PKIServerCommands (and ICSF List)

PKISERVD has a web interface for creating and managing certificate. There is also an API R_PKIServ (IRRSPX00). There is no official command interface.

The PKIServd process for generating a certificate is one of

- create a certificate - with no approvals
- create a certificate request. This requires one or more approvals. Once the required numberof approvers have approved, then the cerificate request is converted into a certificate.

Certificate requests.

There are commands that apply to a certificate request

- list requests
- approve, reject, delete-
- display details

Certificate

There are commands that apply to certificates. When the certificate it created, only then does it update ICSF, and so be visible in ICSF.

You can

- change status, revoke, delete, resume
- list certificates
- display certificate details

Batch commands for PKIServer on z/OS

The programs in this package call R_PKIServ to allow you to issue commands in batch. You can use the Web based interface with Apache Tomcat server.

ICSF List

PKIServer uses a token in ICSF. There is no easy way to get a list of tokens, and certificates, so I wrote a program to extract them.

```
//COLINL  JOB 'COLIN',CLASS=A,REGION=0M,COND=(4,LE)
//RUN      EXEC PGM=TLIST,REGION=0M,PARMDD=MYPARMS
//STEPLIB  DD DISP=SHR,DSN=COLIN.LOAD
//MYPARMS DD *
 -detail 3
 -token PKISRVD.PKITOKEN
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=300)
//SYSOUT   DD SYSOUT=*
//SYSERR   DD SYSOUT=*
```

Parameters:

- -token value it only process records with the give token value. If -token is not specified, it processes the whole of ICSF visible to the user.
- -detail

    - 0 just the tokens
    - 1 the certificates available to the user
    - 2 the certificates + private key, _ public key + data
    - 3 all the fields
    - 4 all values - including hexadecimal
- -label value. This is the value of the certificate, for example colinmay30@gmail.com. The values like 1+bBworxjlmN2TbK6o++++++ are for the private, public and data records. There can be many certificates with the same label value.
- -serial value. This is a hexadecimal value like -serial 3E

It produces output like (one object has multiple parts, private key, certificate, public key).

```
List PKISRVD.PKITOKEN                 Sequence:000000AB Type:T
Token:PKISRVD.PKITOKEN                .
   Sequence:000000AB
   ID:T:Clear token object.
      CKA_CLASS:CKO_CERTIFICATE
      CKA_TOKEN:TRUE.
      CKA_TRUSTED:TRUE.
      CKA_CERTIFICATE_TYPE:CKC_X_509
      CKA_SUBJECT(68):CN=ColinMay 30,OU=SSSDOC,O=SSS,C=gb.
      CKA_ID(20): 38F1A8CC DED47043 F08FDA39 EB2324EA 0E3C210E
      CKA_ISSUER(55):CN=PKICA,OU=SSS,O=YOUR2COMPANY.
      CKA_SERIAL_NUMBER:0x3A.
      CKA_LABEL(32):colinmay30@gmail.com.
      CKA_APPLICATION(0):.

List PKISRVD.PKITOKEN                 Sequence:000000A9 Type:T
Token:PKISRVD.PKITOKEN                .
   Sequence:000000A9
   ID:T:Clear token object.
      CKA_CLASS:CKO_PUBLIC_KEY
      CKA_TOKEN:TRUE.
      CKA_TRUSTED:FALSE.
      CKA_START_DATE:.
      CKA_END_DATE:.
      CKA_SUBJECT(0):no data
      CKA_ID(20): 38F1A8CC DED47043 F08FDA39 EB2324EA 0E3C210E
      CKA_LABEL(32):1+bBworxjlmN2TbK6o++++++        .
      CKA_APPLICATION(32):PKISERVICES                     .

List PKISRVD.PKITOKEN                 Sequence:000000AA Type:T
Token:PKISRVD.PKITOKEN                .
   Sequence:000000AA
   ID:T:Clear token object.
      CKA_CLASS:CKO_PRIVATE_KEY
      CKA_TOKEN:TRUE.
      CKA_START_DATE:.
      CKA_END_DATE:.
      CKA_SUBJECT(0):no data
      CKA_ID(20): 38F1A8CC DED47043 F08FDA39 EB2324EA 0E3C210E
      CKA_LABEL(32):1+bBworxjlmN2TbK6o++++++        .
      CKA_APPLICATION(32):PKISERVICES                     .

List PKISRVD.PKITOKEN                 Sequence:000000AC Type:T
Token:PKISRVD.PKITOKEN                .
```

```
Sequence:000000AC
ID:T:Clear token object.
   CKA_CLASS:CKO_DATA
   CKA_TOKEN:TRUE.
   CKA_LABEL(32):000000000000000000000000000003a.
   CKA_APPLICATION(17):Z/OS PKI SERVICES.
   CKA_ID(20): 38F1A8CC DED47043 F08FDA39 EB2324EA 0E3C210E
```

