z/OS



Cryptographic Services PKI Services Guide and Reference

SA22-7693-13

Note
Before using this information and the product it supports, be sure to read the general information under Notices.

This edition applies to Version 1, Release 13, modification 0 of IBM z/OS (product number 5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SA22-7693-12.

© Copyright IBM Corporation 2001, 2011.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Tables ix

Figures xi

About this document xiii
Who should use this document xiii
How to use this document xiii
Where to find more information xvi
Softcopy publications xvi
RACF courses xvi
Other sources of information xvi
IBM discussion area xvii
Internet sources xvii
To request copies of IBM publications xviii

How to send your comments to IBM xxi
If you have a technical problem xxi

Summary of changes xxiii
Changes made in z/OS Version 1 Release 13 . . xxiii
Changes made in z/OS Version 1 Release 12 . . . xxv
Changes made in z/OS Version 1 Release 11 xxviii

Part 1. Planning 1

Chapter 1. Introducing PKI Services . . 3
What is PKI Services? 3
What is a certificate authority? 3
What is PKI? 4
Basic components of PKI Services and related products 4
Component diagram 5
Supported standards 7
Supported certificate types 7
Supported certificate fields and extensions 8

Chapter 2. Planning your implementation 11
Installing PKI Services 11
Requirements for sysplex support 11
Determining prerequisite products 12
z/OS HTTP Server 12
Websphere Application Server (optional) . . . 12
LDAP directory server 12
OCSF (optional) 13
ICSF (optional) 13
sendmail (optional) 13
OCEP (optional) 13
DB2 (optional) 13
Identifying skill requirements 13
Team members 14
Skills for setting up prerequisite products . . . 14
Skills for setting up PKI Services 16
Creating an implementation plan 17
Task roadmap for implementing PKI Services . . 17

Chapter 3. Installing and configuring prerequisite products 19
Tasks to perform before setting up PKI Services . . 19
Installing and configuring the z/OS HTTP Server 19
Installing and configuring Websphere Application Server for z/OS 21
Installing and configuring LDAP 21
Installing and configuring ICSF (optional) . . . 23
Configuring sendmail (optional) 24
Installing and configuring DB2 25
Tasks to perform before configuring PKITP 25
Installing and configuring OCSF 25
Installing and configuring OCEP 26

Part 2. Configuring your system for PKI Services 27

Chapter 4. Running IKYSETUP to perform RACF administration 29
Overview of IKYSETUP 30
Before you begin 30
Variables whose values must change 31
Variables whose values might change depending on setup 34
Variables you can optionally change 42
Steps for performing RACF tasks using IKYSETUP 47
Sample IKYSETUP log data set 50

Chapter 5. Configuring the UNIX runtime environment. 53
Steps for copying files 55
Optionally updating PKI Services environment variables 57
(Optional) Steps for updating PKI Services environment variables 58
Optionally updating the pkiserv.conf configuration file 58
(Optional) Steps for updating the configuration file 60
Updating pkiserv.conf after installing a new release of z/OS 83
Steps for setting up the var directory 84

Chapter 6. Tailoring the LDAP configuration for PKI Services 87
Steps for loading schema.user.ldif 87
Setting up authorization to create and access CRLs and certificates 88
Establishing a secure connection with LDAP (optional) 88

© Copyright IBM Corp. 2001, 2011

Chapter 7. Updating z/OS HTTP Server configuration and starting the server. . 91

Steps for updating the z/OS HTTP Server configuration files 91
Steps for starting the z/OS HTTP Server 94

Chapter 8. Tailoring the PKI Services configuration file for LDAP 97

Excerpt of LDAP section 97
Storing information for encrypted passwords for your LDAP servers 97
Steps for tailoring the LDAP section of the configuration file 98

Chapter 9. Creating the object store and ICL 103

The object store and ICL 103
Creating the object store and ICL using VSAM data sets. 104
Planning VSAM storage requirements 104
(Optional) preliminary steps for establishing VSAM RLS 105
Steps for creating the VSAM object store and ICL data sets and indexes 106
(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS . . . 107
Tuning VSAM performance. 108
Backing up and restoring the VSAM data sets 109
Creating the object store and ICL using DB2 tables 110
Sysplex considerations 111
Planning DB2 storage requirements 111
Steps for creating the object store and ICL DB2 tables 111
Converting the object store and ICL from VSAM to DB2. 113
Columns in the ICL and object store DB2 tables 114

Chapter 10. Starting and stopping PKI Services 117

Steps for starting the PKI Services daemon . . . 117
Stopping the PKI Services daemon 119

Part 3. Customizing PKI Services 121

Chapter 11. Customizing the end-user Web application if you use REXX CGI execs 123

Contents of the pkiserv.tmpl certificates templates file 123
What are substitution variables? 124
What are named fields? 125
INSERT sections 125
The APPLICATION sections 133
Templates that PKI Services provides 134
TEMPLATE sections 136
Summary of fields in certificate templates 143
Examining the pkiserv.tmpl file 148
Examining the APPLICATION section 149
Examining the TEMPLATE section 153
Examining the INSERT section 156
Relationship between CGIs and the pkiserv.tmpl file 158
Steps for performing minimal customization . . . 160
Steps for additional first-time customization . . . 161
Steps for retrofitting release changes into the PKI Services certificate templates 165
Locating code for customizing end-user Web pages 165
Steps for adding a new certificate template . . . 167
Changing the runtime user ID. 168
Steps for changing the runtime user ID for requesting certificates. 168
Steps for changing the runtime user ID for retrieving certificates 169
Customizing the OtherName field 170
Steps for customizing the sample AltOther_<*OID*> INSERTs 171

Chapter 12. Customizing the administration Web pages if you use REXX CGI execs 175

CGIs for administration Web pages 175
Customizing the administration Web pages . . . 177
Steps for customizing the administration Web pages 177
Changing the runtime behavior for accessing administration pages 178
Steps for changing control of access to administration pages 178

Chapter 13. Implementing the Web application using Java server pages . 179

Certificate templates files used with JSPs 179
Examining the pkitmpl.xml file 180
Roadmap for implementing the PKI Services Web application using JSPs 184
Steps for preparing to implement the PKI Services Web application using JSPs 185
Giving Websphere users authorization to use PKI Services functions 185
Allowing Websphere users to renew and revoke browser certificates 189
Customizing the PKI Services Web application 198
Directories for JSP files 204

Chapter 14. Advanced customization 211

Scaling for high volume installations 211
Using certificate policies. 212
Steps for creating the CertificatePolicies extension on a global basis 213
Steps for creating the CertificatePolicies extension on a template basis 214
Updating the signature algorithm 215
Steps for changing the signature algorithm . . 216
Customizing distribution point CRLs 217
Specifying the URI format 218
Determining CRLDistURI*n* 218
Determining CRLDistDirPath 219
Steps for customizing distribution point CRLs 220

How distribution point CRLs work 222
Enabling support for large CRLs 223
Steps for enabling support for large CRLs . . . 224
Using the OCSP responder 224
Creating a distribution point ARL 225
Adding an application domain 226
Creating application domains when you use REXX CGIs to implement the Web application . 227
Creating application domains when you use JSPs to implement the Web application 229
Adding a new CA domain 230
Task overview 231
Task roadmap for adding CA domains 232
Recording your progress adding CA domains 233
Subtask 1: Steps for planning additional CA domains 234
Subtask 2: Steps for reconfiguring your initial CA domain to allow it to coexist with other CA domains 236
Subtask 3: Steps for running the IKYSETUP exec 238
Subtask 4: Steps for configuring the UNIX environment. 240
Subtask 5: Steps for updating the PKI Services template file. 241
Subtask 6: Steps for updating the Web server configuration 243
Subtask 7: Creating the object store and ICL . . 244
Subtask 8: Steps for starting PKI Services . . . 245
Enabling Simple Certificate Enrollment Protocol (SCEP) 246
Overview of SCEP preregistration 246
Overview of certificate request processing for preregistered SCEP clients 247
Checking certificate fingerprints 249
Steps for enabling Simple Certificate Enrollment Protocol (SCEP) 250
Customizing e-mail notifications sent to users . . 252
Steps for customizing e-mail notification forms 257
Setting up automatic renewal of certificates . . . 258
Steps for setting up automatic certificate renewal 258
Setting up PKI Services to generate keys for certificate requests. 259
Steps for setting up PKI Services to generate keys for certificate requests. 259
Adding custom extensions to certificates 261
Steps for adding a custom extension to a certificate template if you are using REXX CGI execs 262
Steps for adding a custom extension to a certificate template if you are using JSPs . . . 262
Forming the CustomExt value for CertPlist for the R_PKIServ callable service. 263

Chapter 15. Customizing with installation exit routines. 267
Exit routine processing for automatic certificate renewal 267
Steps for updating the exit routine code sample 268
Using the exit routine for pre- and post-processing. 268
Scenario for using the exit routine 272
Exit routine processing for the PKI Services CGIs 272
Steps for updating the exit routine code sample 273
Using the exit routine for pre- and post-processing. 273
Scenarios for using the exit routine 285
Exit routine processing for Java server pages (JSPs) 287
Class UserExit 288
Class ExportCert 299
Class QRecover. 299
Class RevokeCert 299
Class UserExitException. 299
Class CertPlist 300
Class PkiCertificate 301
Class QrecoverResultsList 302
Class RpkiservException. 302

Part 4. Using PKI Services 305

Chapter 16. Using the end-user Web pages. 307
Steps for accessing the end-user Web pages . . . 308
Summary of fields. 311
Steps for requesting a new certificate 314
Retrieving your certificate 321
Steps for retrieving your certificate from the bookmarked Web page 321
Steps for retrieving your certificate from the PKI Services home page 322
Steps for retrieving a PKI generated key certificate. 323
Steps for renewing a certificate 324
Steps for revoking or suspending a certificate . . 327
Recovering a certificate whose keys were generated by PKI Services. 328
Steps for recovering a certificate whose keys were generated by PKI Services 328
Steps for preregistering an SCEP client 331

Chapter 17. Using the administration Web pages 335
Steps for accessing the administration home page 335
Fields in the administration Web pages. 338
Processing certificate requests 338
Status of certificate requests 338
Actions on certificate requests 339
Using the PKI Services administration home page 339
Processing certificates 348
Status of certificates 348
Actions for certificates 349
Steps for processing a single certificate 350
Steps for processing certificates by performing searches 351
Relationship between certificate requests and matching certificates 355

Chapter 18. Using PKI Services utilities 357

Using the createcrls utility 358
Using the iclview utility 359
Using the pkiprereg utility 363
Using the postcerts utility 368
Using the TemplateTool utility 370
Using the vosview utility 372
Using the vsam2db2 utility 377

Chapter 19. Using the certificate management protocol (CMP) with PKI Services 379
Support for CMP messages 380
Support for the CMP certificate request message (type cr) 381
Support for the CMP PKCS #10 certificate request message (type p10cr) 383
Support for the CMP certificate response message (type cp) 383
Support for the CMP revocation request message (type rr) 384
Support for the CMP revocation response message (type rp) 384
Support for the CMP error message (type error) 385
Determining the CA domain to which a request is routed 385
How PKI Services interprets distinguished names (DNs) on CMP requests 387
Setting up a client to make CMP requests to PKI Services 388
Steps for setting up a certificate for a CMP requester 388
Setting up PKI Services to process CMP requests 389
Enabling the CMP support 389
Setting up PKI Services to create private keys for CMP clients 389
Setting up the HTTP Server for CMP 392
Tracing the PKI CMP CGI program 397
Messages and codes returned from the CMP functions 397

Part 5. Administering security for PKI Services 401

Chapter 20. RACF administration for PKI Services 403
Authorizing users for the PKI Services administration group 403
Connecting members to the group 403
Deleting members from groups 403
Authorizing users for inquiry access 403
Steps for authorizing users for inquiry access 404
Administering HostIdMappings extensions . . . 404
Steps for administering HostIdMappings extensions 405
Locating your PKI Services certificates and key ring 406
Steps for locating the PKI Services certificates and key ring 406
Establishing PKI Services as an intermediate CA 408
Steps for changing PKI Services from a self-signed CA to an intermediate CA 409
Renewing your PKI Services CA and RA certificates 410
Steps for renewing your PKI Services CA certificate 411
Steps for renewing your PKI Services RA certificate 412
Recovering a CA certificate profile 413
Steps for recovering a CA certificate profile . . 413
Retiring and replacing the PKI Services CA private key 414
Steps to retire and replace the PKI Services CA private key for the PKI templates 414
Steps to retire and replace the PKI Services CA private key for the SAF templates: Scenario 1. . 415
Steps to retire and replace the PKI Services CA private key for the SAF templates: Scenario 2. . 416
R_PKIServ (IRRSPX00) callable service 417
Authorizing end-user functions 417
Authorizing administrative functions 419
Using encrypted passwords for LDAP servers . . 420
Steps for using encrypted passwords 421

Part 6. Using the certificate validation service 423

Chapter 21. PKI Services Trust Policy (PKITP) 425
Overview of PKITP 425
Certificate policies 427
Checking certificate status with PKITP 427
Certificate extensions 428
CRL extensions and CRL entry extensions 429
Files for PKITP 429
Configuring and getting started with PKITP . . . 429
Steps for configuring PKITP 430
Trust Policy API 430
CSSM_TP_PassThrough 431
Building the sample application to invoke the certificate validation service 434

Part 7. Troubleshooting 447

Chapter 22. Using information from SYS1.LOGREC 449
Sample LOGREC data 452

Chapter 23. Using information from the PKI Services logs 455
Viewing SYSOUT information 455
_PKISERV_MSG_LEVEL subcomponents and message levels 459
Changing logging options 459
Displaying log options settings 460

Part 8. Reference information . . . 461

Chapter 24. Messages. 463

Chapter 25. File directory structure 497
Product libraries 497
File system directory and subdirectories 497

Chapter 26. The pkiserv.conf configuration file 501

Chapter 27. Environment variables 507
Environment variables in the environment variables file. 507
The pkiserv.envars environment variables file . . 510

Chapter 28. The IKYSETUP REXX exec 511
Actions IKYSETUP performs by issuing RACF commands 511
Setting up the PKI Services daemon user ID . . 511
Setting up access control to protect PKI Services 511
Establishing your CA and RA certificates . . . 513
Configuring the z/OS HTTP Server for SSL mode 515
Using RACF to obtain a certificate for the Web server 516
Enabling the z/OS HTTP Server for surrogate operation. 516
Allowing PKI Services to generate key pairs for certificate requests. 517
IKYSETUP sample. 517

Chapter 29. Other code samples . . . 545
z/OS HTTP Server configuration directives . . . 545
IKYCDB2. 548
IKYCVSAM 551
IKYRVSAM 555
IKYSBIND 559
IKYVBKUP 560
IKYVREST 561
PKISERVD sample procedure to start PKI Services daemon 562

Chapter 30. SMF recording 565
PKI Services event code 565
Relocate section variable data 565

LDAP directory server requirements 567

Using a gskkyman key database for your certificate store 571
Steps for using a gskkyman key database for your certificate store 571

Configuring PKI Services as an IdenTrust certificate authority 573
Who should use this appendix 573
Related information from IdenTrust 573
Overview of configuring z/OS PKI Services as a CA 573
System prerequisites 574
Task overview 574
Configuring z/OS PKI Services as a CA 575
Steps to modify pkiserv.conf for different certificate types. 576
Steps to modify pkiserv.conf general settings 577
Steps to create IdenTrust specific certificate templates. 577
Code samples 578
Sample PKI Services configuration file directives for IdenTrust compliance 579
Sample browser certificate template for IdenTrust compliance. 579
Sample server certificate template for IdenTrust compliance 581

Using the PKI Services Web application with Internet Explorer on Windows systems 585
User tasks for setting up a Windows system and Internet Explorer to work with the PKI Services Web application 585
Installing CAPICOM on a Microsoft Windows system 586
Installing the PKI Services ActiveX program . . 587
Configuring Internet Explorer to trust PKI Services on a Windows system 591
Installing the PKI Services CA certificate on a Microsoft Windows system. 592
Administrator tasks for setting up a Windows system and Internet Explorer to work with the PKI Services Web application 593
Signing the PKI Services ActiveX programs . . 594

Accessibility. 601
Using assistive technologies 601
Keyboard navigation of the user interface 601
z/OS information 601

Notices 603
Policy for unsupported hardware. 604
Programming interface information 605
Trademarks 605

Bibliography. 607

Index 609

Tables

1. Basic components of PKI Services and related products 4
2. Types of certificates you can request 7
3. File system directory variables 11
4. Tasks and skills needed for installing prerequisite products 14
5. Roles, tasks, and skills for setting up PKI Services 16
6. Task roadmap for implementing PKI Services 17
7. z/OS HTTP Server information you need to record 21
8. LDAP information you need to record . . . 22
9. IKYSETUP—Structure and divisions 30
10. IKYSETUP variables whose values must change 32
11. Decision table for key_backup 35
12. Decision table for key_type 36
13. Decision table for restrict_surrog 37
14. Decision table for unix_sec 38
15. Decision table for db2_repos and db2_subsys 38
16. IKYSETUP variables you might want to change depending on setup 39
17. IKYSETUP variables you can optionally change 43
18. Deciding which files to copy and change 53
19. Information needed for updating the configuration file. 60
20. LDAP information you need for tailoring LDAP configuration. 87
21. Summary of configuration and usage of each Web server instance 91
22. LDAP information you need for tailoring z/OS HTTP Server configuration 92
23. Information needed for updating the LDAP section of the configuration file 98
24. VSAM RLS information you need to record 106
25. Columns in the object store DB2 tables 114
26. Columns in the ICL DB2 tables 114
27. Structure and main divisions of the certificate template file (pkiserv.tmpl) 123
28. Substitution variables 124
29. INSERTs that are common HTML for Web page content 126
30. Named fields in INSERT sections 127
31. Subsections of the APPLICATION sections 133
32. Certificate templates PKI Services provides 135
33. Names, aliases, and nicknames of certificate templates 137
34. Summary of subsections in certificate templates 143
35. Summary of fields for PKI browser certificate templates 144
36. Summary of fields for PKI server certificate templates 145
37. Summary of fields for SAF, SCEP, and PKI generated key certificate templates 147
38. CGI actions for end-user Web pages 158
39. Location of code for various Web pages 165
40. CGI actions for administrative Web pages 175
41. Task roadmap for implementing the PKI Services Web application using the JSPs. . . 184
42. JSP files in the `Customers` directory 205
43. JSP files in the `PKIServ` directory 206
44. JSP files in the `mod.inc` directory 207
45. Supported signature algorithms for each CA certificate key type. 216
46. Task roadmap for creating multiple application domains 227
47. Task roadmap for adding a new CA domain 233
48. Multiple CA domains: Worksheet #1 for recording progress adding new CA domains . 234
49. Multiple CA domains: Worksheet #2 for planning your domain names 235
50. Multiple CA domains: Worksheet #3 for planning your RACF identifiers, z/OS UNIX identifiers, and VSAM data set names or DB2 package names 236
51. IKYSETUP log information you need to record 239
52. Information you need to enable Simple Certificate Enrollment Protocol (SCEP) . . . 250
53. Descriptions of variables for forms 255
54. Summary of substitution variables in forms 256
55. Files for the `pkiexit.c` exit routine 267
56. Values of arguments for pre- and post-processing 269
57. Values of arguments for pre- and post-processing 273
58. Package and class summary for JSP exit processing 288
59. Methods in class UserExit 288
60. Types of certificates you can request 307
61. Summary of fields in end-user Web pages 311
62. Summary of fields in the administration pages 338
63. Statuses of certificate requests 338
64. Summary of actions to perform on requests and required status 339
65. Searches to display certificate requests 344
66. Status of certificates 349
67. Summary of actions to perform and required status to do so 349
68. Searches to display certificates 351
69. Names, aliases, and nicknames of certificate templates 361
70. List of valid field names for use in the preregistration record as input to the pkiprereg utility 365
71. Names, aliases, and nicknames of certificate templates 375
72. Format of tcp-messages 379
73. Supported fields in the `PKIMessage` structure 380

© Copyright IBM Corp. 2001, 2011

74. Supported fields in the PKIHeader structure 381
75. Supported values in the PKIBody structure. These are the CMP message types that PKI Services supports. 381
76. Supported fields in the CMP certificate request message (type cr) 381
77. Supported fields in the CMP PKCS #10 certificate request message (type p10cr) . . . 383
78. Support for fields in the CMP certificate response message (type cp) 383
79. Supported fields in the CMP revocation request message (type rr) 384
80. Supported fields in the CMP revocation response message (type rp) 384
81. Supported fields in the CMP error message (type error) 385
82. HTTP Server environment variables used to determine the CA 392
83. HTTP Server environment variables used to control the content of the certificate within a CA 393
84. HTTP Server environment variables used to configure the certificate recipients 396
85. HTTP Server environment variables used to control tracing 396
86. CMP error codes 397
87. Information you need for locating your PKI Services certificates and key ring 406
88. Information you need for establishing PKI Services as an intermediate CA 409
89. Information you need for renewing your PKI Services certificate authority certificate . . . 411
90. Information you need for renewing your PKI Services RA certificate. 412
91. Information you need for recovering a CA certificate profile 413
92. Summary of access authorities required for PKI Services requests 419
93. Sequence of validation stages for PKITP certificate revocation checking 428
94. Summary of information about important files for PKITP 429
95. PKI Services OCSF Trust Policy (PKITP) error codes 433
96. LOGREC data for PKI Services. 449
97. Summary of information about important files 461
98. Meaning of fourth character in message number 463
99. Meaning of eighth character in message number 463
100. Files contained in subdirectories 497
101. Access required if you plan to have an administrator approve certificate requests . . 513
102. Access required if you plan to use auto-approval 513
103. FACILITY class access needed for protecting administrative functions 513
104. Access PKISRVD needs to use RACF certificates 515
105. SMF event code and event code qualifier for PKI Services 565
106. SMF data elements of the extended-length relocate section for PKI Services 565
107. LDAP objectclasses and attributes that PKI Services sets 567
108. Relationship of named fields to LDAP attributes and object identifiers 568
109. Tasks to perform to set up a Windows system and the Internet Explorer browser to work with PKI Services 585
110. Directory structure for a sample project named PKIXEnrollDeploy created in the C: directory 596

Figures

1. Component diagram of a typical PKI Services system 6
2. Flowchart of the process of updating IKYSETUP 48
3. Partial listing of the AltOther_1_2_3_4_6 sample INSERT showing the lines you are most likely to customize 173
4. A `certreq_template` tag 181
5. Configuring Websphere for application security 186
6. Configuring Websphere for application security using SAF authorization 186
7. Renewing or revoking a browser certificate 189
8. Websphere SSL certificate and key management page 190
9. Websphere page for creating a new key store 191
10. Viewing the signer certificate 191
11. Websphere new SSL configuration page 192
12. Websphere quality of protection settings page 193
13. Defining a new port 194
14. Assigning a port to the application server 194
15. Assigning your new port to the application server 195
16. Selecting Web container transport chains 196
17. Selecting a transport chain template 196
18. All existing transport chains 197
19. Setting the SSL inbound channel properties 197
20. A portion of the JSP file `notbefore.jsp`, without customization 198
21. A portion of the JSP file `notbefore.jsp`, with customization 199
22. The Websphere Shared Libraries window 201
23. Message indicating that changes have been made to your local configuration. 201
24. Window for specifying the EAR file 202
25. Application properties page. 203
26. Shared library mapping for modules page 203
27. Adding the PKI Services shared library's location to the server's library path 204
28. A sample CRLDistributionPoints extension for a certificate authority (CA) certificate . . . 226
29. Illustration of two CA domains, one for employees and one for customers, administered by a single shared administrator who administers both domains 231
30. Sample of readymsg.form 253
31. Sample of rejectmsg.form 253
32. Sample of expiringmsg.form 254
33. Sample of renewcertmsg.form 254
34. Sample of pendingmsg.form 254
35. Sample of recoverymsg.form 255
36. PKI Services end-user home page for certificate generation 309
37. The certificate window for installing the CA certificate 310
38. One-year SSL browser certificate request form 316
39. Supplying the PKCS #10 certificate request for a server or device certificate 318
40. Successful request displays transaction ID 319
41. Web page to retrieve your certificate 320
42. Successful request for a one-year PKI generated key certificate 320
43. Browser certificate installation Web page 321
44. Server certificate installation Web page 322
45. E-mail notification that your PKI generated key certificate is ready for pickup 323
46. Web page for retrieving a PKI generated key certificate 324
47. Window asking whether to open or save the PKCS #12 package 324
48. Popup window listing certificates 325
49. Renew or revoke a certificate Web page 326
50. Web page requesting answers to security questions when you have forgotten the passphrase 329
51. Web page listing certificates that can be recovered 329
52. Web page showing the passphrase for a certificate to be recovered 329
53. Sample e-mail that lists certificates that can be recovered 330
54. Web page to retrieve a recovered certificate 330
55. Window asking whether to open or save the PKCS #12 package 331
56. SCEP preregistration request form 332
57. Successful preregistration request 332
58. PKI Services administration start page 336
59. The certificate popup window for installing the CA certificate 336
60. Entering your user ID and password 337
61. PKI Services administration home page 340
62. Single request approval Web page 341
63. Processing successful Web page 342
64. Restriction note on the modify and approve request Web page 342
65. Modifying the request Web page 343
66. Processing requests after searching 345
67. Request processing was successful Web page 347
68. Request processing was not successful Web page 347
69. Request processing was partially successful Web page 348
70. Processing a certificate from the single certificate Web page 350
71. Processing certificates using searches 352
72. Processing of certificate was successful Web page 354
73. Request processing was not successful Web page 355
74. Request processing was partially successful Web page 355

© Copyright IBM Corp. 2001, 2011

75. Examples of organizations, certificates, and chains 426
76. Sample LOGREC data 453
77. Separating the job files 456
78. Selecting a file to view 457
79. Messages contained in the file 458
80. File download window 588
81. Security warning window 588
82. ActiveX control setup wizard 589
83. Message window 590
84. Confirmation window 590
85. CAPICOM security alert window 590

About this document

This document supports z/OS® (5694-A01). This document contains information about planning, customizing, administering, and using the PKI Services component of the z/OS Cryptographic Services.

PKI Services provides a certificate authority for the z/OS environment and enables you to issue and administer digital certificates, so that you do not have to purchase them from an external certificate authority. This document provides you with the information you need to become productive with PKI Services. It discusses the following topics:

- Procedures for setting up PKI Services on the z/OS platform.
- Using the PKI Services administration and user Web pages, you can easily issue digital certificates to trusted parties and control whether or not a certificate is renewed or revoked.
- Guidelines to help you plan for PKI Services, such as how to integrate PKI Services components with other products installed at your site.

Who should use this document

This document should be used by those who plan, install, customize, administer, and use PKI Services. It should also be used by those who install, configure, or provide support in the following areas:

- Lightweight Directory Access Protocol (LDAP)
- Resource Access Control Facility (RACF)
- z/OS
- z/OS HTTP Server
- z/OS UNIX System Services
- (optional) z/OS HTTP Server
- (optional) Integrated Cryptographic Service Facility (ICSF)
- (optional) Open Cryptographic Enhanced Plug-ins (OCEP)
- (optional) Open Cryptographic Services Facility (OCSF)
- (optional) z/OS Communications Server's sendmail utility
- (optional) Websphere Application Server

This document assumes that you have experience with installing and configuring products in a network environment. You should be knowledgeable about the following concepts and protocols:

- Hardware installation and configuration
- Internet communications protocols, in particular Transmission Control Protocol/Internet Protocol (TCP/IP) and Secure Sockets Layer (SSL)
- Public key infrastructure (PKI) technology, including directory schemas, the X.509 version 3 standard, and the Lightweight Directory Access Protocol (LDAP)

How to use this document

This document contains several parts:

- Part 1, “Planning,” on page 1 includes the following topics:

© Copyright IBM Corp. 2001, 2011

  - Chapter 1, “Introducing PKI Services,” on page 3 introduces PKI Services, describing its basic components and related products. It also describes supported standards, certificate types, fields and extensions.
  - Chapter 2, “Planning your implementation,” on page 11 provides a planning overview for your implementation. It discusses the components that work with PKI Services and the team members you will need to implement PKI Services and the skills they will need.
  - Chapter 3, “Installing and configuring prerequisite products,” on page 19 describes installing and configuring related products: the z/OS HTTP Server, OCSF, LDAP, and optionally ICSF.
- Part 2, “Configuring your system for PKI Services,” on page 27 describes the tasks your team members need to perform to configure PKI Services.
  - Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 29 describes how the RACF® administrator updates and runs IKYSETUP, a REXX exec to perform RACF administration tasks, such as setting up the daemon user ID and giving accesses.
  - Chapter 5, “Configuring the UNIX runtime environment,” on page 53 explains UNIX programmer tasks including how to copy files, update environment variables, update the PKI Services configuration file, and set up the `/var/pkiserv` file system directory.
  - Chapter 6, “Tailoring the LDAP configuration for PKI Services,” on page 87 explains how the LDAP programmer updates LDAP configuration for PKI Services.
  - Chapter 7, “Updating z/OS HTTP Server configuration and starting the server,” on page 91 explains how the Web server programmer updates the z/OS HTTP Server configuration files and starts the z/OS HTTP Server.
  - Chapter 8, “Tailoring the PKI Services configuration file for LDAP,” on page 97 explains how the UNIX programmer updates the **LDAP** section of the PKI Services configuration file.
  - Chapter 9, “Creating the object store and ICL,” on page 103 explains how the MVS™ programmer creates VSAM data sets or the DB2® database administrator creates DB2 objects for the object store and issued certificate list (ICL).
  - Chapter 10, “Starting and stopping PKI Services,” on page 117 explains how the MVS programmer starts and stops the PKI Services daemon.
- Part 3, “Customizing PKI Services,” on page 121 explains how to customize end-user and administration Web pages and perform advanced customization, including customization using an exit routine.
  - Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” on page 123 provides an overview of the `pkiserv.tmpl` file, which contains the certificate templates, and explains how to customize the end-user Web pages.
  - Chapter 12, “Customizing the administration Web pages if you use REXX CGI execs,” on page 175 provides an overview of the CGI scripts and explains how to customize the administration Web pages.
  - Chapter 14, “Advanced customization,” on page 211 explains how to use certificate policies, update the signature algorithm, enable automatic renewal of certificates, set up PKI Services to generate keys for certificates, and add custom extensions to certificates.
  - Chapter 15, “Customizing with installation exit routines,” on page 267 explains how to use the PKI Services exit routine.

- Part 4, “Using PKI Services,” on page 305 explains using the end-user and administration Web pages, using the PKI Services utilities, and using the certificate management protocol (CMP) with PKI Services.

  Chapter 16, “Using the end-user Web pages,” on page 307 shows the end-user Web pages and explains how to request a certificate, obtain the certificate, and renew or revoke a certificate.

  Chapter 17, “Using the administration Web pages,” on page 335 shows the administration Web pages and explains how to process certificate requests and certificates.

  Chapter 18, “Using PKI Services utilities,” on page 357 explains using the PKI Services utilities.

  Chapter 19, “Using the certificate management protocol (CMP) with PKI Services,” on page 379 describes the support for CMP that PKI Services provides.
- Part 5, “Administering security for PKI Services,” on page 401 explains how to perform many RACF administration tasks needed for PKI Services, such as authorizing users, administering extensions, and locating your PKI Services certificate and key ring.
- Part 6, “Using the certificate validation service,” on page 423 describes how to setup and use the PKI Services Trust Policy (PKITP) plug-in for OCSF.
- Part 7, “Troubleshooting,” on page 447 explains using logs:
  - Chapter 22, “Using information from SYS1.LOGREC,” on page 449 describes SYS1.LOGREC — which is used to record unusual runtime events, such as an exception.
  - Chapter 23, “Using information from the PKI Services logs,” on page 455 discusses using the PKI Services logs to debug problems and explains how to change logging options and display log options settings.
- Part 8, “Reference information,” on page 461 provides reference information including messages and important code samples.
  - Chapter 24, “Messages,” on page 463 explains PKI Services messages.
  - Chapter 25, “File directory structure,” on page 497 describes product and file system directories for PKI Services and files contained in them.
  - Chapter 26, “The pkiserv.conf configuration file,” on page 501 provides a code sample of the `pkiserv.conf` configuration file.
  - Chapter 27, “Environment variables,” on page 507 explains the `pkiserv.envars` environment variables file and provides a code sample.
  - Chapter 28, “The IKYSETUP REXX exec,” on page 511 explains the contents of the IKYSETUP REXX exec that performs RACF administration and provides a code sample.
  - Chapter 29, “Other code samples,” on page 545 provides additional code samples.
  - Chapter 30, “SMF recording,” on page 565 describes the SMF record that PKI Services produces.
- There are several appendixes, including:
  - “LDAP directory server requirements” on page 567 explains using a non-z/OS LDAP server.
  - “Using a gskkyman key database for your certificate store” on page 571 explains an alternative method for setting up your key database.
  - “Configuring PKI Services as an IdenTrust™ certificate authority” on page 573 explains configuring PKI Services to operate as an IdenTrust compliant certificate authority (CA).

– "Using the PKI Services Web application with Internet Explorer on Windows systems" on page 585 describes how you might need to set up a Windows system to work with PKI Services.

Where to find more information

Where necessary, this document references information in other documents. For complete titles and order numbers for all elements of z/OS, see *z/OS Information Roadmap*.

Softcopy publications

The PKI Services and RACF libraries are available on the following DVD softcopy collection in both BookManager® and Portable Document Format (PDF) files. The collection includes Softcopy Reader, which is a program that enables you to view the BookManager files. You can view or print the PDF files with an Adobe reader.

SK3T-4271 *z/OS V1R13 and Software Products DVD Collection*

This collection contains the documents for z/OS Version 1 Release 13 and the libraries for multiple releases of more than 400 z/OS-related software products, on DVDs.

The following CD softcopy collection includes books related to RACF and PKI Services:

SK3T-7876 *IBM System z Redbooks Collection*

This softcopy collection contains a set of documents called IBM® Redbooks® that pertain to System z® subject areas ranging from e-business application development and enablement to hardware, networking, Linux, solutions, security, Parallel Sysplex® and many others.

RACF courses

The following RACF classroom courses are available in the United States:

H3917 *Basics of z/OS RACF Administration*

H3927 *Effective RACF Administration*

ES885 *Exploiting the Advanced Features of RACF*

ES840 *Implementing RACF Security for CICS®*

IBM provides a variety of educational offerings for RACF. For more information about classroom courses and other offerings, do any of the following:

- See your IBM representative
- Call 1-800-IBM-TEACh (1-800-426-8322)

Other sources of information

IBM provides customer-accessible discussion areas where PKI Services and RACF can be discussed by customer and IBM participants. Other information is also available through the Internet.

IBM discussion area

IBM provides the *ibm.servers.mvs.racf* newsgroup for discussion of PKI Services and RACF-related topics. You can find this newsgroup on news (NNTP) server *news.software.ibm.com* using your favorite news reader client.

Internet sources

The following resources are available through the Internet to provide additional information about PKI Services, RACF, and many other security-related topics:

- **Online library**

  To view and print online versions of the z/OS publications, use this address:

  `http://www.ibm.com/systems/z/os/zos/bkserv/`

- **Redbooks**

  The Redbooks that are produced by the International Technical Support Organization (ITSO) are available at the following address:

  `http://www.ibm.com/redbooks`

- **Enterprise systems security**

  For more information about security on the zSeries® platform and z/OS, use this address:

  `http://www.ibm.com/systems/z/advantages/security/`

- **PKI Services home page**

  You can visit the PKI Services home page on the World Wide Web using the following address. Check this site for updates regarding PKI Services.

  `http://www.ibm.com/systems/z/os/zos/features/pki/`

- **RACF home page**

  You can visit the RACF home page on the World Wide Web using the following address.

  `http://www.ibm.com/systems/z/os/zos/features/racf/`

- **RACF-L discussion list**

  Customers and IBM participants can also discuss RACF on the RACF-L discussion list. RACF-L is not operated or sponsored by IBM; it is run by the University of Georgia.

  To subscribe to the RACF-L discussion and receive postings, send a note to:

  `listserv@listserv.uga.edu`

  Include the following line in the body of the note, substituting your first name and last name as indicated:

  `subscribe racf-l` *`first_name last_name`*

  To post a question or response to RACF-L, send a note, including an appropriate `Subject:` line, to:

  `racf-l@listserv.uga.edu`

- **RACF sample code**

  You can get sample code, internally developed tools, and exits to help you use RACF. This code works in our environment, at the time we make it available, but is not officially supported. Each tool or sample has a README file that describes the tool or sample and any restrictions on its use.

  To access this code from a Web browser, go to the RACF home page and select the "Downloads" topic from the navigation bar, or go to http://www.ibm.com/systems/z/os/zos/features/racf/goodies.html.

  The code is also available from `ftp.software.ibm.com` through anonymous FTP. To get access:

1. Log in as user **anonymous**.
2. Change the directory, as follows, to find the subdirectories that contain the sample code or tool you want to download:

   ```
   cd eserver/zseries/zos/racf/
   ```

An announcement will be posted on RACF-L discussion list and on newsgroup *ibm.servers.mvs.racf* whenever something is added.

Note: Some Web browsers and some FTP clients (especially those using a graphical interface) might have problems using `ftp.software.ibm.com` because of inconsistencies in the way they implement the FTP protocols. If you have problems, you can try the following:

- Try to get access by using a Web browser and the links from the RACF home page.
- Use a different FTP client. If necessary, use a client that is based on command line interfaces instead of graphical interfaces.
- If your FTP client has configuration parameters for the type of remote system, configure it as UNIX instead of MVS.

Restrictions

Because the sample code and tools are not officially supported,

- There are no guaranteed enhancements.
- No APARs can be accepted.

The z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a Web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your Web browser to the following Web site, which is available to all users (no login required):
http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp

To request copies of IBM publications

Direct your request for copies of any IBM publication to your IBM representative or to the IBM branch office serving your locality.

There is also a toll-free customer support number (1-800-879-2755) available Monday through Friday from 8:30 a.m. through 5:00 p.m. Eastern Time. You can use this number to:

- Order or inquire about IBM publications
- Resolve any software manufacturing or delivery concerns

- Activate the program reorder form to provide faster and more convenient ordering of software updates

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

1. Send an email to mhvrcfs@us.ibm.com
2. Visit the Contact z/OS web page at http://www.ibm.com/systems/z/os/zos/webqs.html
3. Mail the comments to the following address:
   IBM Corporation
   Attention: MHVRCFS Reader Comments
   Department H6MA, Building 707
   2455 South Road
   Poughkeepsie, NY 12601-5400
   U.S.A.
4. Fax the comments to us as follows:
   From the United States and Canada: 1+845+432-9405
   From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:
  z/OS V1R13.0 Cryptographic Services PKI Services Guide and Reference
  SA22-7693-13
- The topic and page number related to your comment
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

If you have a technical problem

Do not use the feedback methods listed above. Instead, do one of the following:

- Contact your IBM service representative
- Call IBM technical support
- Visit the IBM support portal at http://www.ibm.com/systems/z/support/

© Copyright IBM Corp. 2001, 2011

Summary of changes

This document contains terminology, maintenance, and editorial changes to improve consistency and retrievability. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Changes made in z/OS Version 1 Release 13

This document contains information previously presented in *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693-12, which supports z/OS Version 1 Release 12.

New information:

- "Requirements for sysplex support" on page 11 contains new information about sysplex requirements when you implement the object store and issued certificate list (ICL) using DB2 tables.
- "Variables whose values might change depending on setup" on page 34 describes new variables for IKYSETUP:
  - `db2_repos`, which specifies whether you will use VSAM files or DB2 tables for the object store and ICL
  - `db2_subsys`, which specifies the name of the local DB2 subsystem for the object store and ICL
- "Variables you can optionally change" on page 42 describes new variables for IKYSETUP:
  - `ca_exyears`, which specifies the life span of the PKI Services CA certificate
  - `web_exyears`, which specifies the life span of the Web server certificate
- "(Optional) Steps for updating the configuration file" on page 60 documents new keywords added to the configuration file, `EnableLargeCRLPosting` and `LargeCRLPostPath`, that allow you to enable support for CRLs larger than 32KB.
- "(Optional) Steps for updating the configuration file" on page 60 documents new keywords added to the configuration file, `DBType`, `DBPackage`, and `DBSubsystem`, that allow you to use DB2 tables instead of VSAM files for the object store and issued certificate list (ICL).
- "(Optional) Steps for updating the configuration file" on page 60 documents a new keyword added to the configuration file, `CRLIDPExt`, that controls whether the Issuing Distribution Point (IDP) extension is created with CRLs.
- Chapter 9, "Creating the object store and ICL," on page 103 has been retitled and contains new information about implementing the object store and issued certificate list (ICL) using DB2 tables.
- "Named fields in INSERT sections" on page 127 describes a new named field for INSERT sections, InstallCert, for installing an automatically-renewed certificate for a user who is using the Internet Explorer browser.
- "Named fields in INSERT sections" on page 127 describes a new named field for INSERT sections, PublicKye2IE, for selecting the smart card cryptographic service provider for a user who is using the Internet Explorer browser.
- "The APPLICATION sections" on page 133 describes a new subsection for APPPLICATION sections, RENEWEDCERT, for installing an automatically-renewed certificate for a user who is using the Internet Explorer browser.

