**Overview:**

z/OS PKI Services provides a web interface for users to request and administer certificates. But many users would like to have a RACDCERT command like interface to request a certificate from PKI Services, especially when they need a certificate with other extensions that RACDCERT does not support, like extended keyusages, multiple subject alternate names… etc.

This Unix gencert command serves that purpose. You would still need to use the web interface if you want to use other functions.

After you have set up z/OS PKI Services and started the service, you can make use this gencert utility to generate certificates in two ways by supplying:

1) a Certificate Signed Request (CSR) in PKCS10 format OR

2) a key size and a key algorithm for PKI Services to generate the keypair

It has a wide set of parameters which build the certificate content like the subject distinguished name and extensions.

**Syntax overview:**

gencert -K <CSR file> [-c <keyword=value pairs without keysize>] -s <outfile> <other options>

OR

gencert -c rq=<emailAddr>,ks=<keysize>,ka=<keyAlg>,pp=<passPhase>,<other keyword=value pairs> [-s <outfile>] <other options>

Note: the input CSR and the output certificate can either be a Unix file or a z/OS dataset. If it is a dataset, specify like this “//’myserver.cert’”

**Usage:**

gencert {-K <file containing CSR in PKCS#10 format>

[-c rq=…,cn=…,ku=…,eku=…,dom=…,…] -s <output file> }

|

{-c rq=<emailAddr>,ks=<keySize>,ka=<keyAlg>,

pp=<passp>,cn=…,ku=…,eku=…,dom=…,… [-s <output file>]}

[-D <CA Domain>]

[-t <template nickname>]

[-v]

-c Certificate parameter sub options in the format of

keyword=value pair list separated by commas

(keyword with \* can be repeated for multiple values)

(value with blanks needed to be quoted)

Acceptable Keywords:

**Subject Distinguish name values:**

sn - SerialNumber e.g sn=2943780

ua - UnstructAddr e.g ua='Cisco 36xx Router'

ua - UnstructName e.g un='descriptive text'

ea - EmailAddr e.g ea=gumby@loony.toons

m - Mail e.g m=bugs@loony.toons

dq - DNQualifier e.g dq='domain qualifier'

uid - Uid e.g uid=mega

cn - CommonName e.g cn='Foghorn Leghorn'

t - Title e.g t='The Hammer'

dn - DomainName e.g dn=www.loony.toons

ou\* - OrgUnit e.g ou='My Org Unit'

bc - BusinessCat e.g bc=’Health Organization’

o - Org e.g o=’The Bros Org’

jl - JurLocality e.g jl=Poughkeepsie

js - JurStateProv e.g js=’New York’

jc - JurCountry e.g jc=US

st - Street e.g st='2455 Cartoon Lane'

l - Locality e.g l=’Dutchess County’

sp - StateProv e.g sp='New York'

pc - PostalCode e.g pc=12601

c - Country e.g c=US

**Certificate extension values:**

ip\* - AltIPAddr e.g ip=27.26.25.24,ip=27.26.25.23

uri\*- AltURI e.g uri=http://www.pokey.com

ae\* - AltEmail e.g ae=fredflint@bedrock.com

dom\*- AltDomain e.g dom=www.sys1.com,dom=www.sys2.com

ao - AltOther e.g ao='1.2.3.4;acb123xyz'

ku\* - KeyUsage e.g ku=handshake

xku\*- ExtKeyUsage e.g xku=ocspsigning

ce\* - CustomExt e.g ce='1.2.3.4,n,int,1234'

cp - CertPolicy e.g cp='1 3 5'

aia\*- AuthInfoAcc e.g aia='ocsp;URL:http://

<host[:port]>/PKIServ

/public-cgi/caocsp'

crit\*- Critical e.g crit=KeyUsage

him\* - HostIdMap e.g him=pokey@gumbymvs

Other parameter values:

pp - PassPhrase e.g pp=secretPassPhrase

nb - NotBefore e.g nb=1

na - NotAfter e.g na=365

ne - NotifyEmail e.g ne=hammer@loony.toons

usr - Userid e.g usr=joeuser

rq - Requestor e.g rq='My Friendly Name' or

rq=joeuser@aabank.com,

if -K is not specified

**Other options:**

-D <CA Domain> The 1-8 char CA Domain name

-t template nickname

The nickname of the Template to use, as indicated in the PKI template file like pkiserv.tmpl