Generate certificate |Generate requests

If you generate a certificate it is automatically approved. If you generate a requests then you need one or more approvers before the certificate is created.

Syntax:

It needs the '/' to separate the parameters from the run time options. A parameter in column 1 is appended to the previous line. The parameters should start in column two, so they are not concatenated.

```
//ISTEST   EXEC PGM=CGEN,REGION=0M,PARMDD=MYPARMS
//MYPARMS DD *,SYMBOLS=(EXECSYS)
 /
 -detail 0
 -debug 0
 -req
 -log "COLI ZZZ"     #comment this is a comment before log
 -log "&LOG"
 -ae colin@gmail.com
 -aipa 9.20.4.6
 -auri colin.sss.com
 -nb 2
 -cn "ColinTESTX6"
 -c gb
 -pp passphrase2
 -ks 521
 -ka NISTECC
 -ou SSSYYY
 -c gb
 -r colinSeptX@gmail.com
 -ku docsign
 -ku handshake
 -sw PKI:
//OTHER DD *
//STEPLIB  DD DISP=SHR,DSN=COLIN.LOAD
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=200)
//SYSOUT   DD SYSOUT=*
//SYSERR   DD SYSOUT=*
```

The parameters are as in R_PKISERV The parameters which control the program are

```
-req |-cert
-sw PKI:
-sw SAF:CERTAUTH/keyring.  I could not get this to work.
```

```
-debug value defaults to 0.  It prints out internal control block information.
-detail value defaults to 0. Range 0 to 3, which displays detailed data in hexa
-ca value  the CA_DOMAIN value
-log value
```

In the table below you can specify a value like -sn, or -SerialNumber, and the maximum length is 64 characters. The keywords and values are case sensitive

```
 -sn         ,SerialNumber,64
 -nb         ,NotBefore,2
 -na         ,NotAfter,4
 -ea         ,EmailAddr,64
 -ua         ,UnstructAddr ,64
 -un         ,UnstructName ,64
 -ea         ,EmailAddr ,64
 -e          ,Email       ,64
 -dnq        ,DNQualifier ,64
 -u          ,Uid        ,64
 -cn         ,CommonName ,64
 -t          ,Title        ,64
 -dn         ,DomainName ,64
 -ou         ,OrgUnit      ,64
 -o          ,Org          ,64
 -s          ,Street       ,64
 -l          ,Locality     ,64
 -sp         ,StateProv    ,64
 -pc         ,PostalCode ,64
-c          ,Country      , 2
-ku         ,KeyUsage    , 20
-eku        ,ExtKeyUsage , 20
-nb         ,NotBefore    , 2
-na         ,NotAfter     , 4
-aipa       ,AltIPAddr     , 45
-auri       ,AltURI        ,255
-ae         ,AltEmail     ,100
-ad         ,AltDomain    ,100
-ao         ,AltOther     ,255
-ne         ,NotifyEmail , 45
-pk         ,PublicKey    ,65535
-ks         ,KeySize      ,  4
-ka         ,KeyAlg       , 10
-sw         ,SignWith     , 45
-him        ,HostIdMap     ,100
-r          ,Requestor    , 32
-pp         ,PassPhrase , 32
-u          ,Userid       ,  8
-l          ,Label        , 32
-cp         ,CertPolicies , 32
-aic        ,AuthInfoAcc ,255
-critical   ,Critical      , 32
-ce         ,CustomExt ,1024
-bc         ,BusinessCat , 64
-jc         ,JurCountry , 2
-jsp        ,JurStateProv, 64
```

```
-jl         ,JurLocality, 64
```

Some parameters such as -Business are accepted, but the ICSF formatting does not display them properly.