© Copyright IBM Corp. 2001, 2011

- In “TEMPLATE sections” on page 136, the two-year PKI Windows logon certificate now supports requests from Mozilla-based browsers as well as from Internet Explorer.
- “Examining the CUSTOMERS application” on page 150 shows new function added to the Customers application to allow the new PKI Services ActiveX controls to be installed.
- “Relationship between CGIs and the pkiserv.tmpl file” on page 158 describes a new CGI exec, `installcert.rexx`, that displays a Web page in the Internet Explorer browser to install a renewed certificate.
- “Examining the pkitmpl.xml file” on page 180 describes additions to the PKI Services XML template to provide the URLs of the new PKI Services ActiveX controls.
- Table 42 on page 205 describes a new file, `installcert.jsp`, that contains code to install an automatically-renewed certificate for a user who is using the Internet Explorer browser.
- A new topic “Enabling support for large CRLs” on page 223 describes how to enable support for CRLs larger than 32KB.
- “Adding a new CA domain” on page 230 contains new information about creating the object store and ICL for a new CA domain using DB2 tables.
- “Using the vosview utility” on page 372 describes a new state flag displayed by the vosview utility to indicate that a certificate request is synchronous, and new parameters to specify the DB2 subsystem and package for the object store.
- “Using the iclview utility” on page 359 documents new parameters to specify the DB2 subsystem and package for the issued certificate list (ICL).
- “Using the vsam2db2 utility” on page 377 describes a new utility that copies data from the issued certificate list (ICL) and object store VSAM data sets into DB2 tables.
- In “Setting up the HTTP Server for CMP” on page 392, the _PKISERV_CMP_AUTHINFOACC_*domain* environment variable for CMP has been deprecated and replaced by _PKISERV_CMP_AUTHINFOACC*n_domain*, which allows multiple values to be specified.
- In “Messages and codes returned from the CMP functions” on page 397, a new error code 06 has been added.
- New messages:
  - IKYC077I
  - IKYC078I
  - IKYC079I
  - IKYC080I
  - IKYC081I
  - IKYC082I
  - IKYC083I
  - IKYC084I
  - IKYD002I
  - IKYD003I
  - IKYU018I
- “File system directory and subdirectories” on page 497 describes a new directory, `ActiveX`, which contains the unsigned PKI Services ActiveX files.
- In Chapter 26, “The pkiserv.conf configuration file,” on page 501, the `pkiserv.conf` sample includes new keywords.
- In “IKYSETUP sample” on page 517, the IKYSETUP sample allows ECC keys generated by the Crypto Express® 3 cryptographic coprocessor to be specified for the CA's private key. It also allows you to specify that you are using DB2 as the

repository for the object store and issued certificate list (ICL), and if you are it grants the PKI Services daemon access to the DB2 Resource Recovery Services Access Facility (RRSAF). It provides two methods to set the expiration dates for the CA certificate and Web server:
  – Setting them to a specified number of years in the future
  – Setting them to specific dates
- "z/OS HTTP Server configuration directives" on page 545 contains additional HTTP Server directives to support the new PKI Services ActiveX controls.
- "IKYCDB2" on page 548 and "IKYSBIND" on page 559 are new code samples to create DB2 objects for the object store and ICL, and BIND the PKI Services DBRM into a package and plan.
- "Using the PKI Services Web application with Internet Explorer on Windows systems" on page 585 contains information about new ActiveX programs that PKI Services provides to perform functions that CAPICOM has provided.

Changed information:

- In "Variables you can optionally change" on page 42, the default value of the `ca_expires` variable for IKYSETUP has changed to 2030/01/01. The default value of the `web_expires` variable has changed to 2015/01/01.
- In "(Optional) Steps for updating the configuration file" on page 60, the `SharedVSAM` parameter has been renamed to `SharedPLEX`, and applies to both VSAM and DB2 repositories for the object store and issued certificate list (ICL).
- In "Customizing e-mail notifications sent to users" on page 252, the sample notice sent to a user when a certificate is renewed (`renewcertmsg.form`) has been updated.
- Changed messages:
  – IKYP030I
- In "IKYCVSAM" on page 551, the CISZ values on DEFINE CLUSTER statements have been changed from 1024 to 4096 to minimize contention.

Changes made in z/OS Version 1 Release 12

This document contains information previously presented in *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693-11, which supports z/OS Version 1 Release 11.

New information:

- In Chapter 1, "Introducing PKI Services," the component diagram has been updated to show the CMP CGI and the use of Java server pages (JSPs) for the Web application.
- Chapter 4, "Running IKYSETUP to perform RACF administration" documents two new values for the `key_type` variable to specify generation of ECC private keys for the CA. Both the NIST and Brainpool (BP) algorithms are supported.
- Chapter 5, "Configuring the UNIX runtime environment" describes new keywords added to the configuration file, `MaintRunTime`, `MaintRunDays`, and `RunMaintAtStart`, that allow you to specify when the daily maintenance task runs.
- Chapter 5, "Configuring the UNIX runtime environment" describes a new keyword added to the configuration file, `EnableCMP`, that enables support for Certificate Management Protocol (CMP)

- Chapter 5, “Configuring the UNIX runtime environment” describes a new keyword added to the configuration file, `CertValidityConstraint`, that specifies whether the validity period of a certificate should be constrained within the CA's certificate life time.
- Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” describes the risk associated with using the one-year PKI generated key certificate template.
- Chapter 13, “Implementing the Web application using Java server pages” includes additional steps in the instructions for deploying the EAR file to a Websphere application server.
- In Chapter 13, “Implementing the Web application using Java server pages,” Table 44 on page 207 includes a new JSP file for defining custom certificate extensions.
- In Chapter 14, “Advanced customization,” “Updating the signature algorithm” on page 215 documents new support for 5 signing algorithms for ECC keys, and support for 3 additional SHA algorithms for RSA keys.
- In Chapter 14, “Advanced customization,” the sample rejectmsg.form now includes the reason for the rejection of the certificate request.
- In Chapter 14, “Advanced customization,” a new variable %%rejectreason%% is described, which contains the reason for the rejection of a certificate request.
- In Chapter 14, “Advanced customization,” “Adding custom extensions to certificates” on page 261 describes how to create custom extensions for certificates.
- In Chapter 18, “Using PKI Services utilities,” “Using the createcrls utility” on page 358 describes a new utility, `createcrls`, that creates and posts CRLs.
- In Chapter 18, “Using PKI Services utilities,” “Using the postcerts utility” on page 368 describes a new utility, `postcerts`, that posts issued certificates to an LDAP directory.
- In Chapter 18, “Using PKI Services utilities,” “Using the TemplateTool utility” on page 370 documents the requirement for Java 5.0 or higher for the TemplateTool utility.
- Chapter 19, “Using the certificate management protocol (CMP) with PKI Services” describes support for the certificate management protocol (CMP).
- In Chapter 20, “RACF administration for PKI Services,” on page 403, “Recovering a CA certificate profile” on page 413 contains guidelines for setting the certificate serial number incrementer value.
- Chapter 24, “Messages” documents the following new messages:
  - IKYP041E
  - IKYP042E
  - IKYP043I
  - IKYO005I
  - IKYC075I
  - IKYC076I
- Chapter 26, “The pkiserv.conf configuration file” includes OIDs for additional signature algorithms that are now supported.
- Chapter 26, “The pkiserv.conf configuration file” contains new configuration parameters to specify on which days and at which time the daily maintenance task runs, and whether it runs when the PKI daemon starts.
- Chapter 26, “The pkiserv.conf configuration file” contains a new configuration parameter to specify whether support for the certificate management protocol (CMP) is enabled.

- Chapter 26, “The pkiserv.conf configuration file” contains a new configuration parameter to specify whether certificate requests are restricted to the validity period of the CA certificate.

Changed information:

- In Chapter 5, “Configuring the UNIX runtime environment,” the description of the `TimeBetweenCRLs` configuration parameter indicates that changes to the parameter now take effect when you restart PKI Services.
- In Chapter 5, “Configuring the UNIX runtime environment,” the description of the `SigAlgl` configuration parameter shows the new default signing algorithm of SHA256 as the sample value.
- In Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” the maximum value of the KeySize field is increased to 4096 bits for RSA key generation.
- In Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” the AltDomain, AltPAddr, AltURI, and AltEmail fields can now be repeated.
- In Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” a new section “Adding custom extensions to certificates” on page 261 describes support for custom certificate extensions.
- In Chapter 14, “Advanced customization,” “Updating the signature algorithm” on page 215 documents that the default signature algorithm has changed from SHA1 to SHA256.
- Chapter 24, “Messages” contains updated information for the following messages:
  - IKYP001E
  - IKYP025I
  - IKYP031E
  - IKYS016I
- In Chapter 26, “The pkiserv.conf configuration file,” in the `pkiserv.conf` file, the default signature algorithm is changed from SHA1 to SHA256.
- In Chapter 27, “Environment variables,” the default value of the PATH environment variable is changed from `/usr/sbin` to `/bin`, which is the correct value to use if sendmail is configured to use the SMTP server.
- In Chapter 28, “The IKYSETUP REXX exec,” IKYSETUP allows elliptic curve cryptography (ECC)algorithms to be selected for the CA's private key, and allows additional sizes to be selected for the CA's private key.
- The "Readers' Comments - We'd Like to Hear from You" section at the back of this publication has been replaced with a new section “How to send your comments to IBM” on page xxi. The hardcopy mail-in form has been replaced with a page that provides information appropriate for submitting readers comments to IBM

Deleted information:

- In Chapter 20, “RACF administration for PKI Services,” information on restoring the certificate serial number incrementer value when you recover a CA certificate profile has been deleted. It is no longer necessary to do this, because PKI Services does it automatically.

Changes made in z/OS Version 1 Release 11

This document contains information previously presented in *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693-10, which supports z/OS Version 1 Release 10.

New information:

- Chapter 1, “Introducing PKI Services” adds storing key pairs that PKI Services generates to the functions that ICSF performs.
- Chapter 2, “Planning your implementation” documents that ICSF must be installed and the token data set (TKDS) must be set up if you are going to use PKI Services to generate key pairs for certificates.
- In Chapter 3, “Installing and configuring prerequisite products,” the topic “Installing and configuring ICSF (optional)” on page 23 documents that if you want PKI Services to generate key pairs for certificates, after you install ICSF you must set up the TKDS.
- Chapter 4, “Running IKYSETUP to perform RACF administration” describes two new variables for IKYSETUP that specify whether you want to allow PKI Services to generate keys for certificates, and the name of a RACF group for users authorized to use PKCS #11 tokens.
- Chapter 4, “Running IKYSETUP to perform RACF administration” documents a new variable for IKYSETUP, `cacert_dsn`.
- In Chapter 5, “Configuring the UNIX runtime environment,” SHA-256 is added as a supported signature algorithm on the SigAlg1 parameter of the configuration file.
- Chapter 5, “Configuring the UNIX runtime environment” describes new keywords added to the configuration file:
  - RemoveExpiredCertsAndKey, in the **ObjectStore** section
  - TokenName, in the **SAF** section
- Chapter 5, “Configuring the UNIX runtime environment” documents that the "Your certificate is ready" e-mail notification is required if you are setting up PKI Services to generate key pairs for certificate requests.
- Chapter 5, “Configuring the UNIX runtime environment” contains information about a new e-mail notification form, RecoverForm, sent when a user makes a request to recover a certificate for which PKI Services generated the keys, and two new substitution variables that it supports.
- Chapter 5, “Configuring the UNIX runtime environment” contains information about a new certificate template to be used when the requestor wants PKI Services to generate the keys for the certificate.
- Chapter 5, “Configuring the UNIX runtime environment” describes a new field for certificate templates that contains the size of the certificate's keys, for certificates for which PKI Services generates the keys.
- Chapter 5, “Configuring the UNIX runtime environment” contains information about updating your version of the `pkiserv.conf` file after you install a new release of z/OS.
- Chapter 9, “Creating the object store and ICL” contains information about backing up and restoring the VSAM data sets.
- Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” describes new subsections of the APPLICATION sections of the `pkiserv.tmpl` certificate templates file.

- Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” describes a new CGI exec, `cagorcvr.rexx`, that displays security questions for users to answer in order to recover a passphrase.
- Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” describes new named fields for INSERT sections.
- Chapter 13, “Implementing the Web application using Java server pages,” on page 179 is a new chapter describing how to implement the PKI Services Web application using Java server pages (JSPs).
- Chapter 14, “Advanced customization” describes creating the CertificatePolicies extension when you implement the Web application using JSPs.
- In Chapter 14, “Advanced customization,” “Updating the signature algorithm” on page 215 includes SHA-256 as a supported signature algorithm.
- In Chapter 14, “Advanced customization,” contains information about adding a new CA domain when you implement the Web application using JSPs.
- In Chapter 14, “Advanced customization,” “Customizing e-mail notifications sent to users” on page 252 shows a new e-mail notification form, recoverymsg.form, and changes to readymsg.form and expiringmsg.form. It also describes new substitution variables for forms, and customization of the URLs required for recoverymsg.form, readymsg.form, and expiringmsg.form.
- In Chapter 14, “Advanced customization,” “Setting up automatic renewal of certificates” on page 258 describes how to specify that certificates are to be automatically renewed when you implement the Web application using JSPs.
- Chapter 14, “Advanced customization” describes the set up required to allow PKI Services to generate and archive key pairs for certificate requests.
- Chapter 14, “Advanced customization” describes how to create application domains when you implement the Web application using JSPs.
- Chapter 15, “Customizing with installation exit routines” describes the QRECOVER preprocessing and post-processing exits.
- Chapter 15, “Customizing with installation exit routines” includes a scenario illustrating how the PKI Services exit can be used to allow users to recover a PKI key generated certificate when the passphrase is lost.
- Chapter 15, “Customizing with installation exit routines” describes new exit methods that can be used with Java server pages (JSPs).
- Chapter 17, “Using the administration Web pages” documents the new certificate status `Active, NotRenewable`.
- Chapter 17, “Using the administration Web pages” documents a new action for certificates, change requestor e-mail.
- Chapter 18, “Using PKI Services utilities,” on page 357 documents a new utility, TemplateTool.
- Chapter 24, “Messages” contains the following new messages:
  - IKYC074I
  - IKYI005I
  - IKYI006I
  - IKYI007I
  - IKYK001I
  - IKYK002I
  - IKYK003I
  - IKYP040I

- In Chapter 24, “Messages,” the explanation of the 4th character of messages is updated to indicate that I can indicate a message produced by PKISERV JSPs as well as CGIs.
- Chapter 26, “The pkiserv.conf configuration file,” on page 501 documents the following additions to the pkiserv.conf configuration file:
  - An OID for the SHA-256 signature algorithm
  - The `RemoveExpiredCertsAndKeys` variable
  - The `RecoverForm` variable
  - The `TokenName` variable
- Chapter 27, “Environment variables,” on page 507 documents the _PKISERV_ENABLE_JSP environment variable.
- Chapter 28, “The IKYSETUP REXX exec” describes setup to allow PKI Services to generate keys for certificates.
- Chapter 29, “Other code samples” contains new code samples:
  - IKYVBKUP contains JCL to back up the PKI Services VSAM data sets.
  - IKYVREST contains JCL to restore the PKI Services VSAM data sets.

Changed information:

- In Chapter 4, “Running IKYSETUP to perform RACF administration,” the meaning of the IKYSETUP variable `export_dsn` is changed.
- In Chapter 14, “Advanced customization,” “Customizing e-mail notifications sent to users” on page 252 documents changes to the sample e-mail notification forms readymsg.form, expiringmsg.form, and recoverymsg.form, and the additional customization these forms require.
- In Chapter 4, “Running IKYSETUP to perform RACF administration,” the variable `cacert_dsn` is used instead of `export_dsn`.
- Chapter 24, “Messages” contains updated information for the following messages:
  - IKYC032I
  - IKYC037I
  - IKYC068I
  - IKYP035I
  - IKYP036I
  - IKYU015I

Deleted information:

- Information about using the LDAP server provided by z/OS Integrated Security Services has been deleted. It has been replaced by the LDAP server provided by z/OS Tivoli® Directory Server.

Moved information:

- Information on customizing e-mail notifications sent to users has been moved from Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” to Chapter 14, “Advanced customization.”
- Information on setting up automatic certificate renewal has been moved from Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” to Chapter 14, “Advanced customization.”
- Information on using installation exits has been moved from Chapter 14, “Advanced customization” to Chapter 15, “Customizing with installation exit routines.”

Part 1. Planning

The Planning part includes the following:

- Chapter 1, "Introducing PKI Services," on page 3 provides an overview of PKI Services, its components, and related concepts.
- Chapter 2, "Planning your implementation," on page 11 provides a planning overview for your implementation, including a discussion of the components that work with PKI Services. It also discusses the team members you will need to implement PKI Services and the skills they will need.
- Chapter 3, "Installing and configuring prerequisite products," on page 19 describes installing and configuring related products: the z/OS HTTP Server, LDAP, and optionally ICSF and OCSF.

© Copyright IBM Corp. 2001, 2011

Chapter 1. Introducing PKI Services

This topic provides an overview of PKI Services.

It covers the following topics:

- What is PKI Services?
- What is a certificate authority?
- What is PKI?
- Basic components of PKI Services and related products
- Component diagram
- Supported standards
- Supported certificate types
- Supported certificate fields and extensions

What is PKI Services?

z/OS Cryptographic Services PKI Services allows you use z/OS to establish a PKI infrastructure and serve as a certificate authority for your internal and external users, issuing and administering digital certificates in accordance with your own organization's policies. Your users can use a PKI Services application to request and obtain certificates through their own Web browsers, while your authorized PKI administrators approve, modify, or reject these requests through their own Web browsers. The Web applications provided with PKI Services are highly customizable, and a programming exit is also included for advanced customization. You can allow automatic approval for certificate requests from certain users and, to provide additional authentication, add host IDs, such as RACF user IDs, to certificates you issue for certain users. You can also issue your own certificates for browsers, servers, and other purposes, such as virtual private network (VPN) devices, smart cards, and secure e-mail.

PKI Services supports Public Key Infrastructure for X.509 version 3 (PKIX) and Common Data Security Architecture (CDSA) cryptographic standards. It also supports the following:

- The delivery of certificates through the Secure Sockets Layer (SSL) for use with applications that are accessed from a Web browser or Web server.
- The delivery of certificates that support the Internet Protocol Security standard (IPSEC) for use with secure VPN applications or IPSEC-enabled devices.
- The delivery of certificates that support Secure Multipurpose Internet Mail Extensions (S/MIME), for use with secure e-mail applications.

z/OS is certified as IdenTrust™ compliant. This allows z/OS installations to participate in the IdenTrust infrastructure by configuring PKI Services to operate as an IdenTrust compliant certificate authority (CA). For details, see "Configuring PKI Services as an IdenTrust™ certificate authority" on page 573.

What is a certificate authority?

The certificate authority, commonly called a CA, acts as a trusted third party to ensure that users who engage in e-business can trust each other. A certificate authority vouches for the identity of each party through the certificates it issues. In

© Copyright IBM Corp. 2001, 2011

addition to proving the identity of the user, each certificate includes a public key that enables the user to verify and encrypt communications.

The trustworthiness of the parties depends on the trust that is placed in the CA that issued the certificates. To ensure the integrity of a certificate, the CA digitally signs the certificate as part of creating it, using its signing private key. Trying to alter a certificate invalidates the signature and renders it unusable.

Protecting the CA's signing private key is critical to the integrity of the CA. For this reason, you should consider using ICSF to securely store your PKI Services CA's private key.

As a CA using PKI Services, you can do the following:

- Track certificates you issue with an issued certificate list (ICL) that contains a copy of each certificate, indexed by serial number
- Track revoked certificates using certificate revocation lists (CRLs). When a certificate is revoked, PKI Services updates the CRL during the next periodic update. Just as it signs certificates, the CA digitally signs all CRLs to vouch for their integrity.

What is PKI?

The public key infrastructure (PKI) provides applications with a framework for performing the following types of security-related activities:

- Authenticate all parties that engage in electronic transactions
- Authorize access to sensitive systems and repositories
- Verify the author of each message through its digital signature
- Encrypt the content of all communications.

The PKIX standard evolved from PKI to support the interoperability of applications that engage in e-business. Its main advantage is that it enables organizations to conduct secure electronic transactions without regard for operating platform or application software package.

The PKIX implementation in PKI Services is based on the Common Data Security Architecture (CDSA) from Intel Corporation. CDSA supports multiple trust models, certificate formats, cryptographic algorithms, and certificate repositories. Its main advantage is that it enables organizations to write PKI-compliant applications that support their business policies.

Basic components of PKI Services and related products

Table 1. Basic components of PKI Services and related products

Administration Web application	Assists authorized administrators to review requests for certificates, approve or reject requests, renew certificates, or revoke certificates through their own Web browsers. The application consists of sample screens that you can easily customize to display your organization's logo. It also supports the following tasks: • Reviewing pending certificate requests • Querying pending requests to process those that meet certain criteria • Displaying detailed information about a certificate or request • Monitoring certificate information, such as validity period • Annotating the reason for an administrative action
DB2 (optional)	Provides an alternative to VSAM data sets as the repository for the object store (request database) and issued certificate list (ICL).

Table 1. Basic components of PKI Services and related products (continued)

End-user Web application	Guides your users to request, obtain, and renew certificates through their Web browsers. The application consists of sample screens that you can easily customize to meet your organization's needs for certificate content and standards for appearance. It offers several certificate templates that you can use to create requests for a variety of certificate types, based on the certificate's intended purpose and validity period, and supports certificate requests that are automatically approved.
Exit	Provides advanced customization including additional authorization checking, additional validation, changes to parameters on calls to the `R_PKIServ` callable service (IRRSPX00), and capture of certificates for further processing. An exit program can be called from the daemon, for automatic certificate renewal, or from the `PKIServ` CGIs. Exit methods can be called from Java server pages (JSPs). The exit program and methods support both preprocessing and post-processing functions. A code sample in C language code is included.
ICSF (optional)	Securely stores the PKI Services certificate authority's private signing key and key pairs that PKI Services generates for certificates.
LDAP	The directory that maintains information about the valid and revoked certificates that PKI Services issues in an LDAP-compliant format. You can use an LDAP server such as the one provided by IBM Tivoli Directory Server for z/OS.
PKI Services daemon	The server daemon that acts as your certificate authority, confirming the identities of users and servers, verifying that they are entitled to certificates with the requested attributes, and approving and rejecting requests to issue and renew certificates. It includes support for: • An issued certificate list (ICL) to track issued certificates • Certificate revocation lists (CRLs) to track revoked certificates
`R_PKIServ` callable service (IRRSPX00)	The application programming interface (API) that allows authorized applications, such as servers, to programmatically request the functions of PKI Services to generate, retrieve and administer certificates.
RACF (or equivalent)	Controls who can use the functions of the `R_PKIServ` callable service and protects the components of your PKI Services system. RACF creates your certificate authority's certificate, key ring and private key. You can also use it to store the private key, if ICSF is not available.
Websphere Application server (optional)	Serves as the application server if you implement the PKI Services Web application using Java server pages (JSPs).
z/OS HTTP Server (optional)	PKI Services uses the Web server to encrypt messages, authenticate requests, and transfer certificates to intended recipients if you implement the PKI Services Web application using REXX CGI execs.

Component diagram

Figure 1 shows a typical PKI Services system.

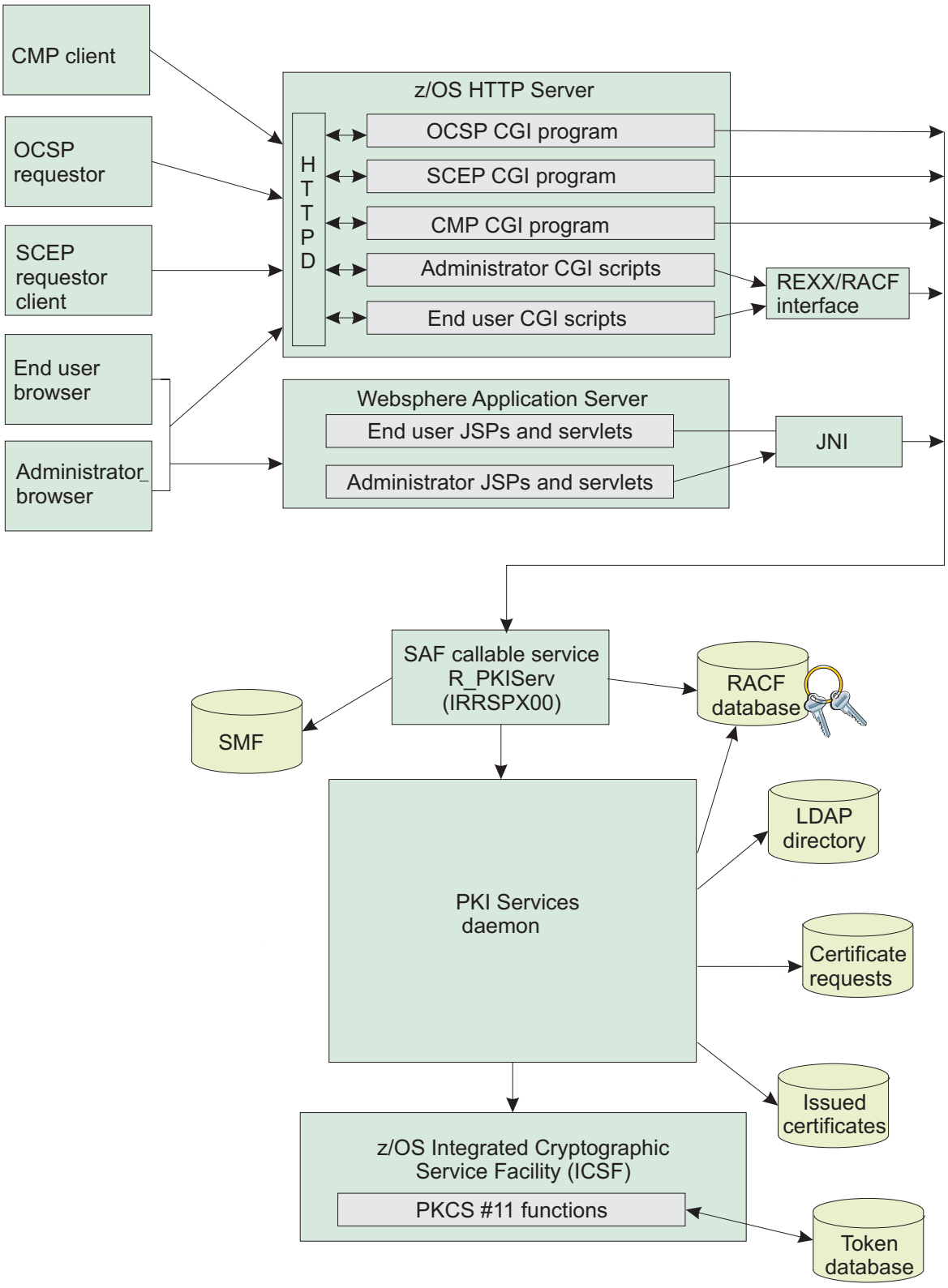


Figure 1. Component diagram of a typical PKI Services system

Supported standards

PKI Services supports the following standards for public key cryptography:

- Secure Sockets Layer (SSL) version 2 and version 3, with client authentication
- PKCS #10 browser and server certificate format, with a base64-encoded response
- IPSEC certificate format
- S/MIME certificate format
- Browser certificates for:
  - 32-bit versions of Microsoft Internet Explorer
  - Mozilla-based browsers such as Mozilla Firefox
- Server certificates
- LDAP standard for communications with the directory
- X.509v3 certificates
- Certificate revocation lists (CRLv2)
- Key lengths up to 4096 bits for the RSA CA signing private keys and up to 1024 bits for DSA keys
- RSA algorithms for encryption and signing
- DSA algorithms for signing
- ECC algorithms for encryption and signing
- RFC 2560: *Online Certificate Status Protocol - OCSP*
- RFC 4291: *IP Version 6 Addressing Architecture*
- RFC 4210: *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*
- RFC 4211 *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*
- Cisco Systems' Simple Certificate Enrollment Protocol (SCEP)

The LDAP standard that PKI Services supports is LDAP Version 2. A directory using LDAP Version 3 (with RFC 1779 syntax) is acceptable if it is backwardly compatible with Version 2.

Supported certificate types

Table 2 lists the types of certificates that you can request, based on the certificate templates that are included with PKI Services. Certificate templates are samples of the most commonly requested certificate types. You can add, modify, and remove certificate templates to customize the variety of certificate types you offer to your users.

Table 2. Types of certificates you can request

Type of certificate	Use
One-year PKI SSL browser certificate	End-user client authentication using SSL
One-year PKI S/MIME browser certificate	Browser-based e-mail encryption
One-year PKI generated key certificate	Generation of public and private keys by PKI Services
Two-year PKI browser certificate for authenticating to z/OS	End-user client authorization using SSL when logging onto z/OS
Two-year PKI Authenticode - code signing server certificate	Software signing

Table 2. Types of certificates you can request (continued)

Type of certificate	Use
Two-year PKI Windows logon certificate	End-user client authentication for an Active Directory user logging in to a Windows desktop using a smart card
Five-year PKI SSL server certificate	SSL Web server certification
Five-year PKI IPSEC server (firewall) certificate	Firewall server identification and key exchange
Five-year PKI intermediate CA server certificate	Subordinate (non-self-signed) certificate-authority certification
Five-year SCEP certificate	Creation of a preregistration record for certificate requestors. (Certificate requestors using Simple Certificate Enrollment Protocol (SCEP) must be preregistered.) Unlike other templates, this template is intended for administration use only.
n-year PKI browser certificate for extensions demonstration	Demonstration of all extensions supported by PKI Services
One-year SAF browser certificate	End-user client authentication where the security product (RACF, not PKI Services) is the certificate provider **Note:** The certificate generated by this template cannot be managed by the PKI Services administrator.
One-year SAF server certificate	Web server SSL certification where the security product (RACF, not PKI Services) is the certificate provider **Note:** The certificate generated by this template cannot be managed by the PKI Services administrator.

Note: You can customize certificate templates to add, modify and remove certificate types.

Supported certificate fields and extensions

PKI Services certificates support most of the fields and extensions defined in the X.509 version 3 (X.509v3) standard. This support lets you use these certificates for most cryptographic purposes, such as SSL, IPSEC, VPN, and S/MIME.

PKI Services supports Basic Latin and Latin-1 supplement characters in the Subject Distinguished Name, Issuer Distinguished Name and Othername in Subject Alternate Name.

PKI Services certificates can include the following types of extensions:

Standard extensions

The standard X.509v3 certificate extensions:

- authority information access
- authority key identifier
- basic constraints
- certificate policies

- certificate revocation list (CRL) distribution points
- extended key usage
- key usage
- subject alternate name
- subject key identifier

Other extensions
Extensions that are unique to PKI Services, such as host identity mapping. This extension associates the subject of a certificate with a corresponding identity on a host system, such as with a RACF user ID.

To support your organization's policies, PKI Services provides the means for you to select and customize the supported certificate extensions. For example, you can change the extensions that are specified in the default certificate templates or create templates that return certificates with different extensions. In addition, you can include your own extensions in your certificates by defining custom extensions.

Chapter 2. Planning your implementation

The implementation of PKI Services requires the interaction of several software products, each with its own required skills. Therefore, it is important to understand the tasks involved and to plan your implementation.

This topic provides the information you need to understand the task of implementing PKI Services, determine which skills are required to complete your implementation team, and create your own implementation plan.

This topic covers the following topics:

- Installing PKI Services
- Determining prerequisite products
- Identifying skill requirements
- Creating an implementation plan.

Installing PKI Services

Your MVS programmer uses SMP/E to install PKI Services into a file system directory. By default, PKI Services is installed in the `/usr/lpp/pkiserv` directory but the MVS programmer can determine whether to change the default for this and other directories. Before your team begins installing and configuring prerequisite products and setting up PKI Services, you will need to know which file system directories were used so you can customize the install process.

Table 3 shows each file system variable with its description and default value. Your MVS programmer should review the rightmost column of this table, crossing out any defaults that have changed and recording the correct directory names.

Table 3. File system directory variables

Variable name	Description	Default value or customized value
variables-dir	The file system directory where PKI Services creates working files.	**/var/pkiserv**
install-dir	The file system directory where PKI Services is installed.	**/usr/lpp/pkiserv**
runtime-dir	The file system directory where PKI Services looks for configuration files.	**/etc/pkiserv**

Requirements for sysplex support

If your installation plans to use sysplex support (running multiple independent instances of PKI Services, one per image, that work in unison):

- If you are using VSAM data sets for the object store and ICL, all systems in the sysplex that run PKI Services must have the same release of z/OS installed, and all instances of PKI Services must share the same VSAM data sets. To do so, they use VSAM record-level sharing (RLS). This requires setting up a coupling facility for data sharing (lock and cache).
- If you are using DB2 tables for the object store and ICL, all systems in the sysplex that run PKI Services must be at z/OS V1R13 or later and have DB2

© Copyright IBM Corp. 2001, 2011

Version 9 installed, and the DB2 subsystems that share data must belong to a DB2 data sharing group that runs on a sysplex cluster.

See "(Optional) preliminary steps for establishing VSAM RLS" on page 105 for information about creating VSAM data sets suitable for VSAM RLS. For information on establishing a Parallel Sysplex environment with a coupling facility, see *z/OS MVS Programming: Sysplex Services Guide*. For more information about establishing data sharing for VSAM RLS, see *z/OS DFSMS Introduction* and *z/OS DFSMSdfp Storage Administration*.

Determining prerequisite products

The installation and use of PKI Services requires the following products:

- z/OS HTTP Server
- Websphere Application Server (optional)
- LDAP directory server
- OCSF (optional)
- ICSF (optional)
- sendmail (optional)
- OCEP (optional).
- DB2 (optional).

The installation and use of RACF, or an equivalent security product, is required.

z/OS HTTP Server

In a PKI Services system, if you implement the Web application using REXX CGI execs the z/OS HTTP Server handles all requests that it receives from a Web browser. This includes requests for new certificates and requests to renew or revoke existing certificates. If needed, it performs authentication before allowing any exchange of information to take place.

z/OS HTTP Server must be installed on the same system where PKI Services is installed. SSL-enablement is required. If your HTTP server is SSL-enabled, your key file can be a RACF key ring, or a key file created by another product. For more information, see "Steps for installing and configuring the z/OS HTTP Server to work with PKI Services " on page 20.

Websphere Application Server (optional)

If you implement the PKI Services Web application using Java server pages (JSPs), you must use Websphere Application Server 6.1 or higher. An application server is a Java virtual machine (JVM) running user applications. Websphere Application Server provides application servers that handle Web application requests.

If you implement the PKI Services Web application using REXX CGI execs, Websphere Application Server is not required.

LDAP directory server

Use of an LDAP server is required to maintain information about PKI Services certificates in a centralized location. The z/OS LDAP server provided by IBM Tivoli Directory Server for z/OS is preferred, but you can use a non-z/OS LDAP server if it can support the objectclasses and attributes that PKI Services uses. Typical PKI Services usage requires an LDAP directory server that supports the

LDAP (Version 2) protocol (and the PKIX schema). If you use the z/OS LDAP server provided by IBM Tivoli Directory Server for z/OS, configure it for either the TDBM or LDBM backend.

Through the integration of the z/OS LDAP server with DB2, the directory can support millions of directory entries. It also allows client applications, such as PKI Services, to perform database storage, update, and retrieval transactions. For more information, see "Steps for installing and configuring LDAP" on page 21.

OCSF (optional)

You need to install and configure OCSF if your installation plans to write an application to implement the use of PKI Trust Policy (PKITP). For more information, see "Installing and configuring OCSF" on page 25.

ICSF (optional)

ICSF is preferred but not required. You can begin using PKI Services without installing ICSF and install it later without reinstalling PKI Services. ICSF is strongly suggested to store and protect your certificate authority's private key. If you plan to have PKI Services generate key pairs for certificates, you must install ICSF and set up the ICSF PKCS #11 token data set (TKDS). For more information, see "Installing and configuring ICSF (optional)" on page 23.

sendmail (optional)

You need to configure sendmail if your installation plans to send e-mail notifications to users for certificate-related events, such as certificate expiration. For more information, see "Configuring sendmail (optional)" on page 24.

OCEP (optional)

You need to install and configure OCEP if your installation plans to write an application to implement the use of PKI Trust Policy (PKITP). For more information, see "Configuring and getting started with PKITP" on page 429.

DB2 (optional)

You need to install and configure DB2 if your installation chooses to implement the object store and issued certificate list (ICL) using DB2 instead of VSAM. For more information, see Chapter 9, "Creating the object store and ICL," on page 103.

Identifying skill requirements

The implementation of PKI Services requires the interaction of several software products, each with its own required skills. This means that your team might consist of people from several different disciplines, particularly if you work with a large organization.

This section provides the information you need to determine which skills are required to complete your implementation. These skills are presented in terms of job titles for people who specialize in those skills. For example, a task requiring MVS skills is referred to as a task for an MVS programmer. Therefore, if some of your team members have multiple skills, you might require fewer individuals to complete your team.

Team members

Your team for installing and configuring prerequisite products and setting up PKI Services should include the following members:

- DB2 database administrator (DBA) (optional)
- ICSF programmer (optional)
- LDAP programmer
- MVS programmer
- OCEP programmer (optional)
- OCSF programmer (optional)
- RACF administrator
- UNIX programmer
- Web server programmer

You might wish to include a Web page designer to customize your PKI Services Web applications. This task is listed in the topic as a task for a Web server programmer.

One or more PKI administrators are needed to manage your ongoing operation as a certificate authority, once your PKI Services system is set up. The responsibilities of these administrators include approving, modifying and rejecting certificate requests and revoking certificates. It might be advisable to appoint a PKI administrator early, and involve this person in your planning.

Important: PKI administrators play a very powerful role in your organization. The decisions they make when managing certificates and certificate requests determine who will access your computer systems and what privileges they will have when doing so. Assign PKI administration duties to only highly trusted individuals.

Skills for setting up prerequisite products

The following table lists team members (alphabetically) and tasks and required skills needed for installing and configuring prerequisite products:

Table 4. Tasks and skills needed for installing prerequisite products

Role	Tasks	Required skills	Documented in:
DB2 database administrator (DBA)	(Optionally) installing and configuring DB2 (if not already done)	DB2 installation and configuration skills	• *Installation Guide* for your version of DB2.
ICSF programmer	(Optionally) installing and configuring ICSF (if not already done)	ICSF installation and configuration skills	• *z/OS Cryptographic Services ICSF Administrator's Guide* • *z/OS Cryptographic Services ICSF Application Programmer's Guide* • *z/OS Cryptographic Services ICSF System Programmer's Guide*
	(Optionally) setting up the ICSF PKCS #11 token data set (TKDS) (if not already done)	ICSF installation and configuration skills	• *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*

Table 4. Tasks and skills needed for installing prerequisite products (continued)

Role	Tasks	Required skills	Documented in:
LDAP programmer	Installing and configuring LDAP (if not already done) and recording information	LDAP installation and configuration skills	• *IBM Tivoli Directory Server Administration and Use for z/OS*
OCEP programmer	(Optionally) Installing and configuring OCEP for use with PKITP	OCEP installation and configuration skills	• *z/OS Integrated Security Services Open Cryptographic Enhanced Plug-ins Application Programming*
OCSF programmer	(Optionally) Installing and configuring OCSF for use with PKITP	OCSF installation and configuration skills	• *z/OS Open Cryptographic Services Facility Application Programming*
UNIX programmer	(Optionally) Configuring sendmail if your installation is planning to send e-mail notifications to users about certificates	• Basic UNIX commands such as the **cp** (copy) command and **mkdir** (make directory) command • sendmail configuration skills	• *z/OS Communications Server: IP Configuration Guide*
Web server programmer	**If you are implementing the PKI Services Web application using REXX CGI execs:** Installing and configuring the z/OS HTTP Server (if not already configured for at least non-SSL pages) and recording information	z/OS HTTP Server installation and configuration skills	• *z/OS HTTP Server Planning, Installing, and Using*
	If you are implementing the PKI Services Web application using Java server pages (JSPs): Installing and configuring Websphere Application Server	Websphere Application Server installation and configuration skills	• The information center for your release of Websphere Application Server, at http://www.ibm.com/software/webservers/appserv/was/library/

Your team needs to install and configure prerequisite products before setting up PKI Services:

1. The Web server programmer installs and configures the z/OS HTTP Server (if you are implementing the PKI Services Web application using REXX CGI execs) or Websphere Application Server (if you are implementing the PKI Services Web application using Java server pages).
2. The LDAP programmer installs and configures LDAP.
3. Optionally, the ICSF programmer installs and configures ICSF, and optionally sets up the token data set (TKDS).
4. Optionally, the OCEP programmer installs and configures the OCEP.
5. Optionally, the OCSF programmer installs and configures the OCSF.
6. Optionally, the DB2 database administrator installs and configures DB2.

