cert Cancel

Configure the **validity** for that certificate, the **certificate usage** and optionally the **alternative names**.

WebADM Enterprise Edition v2.1.0
 Copyright © 2010-2021 RCDevs Security, All Rights Reserved

Home | Admin | Cluster | Create | Search | Import | Databases | Statistics | Applications | About | Logout

New User Certificate Value(s) for CN=valery,OU=SUPAdmins,DC=support,DC=rcdevs,DC=com

Certificate Validity (in days): 365

Certificate Export Format: PKCS12

Email Address: yoann@rcdevs.com

Send Email Notification: Yes No

*Admin certificates are used to enter Admin Portal with PKI mode.
 User certificates are used to enter WebApps requiring certificates.*

Certificate Usage: Admin User

Alternative Name(s):

Login Name: valery

User Domain: SUPPORT

Create Cert Import Cert Cancel

The certificate usage can be configured to **Admin** or **User**. **Admin** certificates are allowed only for super_admin or other_admin users. **User** certificates are for all regular users.

Admin certificates allow you to log in on WebADM Admin GUI, not **user** certificate.

Click **Create Cert** and the certificate will be created.

WebADM Enterprise Edition v2.1.0
Copyright © 2010-2021 RCDevs Security, All Rights Reserved

Home | Admin | Cluster | Create | Search | Import | Databases | Statistics | Applications

New User Certificate for CN=valery,OU=SUPAdmins,DC=support,DC=rcdevs,DC=com

Creating private key... **Success**
Reading infos from LDAP user... **Success**

Certificate details:
- emailAddress: **yoann@rcdevs.com**
- description: **USER**
- commonName: **SUPPORT\valery**
- userId: **valery**
- domainComponent: **SUPPORT**

Creating a certificate request based on the above details... **Success**
Calling WebADM CA for certificate request signing... **Success**
Checking certificate data... **Success**
Storing certificate in LDAP... **Success**
Updating certificate cache... **Success**
Creating a PKCS12 package... **Success**

Certificate installation password: jMMmGGsC

The certificate and private key have been bundled into a PKCS12 package.
Click the button below to download the new certificate package.

[Download](#) [Ok](#)

Copy the password and download the certificate bundle. When you will try to import it into your certificate store, the password will be required.

Below, the details of that generated certificate :

SUPPORT\valery (USER)



SUPPORT\valery (USER)

Délivré par: webadm2.support.rcdevs.com

Expire le jeudi 8 décembre 2022 à 18:13:14 heure normale d'Europe centrale

• Ce certificat est marqué comme fiable pour ce compte

> Se fier

▼ Détails

Sujet	
Adresse e-mail	yoann@rcdevs.com
Description	USER
Nom	SUPPORT\valery
Identifiant de l'utilisateur	valery
Composante de domaine	SUPPORT
Nom de l'émetteur	
Pays ou région	LU
Région/Province	Luxembourg
Localité	Belval
Organisation	Support RCDevs
Unité d'organisation	IT
Nom	webadm2.support.rcdevs.com
Adresse e-mail	yoann@support.rcdevs.com
Numéro de série	61
Version	3
Algorithme de signature	SHA-256 avec chiffrement RSA (1.2.840.113549.1.1.1)
Paramètres	Aucun
Non valide avant	mercredi 8 décembre 2021 à 18:13:14 heure normale d'Europe centrale
Non valide après	jeudi 8 décembre 2022 à 18:13:14 heure normale d'Europe centrale
Infos de clé publique	
Algorithme	Chiffrement RSA (1.2.840.113549.1.1.1)
Paramètres	Aucun
Clé publique	256 octets : C7 0B 0C BF 90 49 E3 23 ...
Exposant	65537
Dimension de clé	2 048 bits
Utilisation de la clé	Vérifier, Ajuster
Signature	128 octets : 50 D5 A8 11 8D 6D 84 17 ...
Extension	Utilisation de la clé (2.5.29.15)
Critique	NON
Utilisation	Signature numérique, Non répudiation, Chiffrement de clé
Extension	Nom alternatif du sujet (2.5.29.17)
Critique	NON
Nom RFC 822	yoann@rcdevs.com
Empreintes	
SHA-256	EE 1E 72 BE 43 25 16 4F 6C 97 07 82 76 6E 94 F9 6E 12 62 1B F2 1D DE C1 9D 01 1B E6 F1 2F FF A2
SHA-1	D8 6A BA BB AD F2 73 BC B7 F8 63 AB 0B E9 C0 34 4F 28 1C 5F

This manual was prepared with great care. However, RCDevs Security S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alternation without notice. RCDevs Security S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs Security S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language without the prior written permission of RCDevs Security S.A. The latter especially applies for data processing systems. RCDevs Security S.A. also reserves all communication rights (lectures, radio and television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs Security. © 2024 RCDevs Security S.A., All Rights Reserved