Certificate

When a certificate is generated, the program displays

```
RACDCERT IMPORT(TOKEN(PKISRVD.PKITOKEN                ) -
  SEQNUM(0000034B)) -
  ID(...) WITHLABEL('...')
```

Which you can use to import the certificate into a RACF, and so connect it to a keyring.

List the PKIServer server contents

For example

```
//ISTEST   EXEC PGM=CLIST,REGION=0M,
//   PARM=' -status pending                                   '
//STEPLIB  DD DISP=SHR,DSN=COLIN.LOAD
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=200)
//SYSOUT   DD SYSOUT=*
//SYSERR   DD SYSOUT=*
```

Where the parameters can be

- -status value where value can be one of all, pending, approved, completed, rejected, rejected,preregistered
- -ca name which CA_DOMAIN to use
- -days nnnn Value indicating the recent activity time period to use as additional search criteria. The time period is the number of days in the past that should be scanned for requests that have been created or modified. If zero (x'00000000'), recent activity will not be used as additional search criteria.
- -expiry yy/mm/dd report certificates that expire before this date
- -name value requestor's name, as specified in the -r or -Requestor parameter
- -certid value such as -certid 1+bBwH5fli5l2TbK6o++++++

It produces output like

```
   1 CertId         : 1+bB4R1dzuOP2TbK6o++++++
   1 Requestors name: colinSeptX@gmail.com
   1 Subject DN     : CN=ColinTESTX6,OU=SSSYYY,O=SSS,C=gb
   1 Issuer  DN     : CN=PKICA,OU=SSS,O=YOUR2COMPANY
   1 Validity       : 2023/09/20 00:00:00 - 2023/12/26 23:59:59
   1 Key usage      : docsign digitalsig
   1 Status         : Approved
   1 Created        : 2023/09/18
   1 Modified       : 2023/09/18
   1 ApplData       : LOG LOG
   1 Serial         : E0
   1 Prev Serial    :
   1 ExtKeyUsage    : not specified
   1 QueryTime      : 2023/09/18 10:47:52
   1 No approvers

   2 CertId          : 1+bB4R24WmsS2TbK6o++++++
   2 Requestors name: colinSeptX@gmail.com
   2 Subject DN      : CN=ColinTESTX6,OU=SSSYYY,O=SSS,C=gb
   2 Issuer  DN      : CN=PKICA,OU=SSS,O=YOUR2COMPANY
   2 Validity        : 2023/09/20 00:00:00 - 2023/12/26 23:59:59
   2 Key usage       : docsign digitalsig
   2 Status          : Approved
   2 Created         : 2023/09/18
   2 Modified        : 2023/09/18
   2 ApplData        : LOG LOG
   2 Serial          : E1
   2 Prev Serial     :
   2 ExtKeyUsage     : not specified
   2 QueryTime       : 2023/09/18 10:47:52
   2   Approver Userid: COLIN
   2   Approver Action: approved
   2   Approver Time  : 2023/09/18 08:07:02
```

The first column is a sequence number to help you manage the output. For example you might sort on created data, extract records with a range of values, and then be able to identify the certificate from the CertId.

You could use PARM=' -status pending ' to see which certificates need some action.