See Chapter 3, “Installing and configuring prerequisite products,” on page 19 for details about performing these tasks.

Skills for setting up PKI Services

The following table lists team members (alphabetically) and the tasks and skills needed for setting up PKI Services:

Table 5. Roles, tasks, and skills for setting up PKI Services

Role	Tasks	Required skills	Documented in:
DB2 database administrator	• Creates DB2 tables, package, and plan for the object store and issued certificate list (ICL).	• DB2 programming skills • DB2 administration skills	• *Application Programming and SQL Guide* • *Command Reference* • *SQL Reference* for your version of DB2.
LDAP programmer	• Customizes LDAP configuration for PKI Services	• LDAP customization skills	• *IBM Tivoli Directory Server Administration and Use for z/OS*
MVS programmer	• (Optionally) Creates VSAM object store and ICL data sets and indexes • (Optionally) sets up VSAM RLS • Starts the PKI Services daemon	• Basic MVS skills – Editing a data set – ISPF COPY command – MVS console START command • JCL knowledge to change job card • Basic browser and Web skills	• *z/OS MVS System Commands*
RACF administrator	• Adds groups and user IDs • Sets up access control • Creates certificates • Sets up daemon security	• RACF administration • REXX skills (for working with IKYSETUP REXX exec) • RACF commands such as the following: ADDGROUP ADDSD ADDUSER RACDCERT RDEFINE PERMIT SETROPTS • TSO skills	• *z/OS TSO/E REXX Reference* • *z/OS UNIX System Services Planning* • *z/OS Security Server RACF Security Administrator's Guide*
UNIX programmer	• Copies files • (Optionally) customizes environment variables • (Optionally) customizes (non-**LDAP** sections of) `pkiserv.conf` configuration file • Sets up `/var/pkiserv` directory • Updates the **LDAP** section of the `pkiserv.conf` configuration file	• Basic UNIX commands, such as the **cp** (copy) command • Getting superuser authority	• *z/OS UNIX System Services Command Reference* • *z/OS UNIX System Services Planning*

Table 5. Roles, tasks, and skills for setting up PKI Services (continued)

Role	Tasks	Required skills	Documented in:
Web server programmer	• Helps set up PKI Services – If you are implementing the PKI Services Web application using REXX CGI execs: Updates the z/OS HTTP Server configuration files and starts the z/OS HTTP Server – If you are implementing the PKI Services Web application using Java server pages (JSPs), updates the enterprise archive (EAR) file and deploys updated JSP files to a Websphere application server. • Customizes the PKI Services Web pages	• z/OS HTTP Server customization skills • Websphere Application Server customization and administration skills • Editing configuration files • Customizing the Web pages	• *z/OS HTTP Server Planning, Installing, and Using* • The information center for your release of Websphere Application Server, at http://www.ibm.com/software/webservers/appserv/was/library/

Creating an implementation plan

Your implementation plan should include major subtasks, responsible parties, and a realistic estimate of time and effort required. The major tasks for implementing PKI Services are provided here as a basis for you to build your own plan.

Task roadmap for implementing PKI Services

Table 6 shows the subtasks and associated procedures for implementing PKI Services. These tasks will comprise the major part of your implementation plan.

Table 6. Task roadmap for implementing PKI Services

Subtask	Associated procedure (See ...)
Installing and configuring prerequisite products:	Chapter 3, "Installing and configuring prerequisite products"
• z/OS HTTP Server	• Steps for installing and configuring the z/OS HTTP Server to work with PKI Services
• OCSF (optional)	• Installing and configuring OCSF
• LDAP directory server	• Steps for installing and configuring LDAP
• ICSF (optional)	• Installing and configuring ICSF (optional)
• OCEP (optional)	• Configuring and getting started with PKITP
• sendmail (optional)	• Configuring sendmail (optional)
• DB2 (optional)	• Installing and configuring DB2
Configuring your system for PKI Services:	Part 2, "Configuring your system for PKI Services"

Table 6. Task roadmap for implementing PKI Services (continued)

Subtask	Associated procedure (See ...)
• RACF	• Chapter 4, "Running IKYSETUP to perform RACF administration"
• z/OS UNIX	• Chapter 5, "Configuring the UNIX runtime environment"
• LDAP configuration	• Chapter 6, "Tailoring the LDAP configuration for PKI Services"
• z/OS HTTP Server	• Chapter 7, "Updating z/OS HTTP Server configuration and starting the server"
• LDAP	• Chapter 8, "Tailoring the PKI Services configuration file for LDAP"
• Set up the object store and issued certificate list (ICL)	• Chapter 9, "Creating the object store and ICL" • Chapter 10, "Starting and stopping PKI Services"
Customizing PKI Services:	Part 3, "Customizing PKI Services"
• Customizing end-user Web pages	• Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs"
• Customizing administration Web pages	• Chapter 12, "Customizing the administration Web pages if you use REXX CGI execs"
• Advanced customizing	• Chapter 14, "Advanced customization"
Testing PKI Services:	Part 4, "Using PKI Services"
• Using end-user Web pages	• Chapter 16, "Using the end-user Web pages"
• Using administration Web pages	• Chapter 17, "Using the administration Web pages"
Administering PKI Services:	Part 5, "Administering security for PKI Services"
• RACF	• Chapter 20, "RACF administration for PKI Services"

Chapter 3. Installing and configuring prerequisite products

After the MVS programmer installs PKI Services using SMP/E (but before team members set up PKI Services, see Chapter 4, "Running IKYSETUP to perform RACF administration" through Chapter 9, "Creating the object store and ICL"), your team needs to set up prerequisite products:

- z/OS HTTP Server - to handle requests through a Web Server
- Websphere Application Server for z/OS (optional)- provides the Web server if you implement the PKI Services Web application using Java server pages (JSPs)
- LDAP - for posting certificates and CRLs
- ICSF (optional) - to store the CA's private key in hardware, and to generate and store key pairs for PKI Services certificate requests
- sendmail (optional) - for sending e-mail notifications to certificate requestors and administrators
- OCSF (optional) - used by the PKI Trust Policy (PKITP)
- OCEP (optional) - used by the PKI Trust Policy (PKITP)
- DB2 (optional) - can be used to store the object store and issued certificate list (ICL) if VSAM data sets are not used

You need to install and configure the z/OS HTTP Server and LDAP only if you are setting up prerequisite products for PKI Services for the first time.

Tasks to perform before setting up PKI Services

Before you can set up PKI Services, your team needs to set up prerequisite software products by completing the following tasks, if not already done:

1. "Installing and configuring the z/OS HTTP Server"
2. "Installing and configuring Websphere Application Server for z/OS" on page 21
3. "Installing and configuring LDAP" on page 21
4. "Installing and configuring ICSF (optional)" on page 23
5. "Configuring sendmail (optional)" on page 24
6. "Installing and configuring DB2" on page 25

This topic explains these tasks in more detail.

Installing and configuring the z/OS HTTP Server

You need to perform this task only if you are setting up prerequisite products for PKI Services for the first time.

PKI Services requires that you have the z/OS HTTP Server installed and configured for at least non-SSL page retrieval. Tasks of other team members, such as the RACF administrator and Web server programmer (see Chapter 4, "Running IKYSETUP to perform RACF administration," on page 29 and Chapter 7, "Updating z/OS HTTP Server configuration and starting the server," on page 91) assume that this is already done.

© Copyright IBM Corp. 2001, 2011

Steps for installing and configuring the z/OS HTTP Server to work with PKI Services

Before you begin:

1. You need Web server programming skills to complete this procedure.
2. You might need to refer to the following document:
   z/OS HTTP Server Planning, Installing, and Using

Perform the following steps to install and configure the z/OS HTTP Server to work with PKI Services:

1. Use the following table to decide what you need to do:

If ...	Then ...	Notes
The z/OS HTTP Server is not installed and configured ...	Install and configure z/OS HTTP Server by following the instructions in the installation section of *z/OS HTTP Server Planning, Installing, and Using*.	**Guideline:** For PKI Services, when you install the z/OS HTTP Server, do not use a password file.
The z/OS HTTP Server is installed but not configured for SSL ...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 4, "Running IKYSETUP to perform RACF administration.")	—
The z/OS HTTP Server is installed and configured for SSL using a RACF key ring ...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 4, "Running IKYSETUP to perform RACF administration.")	—
The z/OS HTTP Server is installed and configured for SSL using gskkyman ...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 4, "Running IKYSETUP to perform RACF administration." The RACF programmer also needs to add your CA certificate to an existing keyfile; see Using a gskkyman key database for your certificate store for information about gskkyman steps.)	—

You can now perform the steps for the decision you have made.

---

2. Fill in the rightmost column of the following table with information from the configuration:

Table 7. z/OS HTTP Server information you need to record

z/OS HTTP Server information	Explanation	Value
z/OS HTTP Server fully qualified domain name	A fully qualified domain name is the name of a host system. It includes a series of subnames (each of which is a domain name). For example, `ralvm7.vnet.ibm.com` is a fully qualified domain name that includes the domain names `ibm.com` and `vnet.ibm.com`. (The RACF administrator needs to know the fully qualified domain name when setting up PKI Services.)	
The full UNIX pathname of your `httpd.conf` configuration file	(The Web server programmer needs to know the full UNIX pathname when updating the `httpd.conf` configuration file to support PKI Services.)	

Installing and configuring Websphere Application Server for z/OS

You need to perform this task only if you are setting up prerequisite products for PKI Services for the first time, or if you are implementing the PKI Services Web application using Java server pages (JSPs) for the first time.

If you implement the PKI Services Web application using Java server pages (JSPs), you must install Websphere Application Server 6.1 or higher and configure an SSL configuration that uses client authentication. For more information, see Chapter 13, "Implementing the Web application using Java server pages," on page 179 and the online information center for your release of Websphere Application Server at http://www.ibm.com/software/webservers/appserv/was/library/.

You do not need to install Websphere Application Server if you implement the PKI Services Web application using REXX CGI execs.

Installing and configuring LDAP

The LDAP programmer installs and configures LDAP for the TDBM DB2 backend and records entries that will be needed later.

Steps for installing and configuring LDAP

You need to perform this task only if you are setting up prerequisite products for PKI Services for the first time.

Although it can be configured otherwise, typical PKI Services usage requires access to an LDAP directory server. Install the LDAP directory server separately from PKI Services. After the installation is complete, LDAP needs to be configured for PKI Services. The directory stores issued certificates and certification revocation lists. The z/OS LDAP server provided by IBM Tivoli Directory Server for z/OS is preferred but not required. The remainder of this topic assumes you will use the IBM Tivoli Directory Server for z/OS LDAP server.

Note: The default name of the LDAP server configuration file is `ds.conf` .

You can use a non-z/OS LDAP server if it can support the object classes and attributes that PKI Services uses. For information about using a non-z/OS LDAP server, see "LDAP directory server requirements" on page 567.

Before you begin:

1. You need LDAP programming skills to complete this procedure.
2. You will need to refer to *IBM Tivoli Directory Server Administration and Use for z/OS*.

Perform the following steps to install and configure LDAP to work with PKI Services:

1. Use the following table to decide what you need to do:

If ...	Then...	Notes
You do not have LDAP installed and configured ...	Follow the instructions in the Administration section of *IBM Tivoli Directory Server Administration and Use for z/OS*.	
You have LDAP installed and configured but not for the TDBM or LDBM backend ...	You need to migrate to the TDBM or LDBM backend. See *IBM Tivoli Directory Server Administration and Use for z/OS* for details about how to do this.	—
You have LDAP installed and configured for the TDBM or LDBM backend ...	Go to the next step.	—

You can now perform the steps for the decision you have made.

---

2. Record the entries and values from the LDAP configuration step in the following table. (Your team will need this information when setting up PKI Services.)

Table 8. LDAP information you need to record

LDAP information	Explanation	Value

Distinguished name	This is the distinguished name to use for LDAP binding. A distinguished name is the unique name of a data entry that identifies its position in the hierarchical structure of the directory. A distinguished name consists of the relative distinguished name (RDN) concatenated with the names of its ancestor entries. For example, an entry for Tim Jones could have an RDN of `CN=Tim Jones` and a DN of:
	`CN=Tim Jones,O=IBM,C=US`
	
	Any RDN type supported by the LDAP server can be used.
	
	The distinguished name can be a RACF-style distinguished name. For information about RACF-style distinguished names, see *IBM Tivoli Directory Server Administration and Use for z/OS*. For example, an entry for RACF user ID `timjones` will be:
	`RACFID=timjones,PROFILETYPE=user,O=racfdb,C=us`	

Distinguished name password	This is the password defined for the distinguished name above, for use by PKI to bind to the LDAP server. RACF passwords can be case-sensitive, so make sure that the password specified for a RACF-style distinguished name in the pkiserv.conf file or in the LDAPBIND profile matches the RACF password exactly.	

Table 8. LDAP information you need to record (continued)

LDAP information	Explanation	Value
LDAP fully qualified domain name and port	This is the domain name on which the LDAP server is listening. For example, for `ldap.widgets.com:389`, the fully qualified domain name is `ldap.widgets.com` and the port is 389. See Table 7 on page 21 for a definition of fully qualified domain name.	

Suffix	A suffix in LDAP is the top level name of the subtree. For example, for the following distinguished name:
	`OU=your-CA's-friendly-name,O=your-organization, C=your-country-abbreviation`
	
	the suffix could be either "O=*your company*,C=*your-country-abbreviation*" or "C=*your-country-abbreviation*"
	
	The suffix value is specified after the suffix keyword in the LDAP server configuration file:
	`suffix "O=your-company,C=your-country-abbreviation"`
	
	Note: If you have more than one suffix, record the suffix you intend to use as the root for storing the PKI Services CA certificate.	

3. The topics that follow require the LDAP server to be running. Follow the instructions in the topic about running the LDAP server in *IBM Tivoli Directory Server Administration and Use for z/OS*.

Installing and configuring ICSF (optional)

You can install and configure ICSF the first time you are setting up PKI Services or at a later time. Using ICSF is suggested, but it is not required for most PKI Services functions. However, ICSF is required for the following functions:

- RACF can use ICSF's public key data set (PKDS) to securely store the PKI Services CA signing key if directed to do so. For this to be successful, the ICSF programmer must install and configure ICSF for Public Key Algorithms (PKA), and ICSF must be running. (The RACF administrator uses the IKYSETUP REXX exec to set up any RACF profiles needed to control access to ICSF services and keys. For more information, see Chapter 4, "Running IKYSETUP to perform RACF administration," on page 29.)
- PKI Services uses ICSF's PKCS #11 token data set (TKDS) to store key pairs that PKI Services generates for non-CMP (certificate management protocol) certificate requests. If ICSF is not running and the TKDS is not set up, PKI Services cannot generate key pairs.
- PKI Services uses ICSF's PKCS #11 support to handle certificate requests that involve elliptic curve cryptography (ECC) keys. If ICSF is not running, PKI Services cannot process a certificate request that has an ECC public key, or for which the signing key is an ECC private key. (The TKDS is not required for these functions.)
- The PKI Services certificate management protocol (CMP) CGI uses ICSF's PKCS #11 support to generate key pairs. If ICSF is not running, the CMP CGI cannot generate key pairs. (The TKDS is not required, because the CMP CGI does not store keys.)

Note: You do not have to choose whether or not to install ICSF and perform the installation and configuration at this point. You can do so later in the process.

Before you begin:

- You need ICSF programming skills to complete this procedure.
- You might need to refer to the following documents:

  z/OS Cryptographic Services ICSF Administrator's Guide

  This document provides information about managing cryptographic keys, setting up and maintaining the PKDS, controlling who can use cryptographic keys and services, and general information about ICSF and cryptographic keys.

  z/OS Cryptographic Services ICSF Writing PKCS #11 Applications

  This document describes the ICSF support for PKCS #11, and provides information about setting up the TKDS.

If ICSF is not already installed and configured for PKA, do this by following the instructions in *z/OS Cryptographic Services ICSF Administrator's Guide*. If you want PKI Services to generate key pairs for certificate requests, set up the TKDS by following the instructions in *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*.

Configuring sendmail (optional)

The UNIX programmer needs to configure sendmail if your installation plans to do any of the following:

- Send e-mail notifications to users whose certificate request is rejected, ready for retrieval, or about to expire
- Send e-mail notifications to administrators who have requests pending
- Use automatic certificate renewal

Before you begin: You need the following document:

- *z/OS Communications Server: IP Configuration Guide*

Follow the instructions in *z/OS Communications Server: IP Configuration Guide* for configuring z/OS UNIX sendmail. In general, you need to perform the following steps:

1. Create an alias file to define the postmaster and MAILER-DAEMON user IDs and the nobody alias (`/dev/null`).

   _____

2. Create the sendmail configuration file using the `m4` macro preprocessor.

   _____

3. Load this configuration file into sendmail.

   _____

Notes:

1. Because PKI Services always provides the return e-mail address, you do not need to configure sendmail to provide it. This simplifies your setup.
2. If you configure sendmail to run on the local system as a server, you need to change the PATH statement in the `pkiserv.envars` file to `PATH=/bin` instead of `PATH=/usr/sbin`.

Perform the following steps to test your sendmail configuration:

1. From the UNIX command line, create a mail file with some information in it. The following example is called `mail.txt`. (You need this name in the next step.)

   Example:

   ```
   To:target-email@address.com
   From:source-email@address.com
   Subject:This is a test
   ```

   ---

2. If sendmail is configured to run as a server on the local system, enter the command:

   ```
   /bin/sendmail -t <mail.txt
   ```

   Otherwise, enter the command:

   ```
   /usr/sbin/sendmail -t <mail.txt
   ```

Installing and configuring DB2

You need to perform this task only if you are implementing the PKI Services object store and issued certificate list (ICL) using DB2 tables for the first time.

For information about installing and configuring DB2, see the online information center for your release of DB2 at http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp or the DB2 library at http://www.ibm.com/support/docview.wss?rs=64&uid=swg27011656.

You do not need to install DB2 if you implement the PKI Services object store and issued certificate list (ICL) using VSAM data sets.

For information about implementing the PKI Services object store and ICL, see Chapter 9, “Creating the object store and ICL,” on page 103.

Tasks to perform before configuring PKITP

If you plan to use the PKI Services Trust Policy (PKITP), your team first needs to set up prerequisite software products by completing the following tasks, if not already done:

1. Installing and configuring OCSF
2. Installing and configuring OCEP

Installing and configuring OCSF

PKI Services Trust Policy (PKITP) requires OCSF to be installed and configured. If OCSF is not already installed and configured, follow the instructions for doing so in *z/OS Open Cryptographic Services Facility Application Programming*.

Before you begin:

1. Although the base feature of z/OS includes OCSF , if you are in the United States or Canada, make sure that you have ordered and installed the additional OCSF Security Level 3 feature. (There is no charge for this feature.)
2. You need OCSF programming skills to complete this procedure.

Installing and configuring OCEP

To install and configure OCEP, follow the instructions in *z/OS Integrated Security Services Open Cryptographic Enhanced Plug-ins Application Programming*. Then follow the instructions in "Configuring and getting started with PKITP" on page 429.

You need to perform this task only if you are setting up prerequisite products for PKITP for the first time.

Part 2. Configuring your system for PKI Services

After the MVS programmer installs PKI Services into the file system directory, your team needs to perform additional tasks to configure PKI Services, including the following tasks:

- Chapter 4, "Running IKYSETUP to perform RACF administration," on page 29 describes how the RACF administrator updates and runs IKYSETUP, a REXX exec to perform RACF administration tasks, such as setting up the daemon user ID and giving accesses.
- Chapter 5, "Configuring the UNIX runtime environment," on page 53 explains:
  - Copying files, such as the PKI Services configuration file
  - Updating environment variables
  - Updating the PKI Services configuration file
  - Setting up the `/var/pkiserv` file system directory.
- Chapter 6, "Tailoring the LDAP configuration for PKI Services," on page 87 explains how to update your LDAP configuration (performed earlier, see "Installing and configuring LDAP" on page 21) for PKI Services.
- Chapter 7, "Updating z/OS HTTP Server configuration and starting the server," on page 91 describes updating the z/OS HTTP Server configuration files and starting the z/OS HTTP Server.
- Chapter 8, "Tailoring the PKI Services configuration file for LDAP," on page 97 explains how to update the **LDAP** section of the PKI Services configuration file.
- Chapter 9, "Creating the object store and ICL," on page 103 explains how to create the VSAM data sets or DB2 objects for the PKI Services object store and issued certificate list (ICL).
- Chapter 10, "Starting and stopping PKI Services," on page 117 explains how to start and stop the PKI Services daemon.

© Copyright IBM Corp. 2001, 2011

Chapter 4. Running IKYSETUP to perform RACF administration

You need to perform this task if you are configuring PKI Services for the first time or adding a new CA domain.

PKI Services provides SYS1.SAMPLIB(IKYSETUP), a REXX exec, to perform RACF administration tasks for setting up PKI Services. The RACF administrator updates and runs this REXX exec, which issues RACF commands to perform the following tasks:

- Adding groups and user IDs
  - Setting up the PKI Services administration group
  - Creating the PKI Services daemon user ID
  - Giving appropriate access to the RACF group
  - Creating the surrogate user ID and giving the surrogate user ID authority to generate certificates

    A surrogate user ID is the identity assigned to client processes when they are requesting certificate services. A surrogate user ID is required for external clients. **Guideline:** For simplicity, use surrogate user IDs for internal clients as well, rather than allowing them to access PKI Services under their own identities.
  - Associating the PKI Services daemon user ID with the PKI Services started procedure.
- Setting up access control to protect end-user and administrative functions of PKI Services:
  - Authorizing the PKI Services daemon user ID for CA functions
  - Authorizing the PKI Services daemon user ID to access the Resource Recovery Services access facility (RRSAF), if you use DB2 as the repository for the object store and ICL
  - Giving administrators access to VSAM data sets, if you use VSAM as the repository for the object store and ICL
  - Optionally authorizing PKI Services for ICSF resources.
- Creating certificate authority (CA), registration authority (RA), and SSL certificates:
  - Creating a CA certificate and private key
  - Backing them up to a password-protected MVS data set
  - Optionally migrating the private key to ICSF
  - Optionally creating an RA certificate and private key for Simple Certificate Enrollment Protocol (SCEP)
  - Creating a SAF key ring and associating it with the certificate
  - Exporting the CA certificate to an MVS data set and file system file
  - Generating a server certificate signed by the new CA
  - Creating a key ring for the Web server
  - Associating the Web server and any trusted CA certificates to the key ring.
- Setting up the z/OS HTTP Server for surrogate operation.
- Allowing PKI Services to generate key pairs for certificate requests

© Copyright IBM Corp. 2001, 2011

Overview of IKYSETUP

IKYSETUP consists of several parts:

- Configurable section—This section assigns values to variables.
- A section that issues RACF commands to perform RACF administration tasks. (See "Actions IKYSETUP performs by issuing RACF commands" on page 511 for details about the actions that various sections of code perform.)
- A section that writes information (such as the name of the PKI Services administration group) to the log data set. The log itself consists of two parts: commands issued and other information. (See "Sample IKYSETUP log data set" on page 50.)

  Note: By default, IKYSETUP creates the log. You can disable recording information to the log by changing the value of one of the variables in IKYSETUP (`log_dsn`) to null.

The configurable section contains three parts:

- Values you must change (by making them specific to your company, such as your company's name)
- Values you might change depending on how you want PKI Services set up (for example, whether your setup will include ICSF)
- Values you can optionally change (these defaults are acceptable without change, but you might want to change them to make them more specific to your company, for example the name of the PKI Services administration group, which by default is PKIGRP)

The following table illustrates the structure and divisions of IKYSETUP:

Table 9. IKYSETUP—Structure and divisions

Configurable section—assigns values to variables • Values you must change to customize (See Table 10 on page 32.) • Values you might change that are related to setup (See Table 16 on page 39.) • Values you can optionally change (See Table 17 on page 43.)
Issues RACF commands
Records information in the log data set

Before you begin

Important: Update and run IKYSETUP *only* if you have not done so previously for an earlier release (or if you are changing the value of one or more variables).

You need to collect the following documents:

- *z/OS Security Server RACF Command Language Reference*
- *z/OS Security Server RACF Security Administrator's Guide*
- *z/OS TSO/E REXX Reference*

The RACF administrator needs to decide the values of variables in IKYSETUP and to record these values for future reference. Review and update as necessary the following three variables tables. There are three tables because there are three categories of variables:

- Variables whose values you are *required* to change, such as ones containing your company name
- Variables whose values you might want to change, depending based on how you are setting up PKI Services
- Variables whose values you can optionally change.

There is some overlap between the three types of variables, for example, if you are already using the RACF sample Web application, PKISERV.

Guideline: If you are running IKYSETUP for the first time, at a minimum, you need to complete the following:

- Table 10 on page 32
- Table 14 on page 38
- The rows of Table 16 on page 39 concerning z/OS UNIX level security:
  - `unix_sec`
  - `bpx_userid.` and `pgmcntl_dsn.` (if z/OS level security is already set up)
- Review the default values in *all* the tables.

Several values described in this topic, particularly for variables in Table 10 on page 32 and Table 17 on page 43, can be qualified with a *ca_domain* value based on whether you implement multiple CA domains. For information about implementing multiple CA domains, see “Adding a new CA domain” on page 230.

Variables whose values must change

Fill in the blank lines in the rightmost column with your company's information (and cross out the defaults in these cells).

Table 10. IKYSETUP variables whose values must change

Variable name	Description	Referenced elsewhere	Default value and your company's information
`ca_dn`	The CA's distinguished name. (For a definition of distinguished name, see Table 8 on page 22.) If you already have your CA certificate and private key set up in RACF, set `ca_dn=""`, set `ca_label` (in the following row) to the value of your CA's label, and update `ca_expires` (in Table 17 on page 43) to reflect the expiration date of your CA certificate. If you do not already have your CA certificate and private key set up in RACF, cross out the default in the rightmost cell of this row and record the information for your company-specific information for distinguished name on the blank line.	The suffix of the PKI Services CA's distinguished name must match the LDAP suffix. (The LDAP suffix is in the LDAP server configuration file. See Table 8 on page 22 for a definition of suffix.) **Note:** However, do not specify a C('*value*') if it is not present in your LDAP suffix.	When you also set *ca_domain*: **OU('***ca_domain* **Human Resources Certificate Authority')** When you do not set *ca_domain*: **OU('Human Resources Certificate Authority')** **O('Your Company')** **C('Your Country 2 Letter Abbreviation')** _____
`ca_label`	The CA certificate label. If you already have your CA certificate and private key set up in RACF (and your CA certificate's label differs from the default), you need to set `ca_label` to your CA certificate's label.	No	When you also set *ca_domain*: ***ca_domain* Local PKI CA** When you do not set *ca_domain*: **Local PKI CA** (Replace the default if you already have your CA certificate and private key set up in RACF.) _____
`daemon_uid`	The z/OS UNIX user identifier (UID) associated with the PKI Services daemon user ID.	No	**554** _____
`pki_gid`	The z/OS UNIX group identifier (GID) for the PKI Services administration group.	No	**655** _____

Table 10. IKYSETUP variables whose values must change (continued)

Variable name	Description	Referenced elsewhere	Default value and your company's information
pkigroup_mem.	Members of the PKI administration group are responsible for administering PKI Services functions. **Guideline:** Assign PKI administration duties to only highly trusted individuals. pkigroup_mem. is a list in which pkigroup_mem.0 is the number of members in the list and the rest of the entries are their user IDs. You must change the pkigroup_mem.0 to at least 1, and change pkigroup_mem.1 through pkigroup_mem.*n* to the member user IDs.	No	**0** (default for pkigroup_mem.0, the number of member user IDs) _____ **Note:** You must change the default to at least 1. (Record the member IDs:) _____ _____ _____ _____ _____
ra_dn	The RA's distinguished name for use with Simple Certificate Enrollment Protocol (SCEP). (For a definition of distinguished name, see Table 8 on page 22.) This name should be similar but not identical to your CA's distinguished name. If you do not wish to have PKI Services operate with a separate RA certificate, set ra_dn="".	No	**CN('Registration Authority')** **OU('Human Resources Certificate Authority')** **O('Your Company')** **C('Your Country 2 Letter Abbreviation')** _____
ra_label	The certificate label of your RA certificate in RACF.	No	When you also set ca_domain:*ca_domain***Local PKI RA** When you do not set ca_domain: **Local PKI RA** _____
surrog_uid	The UID associated with the surrogate user ID.	No	**555** _____

Table 10. IKYSETUP variables whose values must change (continued)

Variable name	Description	Referenced elsewhere	Default value and your company's information
web_dn	Your Web server's distinguished name. (For a definition of distinguished name, see Table 8 on page 22.) **Notes:** 1. The RACF administrator copies the fully qualified domain name from an earlier table: Table 7 on page 21. 2. If you already have your Web server configured for SSL: • Set web_dn="" • Update the web_ring row (You need to connect your PKI Services CA certificate to your key ring. See the web_ring row for directions.)	The value of the Web server's common name (CN), which is your server's symbol IP address. For example, www.*YourCompany*.com must match your Web server's fully qualified domain name.	**CN('www.YourCompany.com')** **O('Your Company')** **L('Your City')** **SP('Your Full State or Province Name')** **C('Your Country 2 Letter Abbreviation')** _____
web_ring	The name of the Web server's SAF key ring. If your Web server is configured for SSL and you are using a RACF key ring, set web_ring to the value of the RACF key ring. If your Web server is configured for SSL and you are using gskkyman, set web_ring="" and see Using a gskkyman key database for your certificate store for additional directions.	httpd*.conf— KeyFile directive	**SSLring** _____

Variables whose values might change depending on setup

To help in completing the next table of variables (see Table 16 on page 39) fill out the decision tables in "Deciding the value of key_backup," "Deciding the value of key_type" on page 35, "Deciding the value of restrict_surrog" on page 37, "Deciding the value of unix_sec" on page 37, and "Deciding the value of db2_repos and db2_subsys" on page 38.

Deciding the value of key_backup

Use the following decision table to determine the value of key_backup in Table 16 on page 39. The key_backup variable determines whether the PKI Services CA certificate and private key should be backed up to an encrypted data set.

Table 11. Decision table for key_backup

If ...	Then ...	Notes
You want to back up your CA's certificate and private key to a passphrase encrypted data set ...	Do not change the default `key_backup=1`	When you use IKYSETUP, you need to enter a passphrase whose display is not inhibited—it appears on the screen in the clear. You cannot back up PCICC keys (`key_type=2`) and hardware ECC keys (`key_type=6` or `key_type=7`).
You do *not* want to back up your CA's certificate and private key to a passphrase encrypted data set ...	Set `key_backup=0`	—

Deciding the value of key_type

Use the following decision table to determine the value of `key_type` in Table 16 on page 39. The `key_type` variable determines whether you are using RSA, ICSF, PCICC, DSA, or ECC for private key protection.

By default, IKYSETUP does not use ICSF. **Guideline:** Do not change the default the first time you run IKYSETUP but change it before going into a production environment. (For information about installing and configuring ICSF, see "Installing and configuring ICSF (optional)" on page 23.)

Table 12. Decision table for key_type

If ...	Then	Notes
You want to use software cryptography and you want a key generated using the RSA algorithm ...	Do not change the default `key_type=0`	—
You want to use ICSF for private key protection but do not want the key generated by the PCI cryptographic coprocessor (PCICC) ...	Set `key_type=1`	Review and possibly change the following additional variables in Table 16 on page 39: • `csfkeys_profile` • `csfserv_profile` • `csfusers_grp`
You want to use ICSF for private key protection and you want the key generated by PCICC ...	Set `key_type=2`	PKI Services does not automatically back up the private key when you select the 2 value. Review and possibly change the following additional variables in Table 16 on page 39: • `csfkeys_profile` • `csfserv_profile` • `csfusers_grp`
You want to use software cryptography and you want a key generated using the DSA algorithm ...	Set `key_type=3`	The key cannot be saved in ICSF.
You want to use software cryptography and you want a key generated using the NIST ECC algorithm ...	Set `key_type=4`	The key cannot be saved in ICSF.
You want to use software cryptography and you want a key generated using the Brainpool (BP) ECC algorithm ...	Set `key_type=5`	The key cannot be saved in ICSF.
You want to use ICSF for private key protection and you want a key generated using the NIST ECC algorithm by the Crypto Express 3 cryptographic coprocessor ...	Set `key_type=6`	Review and possibly change the following additional variables in Table 16 on page 39: • `csfkeys_profile` • `csfserv_profile` • `csfusers_grp` PKI Services does not automatically back up the private key.
You want to use ICSF for private key protection and you want a key generated using the Brainpool (BP) ECC algorithm by the Crypto Express 3 cryptographic coprocessor ...	Set `key_type=7`	Review and possibly change the following additional variables in Table 16 on page 39: • `csfkeys_profile` • `csfserv_profile` • `csfusers_grp` PKI Services does not automatically back up the private key.

Deciding the value of restrict_surrog

Use the following decision table to determine the value of `restrict_surrog` in Table 16 on page 39. The `restrict_surrog` variable determines if the RESTRICTED attribute is assigned to the surrogate user ID. The RESTRICTED attribute limits the resources available to this user ID.

By default, IKYSETUP does not assign the RESTRICTED attribute to the surrogate user ID. **Guideline:** Do not change the default the first time you run IKYSETUP but change it before going into a production environment. For more information, see the topic about defining groups and users in *z/OS Security Server RACF Security Administrator's Guide*.

Table 13. Decision table for restrict_surrog

If ...	Then ...
You want to assign the RESTRICTED attribute to the surrogate user ID ...	Set `restrict_surrog=1`
You do *not* want to assign the RESTRICTED attribute to the surrogate user ID ...	Do not change the default `restrict_surrog=0`

Deciding the value of unix_sec

Use the following decision table to determine the value of `unix_sec` in Table 16 on page 39. The `unix_sec` variable determines whether you want to use z/OS UNIX security, which is a higher level of security. z/OS UNIX provides two levels of security:

UNIX level security
: This is a less stringent level of security than z/OS UNIX level security. It is for installations where system programmers have been granted superuser authority. Programs that run with superuser authority have daemon level authority and can issue MVS identity-changing services without entering a `_passwd()` for the target user ID. With this level of security, the BPX.DAEMON profile in the FACILITY class is not defined.

z/OS UNIX level security
: This is a higher level of security than z/OS UNIX level security. It lets your system exercise more control over superusers. With this level of security, the BPX.DAEMON profile in the FACILITY class is defined.

Table 14. Decision table for unix_sec

If ...	Then ...	Notes
You already have z/OS UNIX security set up ...	Set `unix_sec=1`	—
You do not have z/OS UNIX security set up and you do not want to set it up ...	Do not change the default of `unix_sec=0`	—
You do not have z/OS UNIX security set up and you want to set it up for the first time ...	Set `unix_sec=2`	1. For information about additional manual configuration, see the section about establishing z/OS UNIX security in the *z/OS UNIX System Services Planning*. 2. If you are setting `unix_sec=2`, you must update the following variables: • `bpx_userid.` • `pgmcntl_dsn.`

Deciding the value of db2_repos and db2_subsys

Use the following decision table to determine the value of `db2_repos` and `db2_subsys` in Table 16 on page 39. You have the option of using either VSAM data sets or DB2 tables as the repositories for the PKI Services object store and ICL. For information about these options, see Chapter 9, "Creating the object store and ICL," on page 103. If you use DB2, the PKI Services daemon needs authorization to the DB2 Resource Recovery Services Access Facility (RRSAF). If you use VSAM, the PKI Services daemon and administrator need authorization to the VSAM data sets.

Table 15. Decision table for db2_repos and db2_subsys

If ...	Then ...	Notes
You plan to use VSAM ...	Do not change the default of `db2_repos=0`. Do not change `db2_subsys`; it will be ignored.	The value of `db2_repos` should be consistent with the value of the `DBType` parameter in the `pkiserv.conf` configuration file.
You plan to use DB2 ...	Set `db2_repos=1`. Set `db2_subsys` to the name of the local DB2 subsystem.	The value of `db2_repos` should be consistent with the value of the `DBType` parameter in the `pkiserv.conf` configuration file. The value of `db2_subsys` should be the same as the value of the `DBSubsystem` parameter in the `pkiserv.conf` configuration file.

Table of IKYSETUP variables that you might want to change

Update Table 16 on page 39 based on your answers in the decision tables in "Deciding the value of key_backup" on page 34, "Deciding the value of key_type" on page 35, "Deciding the value of restrict_surrog" on page 37, "Deciding the value of unix_sec" on page 37, and "Deciding the value of db2_repos and db2_subsys." If you have decided to change any of the defaults in the rightmost column, cross out the defaults and enter your company's information:

Table 16. IKYSETUP variables you might want to change depending on setup

Variable name	Description	Referenced elsewhere	Default value or your company's information
`bpx_userid.`	A list of user IDs with daemon and server authority. The `bpx_userid.0` is the number of items in the list and the rest of the entries are the z/OS UNIX user IDs. (This is non-applicable if `unix_sec ¬=2`.)	No	**1** (default for number of items) **OMVSKERN**
`ca_keysize`	The size in bits of the certificate-authority's private key. • When `key_type=0` (RSA software-generated key) or `key_type=2` (RSA PCICC-generated key), the acceptable range is 512 to 4096. • When `key_type=1` (RSA software-generated key stored in ICSF) or `key_type=3` (DSA software-generated key), the acceptable range is 512 to 1024. • When `key_type=4` (NIST ECC software-generated key) or `key_type=6` (NIST ECC hardware-generated key), the acceptable values are 192, 224, 256, 384, and 521. • When `key_type=5` (Brainpool ECC software-generated key) or `key_type=7` (Brainpool ECC hardware-generated key), the acceptable values are 160, 192, 224, 256, 320, 384, and 512.	No	**1024**
`cryptoz_grp`	The name of a RACF group for users authorized to use PKCS #11 tokens. Applies only if you set `key_gen = 1`.	No	**""**
`csfkeys_profile`	A profile to protect the PKI Services key in ICSF. (This variable is non-applicable if `key_type` has the value 0, 3, 4, or 5.) If you do not want IKYSETUP to create the profile, set `csfkeys_profile=""`. **Note:** When RACF stores the private key in the PKDS, it generates the label as: `'IRR.DIGTCERT.CERTIFAUTH.`*`unique-time-stamp`*`'`	No	**IRR.DIGTCERT.CERTIFAUTH.***
`csfserv_profile`	A profile to protect ICSF services. (This is non-applicable if `key_type=0` or `key_type=3` and `key_gen=0`.)	No	**CSF***

Table 16. IKYSETUP variables you might want to change depending on setup (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
csfusers_grp	A group of authorized ICSF service users. (This is non-applicable if csfkeys_profile="" and csfserv_profile="" .)	No	
db2_repos	Specifies whether the repository for the PKI Services object store and ICL is DB2 or VSAM. **0** VSAM **1** DB2	pkiserv.conf – DBType parameter	**0** (VSAM)
db2_subsys	Specifies the name of the local DB2 subsystem for the object store and ICL repositories, if db2_repos is **1**. This is a name of 1 - 4 alphanumeric characters.	pkiserv.conf – DBSubsystem parameter	**DSN9**
key_backup	Specifies whether the PKI Services CA certificate and private key should be backed up to an encrypted data set. The value can be: • 1 (yes - the default) • 0 (no). **Note:** This value is ignored when the value of key_type is 2, 6, or 7. When you use IKYSETUP with key_backup=1, you need to enter a passphrase whose display is not inhibited - it appears on the screen in the clear.	No	**1** (yes)
key_gen	Specifies whether you want PKI Services to generate key pairs (public key and private key) for certificate requests, if asked to do so. Set to **1** to allow PKI Services to generate key pairs for certificate requests.	No	**0**

Table 16. IKYSETUP variables you might want to change depending on setup (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
key_type	Specifies whether PKI Services should use RSA, ICSF, PCICC, DSA, or ECC for private key operations. The value can be: • 0 (use software cryptography with the RSA algorithm - the default) • 1 (use ICSF but not PCICC) • 2 (use ICSF and PCICC) • 3 (use software cryptography with the DSA algorithm). • 4 (use software cryptography with the NIST ECC algorithm) • 5 (use software cryptography with the BP ECC algorithm) • 6 (use hardware cryptography with the NIST ECC algorithm) • 7 (use hardware cryptography with the Brainpool (BP) ECC algorithm) If you are changing key_type to 1, 2, 6, or 7, see also the csfkeys_profile, csfserv_profile, and csfusers_grp rows. **Guideline:** Do not change the default the first time you run IKYSETUP, but change it before going into a production environment.	If choosing option 1 or 2, ICSF must be configured for RSA (PKA) operations and running. If choosing option 4 or 5, the ICSF token data set (TKDS) must be set up and ICSF must be running. If choosing option 6 or 7, ICSF must be set up and running, and a Crypto Express 3 cryptographic coprocessor must be available and configured with the PKDS.	**0**
pgmcntl_dsn.	A list in which pgmcntl_dsn.0 is the number of items in the list and the rest of the entries are a list of load libraries to be program controlled. **Rule:** If you set unix_sec=2, you must update the list of data sets.	No	**8** (default for number of items) • '**CEE.SCEERUN**' • '**CBC.SCLBDLL**' • '**SYS1.SIEALNKE**' • '**SYS1.CSSLIB**' • '**TCPIP.SEZALOAD**' • '**SYS1.LINKLIB**' • '**CSF.SCSFMOD0**' • '**CSF.SCSFMOD1**'
restrict_surrog	Specifies whether the surrogate user ID should be marked restricted. The value can be: • 0 (no - the default) • 1 (yes) **Guideline:** Do not change the default the first time you run IKYSETUP, but change it before going into a production environment.	No	**0** (no)

Table 16. IKYSETUP variables you might want to change depending on setup (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
unix_sec	Specifies whether to set up z/OS UNIX level security. (See "Deciding the value of unix_sec" on page 37 for a definition of z/OS UNIX level security.) The value can be: • 0 (do not set up - the default) • 1 (is already set up) • 2 (add this level of security) **Rule:** If you are changing unix_sec to 1 or 2, you must update the bpx_userid. and pgmcntl_dsn. rows. **Guideline:** Do not set unix_sec=2 the first time you are running IKYSETUP.	For unix_sec=2, the names of the load libraries need to change.	**0** (no)

Variables you can optionally change

To help in filling out the next table of variables (see Table 17 on page 43, see "Specifying when the CA certificate and Web server certificates expire."