-v Verbose mode - extra info displayed

**Examples:**

1. Provide a PKCS10(CSR) for certificate generation:

gencert -K myserver.csr -D subca1 -s myserver.cer

2. Let PKI Services generate the key pair for the certificate

gencert -c rq=joeuser@aabank.com,ks=2048,ka=rsa,

cn=myserver,o=test,c=us,

ku=handshake,xku=clientauth,ip=1.2.3.4,ip=5.6.7.8,

dom=myserver1.com,dom=myserver2.com,pp=secret

-D subca1 -s myserver.p12

**Steps to create this Unix gencert utility:**

1) Copy the content of gencert.c

2) Copy the makefile

3) Issue the command: make gencert

4) Issue the command: chmod 755 gencert

**Use scenarios:**

A) Provide a Certificate Signed Request (CSR) to request a certificate

1. RACDCERT ID(<serverID>) GENCERT SUBJECT(cn(‘<common name>’)…) WITHLABEL(‘MyServer’)

2. RACDCERT ID(<serverID>) GENREQ(LABEL(‘MyServer’)) DSN(<output dataset contains the PKCS10 CSR>)

3. Issue the gencert utility and specify a dataset for output like “//’mycert.cer’”

a. if you want to keep the original subject name and keyusge in the CSR:

gencert -K <dataset from step 2> -D <CA domain> -s <cert output dataset>

b. if you want new values for subject name and keyusage: (don’t specify keysize(ks) in -c)

gencert -K <file from step 3> -c <cn=…,eku=…,dom=…,…> -D <CA domain> -s <cert output dataset>

4. Optional, if you want to list the content of the returned cert package

RACDCERT CHECKCERT(<dataset from step 3>)

5. RACDCERT ID(<serverID>) ADD(<dataset from step 3>)

B) Let PKI Services to create the key pair

1. Issue the gencert utility

a. if you want to save the cert in a dataset, specify the dataset like “//’mycert.p12’”

gencert -c rq=admin@abc.com,ka=rsa,ks=2048,cn=...,pp=secret... -D <CA domain> -s <cert output dataset>

b. if you don’t want to save a copy of the cert, omit -s

gencert -c rq=admin@abc.com,ka=rsa,ks=2048,cn=...,pp=secret... -D <CA domain>

2. Install the cert in RACF

a. if you have the output dataset

if you want to list the content of the returned cert package, use RACDCERT CHECKCERT specifying the same password entered for the gencert utility.

RACDCERT CHECKCERT(<dataset from step 1>) password(‘secret’)

RACDCERT ID(<serverID>) ADD(<dataset contains the cert>) password(‘secret’) WITHLABEL(‘MyServer’)

b. if you did not have the output dataset, you need to find out the sequence number of the cert saved in the issuing CA’s token

RACDCERT LISTTOKEN(<token name used by the CA>)

(as indicated in TokenName in pkiserv.conf)

RACDCERT ID(<serverID>) IMPORT( TOKEN(<token name>) SEQNUM(<sequence num>) ) WITHLABEL(‘MyServer’)

**References:**

R\_PKIServ GENCERT parameter lists (-c is based on):

https://www.ibm.com/docs/en/zos/2.5.0?topic=rpiirpkips-parameters

RACDCERT functions:

https://www.ibm.com/docs/en/zos/2.5.0?topic=syntax-racdcert-manage-racf-digital-certificates