Approve|Reject|Delete|request

You can use

```
//ISTEST   EXEC PGM=CAPPROVE,REGION=0M,PARMDD=MYPARMS
//MYPARMS DD *
 /
 -certid 1+bB4R24WmsS2TbK6o++++++
 -action approve
 -comment "zz3znerapprove"
/*
```

Where - -action is approve|delete|reject

With action approve, you can override parameters as for gencert for example -enddate 2025/12/31

Request details

For example

```
//ISTEST   EXEC PGM=CREQDETA,REGION=0M,PARMDD=MYPARMS
//MYPARMS DD *
/
 -cert 1+bByQYzvoda2TbK6o++++++
//STEPLIB  DD DISP=SHR,DSN=COLIN.LOAD
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=200)
//SYSOUT   DD SYSOUT=*
```

```
//SYSERR   DD SYSOUT=*
```

This produces output like

```
====REQDETAILS====
request ID        : 1+bB4T2+jsca2TbK6o++++++
Requestors name   : colinSeptX@gmail.com
Subject DN        : CN=ColinTESTX6,OU=SSSYYY,BUSINESSCATEGORY=Colins business c
Issuer  DN        : CN=PKICA,OU=SSS,O=YOUR2COMPANY
Validity          : 2023/09/20 00:00:00 - 2023/12/26 23:59:59
Key usage         : docsign digitalsig
Status            : Approved
Created           : 2023/09/18
Modified          : 2023/09/18
ApplData          : LOG LOG
Serial            : E2
Prev Serial       :
Last action       :
PassPhrase        : passphrase2
Email             : colinSeptX@gmail.com
ExtKeyUsage       : not specified
FingerprintSHA1   :
FingerprintSHA256 :
Signature         :
KeyType           : NISTECC
KeySize           : 521
QueryTimestamp    : 2023/09/18 11:28:12
No approvers
CommonName  :   ColinTESTX6
OrgUnit     :   SSSYYY
BusinessCat :   Colins business cat
Org         :   SSS
Country     :   gb
KeyUsage    :   DIGITALSIG
KeyUsage    :   DOCSIGN
AltEmail    :   colin@gmail.com
AltURI      :   colin.sss.com
AltIPAddr   :   9.20.4.6
AutoRenew   :
StartDate   :   2023/09/20
EndDate     :   2023/12/26
```

Query certificates

For example, list all certificates

```
//COLQUER JOB 'COLIN',CLASS=A,REGION=0M,COND=(4,LE)
//JOBLIB JCLLIB ORDER=(COLIN.PKIICSF.C,CBC.SCCNPRC) fere
//*
//ISTEST   EXEC PGM=CQUERY,REGION=0M,
//   PARM='          '
//*  PARM=' -key A2 '
//STEPLIB  DD DISP=SHR,DSN=COLIN.LOAD
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=200)
```

```
//SYSOUT   DD SYSOUT=*
//SYSERR   DD SYSOUT=*
```

You can specify

- -key value
- -name value
- -status value. Where value can be one of all, revoked, expired, active, CRL, crl, spspended, auto, noauto, email . Where CRL|crl is return non-expired revoked or suspended certificates only
- -days value. Where value is negativ report the certificates expiring in the next value days. When the value is positive, report the certificates that were created or modified in the time period.

Certificate details using serial number

Each issued certificate has a serial number. You can display information about it using

```
//CGEN     EXEC  CCPROC,PROG=CDETAIL
//ISTEST   EXEC PGM=CDETAIL,REGION=0M,
//   PARM=' -serial E2               '
//STEPLIB  DD DISP=SHR,DSN=COLIN.LOAD
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=200)
//SYSOUT   DD SYSOUT=*
//SYSERR   DD SYSOUT=*
```

Modify a certificate

I see this as more of a change status of an issued certificate.

```
//ISTEST   EXEC PGM=CMODIFYC,REGION=0M,PARMDD=MYPARMS
//MYPARMS DD *
 /

 -serial E2
 -action revoke
 -reason none
//STEPLIB  DD DISP=SHR,DSN=COLIN.LOAD
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=200)
//SYSOUT   DD SYSOUT=*
//SYSERR   DD SYSOUT=*
```

Where -action is one of revoke, delete, resume, disable, enable, crl. -reason is one of none, userkey, cakey, CAkey, userchanged, super, superseded, invalid, suspend.