Specifying when the CA certificate and Web server certificates expire

By default, IKYSETUP creates a CA certificate that expires in 20 years and a Web server SSL certificate that expires in five years from the date when the IKYSETUP script is run. If these expiration times are compliant with your security guidelines, no changes are needed to the ca_exyears, ca_expires, web_exyears, and web_expires variables in Table 17 on page 43.

You can shorten or extend the lifetime of the CA certificate by altering the value of the ca_exyears variable. This variable specifies the lifetime of the CA certificate in years. The default value is 20. You can shorten or extend the lifetime of the Web server certificate by altering the value of the web_exyears variable. The default value is 5. These variables are listed in IKYSETUP in "Part 3", in the subsection titled "Method 1". Ensure that the value for web_exyears is less than the value of ca_exyears. If it is not, the Web server certificate might be added to the RACF database with the NOTRUST option.

If your security guidelines require that the CA certificate and Web server certificate expire at specific dates, you can set these expiration dates in IKYSETUP. Ensure that the Web server certificate expires before the date that the CA certificate expires. If you don't, the Web server certificate might be added to the RACF database with the NOTRUST option.

Steps for setting the expiration dates for the CA certificate and Web server certificate: Perform the following steps to set the expiration dates for the CA certificate and Web server certificate in IKYSETUP.

1. Comment out all instructions in "Part 3, Method 1" of IKYSETUP. These are the instructions that calculate expiration dates by adding the life spans

specified in the ca_exyears and web_exyears variables to the date that IKYSETUP runs. You need to disable these instructions in order to set the expiration dates to specific dates.

2. Remove the comment delimiters from the sample instructions listed in "Part 3, Method 2" of IKYSETUP. These are the instructions that set the expiration dates to specific date values. You must enable these instructions.
3. Change the value of the ca_expires variable to the date when the CA certificate should expire, in the format *yyyy/mm/dd*.
4. Change the value of the web_expires variable to the date when the Web server certificate should expire, in the format *yyyy/mm/dd*. Specify a date that is before the date specified by the ca_expires variable.
5. Save your changes.

When you are done, you have set the CA certificate expiration date and the Web server certificate expiration date that will take effect the next time that you run IKYSETUP.

Table of IKYSETUP variables you can optionally change

Review the values of the variables in Table 17 to determine if you want to change any of the defaults in the rightmost column. If you decide to change any value, cross out the default in the rightmost column and record your company's information.

Table 17. IKYSETUP variables you can optionally change

Variable name	Description	Referenced elsewhere	Default value or your company's information
backup_dsn	The data set that will contain a backup copy of the PKI Services certificate and private key.	No	When you also set *ca_domain*: '***daemon.ca_domain***.KEY.BACKUP.P12BIN' When you do not set *ca_domain*: '***daemon***.KEY.BACKUP.P12BIN' **Note:** The *daemon* refers to the daemon variable in this table.
ca_domain	The unique name for the CA when you establish multiple PKI Services CAs. If specified, the first eight characters must uniquely identify the CA. The characters of the CA_domain value are limited to the following character set: alphanumeric characters (a - z, A - Z, 0 - 9) and the hyphen (-). In addition, the first character must not be a number or hyphen.	No	"" **Guideline:** Do not change the default (null) value until you perform advanced customization. (See "Adding a new CA domain" on page 230.)

Table 17. IKYSETUP variables you can optionally change (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
ca_expires	The date that the PKI Services CA certificate expires. By default, IKYSETUP calculates the CA certificate expiration date based on the value of ca_exyears. For information about setting this variable, see "Specifying when the CA certificate and Web server certificates expire" on page 42.	No	**2030/01/01** The date format is *yyyy/mm/dd*.
ca_exyears	The life span of the PKI Services CA certificate, expressed in years. By default, IKYSETUP calculates the expiration date for the CA certificate by adding the number of years specified in ca_exyears to the date that IKYSETUP is run. For information about setting this variable, see "Specifying when the CA certificate and Web server certificates expire" on page 42.	No	20
ca_ring	The name of the PKI Services SAF key ring.	pkiserv.conf—**SAF** KeyRing value	When you also set *ca_domain*: **CAring.*ca_domain*** When you do not set *ca_domain*: **CAring**
cacert_dsn	The data set that will contain the PKI Services certificate to assist the backup process.	No	When you also set *ca_domain*: **'*daemon*.*ca_domain*.CACERT.DERBIN'** When you do not set *ca_domain*: **'*daemon*.CACERT.DERBIN'** **Note:** *daemon* refers to the daemon variable in this table.
daemon	The PKI Services daemon user ID. If you also set *ca_domain*, you can choose to assign a unique user ID to the daemon for each CA domain. **Example:** For a *ca_domain* called BankA, you might choose user ID PKISRVDA.	pkiserv.conf—**SAF** KeyRing value	**PKISRVD**

Table 17. IKYSETUP variables you can optionally change (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
export_dsn	The data set that will contain the Web server's root CA certificate for copying to file system.	No	When you also set *ca_domain*: '***daemon.ca_domain*.WEBROOT.DERBIN'** When you do not set *ca_domain*: '***daemon*.WEBROOT.DERBIN'** **Note:** *daemon* refers to the daemon variable in this table.
log_dsn	The log data set name.	No	When you also set *ca_domain*: '***your-id.ca_domain*.IKYSETUP.LOG'** When you do not set *ca_domain*: '***your-id*.IKYSETUP.LOG'** **Notes:** 1. The *your-id* refers to the RACF ID of the person running IKYSETUP. (You do not need to add this; MVS adds this for you.) 2. Changing the default is not suggested.
pkigroup	The PKI Services administration group. This is a RACF group containing the list of user IDs that are authorized to use PKI Services administration functions. If you also set *ca_domain*, you can choose to assign a unique group name to the administration group for each CA domain. **Example:** For a *ca_domain* called BankA, you might choose group name PKIGRPA.	No	**PKIGRP**
ra_backup_dsn	The data set that will contain a backup copy of the PKI Services RA certificate and private key. This name should be similar but not identical to the backup_dsn value.	No	When you also set *ca_domain*: '***daemon.ca_domain*.RAKEY.BACKUP.P12BIN'** When you do not set *ca_domain*: '***daemon*.RAKEY.BACKUP.P12BIN'** **Note:** The *daemon* refers to the daemon variable in this table.

Table 17. IKYSETUP variables you can optionally change (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
`signing_ca_label`	The label of the CA certificate that is the superior (signer) of the PKI Services CA. If specified, the value must match the label of an existing CERTAUTH certificate in RACF that has a private key. Use this value to create a CA hierarchy when you establish multiple PKI Services CAs.	No	""
`surrog`	The surrogate user ID for PKI Services. If you also set *ca_domain*, you can choose to assign a unique user ID as the surrogate user ID for each CA domain. **Example:** For a *ca_domain* called BankA, you might choose user ID PKISERVA. **Note:** This cannot be an existing user ID (because IKYSETUP creates the user ID with the NOPASSWORD attribute).	Surrogate user ID in `httpd*.conf`	**PKISERV**
`vsamhlq`	The high-level qualifier of the VSAM data sets for PKI Services. **Note:** The RACF administrator gets this information from the MVS programmer.	• **ObjectStore** `*DSN` values in `pkiserv.conf` • Data sets names in IKYCVSAM	Same as the `daemon` variable earlier in this table.
`web_expires`	The date that the Web server certificate expires. By default, IKYSETUP calculates the Web server certificate expiration date based on the value of `web_exyears`. For information about setting this variable, see "Specifying when the CA certificate and Web server certificates expire" on page 42.	No	**2015/01/01** The date format is *yyyy/mm/dd*.

Table 17. IKYSETUP variables you can optionally change (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
web_exyears	The life span of the Web server certificate, expressed in years. By default, IKYSETUP calculates the expiration date for the Web server certificate by adding the number of years specified in web_exyears to the date when IKYSETUP is run. For information about setting this variable, see "Specifying when the CA certificate and Web server certificates expire" on page 42.	No	5
web_label	The label for the Web server's certificate.	No	**SSL Cert**
webserver	The Web server's daemon user ID.	See Web server documentation.	**WEBSRV**

Steps for performing RACF tasks using IKYSETUP

Use the following directions to run IKYSETUP only if you have not done so for a previous release (or if you are changing values).

You can use the following directions to run IKYSETUP with minimal changes or to extensively customize it.

Guideline: If this is your first attempt to use IKYSETUP, change only the IKYSETUP variables in the section Things you must change. You can refine IKYSETUP later, after you are familiar with the process of updating and running it.

The following flowchart illustrates the iterative nature of the process of updating IKYSETUP:

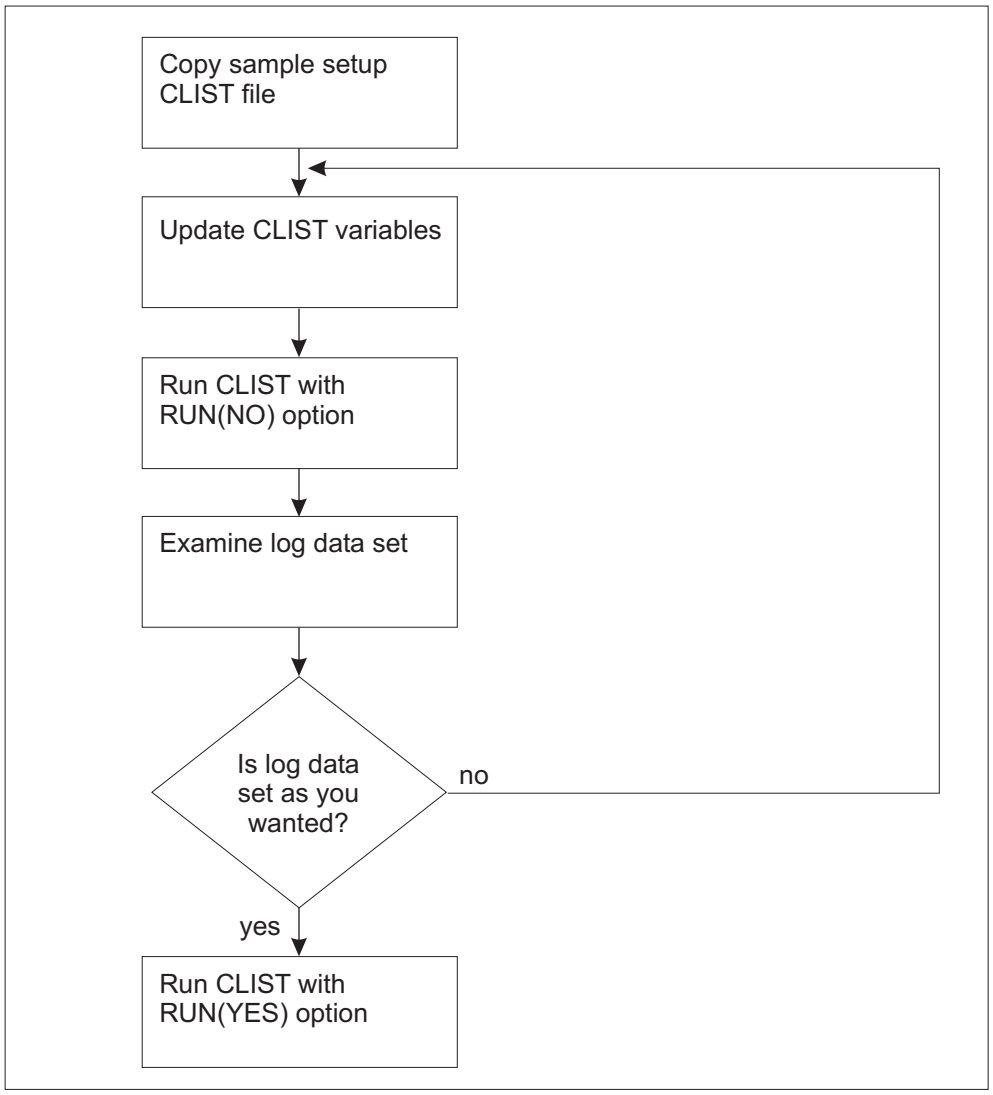


Figure 2. Flowchart of the process of updating IKYSETUP

Perform the following steps to use IKYSETUP to perform RACF administration tasks:

1. Copy `SYS1.SAMPLIB(IKYSETUP)` to a data set you are permitted to edit.

---

2. Edit the IKYSETUP code to update the values of variables you changed in Table 10 on page 32.

   The following example shows how to change the `pkigroup_mem.` variables. (Remember that for `pkigroup_mem.`, you set `pkigroup_mem.0` to the number of items in the list and `pkigroup_mem.1` through `pkigroup_mem.n` to the PKI Services administration group member IDs.)

   Example:

```
pkigroup_mem.0=3       /* Number of pkigroup members to connect */
pkigroup_mem.1="TOM"
pkigroup_mem.2="DICK"
pkigroup_mem.3="HARRY"
```

---

3. If necessary, update the values of variables you changed in Table 16 on page 39.

   The following example shows how to change the `key_type` variable.

   Example:

   ```
   key_type=1
   ```

---

4. Optionally update any variables you changed in Table 17 on page 43.

   The following example shows how to change the `log_dsn` variable.

   Example:

   ```
   log_dsn="PRIVATE.IKYSETUP.LOG"
   ```

   If you are changing the values of `ca_exyears`, `ca_expires`, `web_exyears`, or `web_expires`, see "Specifying when the CA certificate and Web server certificates expire" on page 42 for instructions.

---

5. Run IKYSETUP by entering the following command:

   ```
   EX 'data-set-name(IKYSETUP)' 'RUN(NO)'
   ```

   Notes:

   a. The user ID that runs IKYSETUP must be a RACF SPECIAL user ID.

   b. When IKYSETUP runs, it prompts you to enter your secret passphrase. (This is for encrypting the backup copy of your CA certificate and private key.) Be aware that asterisks do not replace the secret passphrase; it appears on the screen in the clear.

      Important: Make a note of this passphrase. If you forget it, your backup will be useless.

   c. The NO option in the command specifies displaying the commands only. (This creates a log data set listing the commands and other information. Alternative parameters are: YES, which indicates running IKYSETUP as is, and PROMPT, indicates prompting the user before running each command.)

---

6. Review the log data set. (See "Sample IKYSETUP log data set" on page 50 for an example of the data that appears on your display when you are running IKYSETUP; this is similar to the contents of the log data set.) The top part identifies the tasks and shows the commands that run to perform those tasks. Review this to ensure that the issued commands match your expectations. (For more information about these commands, see "Actions IKYSETUP performs by issuing RACF commands" on page 511.) The bottom part provides a record of important information that you will need for later steps, such the name of your daemon user ID. Review this information to ensure that the values are the ones you want.

   If you want to change any of the commands or information in the log data set, you need to change additional values in IKYSETUP. Remember to record any additional changes in Table 10 on page 32, Table 16 on page 39, and Table 17 on page 43. Then go back to Step 3.

---

7. If the log data set includes the commands and information you want, rerun the IKYSETUP code by entering the following command:

   ```
   EX 'data-set-name(IKYSETUP)' 'RUN(YES)'
   ```

8. After running IKYSETUP with RUN(YES), examine the results recorded in the log data set. Investigate and rerun (potentially by hand) any failing commands. Investigate informational messages and make any necessary corrections. (Informational messages usually indicate a setup problem that might affect operations later. For example, any informational message from the RACDCERT commands that indicate that the certificate has been marked NO TRUST is an error.)

9. If you intend to use encrypted LDAP passwords, you need to perform additional RACF administration tasks; see "Using encrypted passwords for LDAP servers" on page 420.

Sample IKYSETUP log data set

Here is an example of the data that appears when you run IKYSETUP.

```
Creating users and groups ...
ADDUSER  PKISRVD name('PKI Srvs Daemon')  nopassword  omvs(uid(554)
assize(256000000)  threads(512))
ADDUSER  PKISERV nopassword   omvs(uid(555))  name('PKI Srvs Surrogate')
ADDGROUP  PKIGRP OMVS(GID(655))

Granting DB2 RRSAF access to the PKI Services daemon.
RDEFINE DSNR DSN9.RRSAF UACC(NONE)
PERMIT DSN9.RRSAF CLASS(DSNR) ACCESS(READ) ID(PKISRVD)
SETROPTS CLASSACT(DSNR)

Creating the CA certificate ...
RACDCERT GENCERT CERTAUTH SUBJECTSDN(OU('Human Resources Certificate Authority')
O('Your Company') C('Your Country 2 Letter Abbreviation'))  WITHLABEL('Local PKI
CA') NOTAFTER(DATE(2020/01/01))  SIZE(1024)

Backing up the CA certificate ...
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA'))
DSN('PKISRVD.KEY.BACKUP.P12BIN') FORMAT(PKCS12DER)  PASSWORD('******')

Marking CA certificate as HIGHTRUST ...
RACDCERT CERTAUTH ALTER(LABEL('Local PKI CA')) HIGHTRUST

Saving the CA certificate to a data set ...
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA'))  DSN('PKISRVD.CACERT.DERBIN')
FORMAT(CERTDER)

Creating the RA certificate ...
RACDCERT ID(PKISRVD) GENCERT SUBJECTSDN(CN('Registration Authority') OU('Human
Resources Certificate Authority') O('Your Company') C('Your Country 2 Letter
Abbreviation'))  KEYUSAGE(HANDSHAKE) SIGNWITH(CERTAUTH LABEL('Local PKI CA'))
NOTAFTER(DATE(202
0/01/01)) WITHLABEL('Local PKI RA')

Backing up RA certificate ...
RACDCERT ID(PKISRVD) EXPORT(LABEL('Local PKI RA'))
DSN('PKISRVD.RAKEY.BACKUP.P12BIN') FORMAT(PKCS12DER)  PASSWORD('******')

Creating the PKI Services keyring ...
RACDCERT ADDRING(CAring) ID(PKISRVD)
RACDCERT ID(PKISRVD) CONNECT(CERTAUTH  LABEL('Local PKI CA')  RING(CAring)
USAGE(PERSONAL) DEFAULT)
RACDCERT ID(PKISRVD) CONNECT(LABEL('Local PKI RA')  RING(CAring)
USAGE(PERSONAL))

Creating the Webserver SSL certificate and keyring ...
RACDCERT GENCERT ID(WEBSRV) SIGNWITH(CERTAUTH  LABEL('Local PKI CA'))
WITHLABEL('SSL Cert') SUBJECTSDN(CN('www.YourCompany.com') O('Your Company')
L('Your City') SP('Your Full State or Province Name') C('Your Country 2 Letter
Abbreviation'))  NOTAFTER
(DATE(2020/01/01))
RACDCERT ADDRING(SSLring) ID(WEBSRV)
RACDCERT  ID(WEBSRV) CONNECT(ID(WEBSRV)  LABEL('SSL Cert') RING(SSLring)
USAGE(PERSONAL) DEFAULT)
```

```
RACDCERT  ID(WEBSRV) CONNECT(CERTAUTH  LABEL('Local PKI CA') RING(SSLring))

Saving the webserver's root CA certificate to a  data set for OPUT ...
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA'))  DSN('PKISRVD.WEBROOT.DERBIN')
FORMAT(CERTDER)

Giving PKISRVD access to BPX.SERVER ...
RDEFINE FACILITY BPX.SERVER
PERMIT BPX.SERVER CLASS(FACILITY)  ID(PKISRVD) ACCESS(READ)

Allowing the PKI Services daemon to act as a CA ...
RDEFINE FACILITY IRR.DIGTCERT.GENCERT
RDEFINE FACILITY IRR.DIGTCERT.LISTRING
RDEFINE FACILITY IRR.DIGTCERT.LIST
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY)  ID(PKISRVD) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)  ID(PKISRVD) ACCESS(READ)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)  ID(PKISRVD) ACCESS(READ)

Allowing the Webserver to access its keyring ...
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)  ID(WEBSRV) ACCESS(READ)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)  ID(WEBSRV) ACCESS(READ)

Allowing the Webserver to switch identity to PKISERV ...
SETROPTS CLASSACT(SURROGAT)
RDEFINE SURROGAT BPX.SRV.PKISERV
PERMIT BPX.SRV.PKISERV CLASS(SURROGAT)  ID(WEBSRV) ACCESS(READ)
SETROPTS RACLIST(SURROGAT) REFRESH

Allowing the PKI Services daemon to use ICSF ...
SETROPTS  GENERIC(CSFKEYS CSFSERV)
SETROPTS  GENERIC(CSFKEYS CSFSERV) REFRESH
RDEFINE CSFKEYS IRR.DIGTCERT.CERTIFAUTH.* UACC(NONE)
PERMIT IRR.DIGTCERT.CERTIFAUTH.* CLASS(CSFKEYS)  ID(PKISRVD) ACCESS(READ)
SETROPTS CLASSACT(CSFKEYS) RACLIST(CSFKEYS)
SETROPTS RACLIST(CSFKEYS) REFRESH
RDEFINE CSFSERV CSF* UACC(NONE)
PERMIT CSF* CLASS(CSFSERV)  ID(PKISRVD) ACCESS(READ)
PERMIT CSF* CLASS(CSFSERV)  ID(PKISERV) ACCESS(READ)
SETROPTS CLASSACT(CSFSERV) RACLIST(CSFSERV)
SETROPTS RACLIST(CSFSERV) REFRESH

Allowing the PKI Services daemon to  generate key pairs ...
SETROPTS CLASSACT(CRYPTOZ)
SETROPTS GENERIC(CRYPTOZ)
SETROPTS RACLIST(CRYPTOZ)
RDEFINE CRYPTOZ SO.PKISRVD.* UACC(NONE)
RDEFINE CRYPTOZ USER.PKISRVD.* UACC(NONE)
PERMIT SO.PKISRVD.* CLASS(CRYPTOZ)  ID(PKISRVD) ACCESS(UPDATE)
PERMIT USER.PKISRVD.* CLASS(CRYPTOZ)  ID(PKISRVD) ACCESS(CONTROL)
SETROPTS RACLIST(CRYPTOZ) REFRESH
SETROPTS GENERIC(CRYPTOZ) REFRESH

Creating the STARTED class profile for the daemon ...
RDEFINE STARTED PKISERVD.* STDATA(USER(PKISRVD))
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
SETROPTS RACLIST(STARTED) REFRESH

Allowing PKISERV to request certificate functions ...
SETR GENERIC(FACILITY)
RDEFINE FACILITY IRR.RPKISERV.**
PERMIT IRR.RPKISERV.** CLASS(FACILITY)  ID(PKISERV) ACCESS(CONTROL)

Creating the profile to protect PKI Admin functions ...
RDEFINE FACILITY IRR.RPKISERV.PKIADMIN
PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY)  ID(PKIGRP) ACCESS(UPDATE)
PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY)  ID(PKISERV) ACCESS(NONE)
SETROPTS RACLIST(FACILITY) REFRESH

-------------------------------------------------
Information needed for PKI Services UNIX set up:
-------------------------------------------------

The daemon user ID is:
  PKISRVD

The VSAM high level qualifier is:
  PKISRVD
This is needed for the [ObjectStore] section in pkiserv.conf

The PKI Services' DER encoded certificate is in data set:
```

```
  'PKISRVD.CACERT.DERBIN'

The webserver's DER encoded root
CA certificate is in data set:
  'PKISRVD.WEBROOT.DERBIN'
This must be OPUT to /var/pkiserv/cacert.der with  the BINARY option

The fully qualified PKI Services' SAF keyring is:
  PKISRVD/CAring
This is needed for the [SAF] section in pkiserv.conf

The label of the PKI Services' RA certificate is:
  Local PKI RA
This is needed for the [SAF] section in pkiserv.conf


The label of the PKI Services' RA certificate is:
  Local PKI RA
This is needed for the [SAF] section in pkiserv.conf

The PKI Services CA DN is:
  OU=Human Resources Certificate Authority,O=Your Company,C=Your Country 2
Letter Abbreviation
The suffix must match the LDAP suffix in slapd.conf

The PKI Services RA DN is:
  CN=Registration Authority,OU=Human Resources Certificate Authority,O=Your
Company,C=Your Country 2 Letter Abbreviation
The suffix must match the LDAP suffix in slapd.conf

The recommended location for the pkiserv.conf and pkiserv.tmpl is:
  /etc/pkiserv

Set the following environment variables in pkiserv.envars:
  _PKISERV_CONFIG_PATH=/etc/pkiserv

Set the following environment variable in your httpd envvars files:
  _PKISERV_CONFIG_PATH=/etc/pkiserv

The webserver's SAF keyring is:
  SSLring
This is needed for the KeyFile directive in httpd*.conf files

The Webserver's DN is:
  CN=www.YourCompany.com,O=Your Company,L=Your City,ST=Your Full State or
Province Name,C=Your Country 2 Letter Abbreviation
The left most RDN must be the webserver's fully qualified domain name
```

Chapter 5. Configuring the UNIX runtime environment

You need to perform all of the tasks in this topic if you are configuring PKI Services for the first time. If you have already configured PKI Services for an earlier release, you might need to perform some of the tasks in this topic if you are:

- Using a sysplex for PKI Services daemons
- Adding an additional CA domain
- Planning to use a PKI Services utility
- Enabling Simple Certificate Enrollment Protocol (SCEP)
- Sending e-mail notification for certificates ready for retrieval or expiration or rejected certificate requests
- Sending e-mail notification to the PKI Services administrator for certificate approval requests
- Automatically renewing certificates that are about to expire
- Customizing certificate revocation list (CRL) distribution point processing
- Allowing certificate requestors to ask PKI Services to create public and private key pairs

After the RACF administrator performs the tasks necessary to set up PKI Services, the UNIX programmer needs to perform the following tasks:

- If necessary, copy files.
- If necessary, update the environment variables file.
- If necessary, update the configuration file.
- If configuring PKI Services for the first time or adding a new CA domain, set up the `/var/pkiserv` directory.

The following table summarizes information about copying and updating files. To view the contents of any of these files, see Chapter 29, "Other code samples," on page 545.

Table 18. Deciding which files to copy and change

File	Purpose	Need to copy?	Need to change?
`expiringmsg.form`	The form for an e-mail sent to a user when a certificate is going to expire.	Only if your company sends an e-mail notification to a user about a certificate that is going to expire	**Guideline:** Make no changes to this file until later. See "Customizing e-mail notifications sent to users" on page 252 for details about making changes.
`pendingmsg.form`	The form for an e-mail sent to an administrator when requests are pending approval	Only if your company sends an e-mail notification to an administrator about requests that are pending approval	**Guideline:** Make no changes to this file until later. See "Customizing e-mail notifications sent to users" on page 252 for details about making changes.

© Copyright IBM Corp. 2001, 2011

Table 18. Deciding which files to copy and change (continued)

File	Purpose	Need to copy?	Need to change?
`pkiserv.conf`	Configuration file. Contains various settings and values PKI Services needs.	Only if you are configuring PKI Services for the first time.	The UNIX programmer might need to change the **LDAP** section of this file. **Guideline:** Do not change it now but change it later when you perform "Steps for tailoring the LDAP section of the configuration file" on page 98. The UNIX programmer needs to update the non-LDAP section of the `pkiserv.conf` configuration file if any of the following is true: • You intend to run multiple instances of PKI Services in a sysplex. • You intend to run multiple CA domains on a single z/OS image. (See "Adding a new CA domain" on page 230.) • You are customizing certificate revocation list (CRL) distribution point processing. • You are configuring PKI Services for the first time and do not intend to send e-mail notifications for certificate-related events. • You intend to have certificates that are about to expire be automatically renewed and sent to their owners. • You intend to have e-mail notifications sent to the PKI administrator when there are requests pending for approval. • You intend to allow certificate requestors to ask PKI Services to generate keys for certificates. • You intend to allow CMP clients to request, revoke, suspend, or resume certificates • You would like to change when and how often the PKI Services maintenance processing is done

Table 18. Deciding which files to copy and change (continued)

File	Purpose	Need to copy?	Need to change?
`pkiserv.envars`	The environment variables file.	Only if you are configuring PKI Services for the first time and the file needs changes.	UNIX programmer might have to update this file. See "Optionally updating PKI Services environment variables" on page 57.
`pkiserv.tmpl`	The certificate templates file used with REXX CGI execs. It contains HTML-style code that builds the Web pages underlying certificate requests.	Only if you are configuring PKI Services for the first time and using the REXX CGI execs to implement the PKI Services Web application.	**Guideline:** Make no changes to this file until later. See Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs," on page 123 for details about making changes.
`PKIServ.xsd`	The XML schema that defines the syntax of the XML certificate templates file `pkitmpl.xml`.	Only if you are configuring PKI Services for the first time and using the Java server pages (JSPs) to implement the PKI Services Web application.	**Rule:** Do not make changes to this file.
`pkitmpl.xml`	The certificate templates file used with Java server pages (JSPs). It defines applications and certificates in XML.	Only if you are configuring PKI Services for the first time and using the JSPs to implement the PKI Services Web application.	**Guideline:** Make no changes to this file until later. See Chapter 13, "Implementing the Web application using Java server pages," on page 179 for details about making changes.
`readymsg.form`	The form for an e-mail sent to a user when the PKI Services administrator has approved a certificate request and the certificate is ready for retrieval.	Only if your company sends an e-mail notification to a user after the PKI Services administrator has approved a certificate request and the certificate is ready for retrieval.	**Guideline:** Make no changes to this file until later. See "Customizing e-mail notifications sent to users" on page 252 for details about making changes.
`rejectmsg.form`	The form for an e-mail sent to a user when the PKI Services administrator has rejected a certificate request.	Only if your company sends an e-mail notification to a user after the PKI Services administrator has rejected a certificate request.	**Guideline:** Make no changes to this file until later. See "Customizing e-mail notifications sent to users" on page 252 for details about making changes.
`renewcertmsg.form`	The form for an e-mail sent to a user when PKI Services has automatically renewed an expiring certificate.	Only if your company enables automatic renewal of certificates.	**Guideline:** Make no changes to this file until later. See "Customizing e-mail notifications sent to users" on page 252 for details about making changes.
`recoverymsg.form`	The form for an e-mail sent to a user who has requested that PKI Services recover a certificate for which PKI Services generated the key pair.	Only if your company allows users to request that PKI Services generate key pairs for certificate requests.	**Guideline:** Make no changes to this file until later. See "Customizing e-mail notifications sent to users" on page 252 for details about making changes.

Steps for copying files

Before you begin:

- You need to obtain the following document:

  z/OS UNIX System Services Planning

- You need to know the file system directory where the MVS programmer installed PKI Services and the runtime directory, *install-dir* and *runtime-dir* in the commands that follow. The defaults are `/usr/lpp/pkiserv/` and `/etc/pkiserv` respectively. The MVS programmer was asked to record any changes to these defaults; see Table 3 on page 11.
- The user ID you use for copying files must have superuser authority.

Perform the following steps to copy the files:

1. If you are configuring PKI Services for the first time, copy the configuration file by entering the following command from the UNIX command line:

   ```
   cp -p /install-dir/samples/pkiserv.conf runtime-dir
   ```

---

2. If you are configuring PKI Services for the first time, copy the templates files. Do either Step 2a or Step 2b.
   a. If you are using REXX CGIs to implement the PKI Services Web application, copy the text template file by entering the following command from the UNIX command line:

      ```
      cp -p /install-dir/samples/pkiserv.tmpl runtime-dir
      ```

   b. If you are using Java server pages (JSPs) to implement the PKI Services Web application, copy the XML template file and the XML schema file by entering the following commands from the UNIX command line:

      ```
      cp -p /install-dir/samples/pkitmpl.xml runtime-dir
      cp -p /install-dir/samples/PKIServ.xsd runtime-dir
      ```

---

3. If your company is sending e-mail notifications to users (when certificate requests are rejected or when certificates are ready for retrieval or expiring), copy the appropriate notification files from the samples directory to the runtime directory by entering commands such as the following:

   ```
   cp -p /install-dir/samples/rejectmsg.form runtime-dir
   cp -p /install-dir/samples/readymsg.form runtime-dir
   cp -p /install-dir/samples/expiringmsg.form runtime-dir
   ```

---

4. If your company is sending e-mail notifications to administrators when certificate requests are pending approval, copy the `pendingmsg.form` notification file from the samples directory to the runtime directory by entering a command such as the following:

   ```
   cp -p /install-dir/samples/pendingmsg.form runtime-dir
   ```

---

5. If your company allows users to request that PKI Services create key pairs (private key and public key) for certificate requests, copy the notification file used when a user requests that PKI Services recover a certificate for which it created the keys by entering a command such as the following:

   ```
   cp -p /install-dir/samples/recoverymsg.form runtime-dir
   ```

---

6. If you are configuring PKI Services for the first time, examine the values in the environment variables file (by default, `pkiserv.envars`). If any values need to change, copy this file by entering the following command:

   ```
   cp -p /install-dir/samples/pkiserv.envars runtime-dir
   ```

Optionally updating PKI Services environment variables

You need to perform this task only if any one of the following conditions is true:

- You are configuring PKI Services for the first time.
- You are adding an additional CA domain.
- You want to send e-mail notifications (for rejected certificate requests or certificates that are ready for retrieval or expiring) and you did not use the default location for sendmail (`/usr/sbin/sendmail`).
- You intend to use automatic certificate renewal.
- You are implementing an **autorenew** exit.
- You intend to use Java server pages (JSPs) instead of REXX CGIs for the PKI Services Web pages.

You need to define certain environment variables (such as LIBPATH) for the PKI Services daemon to run. There are two files related to environment variables:

- A sample environment variables file, `pkiserv.envars` (by default in `/usr/lpp/pkiserv/samples/`)
- `SYS1.PROCLIB` member `PKISERVD` (You can use the ENVAR parameter to point to the environment variables file.)

You can use `pkiserv.envars` to set environment variables for the PKI Services daemon. This file contains most of the environment variables needed to run the daemon.

You need to change the file if you did not use the default for any of the following:

- The install directory for PKI Service (`/usr/lpp/pkiserv`)
- The message level
- The location for sendmail (`/usr/sbin/sendmail`)

Guideline: If you need to make changes to the `pkiserv.envars` file, copy the file another directory (such as `/etc/pkiserv`) and make changes only to the copy.

PKISERVD is the sample procedure to start PKI Services. (For sample code, see "PKISERVD sample procedure to start PKI Services daemon" on page 562.) PKISERVD sets the **TZ** (time zone) environment variable because it is very likely that the value of this variable needs to change. PKISERVD also includes parameters specifying the directory containing the environment variables file (`DIR`) and the file name of the environment variables file (`FN`). If you make a copy of `pkiserv.envars` as suggested, you also need to change the name of the directory in PKISERVD (for example, `DIR="/etc/pkiserv"`) and possibly the file name (for example, `FN="pki.env"`).

Note: You can change all of the following on the START command:

- Environment variables directory
- File name
- Job output class
- Region size
- Standard output
- Standard error
- Time zone

See "Steps for starting the PKI Services daemon" on page 117.

Because of the limitation of the number of characters allowed in the PARM=*operand* on the JCL EXEC card, take care to ensure that the total length of the environment variables directory and file name, **TZ** value, and `stdout` and `stderr` redirection values do not exceed the 100 character maximum.

You must specify any environment variables that PKI Services requires either in the PKISERVD procedure or in the environment variables file (`pkiserv.envars`). **Guideline:** Make your additions and changes to the environment variables file, rather than to the PKISERVD procedure.

(Optional) Steps for updating PKI Services environment variables

Before you begin: Examine the values in the environment variables file (by default, `pkiserv.envars`) and determine whether you need to update any values. (See "Environment variables in the environment variables file" on page 507 for a description of the environment variables and "The pkiserv.envars environment variables file" on page 510 for a code sample of the environment variables file.)

Perform the following steps to update PKI Services environment variables:

1. If you did not install sendmail in its default location (`/usr/sbin`), update the `PATH` environment variable to the location of sendmail. If you are running sendmail as a server, update the `PATH` environment variable to `/bin`.
2. Make any needed changes to PKISERVD, such as updating the path name of the environment variables file (FN and DIR parameters). (See "PKISERVD sample procedure to start PKI Services daemon" on page 562 for a code sample of the PKISERVD procedure.)

   _____
3. _PKISERV_VARDIR specifies the path name for a directory in which PKI Services will write persistent data. The maximum length of the path name is 256 characters, including the trailing /. The default value (if you do not set the environment variable) is `/var/pkiserv`. For example:

   ```
   _PKISERV_VARDIR=/var/mypkiserv/
   ```

   _____
4. If you do not define _PKISERV_EXIT or if it contains a null value, the PKI exit processing is disabled. If you have implemented an AUTORENEW exit, set _PKISERV_EXIT to the absolute path name of the exit program name. The maximum length is 256 characters including the program name. For example:

   ```
   _PKISERV_EXIT=/mydir/renewexit
   ```

   _____
5. Update any other values as necessary.

   _____

Optionally updating the pkiserv.conf configuration file

You need to update the `pkiserv.conf` configuration file if you meet any of the following conditions:

- You are configuring PKI Services for the first time
- You are adding support for:
  - Running a sysplex for PKI Services daemon

  - Sending e-mail notifications to users if the PKI Services administrator rejects certificate requests or certificates are ready for retrieval or expiring
  - Customizing certificate revocation list (CRL) distribution point processing. (See "Customizing distribution point CRLs" on page 217 for details.)
  - Automatic renewal of expiring certificates
  - Sending e-mail notifications to administrators if any requests are pending approval
  - A timeout value for the PKI Services exit.
  - Generation of key pairs (public and private key) for certificates
  - Setting the time that the daily maintenance task runs, or the days that it runs, or specifying that it is not to run when the PKI Services daemon starts
  - The certificate management protocol (CMP)
  - Using DB2 tables instead of VSAM files for the object store and ICL.
  - Creating CRLs without the Issuing Distribution Point extension.
- You installed a new release of z/OS and had configured PKI Services on the earlier release. (For more information see "Updating pkiserv.conf after installing a new release of z/OS" on page 83.

You can also optionally update the file if you want to change certain default values.

The `pkiserv.conf` configuration file for the PKI Services daemon consists of sections of name-value pairs. **Important:** Everything in the `pkiserv.conf` file, including section names, keys, and values, is case-sensitive.

Each section of the `pkiserv.conf` configuration file has a title enclosed in square brackets. The configuration file includes the following sections:

[OIDs]

The OIDs section specifies the object identifiers for various nicknames PKI Services uses internally. The OIDs are specified in the following form:

```
name=dotted-decimal
```

The following excerpt is from the OIDs section:

```
[OIDs]
.
.
.
MyPolicy=1.2.3.4
```

[ObjectStore]

The ObjectStore section specifies operational information for the object store and issued certificate list (ICL).

The following excerpt is from the ObjectStore section:

```
[ObjectStore]
ObjectDSN='pkisrvd.vsam.ost'
.
.
.
```

[CertPolicy]

The CertPolicy section is for CA policy information.

The following excerpt is from the CertPolicy section:

```
[CertPolicy]
SigAlg1=sha-256WithRSAEncryption
⋮
```

[General]

The General section is for general information.

The following excerpt is from the General section:

```
[General]
InitialThreadCount=10
⋮
```

[SAF]

The SAF section is for information about the SAF (RACF) key ring that is used for CA certificate and private key storage.

The following excerpt is from the SAF section:

```
[SAF]
KeyRing=PKISRVD/CAring
```

[LDAP]

The LDAP section contains information about the LDAP server for posting certificates and CRLs.

The following excerpt is from the LDAP section:

```
[LDAP]
NumServers=1
⋮
```

The UNIX programmer needs to update the **LDAP** section of this file. **Guideline:** Do not change it now but change it later when you perform "Steps for tailoring the LDAP section of the configuration file" on page 98.

(Optional) Steps for updating the configuration file

Before you begin: The following table provides information about parameters in the `pkiserv.conf` configuration file. (It omits parameters for the **LDAP** section. For information about these parameters, see Table 23 on page 98.) Read the parameter descriptions, and examine the values provided in the sample configuration file, shown in the rightmost column, to ensure that the values meet your company's requirements. As necessary, cross out the sample values and enter the information appropriate to your own organization's needs and policies.

Table 19. Information needed for updating the configuration file

Parameter	Information needed	Where to get this information	Sample value or your customized value
OIDs section			
`MyPolicy`	A registered Object ID identifying your organization's usage policy, for example: `1.2.3.4`	If you are creating your own certificate policy, see "Using certificate policies" on page 212 for information on creating certificate policies. Otherwise, do not change this information.	**1.2.3.4** If you need to use the CertificatePolicies extension, replace `1.2.3.4` with the value of your Object ID: _____
ObjectStore section			

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
DBType	Repository for the object store and issued certificate list (ICL). Valid values are: • VSAM • DB2 The default value is VSAM. If DBType is VSAM, specify values for the parameters ObjectDSN, ObjectTidDSN, ObjectStatusDSN, ObjectRequestorDSN, ICLDSN, ICLStatusDSN, and ICLRequestorDSN. If DBType is DB2, these parameters are ignored. If DBType is DB2, specify values for the parameters DBPackage and DBSubsystem. If DBType is VSAM, these parameters are ignored.	UNIX programmer decides this value.	**VSAM**
DBPackage	Name of the DB2 package this instance of PKI Services uses for the object store and ICL in the DB2 subsystem specified by the DBSubsystem parameter. If DBType is VSAM, this parameter is ignored.	DB2 programmer decides this value.	**MasterCA**
DBSubsystem	Name of the DB2 subsystem used by this instance of PKI Services. If DBType is VSAM, this parameter is ignored.	DB2 programmer decides this value.	**DSN9**
ObjectDSN	VSAM data set name for the object store base cluster. This is the request database. Each VSAM request record consists of a fixed header followed by a variable-length section. If DBType is DB2, this parameter is ignored.	For the high-level qualifier before the period, see the *vsamhlq* variable in Table 17 on page 43. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	**'pkisrvd.vsam.ost'** Note that this begins with the VSAM high-level qualifier.
	Guideline: If you are adding a new CA domain, insert the *ca_domain* value from Table 17 on page 43 as the second qualifier in the data set name. **Example:** 'pkisrvd.employee.vsam.ost'		

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
ObjectTidDSN	VSAM data set name for the object store transaction ID (TID) alternate index. If DBType is DB2, this parameter is ignored.	For the high-level qualifier before the period, see the *vsamhlq* variable in Table 17 on page 43. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	**'pkisrvd.vsam.ost.path'** Note that this begins with the VSAM high-level qualifier.
	Guideline: If you are adding a new CA domain, insert the *ca_domain* value from Table 17 on page 43 as the second qualifier in the data set name. **Example:** `'pkisrvd.employee.vsam.ost.path'`		
ObjectStatusDSN	VSAM data set name for the object store status alternate index. If DBType is DB2, this parameter is ignored.	For the high-level qualifier before the period, see the *vsamhlq* variable in Table 17 on page 43. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	**'pkisrvd.vsam.ost.status'** Note that this begins with the VSAM high-level qualifier.
	Guideline: If you are adding a new CA domain, insert the *ca_domain* value from Table 17 on page 43 as the second qualifier in the data set name. **Example:** `'pkisrvd.employee.vsam.ost.status'`		
ObjectRequestorDSN	VSAM data set name for the object store requestor alternate index. If DBType is DB2, this parameter is ignored.	For the high-level qualifier before the period, see the *vsamhlq* variable in Table 17 on page 43. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	**'pkisrvd.vsam.ost.requestr'** Note that this begins with the VSAM high-level qualifier.
	Guideline: If you are adding a new CA domain, insert the *ca_domain* value from Table 17 on page 43 as the second qualifier in the data set name. **Example:** `'pkisrvd.employee.vsam.ost.requestr'`		

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
ICLDSN	VSAM data set name for the ICL base cluster. This data set contains the certificates that have been issued. Each VSAM ICL record consists of a fixed header followed by a variable-length section containing the BER-encoded certificates. If DBType is DB2, this parameter is ignored.	For the high-level qualifier before the period, see the *vsamhlq* variable in Table 17 on page 43. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	**'pkisrvd.vsam.icl'** Note that this begins with the VSAM high-level qualifier.
	Guideline: If you are adding a new CA domain, insert the *ca_domain* value from Table 17 on page 43 as the second qualifier in the data set name. **Example:** `'pkisrvd.employee.vsam.icl'`		
ICLStatusDSN	VSAM data set name for ICL status alternate index. If DBType is DB2, this parameter is ignored.	For the high-level qualifier before the period, see the *vsamhlq* variable in Table 17 on page 43. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	**'pkisrvd.vsam.icl.status'** Note that this begins with the VSAM high-level qualifier.
	Guideline: If you are adding a new CA domain, insert the *ca_domain* value from Table 17 on page 43 as the second qualifier in the data set name. **Example:** `'pkisrvd.employee.vsam.icl.status'`		
ICLRequestorDSN	VSAM data set name for ICL requestor alternate index. If DBType is DB2, this parameter is ignored.	For the high-level qualifier before the period, see the *vsamhlq* variable in Table 17 on page 43. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	**'pkisrvd.vsam.icl.requestr'** Note that this begins with the VSAM high-level qualifier.
	Guideline: If you are adding a new CA domain, insert the *ca_domain* value from Table 17 on page 43 as the second qualifier in the data set name. **Example:** `'pkisrvd.employee.vsam.icl.requestr'`		

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`RemoveCompletedReqs`	Time period that completed certificate requests remain in the object store before automatic deletion.This is a number followed by `d` (days) or `w` (weeks). If not specified, the default is `1w` (one week). The value `0d` disables the deletion of completed requests (not suggested).	UNIX programmer decides this value.	**1w**
`RemoveInactiveReqs`	Time period that incomplete, inactive certificate requests remain in the object store before automatic deletion. This is a number followed by `d` (days) or `w` (weeks). If not specified, the default is `4w` (four weeks). The value `0d` disables the deletion of inactive requests (not suggested).	UNIX programmer decides this value.	**4w**
`RemoveExpiredCertsAndKeys`	Time period that keys and expired certificates with keys generated by PKI Services remain in the ICL and TKDS before automatic deletion. This is a number followed by `d` (days) or `w` (weeks). If you do not specify this parameter, or you set the value to `0d`, expired certificates will not be removed.	UNIX programmer decides this value.	**520w**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`RemoveExpiredCerts`	Time period that expired certificates with keys that were not generated by PKI Services remain in the ICL before automatic deletion. This is a number followed by `d` (days) or `w` (weeks). If you do not specify this parameter, or you set the value to `0d`, expired certificates will not be removed.	UNIX programmer decides this value.	**0d**
`SharedPLEX`	Indicates whether you intend to share a single copy of the PKI Services object store and the issued certificate list (ICL) among multiple images in a sysplex. This is `T` (True) or `F` (False). **Note:** This keyword has the same meaning as the `SharedVSAM` keyword in releases prior to z/OS V1R13. If the SharedVSAM parameter is present from an earlier release, it will continue to work. If both SharedVSAM and SharedPLEX are present, SharedPLEX takes precedence.	UNIX programmer decides this value.	**F**
CertPolicy section			
`AdminNotifyNew`*n*	The e-mail address to which notification should be sent immediately when a request is created and requires approval. The notification is only sent once. There can be multiple entries, where *n* is 1 for the first entry and increases sequentially for additional entries. The mailing address is in the form `<userid>@<system>`.	UNIX programmer decides this value. Do not change this information until you set up administrator notification of requests pending approval.	**abigail@***mycompany***.com**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`AdminNotifyReminder`*n*	The e-mail address to which reminder notifications of requests pending approval should be sent when the daily maintenance task runs. There can be multiple entries, where *n* is 1 for the first entry and increases sequentially for additional entries. The mailing address is in the form `<userid>@<system>`.	UNIX programmer decides this value. Do not change this information until you set up administrator notification of requests pending approval.	**abigail@***mycompany***.com**
`ARLDist`	Indicates whether an authority revocation list (ARL) distribution point will be created. `F` (the default) indicates no ARL distribution point will be created. `T` indicates that an ARL distribution point will be created if `CRLDistSize` is greater than zero.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Creating a distribution point ARL" on page 225 for more information.	**F**
`CertValidityConstraint`	Specifies whether the validity period of a certificate should be constrained within the CA's certificate life time. If `T` (True), requests with a validity period that exceeds the CA's validity period fail. If `F` (False), requests are not constrained to the CA's validity period. `F` is the default.	UNIX programmer decides this value.	**F**
`CPS1`	The Uniform Resource Identifier (URI) where your organization's Certification Practice Statement (CPS) is located. This is in the form: `http://www.`*`mycompany`*`.com/cps.html`	Do not change this information until you perform advanced customization. See "Using certificate policies" on page 212 for more information.	**http://www.***mycompany***.com/cps.html** If you changed `PolicyRequired=F` to `PolicyRequired=T`, you need to replace the variable *mycompany* with your own value for this: **http://www._____.com/cps.html**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`CreateInterval`	How often the certificate creation thread scans the database for approved requests. This is a number followed by `w` (weeks), `d` (days), `h` (hours), `m` (minutes), or `s` (seconds).	UNIX programmer decides this value.	**3m**
`CRLDistDirPath`	The full path for the file system directory where PKI Services is to save each DP CRL, as specified by the HTTP URI in the CRLDistributionPoints extension. This value is ignored if you do not create a CRLDistributionPoints extension or if the URI protocol is `ldap`. This value can be specified with or without the trailing slash. The default value is `/var/pkiserv/`.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Customizing distribution point CRLs" on page 217 for more information.	**/var/pkiserv/**
`CRLDistName`	Constant portion of the (leaf-node) relative distinguished name for a distribution point (DP) CRL, if DP CRL processing is being performed. The default value is `CRL`.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Customizing distribution point CRLs" on page 217 for more information.	**CRL**
`CRLDistSize`	An integer value that represents the maximum number of certificates that can appear on one DP CRL. If you do not specify this parameter, or you set the value to `0`, DP CRLs will not be created.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Customizing distribution point CRLs" on page 217 for more information.	**500**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`CRLDistURIn`	*Optional:* Specifies a URI format name for the DP CRL. You can specify multiple names using parameters `CRLDistURI1`, `CRLDistURI2`, and so forth. This value is ignored if you do not create DP CRLs by specifying `CRLDistSize` with a value greater than zero. Specify this only if you want a URI-format name, in addition to the distinguished name format, built in the CRLDistributionPoints extension.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Customizing distribution point CRLs" on page 217 for more information.	—
`CRLDuration`	The amount of time that a certificate revocation list is valid. This is a number followed by `w` (weeks), `d` (days), `h` (hours), `m` (minutes), or `s` (seconds).	UNIX programmer decides this value.	**2d**
`CRLIDPExt`	Specifies whether certificate revocation lists (CRLs) should be created with a critical Issuing Distribution Point (IDP) extension. If `T` (True), CRLs are created with a critical IDP extension. If `F` (False), CRLs are created without the IDP extension. The default is T.	UNIX programmer decides this value.	**T**
`EnableCMP`	Specifies whether support for certificate management protocol (CMP) messages is enabled. If `T` (True), CMP messages that are supported are accepted. If `F` (False), all CMP messages are rejected. `F` is the default.	UNIX programmer decides this value.	**F**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
EnableLargeCRLPosting	Specifies whether large CRL posting is enabled. If T, CRLs are saved in a z/OS UNIX directory before the LDAP posting thread processes them. If F, CRLs are saved in the object store (either VSAM data set or DB2 table, depending on which you are using), and are subject to a size limit of approximately 32KB. The default is F.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Enabling support for large CRLs" on page 223 for more information.	**F**
EnableSCEP	Specifies whether Simple Certificate Enrollment Protocol (SCEP) is allowed. This is T (True) or F (False).	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246 for more information.	**F**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`ExpireWarningTime`	**Note:** You need a value for this parameter only if you are sending e-mail notifications to users when certificates are expiring, or automatically renewing certificates when they are expiring and sending them to the owners. This parameter indicates how soon before certificate expiration to send a warning message or a renewed certificate (that is, the number of days or weeks before the day and time the certificate expires). If automatic certificate renewal is active, this parameter indicates how soon before certificate expiration to renew the certificate and send it to the owner. This name-value pair is optional. Its absence indicates no expiration checking is performed and no automatic certificate renewal occurs. Also, if the name-value pair is present but has an incorrect value or if PKI Services is configured to operate without LDAP, no expiration checking or automatic certificate renewal is done.	UNIX programmer decides this value.	**4w**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
LargeCRLPostPath	The full path for the file system directory where PKI Services is to save each CRL for posting to LDAP, if support for large CRLs is enabled. This value can be specified with or without the trailing slash, and can be the same as the value of CRLDistDirPath. The default value is /var/pkiserv/.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Enabling support for large CRLs" on page 223 for more information.	**/var/pkiserv/crls**
MaxSuspendDuration	The length of the certificate suspension grace period in weeks or days. This is a number followed by w (weeks) or d (days). Certificates that remain suspended for longer than this period are automatically revoked. If you do not specify this parameter, or you set it to 0d, the grace period is unlimited.	UNIX programmer decides this value.	**120d**
OCSPType	The type of OCSP responder support desired: • none (the default) • basic If you do not specify this parameter, or you set the value to none, the responder is not enabled.	Change to basic if you want to enable the responder.	**none**
PolicyCritical	Indicates whether the CertificatePolicies extension created on a global basis (if PolicyRequired=T was specified) should be marked critical. This is T (True) or F (False). This field is ignored if PolicyRequired=F is specified.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Using certificate policies" on page 212 for more information.	**F**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`PolicyRequired`	Indicates whether the CertificatePolicies extension should be created on a global basis. This is `T` (True) or `F` (False). `PolicyRequired=T` indicates that the CertificatePolicies extension will be created with the same value for all certificate templates based on the keywords specified in the **CertPolicy** section of the configuration file. Any policies specified through the CONSTANT subsection in the template file will be ignored. `PolicyRequired=F` indicates that the policies specified through the CONSTANT subsection, if any, will be used.	UNIX programmer decides this value. Do not change this information until you perform advanced customization. See "Using certificate policies" on page 212 for more information.	**F**
`PolicyName1`	The name of the policy. (This is the same policy name used with the MyPolicy parameter of the **OIDs** section.)	Do not change this information until you perform advanced customization. See "Using certificate policies" on page 212 for more information.	*MyPolicy* If you changed `PolicyRequired=F` to `PolicyRequired=T`, replace the name *MyPolicy* with the same policy name used in the **OIDs** section. _____
`Policy1Org`	This is the organization name for the CertificatePolicies extension. For example: `International Business Machines, Inc.`	Do not change this information until you perform advanced customization. See "Using certificate policies" on page 212 for more information.	**My Company, Inc.** If you changed `PolicyRequired=F` to `PolicyRequired=T`, you need to specify your own value for this: _____
`Policy1Notice1`	The first company notice number.	Do not change this information until you perform advanced customization. See "Using certificate policies" on page 212 for more information.	**1** If you changed `PolicyRequired=F` to `PolicyRequired=T`, you need to specify your own value for this: _____

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`Policy1Notice2`	The second company notice number.	Do not change this information until you perform advanced customization. See "Using certificate policies" on page 212 for more information.	**17** If you changed `PolicyRequired=F` to `PolicyRequired=T`, you need to specify your own value for this: _____
`SigAlg1`	The nickname assigned to the Object ID for the signature algorithm in the **OIDs** section. The supported algorithms and their nicknames are listed in Table 45 on page 216. **Guideline:** The MD2 and MD5 hashes have been found to be vulnerable to attack. Avoid specifying `md-5WithRSAEncryption` and `md-2WithRSAEncryption` if possible.	The supported algorithms and their nicknames are listed in Table 45 on page 216. Do not change this information until you perform advanced customization. See "Updating the signature algorithm" on page 215 for more information.	**sha-256WithRSAEncryption**
`TimeBetweenCRLs`	How often a certificate revocation list (CRL) should be created.This is a number followed by `w` (weeks), `d` (days), `h` (hours), `m` (minutes), or `s` (seconds). **Tip:** If you want PKI Services to create a CRL immediately, instead of waiting for the `TimeBetweenCRLs` interval to pass, use the `createcrls` utility. For more information, see "Using the createcrls utility" on page 358.	UNIX programmer decides this value.	**1d**
`UserNoticeText1`	A legal statement about certificate issuance and use. For example: `Certificate for IBM internal use only`	Do not change this information until you perform advanced customization. See "Using certificate policies" on page 212 for more information.	*statement* If you changed PolicyRequired=F to PolicyRequired=T, you need to replace the variable *statement* with your own value for this: _____ _____
General section			

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`ExitTimeout`	Length of time that PKI Services waits for the **autorenew** preprocessing and postprocessing exit to return. If not specified, PKI Services waits for at most 30 seconds. PKI Services cancels the exit program if it runs longer than the specified time. The maximum value allowed is 1 hour. Any time specified greater will be run for the maximum amount of time.	UNIX programmer decides this value.	**10s**
`InitialThreadCount`	Number of threads (at least 2 and no more than 100) the PKI Services daemon should create at program initialization.	UNIX programmer decides this value.	**10**
`MaintRunDays`	The days on which the daily maintenance task is to run. This is a list of digits between 0 and 6, representing the days of the week, with 0 representing Sunday, and 6 representing Saturday. The digits listed represent the days on which the task is to run. No spaces or other characters can be specified, and digits cannot be repeated. The digits can be specified in any order. If not specified, the task runs every day.	UNIX programmer decides this value.	**0123456**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`MaintRunTime`	The time (local time) at which the daily maintenance task is to run, in the format *hh:mm*, where *hh* represents the hour (00 to 23) and *mm* represents the minutes (00 to 59). `00:00` represents midnight. If not specified, the task runs once per day at midnight local time.	UNIX programmer decides this value.	**01:00**
`RunMaintAtStart`	Indicates whether the daily maintenance task should run during PKI Services startup, in addition to the time and days specified by the `MaintRunTime` and `MaintRunDays` parameters. The value `T` (True) indicates that the task should run during PKI Services startup. The value `F` (False) indicates that the task should not run during PKI Services startup. If not specified, the daily maintenance task runs during PKI Services startup.	UNIX programmer decides this value.	**T**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`ReadyMessageForm`	The full path name or data set name containing the 'Your certificate is ready' message form. • If you are not setting up PKI Services to generate keys for certificates, this name-value pair is optional. If you do not specify this name-value pair, no message is sent. • If you are setting up PKI Services to generate keys for certificates, this name-value pair is required. If you do not specify this name-value pair, requests to have PKI Services generate keys for certificates fail.	UNIX programmer decides this value.	**/etc/pkiserv/readymsg.form**
	Guideline: If you are adding a new CA domain, use the *ca_domain* value from Table 17 on page 43 as the second qualifier in the path name. **Example:** `/etc/pkiserv/employees/readymsg.form`		
`RejectMessageForm`	The full path name or data set name containing the 'Your certificate request has been rejected' message form. By default, no message is issued. Using this name-value pair is optional.	UNIX programmer decides this value.	**/etc/pkiserv/rejectmsg.form**
	Guideline: If you are adding a new CA domain, use the *ca_domain* value from Table 17 on page 43 as the second qualifier in the path name. **Example:** `/etc/pkiserv/employees/rejectmsg.form`		

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
ExpiringMessageForm	The full path name or data set name containing the 'Your certificate is about to expire' message form. By default, no message is issued. If your team has specified a value for ExpireWarningTime (see the ExpireWarningTime row in this table), then ExpiringMessageForm is required. Otherwise an error is logged and no expiring message processing is performed.	UNIX programmer decides this value.	**/etc/pkiserv/expiringmsg.form**
	Guideline: If you are adding a new CA domain, use the *ca_domain* value from Table 17 on page 43 as the second qualifier in the path name. **Example:** /etc/pkiserv/employees/expiringmsg.form		
AdminNotifyForm	The full path name or data set name containing the 'request(s) pending for approval' message form. Defaults to no notification sent.	UNIX programmer decides this value.	**AdminNotifyForm=/etc/pkiserv/pendingmsg.form**
	Guideline: If you are adding a new CA domain, use the *ca_domain* value from Table 17 on page 43 as the second qualifier in the path name. **Example:** /etc/pkiserv/employees/pendingmsg.form		
RenewCertForm	The full path name or data set name containing the 'renewed certificate'. Defaults to no certificate sent.	UNIX programmer decides this value.	**RenewCertForm=/etc/pkiserv/renewcertmsg.form**
	Guideline: If you are adding a new CA domain, use the *ca_domain* value from Table 17 on page 43 as the second qualifier in the path name. **Example:** /etc/pkiserv/employees/renewcertmsg.form		

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
`RecoverForm`	The full path name or data set name containing the 'list of certificates that satisfy your search criteria for recovery' message form. Use this name-value pair if you are setting up PKI Services to generate keys for certificate requests, and want users to be able to recover those certificates.	UNIX programmer decides this value.	**RecoverForm=/etc/pkiserv/recoverymsg.form**
	Guideline: If you are adding a new CA domain, use the *ca_domain* value from Table 17 on page 43 as the second qualifier in the path name. **Example:** `/etc/pkiserv/employees/recoverymsg.form`		
SAF section			
`KeyRing`	The fully qualified name of the SAF key ring for PKI Services to use. (This must consist of an uppercase user ID and a case-sensitive ring name separated by a slash (/.)	See the *ca_ring* and *daemon* values in Table 17 on page 43.	**PKISRVD/CAring**
`RA_label`	The label of your PKI Services registration authority (RA) certificate.	See the *ra_label* value in Table 10 on page 32.	**Local PKI RA**
`TokenName`	The name of a token in the ICSF PKCS #11 token data set (TKDS) that PKI Services uses to store key pairs that it generates for certificates. If this keyword is not specified, PKI Services cannot generate key pairs for certificates. If this keyword is specified, the TKDS must be set up before PKI Services starts. For information on setting up the TKDS, see *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*.	UNIX programmer decides this value. It must meet the requirements for a token name: • Up to 32 characters in length • Permitted characters are: – Alphanumeric – National: @ X'5B', # X'7B', or $ X'7C' – Period . X'4B' • The first character must be alphabetic or national • Lowercase letters can be used, but are folded to uppercase • The IBM1047 code page is assumed	**PKISRVD.PKIToken**

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
LDAP section			
For information about the **LDAP** section, see Table 23 on page 98.			

Notes:

1. Keep in mind that everything in the `pkiserv.conf` file, including section names, keys, and values, is case-sensitive.
2. For boolean values, any of the following values are accepted for True: `T`, `t`, `Y`, `y`, or `1`. Any of the following values are accepted for False: `F`, `f`, `N`, `n`, or `0`

Perform the following steps to update the `pkiserv.conf` configuration file:

1. If necessary, update the **ObjectStore** section:

   a. If you want to use DB2 tables for the object store and issued certificate list (ICL) instead of VSAM files, uncomment the following line by removing the "# " at the beginning of the line:

   ```
   # DBType=DB2
   ```

   and leave the following line commented out:

   ```
   # DBType=VSAM
   ```

   If you want to use VSAM files, leave both lines commented out, or uncomment the following line:

   ```
   # DBType=VSAM
   ```

   b. If `DBType` is set to `DB2` (see step 1a), uncomment the following lines, and if necessary change the DB2 package name and subsystem name in the following lines to the names you chose in the `DBPackage` and `DBSubsystem` rows in Table 19 on page 60:

   ```
   # DBPackage=MasterCA

   # DBSubsystem=DSN9
   ```

   c. If `DBType` is set to `VSAM`, or commented out, or not present (see step 1a), if necessary change the data set names in the following lines to the names you chose in the `ObjectDSN`, `ObjectTidDSN`, `ObjectStatusDSN`, `ObjectRequestorDSN`, `ICLDSN`, `ICLStatusDSN`, and `ICLRequestorDSN` rows in Table 19 on page 60:

   ```
   ObjectDSN='pkisrvd.vsam.ost'
   ObjectTidDSN='pkisrvd.vsam.ost.path'
   ObjectStatusDSN='pkisrvd.vsam.ost.status'
   ObjectRequestorDSN='pkisrvd.vsam.ost.requestr
   ICLDSN='pkisrvd.vsam.icl'
   ICLStatusDSN='pkisrvd.vsam.icl.status'
   ICLRequestorDSN='pkisrvd.vsam.icl.requestr'
   ```

   If you are configuring PKI Services for the first time be aware that the high-level qualifier of the VSAM data set names must match the name of the RACF user ID assigned to the PKI Services daemon (by default, PKISRVD). If you change from the default to another user ID, you need to change the high-level qualifier in the `pkiserv.conf` configuration file as well. If the MVS programmer changes the data set names (see Step 2d on page 107), you must make equivalent changes in `pkiserv.conf`.

d. If necessary, change **1w** in the following line to the value in the RemoveCompletedReqs row in Table 19 on page 60:

```
RemoveCompletedReqs=1w
```

e. If necessary, change **4w** in the following line to the value in the RemoveInactiveReqs row in Table 19 on page 60:

```
RemoveInactiveReqs=4w
```

f. If necessary, uncomment the following line and, optionally, change **26w** to the value in the RemoveExpiredCerts row in Table 19 on page 60:

```
RemoveExpiredCerts=26w
```

g. If necessary, uncomment the following line and, optionally, change **520w** to the value in the RemoveExpiredCertsAndKeys row in Table 19 on page 60:

```
RemoveExpiredCertsAndKeys=520w
```

h. If necessary, update the SharedPLEX line:

- If you intend to use a sysplex and you are configuring PKI Services for the first time, change F in the following line to T:

  ```
  SharedPLEX=F
  ```

- If you are not using a sysplex (regardless of whether you are configuring PKI Services for the first time), you do not need to do anything.

---

2. If necessary, update the **CertPolicy** section.

   a. If necessary, change **3m** in the following line to the value in the CreateInterval row in Table 19 on page 60:

   ```
   CreateInterval=3m
   ```

   b. If necessary, update the ExpireWarningTime line(s):

   - If you are sending e-mail notifications and you are configuring PKI Services for the first time, if necessary change the value 4w in the following line to the value in the ExpireWarningTime row of Table 19 on page 60.

     ```
     ExpireWarningTime=4w
     ```

   - If you are not using e-mail notifications and you are configuring PKI Services for the first time, remove the ExpireWarningTime=4w line from the pkiserv.conf file.

   c. If necessary, change **1d** in the following line to the value in the TimeBetweenCRLs row in Table 19 on page 60:

   ```
   TimeBetweenCRLs=1d
   ```

   d. If necessary, change **2d** in the following line to the value in the CRLDuration row in Table 19 on page 60:

   ```
   CRLDuration=2d
   ```

   e. If necessary, change **F** in the following line to the value in the PolicyRequired row in Table 19 on page 60:

   ```
   PolicyRequired=F
   ```

   For more information on this parameter, see "Using certificate policies" on page 212.

f. If necessary, change **F** in the following line to the value in the PolicyCritical row in Table 19 on page 60:

```
PolicyCritical=F
```

For more information on this parameter, see "Using certificate policies" on page 212.

g. If necessary, change **120d** in the following line to the value in the MaxSuspendDuration row in Table 19 on page 60:

```
MaxSuspendDuration=120d
```

h. If necessary, establish distribution point (DP) certificate revocation lists (CRLs) and a DP authority revocation list (ARL). Follow the procedure shown in "Steps for customizing distribution point CRLs" on page 220 to determine the values for Table 19 on page 60.

i. If you wish to enable the OCSP responder, change OCSPType=none to OCSPType=basic.

j. If you want to allow certificate management protocol (CMP) clients to send requests to PKI Services, change EnableCMP=F to EnableCMP=T. For information about support for CMP, see Chapter 19, "Using the certificate management protocol (CMP) with PKI Services," on page 379.

k. If you want to enable support for CRLs larger than 32KB, change F in the following row to T and set LargeCRLPostPath to the full path of the file system directory where PKI Services is to save CRLs for posting to LDAP.

```
EnableLargeCRLPosting=F
```

For more information, see "Enabling support for large CRLs" on page 223.

l. If you want certificate revocation lists (CRLs) to be created without the Issuing Distribution Point (IDP) extension, uncomment the following line by removing the "#" character and change T to F:

```
#IDPExtCRL=T
```

m. If you wish to enable Simple Certificate Enrollment Protocol (SCEP), change EnableSCEP=F to EnableSCEP=T. (See "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.)

---

3. If necessary, update the **General** section:

a. If necessary, change 10 in the following line to the value in the InitialThreadCount row in Table 19 on page 60:

```
InitialThreadCount=10
```

b. If necessary, change 10 in the following line to the value in the ExitTimeout row in Table 19 on page 60:

```
ExitTimeout=30s
```

c. If you choose, you can customize the time at which the daily maintenance task runs, the days on which it runs, and whether it also runs when the PKI Services daemon starts. This task is named daily_Timer, and performs functions such as:

- Removing old and expired certificates
- Removing inactive and completed certificate requests from the object store

- Updating the low CRL distribution point based on expired certificates
- Processing certificate expiration notification warning messages and automatic certificate renewal messages

To specify that the daily maintenance task is to run at a time other than the default of midnight local time, remove the "#" from the following line and change `01:00` to the time that you want the task to run.

```
#MaintRunTime=01:00
```

To specify the days on which the daily maintenance task is to run, remove the "#" from the following line and change `0123456` to the list of digits representing the days on which you want the task to run. For example, specify `15` to run the task every Monday and Friday.

```
#MaintRunDays=0123456
```

To specify that the daily maintenance task is not to run when the PKI Services daemon starts, remove the "#" from the following line and change `T` to `F`.

```
#RunMaintAtStart=T
```

d. If necessary update the ReadyMessageForm, RejectMessageForm, ExpiringMessageForm, AdminNotifyForm, RenewCertForm, and RecoverForm lines:

- If you are sending e-mail notifications and you are configuring PKI Services for the first time, if necessary, change the values of the path name in the following lines to the corresponding values in Table 19 on page 60:

```
ReadyMessageForm=/etc/pkiserv/readymsg.form

RejectMessageForm=/etc/pkiserv/rejectmsg.form

ExpiringMessageForm=/etc/pkiserv/expiringmsg.form

AdminNotifyForm=/etc/pkiserv/pendingmsg.form

RenewCertForm=/etc/pkiserv/renewcertmsg.form
```

- If you are allowing PKI Services to generate key pairs for certificates, if necessary change the value of the path name in the following lines to the corresponding value in Table 19 on page 60:

```
ReadyMessageForm=/etc/pkiserv/readymsg.form

RecoverForm=/etc/pkiserv/recoverymsg.form
```

- If you are not sending e-mail notifications and you are configuring PKI Services for the first time, comment out the following lines in the `pkiserv.conf` configuration file by putting a "#" character in the first position of each line that does not already have a "#" character in it, as shown here:

```
# full pathname or data set name containing the 'your certificate request
# has been rejected' message form. Defaults to no message issued
# RejectMessageForm=/etc/pkiserv/rejectmsg.form

# full pathname or data set name containing the 'your certificate is about
# to expire' message form. Defaults to no message issued
# ExpiringMessageForm=/etc/pkiserv/expiringmsg.form

# full pathname or data set name containing the 'request(s) pending for
# approval' message form. Defaults to no notification sent
# AdminNotifyForm=/etc/pkiserv/pendingmsg.form
```

```
# full pathname or data set name containing the renewed certificate.
# Defaults to no certificate sent
# RenewCertForm=/etc/pkiserv/renewcertmsg.form
```

- If you are not sending e-mail notifications and you are not allowing PKI Services to generate keys for certificates, and you are configuring PKI Services for the first time, comment out the following lines in the `pkiserv.conf` configuration file, as shown here:

  ```
  # full pathname or data set name containing the 'your certificate is ready'
  # message form. Defaults to no message issued
  # ReadyMessageForm=/etc/pkiserv/readymsg.form
  ```

- If you are not allowing PKI Services to generate keys for certificates, and you are configuring PKI Services for the first time, comment out the following lines in the `pkiserv.conf` configuration file, as shown here:

  ```
  # full pathname or data set name containing information on certificate(s)
  # needed to be recovered.
  # RecoverForm=/etc/pkiserv/recoverymsg.form
  ```

- If you are allowing PKI Services to generate keys for certificates, and you are not configuring PKI Services for the first time, and you previously deleted the following lines in the `pkiserv.conf` configuration file, restore the following lines:

  ```
  # full pathname or data set name containing the 'your certificate is ready'
  # message form. Defaults to no message issued
  ReadyMessageForm=/etc/pkiserv/readymsg.form
  ```

---

4. If necessary, update the **SAF** section:

   a. If necessary, change **PKISRVD/CAring** in the following line to the value in the `KeyRing` row in Table 19 on page 60:

      ```
      KeyRing=PKISRVD/CAring
      ```

   b. If you specified `EnableSCEP=T` in Step 2m on page 81, change **Local PKI RA** in the following line to the value in the `ra_label` row in Table 10 on page 32:

      ```
      RALabel=Local PKI RA
      ```

   c. If you want PKI Services to be able to generate and store key pairs for certificate requests, and the ICSF programmer has set up the ICSF PKCS #11 token data set (TKDS), uncomment the following line and change **PKISRVD.PKIToken** to the value you chose in the `TokenName` row in Table 19 on page 60:

      ```
      TokenName=PKISRVD.PKIToken
      ```

---

5. Restart PKI Services. Your changes do not take effect until you do this. For information about starting PKI Services see Chapter 10, “Starting and stopping PKI Services,” on page 117.

Updating pkiserv.conf after installing a new release of z/OS

After you install a new release of z/OS, you have two versions of the `pkiserv.conf` file on your system:

- The version you used on the previous release of z/OS, which contains the changes you made to configure PKI Services.

- The sample version shipped with the new release of z/OS, which contains changes that IBM made to support the new function shipped in the release. This version does not contain any of your configuration changes.

You can continue to use the version from the previous release, but you might not be able to use the new function shipped in the new release until you merge the configuration changes you made in your version of the file with the updates that IBM made in the version shipped in the new release. One approach to merging the files is to compare your version of `pkiserv.conf` with the new sample version, located by default in the `/usr/lpp/pkiserv/samples/` directory. Update your version to match the changes made to the sample version. You should be able to cut and paste between the two versions of the file. To identify the lines that changed, you can refer to the sample of `pkiserv.conf` in Chapter 26, "The pkiserv.conf configuration file," on page 501, which marks the lines that were changed with a bar in the left margin. But be aware that the sample shown might not be identical to the sample shipped with PKI Services.

Steps for setting up the var directory

You need to perform this task only if you are configuring PKI Services for the first time or adding a new CA domain.

Before you begin: Replace the following default values (used in the command examples) with values appropriate for your configuration:

Default value	Your value
PKISRVD	Use your `daemon` value in Table 17 on page 43.
`'pkisrvd.webroot.derbin'`	Use your `export_dsn` value in Table 17 on page 43.
`/var/pkiserv`	**Guideline:** Use your `ca_domain` value from Table 17 on page 43 to qualify the directory location if you are adding a new CA domain. For example, `/var/pkiserv/employees`.

Perform the following steps to set up a UNIX directory and copy certain files that PKI Services needs into that directory:

1. Change ownership of the directory to the user ID of the PKI Services daemon by entering the following command from the UNIX command line:

   Example:

   ```
   chown PKISRVD /var/pkiserv
   ```

   ---

2. Copy the Web Server root certificate from its MVS data set to `cacert.der` in the `/var/pkiserv` directory by entering the following command from the UNIX command line:

   Example:

   ```
   cp "//'pkisrvd.webroot.derbin'" /var/pkiserv/cacert.der
   ```

   ---

3. Change the permission settings of the file by entering the following command from the UNIX command line:

   Example:

   ```
   chmod 644 /var/pkiserv/cacert.der
   ```

   ---

4. Change the ownership of the file by entering the following command from the UNIX command line:

   Example:

   ```
   chown pkisrvd /var/pkiserv/*
   ```

Chapter 6. Tailoring the LDAP configuration for PKI Services

If you are configuring PKI Services for the first time, the LDAP programmer needs to load the LDAP schema file.

If you intend to use a non-z/OS LDAP product, refer to the documentation for that product. See "LDAP directory server requirements" on page 567 for information about installing a non-z/OS LDAP.

If you are configuring PKI Services for the first time, the LDAP programmer needs to set up an LDAP access control list (ACL) to allow any user to read CRLs, and might also need to set up another LDAP ACL to allow the distinguished name used for LDAP binding to create certificates and CRLs. For more information, see "Setting up authorization to create and access CRLs and certificates" on page 88.

You can optionally set up a secure connection with the LDAP server. For more information, see "Establishing a secure connection with LDAP (optional)" on page 88.

Steps for loading schema.user.ldif

Before you begin:

- You will need LDAP programming skills to complete this procedure.
- Make sure that the LDAP server is started before beginning these steps. If you are unsure about this, see "Steps for installing and configuring LDAP" on page 21.
- You need to know the following information from LDAP installation. Copy the information into the following table from (completed) Table 8 on page 22:

Table 20. LDAP information you need for tailoring LDAP configuration

LDAP information	Explanation	Value
Administrator's distinguished name	This is the distinguished name to use for LDAP binding. (For a definition of distinguished name, see Table 8 on page 22. The LDAP administrator defines the administrator's distinguished name with the **adminDN** keyword in the LDAP server configuration file. For example, the value is "cn=Admin" in the following: `adminDN "cn=Admin"`	
Administrator password	This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example: • By specifying the password as a TDBM entry by using the **userPassword** attribute in the **ldif2tdbm** load utility • By using the **adminPW** keyword in the LDAP server configuration file (not suggested)	
LDAP fully qualified domain name and port	This is the IP address and port on which the LDAP server is listening. For example, for `ldap.widgets.com:389`, the fully qualified domain name is `ldap.widgets.com` and the port is `389`. See Table 7 on page 21 for a definition of fully qualified domain name. You can specify this address with or without the preceding string "ldap://" or "ldaps://".	

© Copyright IBM Corp. 2001, 2011

Table 20. LDAP information you need for tailoring LDAP configuration (continued)

LDAP information	Explanation	Value
Suffix	(For a definition of suffix, see Table 8 on page 22.) The suffix value is specified after the suffix keyword in the LDAP server configuration file. `suffix "o=your-company,c=your-country-abbreviation"`	

You need to load the `schema.user.ldif` file only if you are configuring PKI Services for the first time, whether you are using LDBM or TDBM. For more information, see the chapter on LDAP directory schema in *IBM Tivoli Directory Server Administration and Use for z/OS*.

If you are configuring PKI Services for the first time, issue the following command to load the schema. Replace *adminDN*and *passwd*with the adminDN and adminPW values from Table 20 on page 87.

```
ldapmodify -D adminDN -w passwd -f /usr/lpp/ldap/etc/schema.user.ldif
```

Setting up authorization to create and access CRLs and certificates

Certificate revocation lists (CRLs) in an LDAP directory have an attribute of **critical**, which allows only the LDAP administrator to read them. If you are configuring PKI Services for the first time, the LDAP programmer needs to set up an LDAP access control list (ACL) to allow users other than the LDAP administrator to read CRLs. If the ACL is not set up, only the LDAP administrator can retrieve CRLs from LDAP. Other users might get access violation messages if they attempt to retrieve a CRL from LDAP, and LDAP does not return the CRL.

In addition, if the distinguished name to be used for LDAP binding is not the LDAP administrator, the LDAP programmer needs to set up another LDAP ACL to allow that distinguished name to create CRLs and certificates. You define the distinguished name to be used for LDAP binding in the AuthName1 line of the `pkiserv.conf` file. For more information about the AuthName1 line, see Chapter 8, “Tailoring the PKI Services configuration file for LDAP,” on page 97.

For information about setting up LDAP ACLs, see the discussion of access control in *IBM Tivoli Directory Server Administration and Use for z/OS*.

Tips: When setting up an LDAP ACL for PKI Services, consider the following:

- You can use the **entryOwner** attribute to allow an application to read and write LDAP entries without having to use the LDAP administrator bind credentials.
- You can use a propagating ACL (the **aclPropagate** attribute is set to **TRUE**) to allow the defined ACL to cover new CRLs created by PKI Services.

Establishing a secure connection with LDAP (optional)

You can optionally set up a secure connection between PKI Services and the LDAP server to prevent the bind password from flowing in the clear. The secure connection uses the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, provided by z/OS Cryptographic Services System SSL services, to maintain an encrypted communications path between PKI Services and the LDAP server. For information on how to configure LDAP to use a secure connection, see the topic on using SSL/TLS protected communications in *IBM Tivoli Directory Server Administration and Use for z/OS*.

If you are using a secure connection with LDAP, the RACF administrator needs to add a certificate to the PKI Services key ring for validating the LDAP server:

- If the LDAP server you are using is using a self-signed certificate, add that self-signed certificate to the PKI Services key ring.
- If the LDAP server is using a certificate signed by a certificate authority (CA), add the certificate for the CA to the PKI Services key ring, if it is not already there. Use whatever means the CA provides to obtain the CA's certificate.

For the name of the PKI Services key ring, see Table 17 on page 43. The RACF administrator uses the RACF RACDCERT command to add a certificate to the key ring. For information about RACDCERT, see *z/OS Security Server RACF Command Language Reference*.

Chapter 7. Updating z/OS HTTP Server configuration and starting the server

You need to perform the tasks in this topic only if you are configuring PKI Services for the first time and are implementing the PKI Services Web application using the REXX CGI execs. If you are implementing the PKI Services Web application using Java server pages (JSPs), you can skip the tasks in this topic.

Starting the Web server requires having a configuration file for it. This topic describes how the Web server programmer performs the following tasks:

- Updating the z/OS HTTP Server configuration files by cutting and pasting directives from the PKI Services samples directory into them
- Starting the z/OS HTTP Server.

Before you begin:

- The z/OS HTTP Server must have already been configured.
- It would be helpful to have available a copy of *z/OS HTTP Server Planning, Installing, and Using*.

Steps for updating the z/OS HTTP Server configuration files

PKI Services uses two modes of SSL, and these two modes require running two instances of the z/OS HTTP Server. Although the two instances share a single server certificate and private key, they use two different configuration files.

- The first configuration file is your existing configuration file (created earlier, see "Steps for installing and configuring the z/OS HTTP Server to work with PKI Services " on page 20). It specifies port 80 for normal HTTP traffic and port 443 for the SSL traffic port.
- The second configuration file, `/etc/httpd1443.conf`, specifies SSL traffic only on port 1443, with client authentication. (If this file does not exist, you create it by copying the first file.)

The following table summarizes the configuration and usage of each Web server:

Table 21. Summary of configuration and usage of each Web server instance

Server instance	Protocol	SSL	Server authentication	Client authentication	Port number
First instance	HTTP	No	No	No	80
First instance	HTTPS	Yes	Yes	No	443
Second instance	HTTPS	Yes	Yes	Yes	1443

Before you begin:

- **Important:** You need to perform these steps only if you are configuring PKI Services for the first time.
- You need to know the file system install directory (the file system directory where the MVS programmer installed PKI Services), called *`install-dir`* in the

© Copyright IBM Corp. 2001, 2011

commands that follow. The default is /usr/lpp/pkiserv/. The MVS programmer was asked to record any changes to the defaults; see Table 3 on page 11.

- You need to know the following LDAP information. Record the information in the rightmost row of Table 22. Note that the default name of the LDAP server configuration file is ds.conf for the IBM Tivoli Directory Server for z/OS LDAP server.

Table 22. LDAP information you need for tailoring z/OS HTTP Server configuration

LDAP information	Explanation	Value
Administrator's distinguished name	This is the distinguished name to use for LDAP binding. (For a definition of distinguished name, see Table 8 on page 22.) The LDAP administrator defines the administrator's distinguished name with the **adminDN** keyword in the LDAP server configuration file. For example, the value is "cn=Admin" in the following: adminDN "cn=Admin"	
Administrator password	This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example: • By specifying the password as a TDBM entry by using the **userPassword** attribute in the **ldif2tdbm** load utility • By using the **adminPW** keyword in the LDAP server configuration file (not suggested)	
LDAP fully qualified domain name	This is the IP address on which the LDAP server is listening, for example, for ldap.widgets.com. See Table 7 on page 21 for a definition of fully qualified domain name.	
LDAP port	This is the port for LDAP, for example, 389 in ldap.widgets.com:389	

Perform the following steps to update the z/OS HTTP Server configuration files (if you are configuring PKI Services for the first time):

1. If the second configuration file does not yet exist, create it by copying the first configuration file with the following command:

   ```
   cp -p /etc/httpd.conf /etc/httpd1443.conf
   ```

   ---

2. Copy the first set of sample z/OS HTTP Server configuration directives (from the PKI Services samples directory, */install-dir*/samples/httpd.conf file) into the default configuration file, /etc/httpd.conf.

   Note: The *install-dir*, your file system installation directory, by default is /usr/lpp/pkiserv. The MVS programmer determines whether to change this default. (See Table 3 on page 11.)

   a. Copy the keyfile, sslmode, sslport, and normalmode directives as is, replacing any existing values.

   b. If your organization customized the value of web_ring (see Table 10 on page 32), change SSLring in the keyfile directive in the following line to the customized value:

      ```
      keyfile SSLring SAF
      ```

   c. Optionally, copy the userId directive as is, replacing any existing value.

      Guideline: Copy the userId directive (as shown in the following) into your file as is. However, if you already have a value in your file for this, you are not required to change it.

      ```
      UserId %%CLIENT%%
      ```

d. Copy the protection and protect directives after any protection and protect directives you already have. Do not change the order in which these directives appear.
e. Copy the redirect directives after any redirect directives you already have. Do not change the order in which these directives appear. Make sure that the redirect directives are placed before the pass and exec directives.
f. Copy the pass and exec directives before any pass and exec directives you already have.
g. Add the addtype directives to your list of addtypes if they don't already exist.
h. Change all instances of *server-domain-name* to your Web server's fully qualified domain name, for example, www.ibm.com. (For information about your Web server's fully qualified domain name, see Table 7 on page 21.)
i. Change all instances of *application-root* to your file system installation directory, which is `usr/lpp/pkiserv` by default.

Note: Your file system installation directory by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default. (See Table 3 on page 11.)

---

3. Copy the second set of z/OS HTTP Server configuration directives (from the PKI Services samples directory, `/`*`install-dir`*`/samples/httpd2.conf`) into the `/etc/httpd1443.conf` file.

Note: The *`install-dir`*, your file system installation directory, by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default. (See Table 3 on page 11.)

a. If you created this file by copying the first `httpd.conf` file, delete all existing protection, protect, redirect, pass, exec, and FastCGI directives.
b. Copy the `userId`, `keyfile`, `sslmode`, `sslport`, `sslclientauth`, `normalmode`, and `SSLX500CARoots` directives as is, replacing any existing values.
c. If your organization customized the value of `web_ring` (see Table 10 on page 32), change `SSLring` in the keyfile directive in the following line to the customized value:

```
keyfile SSLring SAF
```

d. Add the following directives after the SSLX500CARoots directive:
   - SSLX500Host
   - SSLX500Port
   - SSLX500UserID
   - SSLX500Password

   Replace the <> placeholders with the actual values from Table 22 on page 92.
e. Copy the protection and protect directives after any protection and protect directives you already have. Do not change the order in which these directives appear.
f. Copy the redirect directives after any redirect directives you already have. Do not change the order in which these directives appear.
g. Copy the exec directives before any pass and exec directives you already have.

h. Change all instances of *server-domain-name* to your Web server's fully qualified domain name, for example, www.ibm.com. (For information about your Web server's fully qualified domain name, see Table 7 on page 21.)

i. Change all instances of *application-root* to your file system installation directory.

Note: Your file system installation directory by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default. (See Table 3 on page 11.)

j. If you created `httpd1443.conf` by copying `httpd.conf`, optionally change the directories in `httpd1443.conf` for the report, log, and pid files.

Guideline: Perform this step to ensure the two servers are not using the same files at the same time.

1) Create a new directory for the `httpd1443` files by using the following command:

```
mkdir /etc/internet/logs1443
```

2) Assign ownership to WEBSRV with the following command:

```
chown websrv /etc/internet/logs1443
```

3) Edit the `*Log` directives in the new `httpd1443.conf` file to provide unique pathnames.

For example, if the first `httpd.conf` file has the following:

```
AccessLog     /etc/internet/logs/httpd-log
AgentLog      /etc/internet/logs/agent-log
RefererLog    /etc/internet/logs/referer-log
ErrorLog      /etc/internet/logs/httpd-errors
CgiErrorLog   /etc/internet/logs/cgi-errors
```

change the `httpd1443.conf *Logs` to the following:

```
AccessLog     /etc/internet/logs1443/httpd-log
AgentLog      /etc/internet/logs1443/agent-log
RefererLog    /etc/internet/logs1443/referer-log
ErrorLog      /etc/internet/logs1443/httpd-errors
CgiErrorLog   /etc/internet/logs1443/cgi-errors
```

---

4. If you plan to use an OCSP responder, SCEP, or the certificate management protocol (CMP) CGI programs, add `/usr/lpp/pkiserv/lib` to the LIBPATH environment variable in the file `httpd.envvars`.

Steps for starting the z/OS HTTP Server

Perform the following steps to start the z/OS HTTP Server (if you are configuring PKI Services for the first time):

1. Make sure that the LDAP server is started. (If you are unsure about this, see "Steps for installing and configuring LDAP" on page 21.)

---

2. Issue the following commands from the UNIX command line:

```
httpd
httpd -r /etc/httpd1443.conf
```

Alternately, if you are using the IMWEBSRV started procedure as shipped with the Web server, you can start the two instances by entering the following MVS console commands:

```
S IMWEBSRV
S IMWEBSRV,ICSPARM='-r /etc/httpd1443.conf'
```

Updating z/OS HTTP Server configuration and starting the server

Chapter 8. Tailoring the PKI Services configuration file for LDAP

You need to tailor the **LDAP** section of the `pkiserv.conf` configuration file only if you meet one of the following conditions:

- You are configuring PKI Services for the first time
- You intend to use encrypted passwords for your LDAP servers

Chapter 5, "Configuring the UNIX runtime environment," on page 53 describes tasks the UNIX programmer performs. The other team members perform additional tasks before the UNIX programmer updates the **LDAP** section of the `pkiserv.conf` configuration file (described in this topic) and starts the PKI Services daemon (described in Chapter 10, "Starting and stopping PKI Services," on page 117).

Excerpt of LDAP section

The following excerpt shows the **LDAP** section of the `pkiserv.conf` configuration file as it is shipped:

```
[LDAP]
NumServers=1
PostInterval=5m
Server1=myldapserver.mycompany.com:389
AuthName1=CN=root
AuthPwd1=root
CreateOUValue= Created by PKI Services
RetryMissingSuffix=T
# Name of the LDAPBIND Class profile containing the bind information for LDAP
# server 1. This key is optional. Used in place of keys Server1, AuthName1.
# and AuthPwd1
#BindProfile1=LOCALPKI.BINDINFO.LDAP1
```

You use the **LDAP** section of the `pkiserv.conf` file to provide information for one or more LDAP servers. The NumServers line specifies the number of servers.

Storing information for encrypted passwords for your LDAP servers

You store information about passwords for binding to LDAP directories in the `pkiserv.conf` configuration file. Passwords can be in clear text or encrypted. By default, the `pkiserv.conf` configuration file contains Server1, AuthName1, and AuthPwd1 parameters; these lines are for specifying your LDAP bind information, including passwords, in clear text: (For more than one LDAP server, you add additional lines, Server2, AuthName2, AuthPwd2, Server3, AuthName3, AuthPwd3, and so forth.) If you want to use encrypted passwords for your LDAP servers, you delete all these lines, uncomment (remove the #) from the BindProfile1 line at the bottom of the file, and correct the profile value specified if necessary. (See "Using encrypted passwords for LDAP servers" on page 420 for information about setting up this bind profile in RACF). For more than one LDAP server, you add additional lines: BindProfile2, BindProfile3, and so forth.

PKI Services performs the following processing when locating LDAP bind information:

© Copyright IBM Corp. 2001, 2011

1. The Server*n* line specifies the fully qualified domain and port of your LDAP server. If your file contains a Server*n* line, PKI Services looks for the matching AuthName*n* and AuthPwd*n* lines and uses these values.
2. The BindProfile*n* parameter specifies the name of the LDAPBIND class profile. If your file does not contain a Server*n* line but does contain a BindProfile*n* line, PKI Services looks for the bind information in the LDAPBIND class profile. (If Server*n* is present, PKI Services does not look for bind information in BindProfile*n*, even if the value in Server*n* is incorrect.)
3. If neither is present for a specific server, then PKI Services uses the default from IRR.PROXY.DEFAULTS in the FACILITY class.

Steps for tailoring the LDAP section of the configuration file

Before you begin:

- **Important:** You need to update the **LDAP** section of the `pkiserv.conf` configuration file only if you are configuring PKI Services for the first time or your company is using encrypted passwords for your LDAP servers.
- You will need UNIX programming skills to complete this procedure.
- Table 23 lists some parameters that are in the **LDAP** section of the `pkiserv.conf` configuration file. The rightmost column lists the default values. You need to change some of these values. Fill in the blank lines with your company's information (and cross out these defaults). If you decide to change any of the other defaults, cross out these values and record your company's information.

Table 23. Information needed for updating the LDAP section of the configuration file

Parameter	Information needed	Where to get this information	Default value and your company's information
`NumServers=`	The number of available LDAP servers. These are replicas that can post certificates and CRLs.	From LDAP programmer	**1**
`PostInterval=`	How often the posting thread scans the request database for certificates and CRLs to post to the LDAP server in weeks (`w`), days (`d`), hours (`h`), minutes (`m`), or seconds (`s`) if NumServers > 0.	UNIX programmer decides this. Specify a number followed by `h` (hours), `m` (minutes) or `s` (seconds). Example: `6m`	**5m**
	Notes: 1. If the post is unsuccessful for a given certificate, the post is retried at the next post interval. If the post continues to be unsuccessful after 3 attempts, the post frequency for this certificate is reduced to no more than once per hour. After 26 unsuccessful attempts, it is further reduced to no more than once per day. After 33 unsuccessful attempts, the post request for this certificate is deleted from the request database. 2. Certificates created when NumServers is set to 0 are not posted to LDAP. If the value of NumServers is changed later to enable posting, the new value applies to new certificates only.		

Table 23. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
`Server1=`	You use this parameter only if you are storing LDAP passwords in the clear. This parameter's value is the fully qualified domain name (domain name or IP address and port) for the first LDAP server. If you are using a Secure Sockets Layer (SSL) session, the fully qualified domain name should be preceded by `ldaps://`.	Copy this information from the earlier (completed) table, Table 8 on page 22.	*myldapserver.mycompany*.com:389 _____ **Note:** If the number of servers (the value in the row containing `NumServers=`) is greater than one, you need one value for *each* server.
`AuthName1=`	You use this parameter only if you are storing LDAP passwords in the clear. This parameter's value is the distinguished name to use for LDAP binding. (See Table 8 on page 22 for a definition of distinguished name.)	Copy this information from the earlier (completed) table, Table 8 on page 22.	*CN=root* **Notes:** • If the number of servers (the value in the row containing `NumServers=`) is greater than one, you need one value for *each* server. • The default name of the LDAP server configuration file is `ds.conf` for the IBM Tivoli Directory Server for z/OS LDAP server.

Table 23. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
`AuthPwd1=`	You use this parameter only if you are storing LDAP passwords in the clear. This parameter's value is the password to use for LDAP binding. The LDAP programmer sets this. **Note:** Include this parameter, Server1, and AuthName1 only if you are storing the LDAP password in the clear. Alternately, if you encrypting the password for an LDAP server, use the BindProfile1 parameter. Omitting BindProfile1 and Server1 specifies using the PROXY segment information from the IRR.PROXY.DEFAULTS profile in the FACILITY class. (For more information, see "Using encrypted passwords for LDAP servers" on page 420.)	Copy this information from the earlier (completed) table, Table 8 on page 22.	*root* _____ **Note:** If the number of servers (the value in the row containing `NumServers=`) is greater than one, you need one value for *each* server.
`CreateOUValue=`	Value to use for the OU attribute when creating LDAP entries under the objectclass organizationalUnit. (See Table 107 on page 567.) This is used only when no OU value is specified in the relative distinguished name.	UNIX programmer decides this (after consulting with LDAP programmer)	**Created by PKI Services**
`RetryMissingSuffix=`	True (T) or False (F) setting that indicates whether LDAP post requests should be retried later if the distinguished name suffix does not exist. When set to F, LDAP post requests that fail because of a missing suffix are discarded.	UNIX programmer decides this (after consulting with LDAP programmer)	**T**

Table 23. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
`BindProfile1=`	You use this parameter only if you intend to use an encrypted password for your LDAP server. This parameter's value is the name of the LDAPBIND class profile containing the bind information for the LDAP server. (For more information, see "Using encrypted passwords for LDAP servers" on page 420.)	Get the profile name from the RACF administrator who creates the profile. See "Using encrypted passwords for LDAP servers" on page 420 for more information.	LOCALPKI.BINDINFO.LDAP1 **Note:** If the number of servers (the value in the row containing `NumServers=`) is greater than one, you need one value for *each* server.

Perform the following steps to update the **LDAP** section of the `pkiserv.conf` configuration file (if you are configuring PKI Services for the first time or using encrypted passwords for your LDAP servers):

1. If necessary, change 1 (the default) in the following line to the number of available LDAP servers listed in Table 23 on page 98:

   ```
   NumServers=1
   ```

2. Optionally change `5m` in the following line to the posting interval in Table 23 on page 98:

   ```
   PostInterval=5m
   ```

3. If necessary, update the BindProfile1 line or the Server1, AuthName1, and AuthPwd1 lines:
   - If you intend to use an encrypted password for your LDAP server and you are configuring PKI Services for the first time, perform the following steps:
     a. If you are using an LDAPBIND class profile, remove the comment delimiter (#) from the start of the following line and change `LOCALPKI.BINDINFO.LDAP1` to the name of the LDAPBIND class profile. (See Step 3 on page 421).

        ```
        # BindProfile1=LOCALPKI.BINDINFO.LDAP1
        ```

     b. Delete the following three lines in the **LDAP** section:

        ```
        Server1=myldapserver.mycompany.com:389
        AuthName1=CN=root
        AuthPwd1=root
        ```

   - If you are not using an encrypted password for your LDAP server and are configuring PKI Services for the first time, perform the following steps:
     a. Change *your-ldap-server-address:port* to your fully qualified domain name and port as listed in Table 23 on page 98:

        ```
        Server1=your-ldap-server-address:port
        ```

     b. Change *CN=root* in the following line to the value of the administrator distinguished name in Table 23 on page 98:

```
AuthName1=CN=root
```

c. Change *root* in the following line to the value of the administrator password in Table 23 on page 98:

```
AuthPwd1=root
```

---

4. If the value of `NumServers=` is greater than 1, repeat Step 3 for each additional server. (You will need to increment the number in the parameter names for each additional server, for example `Server2`, `AuthName2`, `AuthPwd2`.

---

5. If necessary, change `Created by PKI Services` in the following line to the OU attribute value in Table 23 on page 98:

```
CreateOUValue=Created by PKI Services
```

---

6. If necessary, change `T` in the following line to the RetryMissingSuffix value in Table 23 on page 98:

```
RetryMissingSuffix=T
```

Chapter 9. Creating the object store and ICL

This topic includes the following procedures:

- Planning VSAM storage requirements
- (Optional) preliminary steps for establishing VSAM RLS
- Steps for creating the VSAM object store and ICL data sets and indexes
- (Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS
- (Optional) steps for adding VSAM buffer space
- "Backing up and restoring the VSAM data sets" on page 109
- Steps for creating the object store and ICL DB2 tables
- Converting the object store and ICL from VSAM to DB2

You need to perform the tasks in this topic if:

- You are configuring PKI Services for the first time or adding a new CA domain.
- Your organization is using a sysplex for PKI Services daemons.
- You wish to tune VSAM performance.
- Your object store and ICL are in VSAM data sets, and you want to convert them to DB2 tables

The object store and ICL

PKI Services maintains two databases containing information about certificate requests and issued certificates:

- The *object store*, or *request database*, holds records to track active certificate requests and posting objects for certificates and certificate revocation lists. Object store records are not permanent. They are deleted when they are no longer needed:
  - CRL posting requests and certificate posting requests are removed when they are successfully posted to LDAP.
  - Revocation requests are deleted at the end of revocation processing.
  - When a certificate is retrieved by the requestor, the certificate request is deleted after the time period specified by the parameter `RemovedCompletedReqs` in the configuration file (by default one week).
  - If a certificate is not retrieved by the requestor, the certificate request is deleted after the time period specified by the parameter `RemovedInactiveReqs` in the configuration file (by default four weeks).
- The *issued certificate list (ICL)* contains a permanent record for each certificate that PKI Services issues. There is one ICL record for each issued certificate.

You have two options for implementation of the object store and ICL:

- VSAM data sets.
- DB2 tables. (This option was introduced in z/OS V1R13).

Some things you should consider when choosing which option to use:

- VSAM is shipped with z/OS, but you must purchase DB2.
- VSAM supports limited query functions and does not allow users to create their own queries. DB2 allows users to create queries.

© Copyright IBM Corp. 2001, 2011

Certificate revocation lists (CRLs) created temporarily in the object store for LDAP posting are limited in size to approximately 32KB. You can avoid this limitation by enabling support for large CRLs, which causes CRLs to be stored in the z/OS UNIX file system instead of the object store. For more information, see "Enabling support for large CRLs" on page 223.

Creating the object store and ICL using VSAM data sets

The MVS programmer performs the following tasks:

- If configuring PKI Services for the first time, create the VSAM object store and ICL data sets and indexes.
- If configuring PKI Services to run in a sysplex as a single PKI instance, perform the preliminary steps for establishing VSAM record-level sharing (RLS).
- If desired, tune VSAM data set performance.

Planning VSAM storage requirements

The MVS programmer uses the IKYCVSAM sample JCL to create two VSAM data sets (clusters):

- A data set for the request database (object store)
- A data set for the issued certificate list (ICL).

The MVS programmer also uses the same sample JCL to create five alternate index data sets (paths):

1. Transaction ID alternate index into the object store
2. Status alternate index into the object store
3. Requestor alternate index into the object store
4. Status alternate index into the ICL
5. Requestor alternate index into the ICL.

The IKYCVSAM sample JCL contains default values for the primary and secondary extent allocations for these data sets. The default allocation for base clusters is `CYL(3,1)`. For alternate indexes, it is `TRK(5,1)`. You need to update these values based on your anticipated future needs. Use the guidelines in "Determining storage needs for the ICL" and "Determining storage needs for the object store" on page 105 to update the space allocation parameters for the DEFINE CLUSTER and DEFINE ALTERNATEINDEX statements. (For more information about IDCAMS, see *z/OS DFSMS Access Method Services for Catalogs*.)

Determining storage needs for the ICL

Unless set up otherwise, the ICL grows continuously over time as more certificates are issued. The size of a certificate varies depending on many factors; for example, the public key type and size, the length of the issuer and subject names, the number and size of Subject Alternative names, and the number and size of certificate extensions. Assume an average size of 1024 bytes of storage for each ICL record. With an allocation of `CYL(3,1)`, the data set can have a maximum of 125 cylinders on a single 3390 volume for a total size of 105 MB. This would mean the data set should be able to hold at least 102,500 certificates. If multiple volume support is used, you can double this amount.

If your anticipated needs differ greatly from the above value, you need to adjust the space allocation parameters `CYL(3,1)` on the DEFINE CLUSTER statement for the ICL. (This is the second DEFINE CLUSTER statement in IKYCVSAM. See IKYCVSAM for a code sample of this file.) You might also want to proportionally

adjust the space allocation parameters `TRK(5,1)` on the DEFINE ALTERNATEINDEX statements for the ICL. These are defined in the DEFALTDX job step. (Their names contain the `icl` qualifier.)

Determining storage needs for the object store

Object store records are not permanent. They are deleted when they are no longer needed. Unlike the ICL, the object store does not grow beyond a certain point, unless there is a sharp increase in certificate request activity. Typically, a certificate request record is less than twice the size of the ICL record. Assume that one object store record and its companion posting record occupy a total of 2560 bytes of storage. With a space allocation of `CYL(3,1)`, the data set can have a maximum of 125 cylinders on a single 3390 volume for a total size of 105 MB, which would hold at least 41,000 concurrent certificate requests. If multiple volume support is used, you can double this amount.

If your anticipated needs different greatly from the preceding value, you need to adjust the space allocation parameters `CYL(3,1)` on the DEFINE CLUSTER statement for the object store. (This is the first DEFINE CLUSTER statement in IKYCVSAM. See "IKYCVSAM" on page 551 for a code sample of this file.) You might also want to proportionally adjust the space allocation parameters `TRK(5,1)` on the DEFINE ALTERNATEINDEX statements for the object store. These are defined in the DEFALTDX job step. (Their names contain the `ost` qualifier.)

If you set the `RemoveInactiveReqs` and `RemoveCompletedReqs` parameters in the configuration file to `0d`, certificate request records are not deleted and you should adjust the space allocation accordingly.

(Optional) preliminary steps for establishing VSAM RLS

Your team can configure PKI Services to take advantage of a Parallel Sysplex environment. This enables you to start multiple instances of the PKI Services daemon (one per image) that work in unison. The daemons are totally independent of each other, but they all act upon a single common data store containing the ICL and object store VSAM data sets.

If you want to run multiple instances of PKI Services in a Parallel Sysplex (one per image), you must first establish the data sharing environment suitable for VSAM record-level sharing (RLS).

Before you begin: The following steps assume that the coupling facility has already been set up. If this is not the case, for information on how to set up the coupling facility, see *z/OS MVS Programming: Sysplex Services Guide*. Also see "(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS" on page 107 for additional information about setting up VSAM data sets to run PKI Services in a sysplex.

Perform the following steps to establish VSAM RLS. For specific information on how to perform these steps, see the topic about administering VSAM record level sharing in *z/OS DFSMSdfp Storage Administration*.

1. Define and activate at least two sharing control data sets (SHCDS) and one spare SHCDS for recovery purposes.

   _____

2. Define CF lock structure to MVS.

   _____

3. Define CF lock structure in the SMS base configuration.

4. Define at least one storage class for VSAM record-level sharing (RLS).

   You must record the name of this storage class for use in creating the VSAM data sets for PKI Services.

   Table 24. VSAM RLS information you need to record

VSAM information you need to record	Your value
Name of storage class for VSAM RLS	

Steps for creating the VSAM object store and ICL data sets and indexes

You need to perform this task only if you are configuring PKI Services for the first time.

PKI Services uses VSAM data sets to store requests in progress and issued certificates. You need to create these data sets manually.

If you expect the object store or ICL data sets to exceed 4GB in size, consider using VSAM extended addressability. For more information, see *z/OS DFSMS Using Data Sets*.

Before you begin: If you also want run multiple instances of PKI Services in a Parallel Sysplex (one per image), you need to have performed the steps described in "(Optional) preliminary steps for establishing VSAM RLS" on page 105.

Perform the following steps to create the VSAM object store and ICL data sets and indexes (if you are configuring PKI Services for the first time):

1. Copy the sample JCL in `SYS1.SAMPLIB(IKYCVSAM)` to your JCL data set. (See "IKYCVSAM" on page 551 for a code sample of this file.)

2. Update your data set as directed in the instructions in the prolog of the sample JCL:
   a. Change the JOB card.
   b. Change the VOL statements.
      - If you are running multiple instances of PKI Services in a Parallel Sysplex, replace the VOL statements with STORCLAS statements that specify the storage class recorded in Table 24, for example:

        `STORCLAS(VSAMRLS)`
      - If you are running without a Parallel Sysplex, replace the `vvvvvv` in the VOL statements with a `VOL=SER` suitable for your VSAM data sets.
   c. If you are running multiple instances of PKI Services in a Parallel Sysplex, remove the SPANNED and CISIZE statements in the file. Make sure that you do not delete a closing parenthesis that is still needed. For example, change:

```
DATA -
(NAME(PKISRVD.VSAM.ICL.DA) -
CISZ(4096) -
SPANNED) -
```

to

```
DATA -
(NAME(PKISRVD.VSAM.ICL.DA)) -
```

d. You can optionally change the data set names but must remember to make equivalent changes in the `pkiserv.conf` file if you do so. (See Step 1c on page 79.)

e. Update the primary and secondary extent allocations based on your anticipated future needs. (See "Planning VSAM storage requirements" on page 104 for guidelines on determining the space you will need.) These are the default allocations for each type of data set:

Base cluster	`CYL(3,1)`
Alternate indexes	`TRK(5,1)`

Guideline: Do not change any numeric values, other than the primary and secondary space allocation values for the base cluster and alternate index data sets.

---

3. Submit the job when your changes are complete.

(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS

To run PKI Services in parallel, the UNIX programmer must specify `SharedPLEX=T` in the `pkiserv.conf` configuration file. (See the `SharedPLEX=T` row in Table 19 on page 60.) The MVS programmer enables the sysplex to access the VSAM data sets.

Before you begin: You need to have performed the steps described in "(Optional) preliminary steps for establishing VSAM RLS" on page 105.

Perform the following steps to enable your existing PKI Services data sets for VSAM record-level sharing (RLS).

1. Copy the sample reallocation JCL in `SYS1.SAMPLIB(IKYRVSAM)` to your JCL data set.

   Attention: Do not run IKYCVSAM by mistake because this JCL will destroy your existing VSAM data sets.

---

2. Update your data set, following the instructions in the prolog of the sample JCL:

   a. Change the JOB card.

   b. Change the STORCLAS statements.

      Note: Before submitting this job, check your access control service (ACS) routines for naming convention definitions that will be used to create the VSAM RLS data sets.

   c. Rename the source data sets to the names of your existing object store and ICL data sets.

d. Change the destination data set names.

   Note: Remember to give the UNIX programmer the data set names so the UNIX programmer can make equivalent changes in the `pkiserv.conf` file. See "(Optional) Steps for updating the configuration file" on page 60.

e. Update the primary and secondary extent allocations based on your anticipated future needs. (See "Planning VSAM storage requirements" on page 104 for guidelines on determining the space you will need.) These are the default allocations for each type of data set:

Base cluster	`CYL(3,1)`
Alternate indexes	`TRK(5,1)`

   Guideline: Do not change any numeric values, other than the primary and secondary space allocation values for the base cluster and alternate index data sets.

---

3. Submit the job when your changes are complete.

Tuning VSAM performance

Depending on your environment, your VSAM performance might be improved by providing buffer space for the VSAM data sets as part of the IKYSPROC (alias PKISERVD) started procedure.

(Optional) steps for adding VSAM buffer space

Perform the following steps to add buffer space for the VSAM data sets as part of the IKYSPROC (alias PKISERVD) started procedure. When completed, you will need to stop and restart PKI Services before your changes will take effect.

1. Make a backup copy of `SYS1.PROCLIB(PKISERVD)`. (See "PKISERVD sample procedure to start PKI Services daemon" on page 562 for a code sample of this file.)

---

2. Edit the JCL in `SYS1.PROCLIB(PKISERVD)` in the following ways.

   a. Append the DD statements shown in the following example to the bottom of the PKISERVD procedure.

   b. Change the names of the VSAM data sets shown in the example to the names that you used when you executed the IKYCVSAM sample job to allocate the data sets. (See"Steps for creating the VSAM object store and ICL data sets and indexes" on page 106.)

   Example:

```
//* ObjectDSN data set
//OST      DD  DSN=PKISRVD.VSAM.OST,DISP=SHR,
//  AMP=('BUFNI=6,BUFND=4')
//* ObjectTidDSN data
//TID      DD  DSN=PKISRVD.VSAM.OST.PATH,DISP=SHR,
//  AMP=('BUFNI=6,BUFND=4')
//* ObjectStatusDSN data set
//OSTAT    DD  DSN=PKISRVD.VSAM.OST.STATUS,DISP=SHR,
//  AMP=('BUFNI=1,BUFND=4')
//* ObjectRequestorDSN data set
//OREQ     DD  DSN=PKISRVD.VSAM.OST.REQUESTR,DISP=SHR,
//  AMP=('BUFNI=1,BUFND=4')
//* ICLDSN data set
```

```
//ICL      DD  DSN=PKISRVD.VSAM.ICL,DISP=SHR,
//  AMP=('BUFNI=6,BUFND=4')
//* ICLStatusDSN data set
//ISTAT    DD  DSN=PKISRVD.VSAM.ICL.STATUS,DISP=SHR,
//  AMP=('BUFNI=1,BUFND=4')
//* ICLRequestorDSN data set
//IREQ     DD  DSN=PKISRVD.VSAM.ICL.REQUESTR,DISP=SHR,
//  AMP=('BUFNI=1,BUFND=4')
```

3. Edit the names in the new DD statements above to match the data sets in the **ObjectStore** section of the PKI Services configuration file (`pkiserv.conf`).

4. Optionally, you might need to adjust the numeric values for BUFNI and BUFND.

   Tip: The STATUS and REQUESTR data sets are accessed sequentially only, while the others are accessed both sequentially and directly. Keep this in mind when adjusting the values. (For more information on VSAM buffer space, see *z/OS DFSMS Using Data Sets*.)

5. Edit the **ObjectStore** section of the PKI Services configuration file (`pkiserv.conf`) to change the existing data set names to the following DD names:

Default value	Suggested value
`ObjectDSN='pkisrvd.vsam.ost'`	`ObjectDSN=DD:OST`
`ObjectTidDSN='pkisrvd.vsam.ost.path'`	`ObjectTidDSN=DD:TID`
`ObjectStatusDSN='pkisrvd.vsam.ost.status'`	`ObjectStatusDSN=DD:OSTAT`
`ObjectRequestorDSN='pkisrvd.vsam.ost.requestr'`	`ObjectRequestorDSN=DD:OREQ`
`ICLDSN='pkisrvd.vsam.icl'`	`ICLDSN=DD:ICL`
`ICLStatusDSN='pkisrvd.vsam.icl.status'`	`ICLStatusDSN=DD:ISTAT`
`ICLRequestorDSN='pkisrvd.vsam.icl.requestr'`	`ICLRequestorDSN=DD:IREQ`

6. Save your changes.

7. Stop and restart PKI Services.

Backing up and restoring the VSAM data sets

You can use the DFSMSdss DUMP and RESTORE commands to back up and restore the VSAM data sets. For information about these commands, see *z/OS DFSMSdss Storage Administration*.

Steps for backing up the VSAM data sets

Perform the following steps to back up the VSAM data sets using the DFSMSdss DUMP command.

1. Copy the sample JCL in `SYS1.SAMPLIB(IKYVBKUP)` to your JCL data set. (See "IKYVBKUP" on page 560 for a code sample of this file.)
2. Update the JCL as directed in the instructions in the prolog of the sample JCL:

a. Change the job card.
b. If you are not using the default data set qualifiers, change all occurrences of "PKISRVD.VSAM" to the qualifiers you are using.
c. Change all occurrences of *vvvvvv* to a VOLSER value that contains sufficient free space to contain a complete backup of both of the PKI Services VSAM data set clusters.
d. Change the primary and secondary allocation values for the backup data set to values that will ensure a complete backup of both VSAM data set clusters. Change the *xxx* value for the primary allocation, and the *yyy* value for the secondary allocation.
e. You can optionally change the data set name for the BACKUPDS DD, but if you do be sure to make the same change to the restore JCL in IKYVREST.

3. Stop the PKI Services address space. For information about how to do this, see Chapter 10, "Starting and stopping PKI Services," on page 117.
4. Submit the job.
5. When the job has completed, restart the PKI Services address space. For information about how to do this, see Chapter 10, "Starting and stopping PKI Services," on page 117.

When you are done, you have created a backup data set containing the PKI Services VSAM files.

Steps for restoring the VSAM data sets

Perform the following steps to restore the VSAM data sets using the DFSMSdss RESTORE command.

1. Copy the sample JCL in `SYS1.SAMPLIB(IKYVREST)` to your JCL data set. (See "IKYVREST" on page 561 for a code sample of this file.)
2. Update the JCL as directed in the instructions in the prolog of the sample JCL:
   a. Change the job card.
   b. If you are not using the default data set qualifiers, change all occurrences of "PKISRVD.VSAM" to the qualifiers you are using.
   c. If you changed the default data set name for the BACKUPDS DD in the IKYVBKUP JCL, change the data set name (DSN) value in the BACKUPDS DD statement to match.
3. Stop the PKI Services address space. For information about how to do this, see Chapter 10, "Starting and stopping PKI Services," on page 117.
4. Submit the job.
5. When the job has completed, restart the PKI Services address space. For information about how to do this, see Chapter 10, "Starting and stopping PKI Services," on page 117.

When you are done, you have restored the PKI Services VSAM files from a backup data set.

Creating the object store and ICL using DB2 tables

PKI Services can use DB2 tables to store requests in progress and issued certificates. You need to create these tables (see "Steps for creating the object store and ICL DB2 tables" on page 111).

Sysplex considerations

Requirements: These requirements apply when you share the DB2 tables in a sysplex environment:

- If PKI Services is set up to run in a sysplex environment and share the DB2 tables for the object store and ICL, all systems in the sysplex must be running z/OS Version 1 Release 13 or higher.
- The DB2 subsystems that share data must belong to a DB2 data sharing group that runs on a sysplex cluster.
- The `SharedPLEX` parameter in the configuration file `pkiserv.conf` must be set to `T` (true). (See the `SharedPLEX=T` row in Table 19 on page 60.)

Planning DB2 storage requirements

The space allocation for DB2 is specified in the CREATE TABLESPACE SQL statements in the IKYCDB2 sample. The PRIQTY value is `144400` in kilobytes, which holds 144,400 ICL records that are 1024 bytes in length. Assuming that an object store certificate request record is approximately twice the size of an ICL record, 144,400 kilobytes holds 72,200 object store records. No SECQTY keyword is specified, because DB2 calculates a secondary quantity value.

If your anticipated needs differ greatly from the amount of storage provided by the PRIQTY keyword in the IKYCDB2 sample, adjust the PRIQTY value.

If you set the `RemoveInactiveReqs` and `RemoveCompletedReqs` parameters in the configuration file to `0d`, certificate request records are not deleted and you should adjust the space allocation accordingly.

Steps for creating the object store and ICL DB2 tables

You need to perform this task only if you are configuring PKI Services for the first time. If you have been using VSAM files for the object store and ICL and want to use DB2 tables instead, follow the steps in “Converting the object store and ICL from VSAM to DB2” on page 113.

Before you begin:

- DB2 must be installed and running.
- You need to know the name of the DB2 subsystem.
- You need to know the name you will use for the DB2 package. The default package name is `MASTERCA`.
- You need to know the name of the PKI Services daemon.

Perform the following steps to implement the object store and ICL using DB2 tables.

1. Update the IKYCDB2 sample and run it to create the DB2 objects. This sample contains SQL statements that define the database, tablespace, tables, and indexes for the ICL and object store.
   a. Copy the sample from `SYS1.SAMPLIB(IKYCDB2)`. (For a code sample of this file, see “IKYCDB2” on page 548.)
   b. As directed in the sample, if you have run the sample before, uncomment the statements in the first section so that the existing indexes, tables, tablespaces, and database are dropped and committed.

c. If you are using a package name other than the default name MASTERCA, change every occurrence of MASTERCA in the sample to the package name you have chosen.
d. If the PKI Services daemon has a name other than the default name of PKISRVD, change PKISRVD in the GRANT statements at the end of the sample to the name of your PKI Services daemon. (The daemon name is determined by the daemon variable in IKYSETUP. For more information, see Chapter 4, "Running IKYSETUP to perform RACF administration," on page 29.)
e. Run your updated copy of IKYCDB2. You can use the DB2 SPUFI facility to run it.

---

2. Update the sample job IKYSBIND and run it to build the PKI Services package and plan.
   a. Copy the sample job from SYS1.SAMPLIB(IKYSBIND). (For a code sample of this file, see "IKYSBIND" on page 559.)
   b. If you are using a package name other than the default name MASTERCA, change every occurrence of MASTERCA in the sample job to the package name you have chosen.
   c. If the PKI Services daemon has a name other than the default name of PKISRVD, change PKISRVD in the two OWNER statements to the name of your PKI Services daemon. (The daemon name is determined by the daemon variable in IKYSETUP. For more information, see Chapter 4, "Running IKYSETUP to perform RACF administration," on page 29.)
   d. Run your updated copy of IKYSBIND.

---

3. Update the IKYSETUP REXX exec to indicate that you are using DB2 tables, so that IKYSETUP gives the PKI Services daemon the access it needs to the DB2 Resource Recovery Services Access Facility (RRSAF). Find the section "Question 6 - Use DB2 as the repository for the Issued Certificate List (ICL) and Object Store". Change the line:

   ```
   db2_repos = 0
   ```

   to:

   ```
   db2_repos = 1
   ```

   Update the line:

   ```
   db2_subsys = 'DSN9'
   ```

   and change 'DSN9' to the name of the DB2 subsystem that will provide the repository. Run the updated copy of IKYSETUP.

---

4. Update the configuration file, pkiserv.conf, to indicate that you are using DB2 tables.
   - In the **ObjectStore** section, uncomment the following line:

     ```
     # DBType=DB2
     ```

     and make sure that the following line is commented out or omitted:

     ```
     # DBType=VSAM
     ```

   - In the **ObjectStore** section, uncomment the following line:

```
# DBPackage=MasterCA
```

If you used a package name other than `MasterCA` in the IKYCDB2 and IKYSBIND samples, change `MasterCA` in this line to the package name you used.

- In the **ObjectStore** section, uncomment the following line:

  ```
  # DBSubsystem=DSN9
  ```

  If you are using a DB2 subsystem with a name other than `DSN9`, change `DSN9` in this line to the name of your DB2 subsystem.
- If the object store and ICL will be shared in a sysplex with other instances of PKI Services, change the line:

  ```
  SharedPLEX=F
  ```

  to

  ```
  SharedPLEX=T
  ```
- Restart PKI Services so that your changes take effect.

When you are done, you have created DB2 objects for the object store and ICL.

Converting the object store and ICL from VSAM to DB2

If you have previously set up PKI Services to use VSAM data sets for the object store and ICL, you can convert it to use DB2 tables instead. To do this you must copy the data in your existing VSAM data sets to DB2 tables.

Steps for converting the object store and ICL from VSAM to DB2

Before you begin:

- DB2 must be installed and running.
- You need to know the name of the DB2 subsystem.
- You need to know the name you will use for the DB2 package. The default package name is `MASTERCA`.
- You need to know the name of the PKI Services daemon.

Perform the following steps to convert the object store and ICL from VSAM data sets to DB2 tables.

1. Stop the instance of PKI Services for which you want to do the conversion. For information on how to do this, see Chapter 10, “Starting and stopping PKI Services,” on page 117.

   ---

2. Update the IKYCDB2 sample and run it to create the DB2 objects. See step 1 on page 111.

   ---

3. Update the sample job IKYSBIND and run it to create the DB2 package and plan. See step 2 on page 112.

   ---

4. Use the utility program `vsam2db2` to copy the data from your VSAM data sets to the corresponding DB2 tables. See “Using the vsam2db2 utility” on page 377.

   ---

5. Update the configuration file, `pkiserv.conf`, to indicate that you are using DB2 tables. See step 4 on page 112.

6. Restart PKI Services. For information on how to do this, see Chapter 10, "Starting and stopping PKI Services," on page 117.

When you are done, you have copied your object store and ICL VSAM data sets to DB2 tables, and set up PKI Services to use the DB2 tables.

Columns in the ICL and object store DB2 tables

Not programming interface information

The information in Table 25 and Table 26 is intended for your use when creating DB2 queries and reports. It is not a programming interface.

Table 25. Columns in the object store DB2 tables

Column name	Format	Contents
Header:		
RECORD_KEY	BINARY(4)	
RECORD_STATE	BINARY(4)	
REQDATA_LEN	INTEGER	
REQUESTOR_NAME	VARCHAR(32)	Requestor name
TRANS_ID	CHAR(24)	Transaction ID
COMMENT	VARCHAR(64)	Comment
ISSUED_TIME	TIMESTAMP	Issued time
LAST_CHANGE_TIME	TIMESTAMP	Last changed time
TEMPLATE_NICKNAME	VARCHAR(8)	Template nickname
SERIAL_NUM	BINARY(4)	Serial number
Body:		
REQDATA	VARBINARY(32512)	Request object in ASN1 format

Table 26. Columns in the ICL DB2 tables

Column name	Format	Contents
Header:		
SERIAL_NUM	BINARY(4)	Serial number
CERT_STATE	BINARY(4)	
CERT_LEN	INTEGER	
REQUESTOR_NAME	VARCHAR(32)	Requestor name
REVOKE_DATE	TIMESTAMP	Revocation date
INVALID_DATE	TIMESTAMP	Expiration date
REVOKE_REASON	INTEGER	Revoke reason
COMMENT	VARCHAR(64)	Comment
ISSUED_TIME	TIMESTAMP	Issued time

Table 26. Columns in the ICL DB2 tables (continued)

Column name	Format	Contents
LAST_CHANGE_TIME	TIMESTAMP	Last changed time
TEMPLATE_NICKNAME	VARCHAR(8)	Template nickname
OBFUS_PW	VARCHAR(32)	Obfuscated passphrase
PROCESS_FLAGS	BINARY(4)	Flags
KEYID	BINARY (20)	SHA1 hash of public key
CRLDP_NUM	INTEGER	CRL DP number
EXPIRE_EPOCH_DAYS	INTEGER	Expiration date, expressed as "days since the epoch " (January 1, 1970)
EXPIRE_DATE	TIMESTAMP	Expiration date, expressed in DB2 TIMESTAMP format
KU_DIGTSIG	BINARY(1)	digitalSignature keyusage
KU_NONRPU	BINARY(1)	nonRepudiation keyusage
KU_KEYENC	BINARY(1)	keyEncipherment keyusage
KU_DATAENC	BINARY(1)	dataEncipherment keyusage
KU_KEYAGR	BINARY(1)	keyAgreement keyusage
KU_CRTSGN	BINARY(1)	keyCertSign keyusage
KU_CRLSGN	BINARY(1)	CRLSign keyusage
KU_ENCONLY	BINARY(1)	encipherOnly keyusage
KU_DECONLY	BINARY(1)	decipherOnly keyusage
EKU_SEVAUTH	BINARY(1)	serverAuth extended keyusage
EKU_CLIAUTH	BINARY(1)	clientAuth extended keyusage
EKU_CODESGN	BINARY(1)	codeSigning extended keyusage
EKU_EMLPROT	BINARY(1)	emailProtection extended keyusage
EKU_TMESTMP	BINARY(1)	timeStamping extended keyusage
EKU_OCSPSGN	BINARY(1)	OCSPSigning extended keyusage
EKU_MSSCLNON	BINARY(1)	Microsoft smart card logon extended keyusage
PREV_SERIAL_NUM	BINARY(4)	Previous serial number if the certificate has been renewed
SUBJ_DN	VARCHAR(1024)	Subject distinguished name
Body:		
X509CERT	VARBINARY(10240)	Certificate in ASN1 format

End of Not programming interface information

Chapter 10. Starting and stopping PKI Services

You start the PKI Services daemon or daemons the first time you are configuring PKI Services or if you are adding sysplex support to run multiple independent instances of PKI Services (one per image) on a sysplex. The MVS programmer performs these tasks.

Steps for starting the PKI Services daemon

You need to start the PKI Services daemon if:

- You are configuring PKI Services for the first time.
- You want to enable Simple Certificate Enrollment Protocol (SCEP).
- You renewed your CA or RA certificate.
- You want to use Parallel Sysplex support and need to run another instance of the PKI Services on a different image in the sysplex.
- You stopped PKI Services and need to restart it.
- You created a new (additional) CA domain and want to start it.
- The PKI Services CA certificate private key is managed by ICSF and ICSF became unavailable. After you fix the problem with ICSF, you need to stop and restart PKI Services.

Before you begin:

- Your z/OS HTTP Server should be SSL-enabled (see Chapter 7, "Updating z/OS HTTP Server configuration and starting the server," on page 91) and the uncustomized PKISERV application ready for use.
- If you are starting PKI Services for the first time, you need to know the runtime directory, called *runtime-dir* in the command examples. The default is `/etc/pkiserv/`. The MVS programmer was asked to record any changes to the default; see Table 3 on page 11.
- If you are starting PKI Services for a new CA domain, you need to know the job name that contains the instance of the daemon you need to start. Do not use these steps. Instead see "Adding a new CA domain" on page 230 for steps to add a new domain and start the new daemon.

Perform the following steps to start the PKI Services daemon and view your Web pages:

1. If you have not done so already, start the Web server and the LDAP server.

   _____

2. If you want to test the configuration to this point before customizing PKI Services (preferred), you need to temporarily prevent PKI Services from posting issued certificates to LDAP because posting to LDAP will not be successful. Have the UNIX programmer perform the following steps to prevent PKI Services from posting issued certificates to LDAP:
   a. Edit the PKI Services configuration file (by default, this is: `/etc/pkiserv/pkiserv.conf`).

      _____

   b. Set `NumServers=0` in the **LDAP** section of the file.

      _____

© Copyright IBM Corp. 2001, 2011

c. Exit to save your changes.

Note: After testing the configuration, you need to stop PKI Services and undo the change in this step (see Step 5) and then restart PKI Services.

---

3. Start the PKI Services daemon from the MVS console by entering the following command:

   ```
   S PKISERVD
   ```

   Notes:

   a. You must start the PKI Services daemon only from a started procedure. PKI Services rejects all other methods of starting the daemon (including INETD, `/etc/rc`, UNIX shell, or submitted JCL job).

   b. Depending on the amount of customization you did, there are various versions of the preceding command to start the PKI Services daemon. For example, if you changed the `pkiserv.envars` file (see Step 6 on page 56), you need to specify its new location as a parameter in the START command:

      ```
      S PKISERVD,DIR='runtime-dir'
      ```

      (Single quotation marks are required to maintain the character case of the values being assigned to the substitution parameters.)

      The command in the following example specifies the runtime directory and the file name of the environment variables file:

      Example:

      ```
      S PKISERVD,DIR='/etc/pkiserv',FN='pkiserv.envars'
      ```

      The default time zone is EST5EDT. If you need to change this, you can supply the new value as a parameter, as in the following examples:

      Examples:

      ```
      S PKISERVD,TZ=PST8PDT
      S PKISERVD,JOBNAME=jobname,DIR='/etc/pkiserv',FN='pkiserv.envars',TZ=PST8PDT
      ```

---

4. Go to your Web pages by entering the following URL from your browser:

   ```
   http://webserver-fully-qualified-domain-name/PKIServ/public-cgi/camain.rexx
   ```

   The *webserver-fully-qualified-domain-name* is the common name (CN) portion of the Web server's distinguished name; see Table 10 on page 32.

   You should be able to go through your Web pages to request, retrieve, and revoke a certificate of type “PKI browser certificate for authenticating to z/OS”. Ensure you can do this before trying to customize the application.

---

5. If you elected to test the configuration, you need to stop PKI Services (see “Stopping the PKI Services daemon” on page 119), undo the change in Step 2, and then restart PKI Services. To undo the change in Step 2:

   a. Edit the PKI Services configuration file (by default `/etc/pkiserv/pkiserv.conf`).

   b. Set `NumServers=n` in the **LDAP** section of the file, where *n* is the same number of LDAP servers indicated in Table 23 on page 98.

   c. Exit to save your changes.

Stopping the PKI Services daemon

To stop the PKI Services daemon, you can use either the MODIFY (or **F**) console command, or the STOP (or **P**) command. Enter one of the following two commands:

```
F PKISERVD,STOP
```

or

```
P PKISERVD
```

Part 3. Customizing PKI Services

PKI Services provides two ways to implement and customize the PKI Services Web interface:

- Using REXX CGI scripts. This method uses a text template file.
- Using Java server pages (JSPs). This method uses an XML template file.

Both methods produce identical Web pages with the default files shipped with PKI Services.

This part includes the following:

- Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs," on page 123 provides an overview of the `pkiserv.tmpl` file, which contains the certificate templates, and explains customizing the end-user Web pages when you are implementing them using the REXX CGI execs.
- Chapter 12, "Customizing the administration Web pages if you use REXX CGI execs," on page 175 provides an overview of the REXX CGI execs and explains how to customize the administration Web pages when you are implementing them using the REXX CGI execs.
- Chapter 13, "Implementing the Web application using Java server pages," on page 179 provides an overview of customizing the PKI Services Web pages when you implement them using the Java server pages.
- Chapter 14, "Advanced customization," on page 211 explains:
  - "Scaling for high volume installations" on page 211
  - "Using certificate policies" on page 212
  - "Updating the signature algorithm" on page 215
  - "Customizing distribution point CRLs" on page 217
  - "Using the OCSP responder" on page 224
  - "Creating a distribution point ARL" on page 225
  - "Adding an application domain" on page 226
  - "Adding a new CA domain" on page 230
  - "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246
  - "Customizing e-mail notifications sent to users" on page 252
  - "Setting up automatic renewal of certificates" on page 258
  - "Setting up PKI Services to generate keys for certificate requests" on page 259
  - "Adding custom extensions to certificates" on page 261
  - "Enabling support for large CRLs" on page 223
- Chapter 15, "Customizing with installation exit routines," on page 267 explains how to use installation exits with PKI Services.

© Copyright IBM Corp. 2001, 2011

Chapter 11. Customizing the end-user Web application if you use REXX CGI execs

This information applies if you are using REXX CGI execs for your PKI Services Web pages. If you are using Java server pages (JSPs) see Chapter 13, "Implementing the Web application using Java server pages," on page 179.

For certificate processing to work, you need to customize the end-user Web pages at least to some degree. Before you begin to customize Web pages, you need to understand the `pkiserv.tmpl` certificate templates file. This file contains certificate templates, which define the fields that comprise a specific certificate request. This topic describes the `pkiserv.tmpl` certificate templates file and explains how to use it to customize the end-user Web pages. This topic also explains the relationship between CGIs and the certificate templates file. Finally, this topic also discusses customizing e-mail notifications. (Sending e-mail notifications is an optional feature.)

Contents of the pkiserv.tmpl certificates templates file

The `pkiserv.tmpl` certificate templates file contains certificate templates that define the fields that comprise a specific certificate request. The file contains a mixture of true HTML and HTML-like tags. The HTML can contain JavaScript for input field verification.

The main sections of the `pkiserv.tmpl` certificate templates file are listed in Table 27:

Table 27. Structure and main divisions of the certificate template file (pkiserv.tmpl)

A prolog section of comments explaining main sections, subsections, named fields, and substitution variables.
A DEBUG flag appears right after the prolog section. (You can change `DEBUG=0` to `DEBUG=1` to get CGI debugging information.)
APPLICATION sections The APPLICATION sections contain subsections, which produce certain Web pages, such as the PKI Services home page shown in Figure 36 on page 309. For details, see "The APPLICATION sections" on page 133.
TEMPLATE sections These are the certificate templates (models) that contain the HTML to produce certificate request forms. They also define the fields that are permissible in the certificate. For details, see "TEMPLATE sections" on page 136.
INSERT sections These contain HTML for certain Web pages, such as the "Request submitted successfully" Web page, and certificate field dialogs, such as text entry boxes (the common name INSERT produces a text box where the user enters this information) and drop-downs. For details, see Figure 40 on page 319.

The `pkiserv.tmpl` file begins with a prolog. This is a section of comments that explains the main sections and subsections of the file. Any line with a # in column 1 is a comment.

© Copyright IBM Corp. 2001, 2011

Only the APPLICATION sections and TEMPLATE sections can contain subsections, but all three can contain named fields and substitution variables.

What are substitution variables?

A substitution variable holds a value that HTML code can reference. At run time, the actual value replaces a substitution variable.

You use square brackets to delineate a substitution variable.

Example:

```
[base64cert]
```

Notes:

1. Substitution variables are case-sensitive.
2. Depending on the section where a substitution variable is present, it might not have a valid meaning. For example, the `[base64cert]` substitution variable is meaningless before the certificate is retrieved. Therefore, in this case, the value of `[base64cert]` would be the null string (an empty string).

Table 28 summarizes valid substitution variables:

Table 28. Substitution variables

Substitution variable	Description
`altrawvalue`	The concatenated value of the AltOther fields.
`base64cert`	The requested certificate, base64-encoded.
`browsertype`	A special substitution variable to qualify named fields only. It enables Mozilla-based browsers and Internet Explorer to perform browser-specific operations, such as generating a public and private key pair. To generate a key pair, a Mozilla-based browser uses a `KEYGEN` HTML tag, while Internet Explorer uses ActiveX controls. For example, suppose you specify `%%PublicKey[browsertype]%%` in a TEMPLATE CONTENT section. If the user referencing this section uses the Mozilla Firefox browser, INSERT PublicKeyNS is included. If the user's browser is Microsoft Internet Explorer, INSERT PublicKeyIE is included.
`cadomain`	The CA domain name used to examine the preregistration record in the Simple Certificate Enrollment Protocol (SCEP).
`errorinfo`	Information such as the return code and reason code related to a failing SAF call.
`iecert`	The requested certificate in a form that Microsoft Internet Explorer accepts.
`keyid`	The SHA1 hash of the generated public key.
`optfield`	A special substitution variable that should be placed in any certificate field name INSERT where the end user can supply the value. It makes the input field optional.
`p12cert`	A PKCS #12 package in DER encoded format, containing a certificate and the private key that PKI Services generated for the certificate.
`printablecert`	This variable contains the certificate details so that the end user can confirm that the certificate is the correct one to renew or revoke. The displayed data is extracted from the ICL entry.

Table 28. Substitution variables (continued)

Substitution variable	Description
`readonly`	This substitution variable is converted to a null string when the INSERT is used for input purposes, such as when requesting a certificate. It is substituted with the string `readonly` when the INSERT is used for output purposes, such as when displaying request or certificate information.
`requestor`	The requestor's e-mail address, when the keys for the certificate were generated by PKI Services.
`serialno`	The serial number of a certificate.
`tmplname`	A certificate template name. This is primed from the HTML tag `<SELECT NAME="Template">` in the `<APPLICATION NAME=PKISERV>` section. The end user selects it on the first Web page.
`transactionid`	A unique value returned from a certificate request.

What are named fields?

Named fields insert common HTML code, such as a common input field or a page header or footer, in a Web page. (Each named field refers to a corresponding INSERT section.) A named field is delineated with `%%`.

Examples:

```
%%Country%%
%%-pagefooter%%
```

Note: Named fields are case-sensitive.

A named field can include or not include a dash. A named field without a dash, such as `%%Label%%` might have a special meaning as a certificate field. Its special meaning depends on the section in which it appears. (See "Relationship between CGIs and the pkiserv.tmpl file" on page 158 for more information.)

A named field with a dash, such as `%%-pagefooter%%`, has no special meaning. PKISERV treats it simply as HTML code to insert. Any special meaning the named field might have, based on the section in which it is contained, is ignored. For example, in a TEMPLATE CONTENT section (see "TEMPLATE sections" on page 136) if you specify `%%-pagefooter%%`, `-pagefooter` is not considered a certificate field name. However, the INSERT section with the name `-pagefooter` is included in the HTML page displayed to the end user.

INSERT sections

Although the INSERT sections are at the end of the `pkiserv.tmpl` certificate templates file, they are explained first because of their relationship to named fields. Any named field used in the `pkiserv.tmpl` file must be defined in a corresponding INSERT section.

Unlike the APPLICATION sections and TEMPLATE sections, INSERT sections can have no subsections. The following is the format of an INSERT section:

<INSERT NAME=*insert-name***>...</INSERT>**

An INSERT contains HTML that either:

- defines a certificate field
- defines other common HTML that can be referenced in other sections.

The following example of an INSERT defines a certificate field.

Example:

```
<INSERT NAME=Country>
<p> <LABEL for="countryfield">Country [optfield]</LABEL> <BR>
<INPUT NAME="Country" TYPE="text" SIZE=2 maxlength="2" id="countryfield">
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidCountry(frm){
 if ("[optfield]" == "" && frm.Country.value == "") {
  alert("Enter required field."); frm.Country.focus();
  return false;
 }
 return true;
}
//-->
</SCRIPT>
</INSERT>
```

The next example defines other common HTML:

Example:

```
<INSERT NAME=-pagefooter>
<p>email: webmaster@your_company.com
</INSERT>
```

To reference an INSERT, you use a named field of the form *%%insert-name%%*, for example %%Country%% or %%-pagefooter%%.

The pkiserv.tmpl certificate templates file contains INSERT sections of several main types:

- INSERTs that are for internal processing. (This is common HTML for Web page content as listed in Table 29.)
- INSERTs that are related to the certificate content. (See Table 30 on page 127.) These include:
  - X.509 fields (for example, OrgUnit)
  - Non-X.509 fields (for example UserId).

Table 29. INSERTs that are common HTML for Web page content. These INSERTS are for internal processing.

INSERT NAME	Contents
-AdditionalHeadIE	ActiveX controls to enable Internet Explorer to generate a key pair
-ChallengePassphrase, -ChallengePassPhrase2	HTML for a Web page that requests the passphrase that was specified when a certificate request was submitted
-RecoverEmail, -RecoverEmail2	HTML for the Web page that requests the e-mail address that was used when a certificate was requested, after a user requests to recover the certificate.
-requestok	HTML for the Web page “Request submitted successfully” after a successful certificate request (for both original requests and renewals). (For a sample of this Web page, see Figure 40 on page 319.)
-requestbad	HTML for the Web page “Request was not successful”
-requestor	HTML for a Web page that requests the name of a certificate requestor.

Table 29. INSERTs that are common HTML for Web page content (continued). These INSERTS are for internal processing.

INSERT NAME	Contents
-requestor2	HTML for a Web page that requests the e-mail address of a certificate requestor.
-renewkeysetIE	ActiveX controls to create a renewal certificate request using the original certificates key pair.
-renewkeysetNS	Mozilla-based browser script for renewal certificate requests.
-renewrevokeok	HTML for the Web page "Request submitted successfully" after a successful attempt to revoke a certificate. (See Figure 49 on page 326 for a sample of the Web page to renew or revoke a certificate.)
-renewrevokebad	HTML for the Web page "Request was not successful" after an unsuccessful attempt to renew or revoke a certificate. (See Figure 49 on page 326 for a sample of the Web page to renew or revoke a certificate.)
-returnp12cert	HTML for a Web page that displays a PKCS #12 package.
-preregok	HTML for the Web page "Preregistration successful" after a successful attempt to preregister a client for a certificate.
-returnpkcs10	This returns the server certificate in B64 format.
returnbrowsercertNS	This contains `[base64cert]`, which is the base64 substitution variable.
returnbrowsercertIE	This contains a script for producing a popup window installing your certificate (if you are using the Microsoft Internet Explorer browser). See Figure 43 on page 321 for a sample of this Web page.

Named fields in INSERT sections

Most of the following fields are X.509 fields. Table 30 summarizes the named fields in INSERT sections. (See **Restrictions** at the end of the table.)

Table 30. Named fields in INSERT sections

Field	Description
AltDomain	The host name of the machine where a certificate will be installed. This is a text field of up to 100 characters. The field can be repeated. **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
AltEmail	The user's e-mail address, including the `@` character and any periods (`.`). This is a text field of up to 100 characters. The field can be repeated. **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.

Table 30. Named fields in INSERT sections (continued)

Field	Description
AltIPAddr	The unique IP address that specifies the location of the server or device on the Internet. The field can be repeated. PKI Services supports both IP version 4 and IP version 6 addresses. The IP address is a text field of up to 45 characters. • For IP version 4, the IP address is in dotted decimal format; for example, `9.67.97.103`. • For IP version 6, the IP address is divided into eight 16-bit hexadecimal blocks separated by colons. Leading zeros in each 16-bit field are optional, and successive fields of zeros can be represented by double colons, but only once; for example `1:2::3:4` is equivalent to `0001:0002:0000:0000:0000:0000:0003:0004`. • In a mixed IP version 4 and IP version 6 environment, the IP address can be expressed in the format *x:x:x:x:x:x:d.d.d.d*, where the *x* values are the hexadecimal values of the six high-order 16-bit pieces of the address, and the *d* values are the decimal values of the four low-order 8-bit pieces of the address in standard IP version 4 representation; for example, `0:0:0:0:0:ABCD:1.2.3.4`, or the equivalent value `::ABCD:1.2.3.4` **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
AltOther [1]	A free form value for the other name of the subject's alternate name. Unlike the other INSERTs, you must customize it before you use it. The name of this INSERT consists of the string `AltOther`, concatenated with an underscore (_), then followed by the OID, specified in the following format: `AltOther_1_2_3_4_5`. (See "Customizing the OtherName field" on page 170.) You can have more than one input field but the total length of these fields together with the length of the OID and the comma cannot exceed 255 bytes. The resulting AltOther field is built by concatenating the dotted-decimal OID that matches the INSERT name, a comma, and the value of the input field. This is a text field of up to 255 characters. **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
AltURI	A name or address referring to an Internet resource; a URL is one kind of uniform resource identifier. This is a text field of up to 100 characters. The field can be repeated. **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
ChallengePassPhrase [1]	The passphrase the user entered when requesting a certificate. The user types the same passphrase, exactly as entered on the request form. This is a case-sensitive text field of up to 32 characters.
ClientName [1]	Name of the person or device being preregistered. This is a text field of up to 64 characters. **Restriction:** The first 32 characters of the name must be unique, irrespective of case, for each preregistered user.
CommonName	For browser certificates, this is your name, such as John Smith. (You can use your first and last name, in that order.) For server certificates, this is name by which the server's administrator wants it to be known. For SSL servers, the SSL protocol requires the CommonName to be the fully qualified domain name of the server, for example, `www.ibm.com`. CommonName is a text field of up to 64 characters. Although CommonName is a constant, no value is assigned to it. This indicates that RACF must determine the value. The user authenticates by specifying a user ID and password. (If `UserId` is listed in the APPL section, this means the application provides the user ID and password.) Providing the user ID and password enables RACF to look up the CommonName value in the user's profile. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.

Table 30. Named fields in INSERT sections (continued)

Field	Description
Country	The country where your organization is located. This is a 2-character text field. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
CustomExt	A custom certificate extension. Use this field to support extensions that PKI Services does not otherwise support. This is a repeatable field. For more information, see "Adding custom extensions to certificates" on page 261.
DNQualifier[1]	The subject's distinguished name qualifier. This is a text field of up to 64 characters.
DomainName[1]	The subject's domain name. It contains all the domain name components in the form <domain component1>.<domain component2>. ... .<domain component*n*>. This is a text field of up to 64 characters.
Email [1]	This is a deprecated insert for the e-mail address for the distinguished name; use the Mail insert instead. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
EmailAddr [1]	The e-mail address for the distinguished name. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
ExtKeyUsage [1]	The intended purpose of the certificate. Possible values are: **clientauth** Client side authentication **codesigning** Code signing **emailprotection** E-mail protection **mssmartcardlogon** Microsoft Smartcard logon **ocspsigning** OCSP response signing **serverauth** Server side authentication **timestamping** Digital timestamping.
HostIdMap [1]	This is the user ID for authorization purposes, in an e-mail type of format: *subject-id@host-name* For example, this could be `dsmith@ibm.com`. This is a text field of up to 100 characters. There are three ways to use `%%HostIdMap%%`: • If you place it in the CONTENT section, the end user can specify the value (or values, because it can be repeated). • You can also place it in the APPL section that the application provides. If you do so, it should have the following form: `%%HostIdMap=@host-name%%` The `host-name` is the hardcoded system name for the current system. The application provides the user ID as the user entered it when prompted for user ID and password. Note that, for this to function properly, the z/OS HTTP Server protection scheme for the request must force a prompt for user ID and password. Thus, only one `HostIdMap` is provided using this method. • A third way to specify `HostIdMap` is to place `%%HostIdMap%%` in the ADMINAPPROVE section. This allows the administrator to fill in the value when approving the certificate request. See "Administering HostIdMappings extensions" on page 404 for more information.
InstallCert	(This field is for the Internet Explorer browser only.) This field contains script for producing a window that installs an automatically-renewed certificate copied from an e-mail notification.
KeyProt [1]	(This is for the Internet Explorer browser only.) This asks if the user wants to enable strong private key protection. The drop-down choices are `Yes` and `No`.

Table 30. Named fields in INSERT sections (continued)

Field	Description
KeySize	The size of the keys (public key and private key) in bits, if they are to be generated by PKI Services. Valid values for each key type are: **RSA** 512, 1024, 2048, 4096 **NISTECC** 192, 224, 256, 384, 521 **BPECC** 160, 192, 224, 256, 320, 384, 512
KeyUsage	The intended purpose of the certificate. Each possible value is shown here with its intended purpose and possible PKIX bits:

KeyUsage value	Intended purpose	PKIX bits
certsign	Certificate and CRL signing	`KeyCertSign` and `cRLSign`
crlsign	CRL signing	`cRLSign`
dataencrypt, **dataencipherment**, or **dataenciph**	Data encryption	`dataEncipherment`
digitalsig or **digitalsignature**	Authentication	`digitalSignature`
docsign or **nonrepudiation**	Document signing	`nonRepudiation`
handshake	Protocol handshaking (for example, SSL)	`digitalSignature` and `keyEncipherment`
keyagree or **keyagreement**	Key agreement	`keyAgreement`
keycertsign	Certificate signing	`keyCertSign`
keyencrypt, **keyencipherment**, or **keyenciph**	Key transport	`keyEncipherment`

	Note: If **certsign**, **crlsign**, or **keycertsign** is specified, the certificate is created with the basic constraints extension to indicate that it is a Certificate Authority certificate, in addition to the key usage extension.
Label [2]	The label assigned to the requested certificate. This is a text field of up to 32 characters.
Locality	The city or municipality where your organization is located, such as Pittsburgh or Paris. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Mail [1]	The e-mail address for the distinguished name. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
NotBefore	Number of days (0 - 30) before the certificate becomes valid.
NotAfter	Number of days (1 - 9999) that the certificate is current. For example, 365 for a one-year certificate.

Table 30. Named fields in INSERT sections (continued)

Field	Description
NotifyEmail [1]	The e-mail address for notification purposes. If automatic certificate renewal is in effect, this is the e-mail address to which PKI Services sends the certificate when it is automatically renewed. This is a text field of up to 64 characters. **Notes:** 1. When a certificate is created and posted to LDAP, the NotifyEmail value, if specified, is posted as the MAIL attribute. If the MAIL attribute already exists in that directory entry, its value is replaced by the new value. If both NotifyEmail and Email appear on one request, they must have the same value. 2. If a certificate for which PKI Services generated the keys is renewed, the NotifyEmail field is ignored, and the renewed certificate is sent to the requestor's e-mail address.
Org	Organization. The legally registered name (or trademark name, for example, IBM) of your organization. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
OrgUnit	The name of your division or department. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
OrgUnit2	The name of your division or department. (There can be more than one organizational unit field on a request form. For example, one could be for your department and another for your division.) This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
PassPhrase [1]	The user decides this and enters and then reenters it when requesting a certificate (and must later supply this value when retrieving the certificate). This is a case-sensitive text field of up to 32 characters. There is no minimum number of characters, and the user can use any characters, but alphanumeric characters (`A - Z`, `a - z`, and `0 - 9`) are suggested.
PostalCode [1]	The zip code or postal code. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.

PublicKey	The base64-encoded #10 certificate request. (This is for server or device enrollment only.) You create a certificate request on behalf of another server (which could be a z/OS server or other type of server) or device for which you are requesting a certificate. You use software specific to that server to generate the #10 request before going to the PKI Services Web site. Save the request in a file. Then open the file in a text editor such as Windows Notepad and copy the and paste the contents into the text box on the enrollment form. A text area of 70 columns and 12 rows is allocated for this certificate request. Here is an example of the certificate request:
	
	`-----BEGIN NEW CERTIFICATE REQUEST-----`
	`MIIBiDCB8gIBADAZMRcwFQYDVQQDEw5Kb2huIFEuIFB1YmxpYzCBnzANBgkqhkiG`
	`9w0BAQEFAAOBjQAwgYkCgYEAsCT1cJHAGPqi6OjAyL+xNbt8z5ngmvqO2V0O3oYu`
	`/mEnQtRM96e+2jbmDCRo5tWVklG4OYf9ZVB5biURMJFLztfa4AVdEVtun8DH2pwc`
	`wiNIZZcC1Zym5adurUmyDk64PgiiIPMQS/t0ttG4c5U8uWSK0b1J4V4f7ps+t1aG`
	`t+cCAwEAAaAwMC4GCSqGSIb3DQEJDjEhMB8wHQYDVR0OBBYEFAlKTovBBvnFqDAO`
	`1oIhtRinwRC9MA0GCSqGSIb3DQEBBQUAA4GBAIbCVpwYvppIX3HHmpKZPNY8Snsz`
	`AJrDsgAEH51WOIRGywhqKcLLxa9htoQai6cdc8RpFVTwk6UfdCOGxMn4aFb34Tk3`
	`5WYdzOiHXg8MhHiB3EruwdWs+S7Fv3JhU3FLwU6lFLfAjbVi+35iEWQymOR6mE5W`
	`CathprmGfKRsDE5E`
	`-----END NEW CERTIFICATE REQUEST-----`

PublicKeyIE [1]	(This is for the Internet Explorer browser only.) This is the cryptographic service provider. The user selects a value from a drop-down list (Microsoft Base Cryptographic Provider or Microsoft Enhanced Cryptographic Provider).

Table 30. Named fields in INSERT sections (continued)

Field	Description
PublicKeyNS [1]	(This is for Mozilla-based browsers only.) This is the key size for your public/private key pair. The user selects a value from the drop-down list. Larger keys are more secure, but they also increase the time needed for connecting to a secure session.
PublicKey2IE	(This field is for the Internet Explorer browser only.) This field is the smart card cryptographic service provider. The user selects a smart card provider from a list.
PublicKey2NS	(This field is for Mozilla-based browsers only.) This field is the keygen HTML tag. It displays a menu of key sizes from which the user must choose one. When the user clicks **submit**, a key pair of the selected size is generated.
RecoverEmail, RecoverEmail2	This field is used to recover a certificate whose keys were generated by PKI Services. It contains the e-mail address of the requestor.
Requestor [1]	The user's name, used for tracking the request. This can be in any format, for example, John Smith or John. J. Smith. (This can differ from the common name, especially if the request is for a server certificate.) The value is saved with the request and issued certificate, but it is not a field in the created certificate. The default value is taken from the leftmost RDN in the subject's distinguished name, truncated to 32 characters.
Requestor2	The e-mail address of the requestor. This field is used to request a certificate with a key pair generated by PKI Services, and to retrieve such a certificate.
Security1, Security 2, ... Security*n*	Security questions used to assist recovering a certificate whose keys were generated by PKI Services. These fields can be used by the GENCERT, REQCERT and QRECOVER exits. You can have as many of these fields as you want, but the number you have must match the number that your exits handle. The fields should be numbered in order, beginning with Security1.
SerialNumber [1]	Serial number of the subject device. This is a text field of up to 64 characters.

SignWith	For PKI the component and for SAF the component and key-label used to sign this certificate, indicating the provider for certificate generation. This is a text field of up to 45 characters. It can be SAF or PKI Services, as shown in the following examples.
	
	Examples:
	`"SAF:CERTAUTH/Local CA Cert"`
	`"PKI:"`
	
	For SAF, the label of the signing certificate must be included. The first example shows the SignWith field in a SAF template. It includes the signing certificate, a CERTAUTH certificate labeled `'Local CA Cert'`.
	
	For PKI, it is an error to include the signing certificate. The second example shows the SignWith field in a PKI template. Notice that this contains no signing certificate.

StateProv	The state or province where your organization is located. Your registration policies determine whether you spell out the full name of the state or province or use an abbreviation. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Street [1]	The street address. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Title	Job title. This is a text field of up to 64 characters. **Note:** The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
TransactionId	PKISERV Web pages assign this after the user requests a certificate. When it is displayed, the user needs to record this number. This is a text field of up to 56 characters.
Uid [1]	The subject's login ID. This is a text field of up to 64 characters.

Table 30. Named fields in INSERT sections (continued)

Field	Description
UnstructAddr [1]	Unstructured address of the subject device. This is a text field of up to 64 characters.
UnstructName [1]	Unstructured device name. This is a text field of up to 64 characters.
UserId	The owning SAF user ID. This is a text field of up to 8 characters.

Restrictions:
1. This field is applicable for only PKI certificates (certificates using the `PKI:` value in the `SignWith` field).
2. This field is applicable for only SAF certificates (certificates using the `SAF:` value in the `SignWith` field).

The APPLICATION sections

The APPLICATION sections identify the application domains supported by PKI Services. The default certificate templates file (`pkiserv.tmpl`) ships with two applications sections, `PKISERV` (for PKI administrators) and `CUSTOMERS` (for general users).

The following is the format of the APPLICATION sections:

<APPLICATION NAME=*appl-name***>...</APPLICATION>**

Each application section begins with an application name definition.

Examples:

```
<APPLICATION NAME="PKISERV">
```

This application contains support for all templates and functions.

```
<APPLICATION NAME="CUSTOMERS">
```

This application contains support for all templates and functions but does not include the button on the bottom of the PKI Services home page that directs users to the administration page.

Each APPLICATION section can contain the subsections shown in Table 31.

Table 31. Subsections of the APPLICATION sections

Section or subsection	Contents
CONTENT	HTML for the "PKI Services Certificate Generation Application" Web page. (For a sample of this Web page, see Figure 36 on page 309.)This subsection should contain one or more named fields (see "What are named fields?" on page 125) identifying certificate templates to use for requesting or managing certificates through this application. These template names should match the HTML selection value associated with them.
FAILURECONTENT	This subsection contains the HTML for the Web page displayed when the certificate request submit failed. Any named fields in this subsection are interpreted as content inserts defined by INSERT sections. For PKISERV, the INSERT sections are included as part of the HTML presented to the end user.
FINDRECOVERCONTENT	This subsection contains the information for recovering a certificate in the case where the user has forgotten the passphrase. **Note:** Only certificates whose keys were generated by PKI Services are recoverable.

Table 31. Subsections of the APPLICATION sections (continued)

Section or subsection	Contents
RECONTENT	HTML for the Web page "Renew or revoke a browser certificate". The Web page displays information about a certificate so that the end user can confirm that this is the correct certificate to renew or revoke. (For a sample of this Web page, see Figure 49 on page 326.)This subsection uses the substitution variable [printablecert], which contains the data extracted from the ICL entry. (See "What are substitution variables?" on page 124.)
RECONTENT2	This subsection is similar to the RECONTENT section except that it applies to a certificate whose key was generated by PKI Services and is to be revoked.
RENEWEDCERT	This subsection contains HTML and Javascript functions to install an automatically-renewed certificate copied from an e-mail notification if using the Internet Explorer browser.
RESUCCESSCONTENT	Contains only the named field `%%-renewrevokeok%%` (whose associated INSERT contains HTML for the Web page "Request submitted successfully").
REFAILURECONTENT	Contains only the named field `%%-renewrevokebad%%` (whose associated INSERT contains HTML for the Web page "Request was not successful").The Web page is displayed to the end user when a renewal or revocation request is unsuccessful.
RETRIEVECONTENT2	This subsection contains the HTML to allow the end user to retrieve a recovered certificate.
RETURNCERT	This subsection contains the HTML for the Web page displayed upon successful retrieval of a recovered certificate. This section contains the named field `%%returnp12cert%%`, which indicates a PKCS #12 format.
ADMINSCOPE	This subsection is for an administration page. It contains only the named field `%%SelectCADomain%%` to prompt the administrator to choose a domain. (For more information, see "Adding a new CA domain" on page 230.)
ADMINHEADER	This subsection contains the general installation-specific HTML content for the header of all administration Web pages. See "Steps for customizing the administration Web pages" on page 177 for more information.
ADMINFOOTER	This subsection contains the general installation-specific HTML content for the footer of all administration Web pages. See "Steps for customizing the administration Web pages" on page 177 for more information.

Templates that PKI Services provides

PKI Services provides the templates to request the following certificates:

- One-year SAF server certificate
- One-year SAF browser certificate
- One-year PKI SSL browser certificate (See Figure 38 on page 316 to see a sample of this Web page.)
- One-year PKI SSL S/MIME browser certificate
- One-year PKI generated key certificate

- Two-year PKI browser certificate for authenticating to z/OS
- Two-year PKI Authenticode - code signing server certificate
- Two-year PKI Windows logon certificate
- Five-year PKI SSL server certificate
- *n*-year PKI browser certificate for extensions demonstration
- Five-year SCEP certificate - Preregistration
- Five-year PKI IPSEC server (firewall) certificate
- Five-year PKI intermediate CA server certificate

The following table describes the certificate templates that PKI Services provides:

Table 32. Certificate templates PKI Services provides

Certificate template	Description
One-year SAF server certificate	This template allows end users to request a server certificate using native SAF certificate generation facilities (rather than PKI Services certificate generation facilities). The certificate is used for handshaking only (for example, SSL). This certificate is auto-approved.
One-year SAF browser certificate	This template allows end users to request a browser certificate. SAF certificate generation facilities (rather than PKI Services certificate generation facilities) create the certificate. The requestor must input a label (see Table 30 on page 127 for descriptions of fields) because the certificate is stored in a RACF database. This certificate is auto-approved.
One-year PKI SSL browser certificate	This template allows end users to request a browser certificate that PKI Services generates. The end user enters the common name. (See Table 30 on page 127 for descriptions of fields.) This template contains an ADMINAPPROVE section. Therefore, certificates requested using this template require administrator approval before being issued. The user ID and password are not required but the passphrase is required.
One-year PKI S/MIME browser certificate	This template allows end users to request a browser certificate that PKI Services generates. This is similar to the one-year PKI SSL browser certificate except the end user selects AltEmail.
One-year PKI generated key certificate	This template allows end users to request a certificate that PKI Services generates, with a public key and private key that PKI Services generates. The user must supply a name, e-mail address, passphrase, and key size. This template requires administrator approval. You need to assess the risk of using this template. The requestor provides the transaction ID and passphrase to retrieve the private key and the certificate. The transaction ID and the passphrase entered by the requestor can be shown on the administrator pages. A malicious administrator could retrieve the certificate and the private key and use them. You should implement measures to minimize the risk of this happening; for example, check the log record on the number of retrievals or create an exit to limit the number of retrievals.
Two-year PKI browser certificate for authenticating to z/OS	This template allows end users to request a browser certificate that PKI Services generates. This is similar to the one-year PKI SSL browser certificate except this includes the `%%HostIdMap%%` INSERT and this certificate is auto-approved. `%%HostIdMap%%` is intended as a replacement for adding (and mapping) the certificate to a RACF user ID. This template specifies `%%HostIdMap=@ host-name%%` and `%%UserId%%` in the APPL section. This template does not require administrator approval but has protection through the user ID and password. (For more information about `%%HostIdMap%%`, see the `HostIdMap` field in Table 30 on page 127.)

Table 32. Certificate templates PKI Services provides (continued)

Certificate template	Description
Two-year PKI Authenticode–code signing server certificate	This template allows end users to request a server certificate be used to sign software that will be downloaded across an untrusted medium. It also demonstrates how to define extensions for template specific certificate policies and third party provided OCSP.
Two-year PKI Windows logon certificate	This template allows end users to request a certificate to use when logging in with a smart card to a Windows desktop as an Active Directory user. This template supports requests from both Internet Explorer and Mozilla-based browsers, and supports the following cryptographic services providers (CSPs). • Datakey • Gemplus • Infineon SICRYPT • Schlumberger Support for additional CSPs can be added when you customize the template.
Five-year PKI SSL server certificate	This template allows end users to request a server certificate that PKI Services generates. This is similar to the SAF server template except that this template contains an ADMINAPPROVE section. Therefore, certificates requested using this template require administrator approval before being issued. The user ID and password are not required but the passphrase is required.
Five-year PKI IPSEC server (firewall) certificate	This template allows end users to request a server certificate that PKI Services generates. This is similar to the five-year PKI SSL server certificate except that KeyUsage constants handshake and dataencrypt are hardcoded. Also, the end user selects AltEmail, AltIPAddr, AltURI, and AltDomain.
Five-year PKI intermediate CA server certificate	This template allows end users to request a server certificate that PKI Services generates. This is similar to the PKI SSL server template except that KeyUsage is hardcoded as certsign. Also, this certificate is auto-approved (because it runs under the user ID of the requestor, that is the person requesting this must be highly authorized). The user ID and password are required, and the units of work should run under the client's ID. In other words, the end user must be someone who can do this using RACDCERT alone, that is, must have CONTROL authority to IRR.DIGTCERT.GENCERT, and so forth. Given this requirement, the administrator need not approve this. The PassPhrase is required.
Five-year SCEP certificate—Preregistration	This template supports certificate preregistration for Simple Certificate Enrollment Protocol (SCEP) clients. The PassPhrase is required.
n-year PKI browser certificate for extensions demonstration	This template creates a browser certificate that has most of its information provided by the user rather than controlled by the administrator. The certificate contains all the supported extensions.

TEMPLATE sections

TEMPLATE sections define the fields that comprise a specific certificate request. They define the certificate templates referenced in the APPLICATION section. The `pkiserv.tmpl` certificate templates file contains a TEMPLATE section for each of the certificates described in Table 32 on page 135.

Each template section begins with one or more template names.

<TEMPLATE NAME=*tmpl-name***>...</TEMPLATE NAME>**

The `pkiserv.tmpl` certificate templates file that ships with PKI Services includes lines like the following:

Example:

```
<TEMPLATE NAME=1-Year PKI SSL Browser Certificate>
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=1YBSSL>
```

The true name of a certificate template is its actual complete name. This is the name in the first line, `1-Year PKI SSL Browser Certificate`, However, you can refer to a single template by more than one name by using an alias. The template name in the second line, `PKI Browser Certificate`, is an alias. An alias simply differentiates browser from server certificates. Finally, renewing a certificate requires recalling the template name, so the template name must be stored with the certificate. The NICKNAME (or short name) serves this purpose.

Notes:

1. You can have more than one alias. (Use an additional <TEMPLATE NAME=*alias*> line for each one.)
2. The value of a NICKNAME is an 8-character string.
3. SAF certificate templates do not include nicknames.

The following table shows the true name, alias, and nickname for each certificate template:

Table 33. Names, aliases, and nicknames of certificate templates

True name	Alias	Nickname
1-Year PKI SSL Browser Certificate	PKI Browser Certificate	1YBSSL
1-Year PKI S/MIME Browser Certificate	PKI Browser Certificate	1YBSM
1-Year PKI Generated Key Certificate	PKI Key Certificate	1YKRC
2-Year PKI Browser Certificate For Authenticating To z/OS	PKI Browser Certificate	2YBZOS
2-Year PKI Authenticode - Code Signing Certificate	PKI Server Certificate	2YIACS
2-Year PKI Windows Logon Certificate	PKI Browser Certificate	2YBWL
5-Year PKI SSL Server Certificate	PKI Server Certificate	5YSSSL
5-Year PKI IPSEC Server (Firewall) Certificate	PKI Server Certificate	5YSIPS
5-Year PKI Intermediate CA Certificate	PKI Server Certificate	5YSCA
5-Year SCEP Certificate - Preregistration	—	5YSCEPP
n-Year PKI Certificate for Extensions Demonstration	PKI Browser Certificate	SAMPLB
1-Year SAF Browser Certificate	SAF Browser Certificate	—
1-Year SAF Server Certificate	SAF Server Certificate	—

The AUTORENEW tag is optional. It determines whether the certificate is to be automatically renewed when it approaches expiration. This tag has the form <AUTORENEW=*value*>, where *value* can have the value `Y`, `y`, `N`, or `n`. If the AUTORENEW tag has any other value, or does not immediately follow the NICKNAME tag, PKI Services operates as if the tag is not present. The tag has the following meanings:

- AUTORENEW tag not present means that the certificate is not set up for automatic renewal.

- AUTORENEW=Y means that the certificate is enabled for automatic renewal.
- AUTORENEW=N means that the certificate is eligible for automatic renewal, but automatic renewal is disabled.

Note: Adding the AutoRenew=Y will not enable all certificates to be AutoRenewed. It will enable only the newly issued certificates after the template has been updated. For more information on how AutoRenew processing should be added, see Chapter 17, "Using the administration Web pages," on page 335.

See "Setting up automatic renewal of certificates" on page 258 for more information.

Example:

```
<TEMPLATE NAME=1-Year PKI SSL Browser Certificate>
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=1YBSSL>
<AUTORENEW=Y>
```

TEMPLATE sections can have the following subsections:

- CONTENT
- APPL
- CONSTANT
- ADMINAPPROVE
- SUCCESSCONTENT
- FAILURECONTENT
- RETRIEVECONTENT
- RETURNCERT
- PREREGISTER

<CONTENT>...</CONTENT>

This subsection contains the HTML to display a Web page to the end user requesting a certificate of a specific type. (See Figure 38 on page 316 for a sample Web page.) Field names on the certificate request (such as a text box where the user enters a value for `Common Name`) match the names of INSERT sections. The following examples show the INSERT sections corresponding to the field names `%%CommonName%%` and `%%Requestor (optional)%%`.

Examples:

```
<INSERT NAME=CommonName>
<p><LABEL for="commonnamefield">Common Name [optfield]</LABEL><BR>
<INPUT NAME="CommonName" TYPE="text" SIZE=64  maxlength="64" id="commonnamefield">

<INSERT NAME=Requestor>
<p><LABEL for="requestorfield">Your name for tracking this request [optfield]</LABEL><BR>
<INPUT NAME="Requestor" TYPE="text" SIZE=32 maxlength="32" id="requestorfield">
```

Named fields in this subsection are optional if the named field contains more that one word within the `%%` delimiters (as in `%%Requestor (optional)%%`). The user need not supply a value for Requestor.

<APPL>...</APPL>

This subsection identifies certificate fields for which the application itself should provide values. This subsection should contain only named fields, one per line. The only supported named fields allowed in this section are:

- UserId
- HostIdMap

Example:

```
<APPL>
 %%UserId%%
 %%HostIdMap=@www.ibm.com%%
<APPL>
```

<CONSTANT>...</CONSTANT>

This subsection identifies certificate fields that have a constant (hardcoded) value for everyone. This subsection should contain only named fields, one per line. The syntax for specifying the values is *%%field-name=field-value%%*:

Example:

```
%%KeyUsage=handshake%%
```

In addition to the named fields listed in Table 30 on page 127, you can also include the following named fields in this subsection only.

Critical

Identifies a certificate extension that is to be marked critical in the issued certificates. This name-value pair can be repeated for each extension to be marked critical. Here is the list of acceptable values for **Critical**:

- BasicContraints (ignored as this extension is always marked critical)
- KeyUsage (ignored as this extension is always marked critical)
- ExtKeyUsage
- SubjectAltName, AltEmail, AltIPAddr, AltDomain, AltURI
- HostIdMappings, HostIdMap
- CertificatePolicies, CertPolicies

Example:

```
%%Critical=ExtKeyUsage%%
```

Rules:

1. If you have specified configuration file setting PolicyRequired=T, then specifying %%Critical=CertPolicies%% will be ignored. The configuration file setting PolicyCritical will determine if the CertificatePolicies extension is marked critical. See "Using certificate policies" on page 212 for more information.
2. When ExtKeyUsage is extracted from an input PKCS #10 certificate request, the critical flag in the request is ignored. Therefore, setting %%Critical=ExtKeyUsage%% is the only way to get the ExtKeyUsage extension marked critical.

CertPolicies

Identifies the certificate policies that are to be included in the issued certificates. The value is a vector of numbers each representing one of the PolicyName*n* values specified in the **CertPolicy** section of the configuration file.

Example:

```
%%CertPolicies=3 6 10%%
```

Rule: If you have specified configuration file setting `PolicyRequired=T`, then specifying `%%CertPolicies=`*`any-value`*`%%` will be ignored. All issued certificates will have the same certificate policies as defined in the configuration file. See "Using certificate policies" on page 212 for more information.

AuthInfoAcc
Indicates the information necessary for the AuthorityInfoAccess extension. The value specifies a two-part, comma-separated string identifying the access method (`OCSP` or `IdentrusOCSP`) and the access URL. The URL must be specified in HTTP-protocol format only. (LDAP protocol is not supported.) The name-value pair can be repeated for each value required in the extension.

Examples:

```
%%OCSP,URL=https://ocsp.vendor.com%%
%%IdentrusOCSP,URI=https://identrus200.identrus.com%%
%%OCSP,URI=http://www.mycompany.com/PKIServ/public-cgi/caocsp%%
```

<ADMINAPPROVE>...</ADMINAPPROVE>
This optional subsection contains the named fields that the administrator can modify when approving certificate requests. (The named fields refer to INSERT sections.) When an end user requests a certificate, the certificate request might contain fields that the end user cannot see. When approving a request, the administrator can modify:

- Fields that are present and visible to the end user in the certificate request, for example `Common Name`
- Fields that are not visible to the end user but are hardcoded (in the CONSTANT subsection) in the template, for example `Organizational unit`
- Fields that are not visible to the end user and that the PKI Services administrator can add, for example, `HostIdMappings extension` or an empty `Organizational Unit` field (these are listed in the <ADMINAPPROVE> section, and either the end user did not fill them in or they are not present on the template request form).

The presence of this section (even if empty) indicates that an administrator must approve this request. The absence of this section indicates using auto-approval.

Note: In the `pkiserv.tmpl` certificate templates file, the only certificate templates that are auto-approved are the following:

- One-year SAF server certificate
- One-year SAF browser certificate
- Two-year PKI browser certificate for authenticating to z/OS
- Five-year PKI intermediate CA server certificate

You can put the following fields in the ADMINAPPROVE section:
- AltDomain

- AltEmail
- AltIPAddr
- AltOther_*OID*
- AltURI
- AuthInfoAcc
- CertPolicies
- CommonName
- Country
- Critical
- CustomExt
- DNQualifier
- DomainName
- EmailAddr
- EndDate
- ExtKeyUsage
- HostIdMap (can repeat)
- KeyUsage
- Locality
- Mail
- Org
- OrgUnit (can repeat)
- PostalCode
- SerialNumber
- StartDate
- StateProv
- Street
- Title
- Uid
- UnstructName
- UnstructAddr

Note: The following fields are not modifiable and are ignored in the ADMINAPPROVE section:

- Label
- PublicKey
- Requestor
- SignWith
- UserId

(For information about fields, see Table 30 on page 127.)

Example:

```
<ADMINAPPROVE>
%%KeyUsage%%
%%CommonName%%
%%OrgUnit%%
%%Org%%
%%Country%%
%%HostIdMap%%
%%HostIdMap%%
%%HostIdMap%%
%%HostIdMap%%
<ADMINAPPROVE>
```

<SUCCESSCONTENT>...</SUCCESSCONTENT>

This subsection contains the HTML to display to the end user a Web page saying that the certificate request was submitted successfully. Any named fields in this subsection are interpreted as

content inserts defined by INSERT sections. For PKISERV, the INSERT sections are included as part of the HTML to display a Web page to the end user.

In all of the templates included with PKI Services, <SUCCESSCONTENT> contains only the named field %%-requestok%%. (See "What are named fields?" on page 125 for an explanation of named fields.) This contains HTML for the Web page "Request submitted successfully". (For a sample of this Web page, see Figure 40 on page 319.)

<FAILURECONTENT>...</FAILURECONTENT>

This subsection contains the HTML to display to the end user a Web page saying the certificate request was not submitted successfully. Any named fields in this subsection are interpreted as content inserts defined by INSERT sections. For PKISERV, the INSERT sections are included as part of the HTML to display a Web page to the end user.

In all of the templates included with PKI Services, <SUCCESSCONTENT> contains only the named field %%-requestbad%%. (See "What are named fields?" on page 125 for an explanation of named fields.) This contains HTML for the Web page that says, "Request was not successful".

<RETRIEVECONTENT>...</RETRIEVECONTENT>

This subsection contains the HTML to display to the end user a Web page to enable certificate retrieval. Any named fields in this subsection are interpreted as content inserts that the INSERT sections define. For PKISERV, the INSERT sections are included as part of the HTML presented to the end user.

For a sample of a Web page this section generates, see Figure 41 on page 320. You might want to look at this Web page while reading the following explanation:

In all of the templates included with PKI Services, <RETRIEVECONTENT> contains the following:

- The named field %%-copyright%%, which displays any copyright information. (See "What are named fields?" on page 125 for an explanation of named fields.)
- The title of the Web page (This appears in the banner of your browser. Figure 41 on page 320 does not include the banner header but shows only the frame containing the content and not the browser window displaying the content.)
- A JavaScript script for processing the fields the user enters the Web page.
- A heading that says "Retrieve Your (name of certificate)". This uses the substitution variable [tmplname]. (See "What are substitution variables?" on page 124 for an explanation of substitution variables.)
- Text: a heading and paragraph about bookmarking this Web page.
- The named field %%TransactionId%% — A field where you enter your transaction ID if it is not already displayed.
- A field where you enter the passphrase you entered on the certificate request form.

<RETURNCERT>...</RETURNCERT>

This subsection contains the HTML to display a Web page upon successful certificate retrieval. The formats are:

- A browser certificate that can be installed into the browser.
- A server certificate that is displayed in B64 format.
- A PKCS #12 package, which contains the private key. For a browser or server certificate, this can be a single certificate or a certificate chain, depending on the authority of the user ID that does the retrieval. For a certificate with a key pair generated by PKI Services, the PKCS #12 package contains the requested certificate and its issuer certificate.

<PREREGISTER>...</PREREGISTER>

This optional subsection indicates the creation of a preregistration record and contains the Simple Certificate Enrollment Protocol (SCEP) rules for approval of a SCEP request. (For details, see "Overview of certificate request processing for preregistered SCEP clients" on page 247.)

Example:

```
<PREREGISTER>
AuthenticatedClient=AutoApprove
SemiauthenticatedClient=AdminApprove
UnauthenticatedClient=Reject
SubsequentRequest=AutoApprove
RenewalRequest=AutoApprove
</PREREGISTER>
```

Summary of subsections contained in certificate templates

Table 34 summarizes the subsections that are present in the various certificate templates in the `pkiserv.tmpl` file (as it is shipped):

Table 34. Summary of subsections in certificate templates

Subsection (in TEMPLATE section)	1-year PKI SSL browser	1-year PKI SSL S/MIME browser	1-year SAF browser	1-year SAF server	1–year PKI key	2-year PKI browser for z/OS	2-year PKI Authenticode signing server	2-year PKI Windows logon browser	5-year PKI SSL server	5-year PKI IPSEC server (firewall)	5-year PKI intermediate CA server	5-year SCEP preregistration	*n*-year PKI browser extensions demo
CONTENT	X	X	X	X	X	X	X	X	X	X	X	X	X
APPL			X	X		X					X		X
CONSTANT	X	X	X	X	X	X	X	X	X	X	X	X	X
ADMINAPPROVE	X	X			X		X	X	X	X			X
SUCCESSCONTENT	X	X	X	X	X	X	X	X	X	X	X	X	X
FAILURECONTENT	X	X	X	X	X	X	X	X	X	X	X	X	X
RETRIEVECONTENT	X	X	X	X	X	X	X	X	X	X	X		X
RETURNCERT	X	X	X	X	X	X	X	X	X	X	X		X
PREREGISTER												X	

Summary of fields in certificate templates

The tables in this topic summarize the fields contained in each certificate template that PKI Services provides.

Fields in the PKI browser certificate templates	Table 35 on page 144
Fields in the PKI server certificate templates	Table 36 on page 145
Fields in the SAF (browser and server), SCEP, and PKI generated key certificate templates	Table 37 on page 147

Table 35, Table 36 on page 145, and Table 37 on page 147 identify each template field as one of the following:

- Required
- Optional
- Provided by the application
- Constant (supplied value is shown)
- Blank (field is not present in either the CONTENT or CONSTANT section)

Table 35. Summary of fields for PKI browser certificate templates

Field name	One-year PKI SSL browser	One-year PKI S/MIME browser	Two-year PKI browser for z/OS	Two-year PKI Windows logon certificate	*n*-year PKI browser extensions demonstration
AltDomain					Optional
AltEmail		Required		Optional	
AltIPAddr					Optional
AltOther_*OID*				Constant[1]	Optional
AltURI					Optional
AuthInfoAcc					Constant[2]
CertPolicies					Constant: 1
ClientName					
CommonName	Required		Constant[3]	Optional	
Country					Optional
Critical					
CustomExt					Optional
DomainName					Optional
DNQualifier					Optional
EmailAddr					Optional
ExtKeyUsage	Constant: `clientauth`		Constant: `clientauth`	Constants: `clientauth` and `mssmartlogon`	Optional
HostIdMap[4]			Application provides		Optional
KeySize					
KeyUsage	Constant: `handshake`			Constant: `digitalSig`	Required
Label					Optional
Locality					Optional
Mail (previously called Email)					Optional
NotAfter	Constant: `365`		Constant: `730`		Optional
NotBefore	Constant: `0`				Optional
NotifyEmail	Optional				
Org	Constant: `The Firm`				Optional

Table 35. Summary of fields for PKI browser certificate templates (continued)

Field name	One-year PKI SSL browser	One-year PKI S/MIME browser	Two-year PKI browser for z/OS	Two-year PKI Windows logon certificate	*n*-year PKI browser extensions demon-stration
OrgUnit	Constant: `Class 1 Internet Certificate CA`				Required
OrgUnit2					Optional
PassPhrase	Required				
PostalCode					Optional
PublicKey	Browser provided[5]				
PublicKey2				Browser provided[5]	
Requestor	Optional				
SerialNumber					Optional
SignWith	Constant: `PKI:`				
StateProv					Optional
Street					Optional
Title					Optional
Uid					Optional
UnstructAddr					Optional
UnstructName					Optional
UserId			Application provides		

Notes:

1. The constant value is `_1_3_6_1_4_1_311_20_2_3`.
2. The constant value is `OCSP,URL=https://IV.OCSP.BankXYZ.com`.
3. Although CommonName is a constant, no value is assigned to it. This indicates that RACF must determine the value. The user authenticates by specifying a user ID and password. (If `UserId` is listed in the APPL section, this means the application provides the user ID and password.) Providing the user ID and password enables RACF to look up the CommonName value in the user's profile.
4. The `HostIdMap` value is formed by concatenating `UserId` with `@`*`host-name`*.
5. The PublicKey and PublicKey2 fields are coded with the *browsertype* substitution variable.

Table 36. Summary of fields for PKI server certificate templates

Field name	Two-year PKI Authenticode code signing server	Five-year PKI SSL server	Five-year PKI IPSEC server (firewall)	Five-year PKI intermediate CA server
AltDomain		Optional		
AltEmail	Required	Optional		
AltIPAddr		Optional		
AltOther_*OID*				
AltURI		Optional		
AuthInfoAcc	Constant[1]			

Table 36. Summary of fields for PKI server certificate templates (continued)

Field name	Two-year PKI Authenticode code signing server	Five-year PKI SSL server	Five-year PKI IPSEC server (firewall)	Five-year PKI intermediate CA server
CertPolicies	Constant: `1`			
ClientName				
CommonName	Constant: `My Company Code Signing Certificate`	Optional		
Country		Optional		
Critical	Constant: `ExtKeyUsage`			
CustomExt				
DNQualifier				
DomainName				
EmailAddr				
ExtKeyUsage	Constant: `codesigning`	Constant: `serverauth`		
HostIdMap[2]				
KeySize				
KeyUsage	Constants: `digitalsig` and `docsign`	Constant: `handshake`	Constants: `handshake` and `dataencrypt`	Constant: `certsign`
Label				
Locality		Optional		
Mail (previously called Email)				
NotAfter	Constant: `730`	Constant: `1825`		
NotBefore	Constant: `0`			
NotifyEmail	Required		Optional	
Org	Constant: `The Firm`	Optional		
OrgUnit	Optional			
OrgUnit2		Optional		
PassPhrase	Required			
PostalCode		Optional		
PublicKey	Required			
PublicKey2				
Requestor	Optional			
SerialNumber				
SignWith	Constant: `PKI:`			
StateProv		Optional		
Street		Optional		

Table 36. Summary of fields for PKI server certificate templates (continued)

Field name	Two-year PKI Authenticode code signing server	Five-year PKI SSL server	Five-year PKI IPSEC server (firewall)	Five-year PKI intermediate CA server
Title				
Uid				
UnstructAddr				
UnstructName				
UserId				Application provides

Notes:

1. The constant value is `OCSP,URL=https://ocsp.vendor.com`.
2. The `HostIdMap` value is formed by concatenating `UserId` with `@host-name`.

Table 37. Summary of fields for SAF, SCEP, and PKI generated key certificate templates

Field name	One-year SAF server	One-year SAF browser	Five-year SCEP preregistration	One-year PKI generated key
AltDomain	Optional			
AltEmail	Optional			
AltIPAddr	Optional			
AltOther_*OID*				
AltURI	Optional		Optional	
AuthInfoAcc				
CertPolicies				
ClientName			Required	
CommonName	Optional	Constant[1]	Optional	Required
Country	Required	Constant: `US`	Optional	
Critical				
CustomExt				
EmailAddr			Optional	
ExtKeyUsage			Optional	
HostIdMap[2]			Optional	
KeySize				Required
KeyUsage			Optional	Constant: `handshake`
Label	Required		Optional	
Locality	Optional		Optional	
Mail (previously called Email)			Optional	
NotAfter	Constant: `365`		Constant: `1825`	Constant: `365`
NotBefore	Constant: `0`			
NotifyEmail			Optional	

Table 37. Summary of fields for SAF, SCEP, and PKI generated key certificate templates (continued)

Field name	One-year SAF server	One-year SAF browser	Five-year SCEP preregistration	One-year PKI generated key
Org	Required	Constant: `The Firm`	Optional	Constant: `The Firm`
OrgUnit	Required	Constants: `OrgUnit=SAF template certificate` and `OrgUnit=Nuts and Bolts Division`	Optional	Constant: `Class 1 Internet Certificate CA`
OrgUnit2	Optional		Optional	
PassPhrase			Required	Required
PostalCode			Optional	
PublicKey	Required[3]	Browser provided[4]	Optional	
PublicKey2				
Requestor			Optional	Required
SerialNumber			Optional	
SignWith	Constant: `SAF:CERAUTH/taca`		Constant: `PKI:`	Constant: `PKI:`
StateProv	Optional		Optional	
Street			Optional	
Title			Optional	
UnstructAddr			Optional	
UnstructName			Optional	
UserId	Application provides			

Notes:

1. Although CommonName is a constant, no value is assigned to it. This indicates that RACF must determine the value. The user authenticates by specifying a user ID and password. (If `UserId` is listed in the APPL section, this means the application provides the user ID and password.) Providing the user ID and password enables RACF to look up the CommonName value in the user's profile.
2. The `HostIdMap` value is formed by concatenating `UserId` with `@host-name`.
3. The PublicKey is the PKCS #10 request.
4. The PublicKey field is coded with the *browsertype* substitution variable.

Examining the pkiserv.tmpl file

This topic contains excerpts from the following sections of the `pkiserv.tmpl` file. Each excerpt contains numbered pointers that describe the important tags in each section.

- "Examining the APPLICATION section" on page 149
- "Examining the TEMPLATE section" on page 153
- "Examining the INSERT section" on page 156

The pkiserv.tmpl file begins with a prolog section of comments explaining main sections, subsections, named fields, and substitution variables. The prolog section is followed by a DEBUG tag that you can change from the default (DEBUG=0) to DEBUG=1 to get CGI debugging information.

Examining the APPLICATION section

The APPLICATION section of the pkiserv.tmpl file contains two sample applications named PKISERV and CUSTOMERS.

- "Examining the PKISERV application"
- "Examining the CUSTOMERS application" on page 150

Examining the PKISERV application

The following example is an excerpt of the PKISERV application in the APPLICATION section of the pkiserv.tmpl file. (The vertical ellipses indicate omitted sections.)

```
# ==================================================================
#
# Application - PKISERV
#
# The installation should customize the CONTENT, ADMINHEADER
# ADMINFOOTER, and ADMINSCOPE subsections as appropriate
#
# ==================================================================
#
<APPLICATION NAME=PKISERV> 1
<CONTENT> 2
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE>PKI Administrators Start Page</TITLE>
%%-copyright%%
</HEAD>
<BODY>
<<H1>PKI Administrators Start Page</H1>
<p>
<A HREF="/PKIServ/cacerts/cacert.der"> 3
Install the CA certificate to enable SSL sessions for PKI Services </A>
<H2>Choose one of the following:</H1>
<p>
<h3>Manage existing certificates and certificate requests</h3>
# The following action will force userid/pw authentication for
# administrators
<FORM name=admform METHOD=GET ACTION="/PKIServ/ssl-cgi/auth/admmain.rexx"> 4
# The following action will force client certificate authentication for
# administrators
#<FORM name=admform METHOD=GET
# ACTION="PKIServ/clientauth-cgi/auth/admmain.rexx">
<p>
<INPUT TYPE="submit" VALUE="Administration Page">
</FORM>
# Multiple CA mode - replicate and modify the following H3 and FORM
# section for each CA domain.
<h3>Go to the Customers' home page </h3>
<FORM name=admform METHOD=GET ACTION="/Customers/public-cgi/camain.rexx"> 5
<p>
<INPUT TYPE="submit" VALUE="Customers' Home Page">
</FORM>
<p> %%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<ADMINHEADER> 6
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE>Web Based Certificate Generation Administration</TITLE>
%%-copyright%%
</HEAD>
<BODY>
</ADMINHEADER>
<ADMINFOOTER>
<p> %%-pagefooter%% 7
</BODY>
</HTML>
</ADMINFOOTER>
<ADMINSCOPE> 8
```

```
# Uncomment the following line to enable multiple CA domains
#%%SelectCADomain%%
</ADMINSCOPE>
</APPLICATION>
```

The numbers in the following list refer to the highlighted tags in the preceding excerpt of the PKISERV application.

1. This is the beginning of the APPLICATION section. The name of the application is PKISERV.
2. This is the beginning of the CONTENT subsection. The CONTENT subsection contains HTML to display the Web page where the administrator begins. The TITLE indicates the main heading of that Web page, "PKI Administrators Start Page." (See Figure 58 on page 336 for a sample of that Web page.)
3. The HREF tag is the link to install the CA certificate in the browser.
4. The ACTION tag indicates where to go when the user clicks the **Administration Page** button. (See Figure 61 on page 340 for a sample of that Web page.)
5. The ACTION tag indicates where to go when the user clicks the **Customers' Home Page** button. (See Figure 36 on page 309 for a sample of that Web page.)
6. The ADMINHEADER subsection references the `%%-copyright%%` named field, which is defined in the INSERT section. This should contain the copyright statement for your company.
7. The ADMINFOOTER subsection references the `%%-pagefooter%%` named field, which is defined in the INSERT section. This named field should specify the e-mail address of your PKI Services administrator.
8. The ADMINSCOPE subsection references the `%%SelectCADomain%%` named field, which is defined in the INSERT section. When you have multiple CA domains, you can use this variable to allow PKI administrators to select a CA domain on the administrator's home page. (See "Adding a new CA domain" on page 230 for details about implementing multiple CA domains.)

Examining the CUSTOMERS application

The following example is an excerpt of the CUSTOMERS application in the APPLICATION section of the `pkiserv.tmpl` file. (The vertical ellipses indicate omitted sections.)

```
# ====================================================================
#
# Application - CUSTOMERS
#
# The installation should customize the CONTENT subsection as appropriate.
#
# ====================================================================
#
<APPLICATION NAME=CUSTOMERS> 1
<CONTENT> 2
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Customers Certificate Generation Application </TITLE>
<SCRIPT LANGUAGE="Javascript">
<!--
//Get browser type
function getBrowserType()
{
// Determine the browser type from where the script is being invoked.
var type = navigator.appName;
if(type == "Microsoft Internet Explorer")
{
}else
{document.getElementById('install').style.display='none';
}
}
// -->
</SCRIPT><SCRIPT LANGUAGE="JavaScript">
<!--
function getOsVersion()
```

```
{
   var OS = navigator.userAgent;
   if (OS.indexOf("Windows NT 5")!=-1)
   {
        document.getElementById('install').href =
"/PKIServ/PKIXEnroll/PKIXEnrollDeploy.msi";
   }
   else
   {
        document.getElementById('install').href =
"/PKIServ/PKICEnroll/PKICEnrollDeploy.msi";
   }
   return true;
}
//-->
</SCRIPT>
%%-copyright%%
</HEAD>
<BODY>
<H1>PKI Services Certificate Generation Application</H1>
<p>
<A HREF="/PKIServ/cacerts/cacert.der"> 3
Install the CA certificate to enable SSL sessions for PKI Services </A>

<br><p>
<A href = "" id = "install" onClick="getOsVersion()">Install the PKI ActiveX
Control to renew certificates</A> 4

<H2>Choose one of the following:</H2>
<ul>
<li><h3>Request a new certificate using a model</h3>
<FORM name=mainform METHOD=GET ACTION="/[application]/ssl-cgi/catmpl.rexx"> 5
<p><LABEL for="seltemplate">Select the certificate template to use as a model</LABEL>
<SELECT NAME="Template" id="seltemplate"> 6
 %%1-Year PKI SSL Browser Certificate%%
     <OPTION>1-Year PKI SSL Browser Certificate
 %%1-Year PKI S/MIME Browser Certificate%%
     <OPTION>1-Year PKI S/MIME Browser Certificate
 %%2-Year PKI Windows Logon Certificate%%
     <OPTION>2-Year PKI Windows Logon Certificate
.
.
.
</HTML>
</CONTENT>
<RECONTENT> 7
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Customers Renew or Revoke a Browser Certificate </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function LoadActiveX()
{
var OS = navigator.userAgent;
if (OS.indexOf("Windows NT 5")!=-1)
{
document.getElementById("xenrollreq").setAttribute("classid","clsid:157B42C3-
25EB-4C6B-A569-27FA081D61EC");
       document.getElementById("osname").value="XP";
  }
  else
  {

document.getElementById("cenrollreq").setAttribute("classid","clsid:65D22D38-
D2D2-421F-BDFE-B7D990DDFE96");
       document.getElementById("osname").value="nonXP";

   }
   return true;
}
//--></SCRIPT>
#-- Create a certmgr object for use in the renew process @01A
<OBJECT
   classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
   CODEBASE="xenroll.cab#Version=5,131,3659,0"
   id="certmgr"
>
</OBJECT>
<OBJECT
#-- Create g_objWCF object for CertEnroll process                  6@LDA>
<OBJECT
   classid="clsid:884e2049-217d-11da-b2a4-000e7bbb2b09"
   id="g_objWCF"
>
</OBJECT>
<OBJECT id="xenrollreq">
```

```
</OBJECT>
<OBJECT id="cenrollreq">
</OBJECT>
%%-copyright%%
</HEAD>
<BODY onLoad="LoadActiveX()">
<H1>Renew or Revoke a Browser Certificate</H1>
⋮
</BODY>
</HTML>
</RECONTENT>
<RESUCCESSCONTENT>  8
 %%-renewrevokeok%%
</RESUCCESSCONTENT>
<REFAILURECONTENT>  9
 %%-renewrevokebad%%
</REFAILURECONTENT>


<FINDRECOVERCONTENT> 10
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Use security questions to locate certificate </TITLE>
⋮
<RETRIEVECONTENT2> 11
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE  Web Based PKIX Certificate Recovery Application</TITLE>
⋮
<RETURNCERT>
%%returnp12cert%%
</RETURNCERT>
<FAILURECONTENT>
%%-requestbad%%
</FAILURECONTENT>
<RENEWEDCERT>  12
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Install Automatic Renewed Certificate</TITLE>
<SCRIPT LANGUAGE="Javascript">
⋮
%%-RenewKeySetIE%%
%%InstallCert%%
</RENEWEDCERT></APPLICATION>
```

The numbers in the following list refer to the highlighted tags in the preceding excerpt of the CUSTOMERS application.

1. This is the beginning of the APPLICATION section. The name of the application is CUSTOMERS.
2. This is the beginning of the CONTENT subsection. The CONTENT subsection contains HTML to display the Web page where the end user requests or retrieves a certificate. The `<H1>` indicates the main heading of that Web page, "PKI Certificate Generation Application." (See Figure 36 on page 309 for a sample of that Web page.)
3. The HREF tag is the link to install the CA certificate in the browser.
4. The HREF tag is the link to install the PKI Services ActiveX control.
5. The ACTION tag indicates where to go when the user clicks the **Request certificate** button.
6. The SELECT tag produces a drop-down that lists the certificate templates the user can request. (The named fields, which are bracketed with %% symbols, are the names of the certificate templates.)
7. The RECONTENT section contains the HTML to display the Web page where the end user renews or revokes a certificate. The main heading on this Web page is "Renew or Revoke a Browser Certificate". It includes a Javascript function that determines which PKI Services ActiveX programs should be loaded. (See Figure 49 on page 326 for a sample of that Web page.)
8. The RESUCCESSCONTENT subsection references the `%%-renewrevokeok%%` named field, which is defined in the INSERT section. This contains HTML for

the Web page displayed when the user's attempt to revoke a certificate is successful. The main heading on this Web page is “Request submitted successfully”. (See Figure 40 on page 319 for a sample of that Web page.)

9. The REFAILURECONTENT subsection references the `%%-renewrevokebad%%` named field, which is defined in the INSERT section. This contains HTML for the Web page displayed when the user's attempt to renew or revoke a certificate fails. The main heading on this Web page is “Request was not successful”.
10. The FINDRECOVERCONTENT subsection displays security questions for users to answer when they want to recover a certificate and have forgotten the passphrase.
11. The RETRIEVECONTENT2 subsection contains the HTML to allow the end user to retrieve a recovered certificate.
12. The RENEWEDCERT subsection references the `%%RenewKeySetIE%%` named field, which is defined in the INSERT section. This field contains the HTML and Javscript functions for the Web page displayed when a user clicks on a link in an e-mail notification to install an automatically renewed certificate.

Examining the TEMPLATE section

The TEMPLATE section follows the APPLICATION section and contains several sample templates. The following example is an excerpt from the TEMPLATE section of the `pkiserv.tmpl` file. (The vertical ellipses indicate omitted sections.)

```
# ==========================================================================
#
# Template Name - 2-Year PKI Browser Certificate For Authenticating
#                 to z/OS 1
#
# Function - Creates a 2-Year certificate good for authenticating to z/OS.
:
# User input fields:
#  Requestor - optional
#  PassPhrase - required
#  PublicKey - required (Provided by the browser itself)
#  NotifyEmail - optional
:
# ==========================================================================
#
<TEMPLATE NAME=2-Year PKI Browser Certificate For Authenticating To z/OS> 2
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=2YBZOS>
<AUTORENEW=Y>
<CONTENT> 3
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE> 4
%%-copyright%% 5
%%-AdditionalHead[browsertype]%%
</HEAD>

<BODY>
<H1>2-Year PKI Browser Certificate For Authenticating To z/OS</H1> 6
<p>
<H2>Choose one of the following:</H2>
:
#<FORM NAME="CertReq" METHOD=POST ACTION= 7
#              "/[application]/ssl-cgi-bin/careq.rexx" onSubmit=
    "return ValidateEntry(this)">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s) 8
<SCRIPT LANGUAGE="JavaScript"> 9
<!--
:
//-->
</SCRIPT>
 %%Requestor (optional)%%
 %%NotifyEmail (optional)%%
 %%PassPhrase%%
 %%PublicKey2[browsertype]%%
```

```
<p>
<INPUT TYPE="Submit" VALUE="Submit certificate request">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="/[application]/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%% 10
</BODY>
</HTML>
</CONTENT>
<APPL> 11
 %%UserId%%
 %%HostIdMap=@host-name%%
</APPL>
<CONSTANT> 12
 %%CommonName=%%
 %%OrgUnit=Class 1 Internet Certificate CA%%
 %%Org=The Firm%%
 %%KeyUsage=handshake%%
 %%ExtKeyUsage=clientauth%%
 %%NotBefore=0%%
 %%NotAfter=730%%
 %%SignWith=PKI:%%
</CONSTANT>
<SUCCESSCONTENT> 13
 %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT> 14
 %%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT> 15
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
%%-copyright%%
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
:
:
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1> 16
<H3>Please bookmark this page</h3>
:
:
#<FORM NAME=retrieveform METHOD=POST ACTION= 17
#      "/[application]/ssl-cgi-bin/cagetcert.rexx" onSubmit=
:
:
</FORM>
:
:
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT> 18
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>
```

The numbers in the following list refer to the highlighted tags in the preceding excerpt of the TEMPLATE section.

1. The template begins with a block comment identifying the template and explaining its use and fields.
2. There are three names for each certificate (except for SAF templates, which do not include nicknames). The first TEMPLATE NAME line defines the true (actual, complete) name of the certificate. The next TEMPLATE NAME line defines an alias. (This simply differentiates browser from server certificates.) The NICKNAME defines an 8-character string. In each template for which you

want certificates to be automatically renewed, insert the AUTORENEW tag immediately following the NICKNAME tag, if it is not already there, and set it to `Y`.

3. The CONTENT subsection contains the HTML to display a Web page to the end user requesting this type of certificate. (The CGI script `catmpl.rexx` displays this content.)
4. The title contains the heading that appears at the very top of the browser when the Web page is displayed.
5. The `%%-copyright%%` named field displays the copyright statement.
6. The heading is the main heading on the Web page for requesting the selected certificate.
7. The ACTION tag indicates that the CGI script that gets control when the user clicks the **Submit certificate request** button is `careq.rexx`.
8. Fields for which the user can supply input include `%%Requestor%%`, `%%PassPhrase%%`,`%%NotifyEmail%%`, and `%%PublicKey2%%`. (These fields are named fields that are defined in the INSERT section, which is shown later.) All fields not marked optional are required. `%%PublicKey2%%` contains the substitution variable, `[browsertype]`. This is replaced at run time with `IE` or `NS`, depending on the browser the user has. This is necessary because the browsers behave differently for key generation and certificates.
9. This JavaScript script provides the underlying logic for the text entry that the user must perform.
10. The `%%-pagefooter%%` named field is defined in the INSERT section (shown later). This contains the e-mail address of the PKI Services administrator.
11. The APPL subsection indicates the fields that `careq.rexx` itself provides, in this case, `%%UserId%%` and `%%HostIdMap%%`. (These are set from the z/OS HTTP Server environment variable `REMOTE_USER`.)
12. The CONSTANT subsection has hardcoded values to use, for example (for the non-SAF certificates), the signing certificate is `PKI:`.
13. The SUCCESSCONTENT subsection contains the HTML to display upon successfully requesting the certificate. It includes the `%%-requestok%%` named field. (This is defined in the INSERT section, shown in "Examining the INSERT section" on page 156. See list item 1 on page 157.)
14. The FAILURECONTENT subsection contains the HTML to display when the certificate request is unsuccessful. This subsection contains the `%%-requestbad%%` named field. (This named field is defined in the INSERT section, shown in "Examining the INSERT section" on page 156.)
15. The `-requestok` INSERT (mentioned in list item 13) includes an ACTION that calls `caretrieve.rexx`, which displays the HTML in the RETRIEVECONTENT subsection. The first time the Web page is displayed, it includes the transaction ID associated with the certificate request. If the user leaves the Web page and then returns, the transaction ID field must be filled in. Entering the transaction ID and clicking the **Continue** button calls `cagetcert.rexx`.
16. The main heading on the Web page is "Retrieve Your (Name of Certificate)".
17. The ACTION is to call `cagetcert.rexx` as list item 15 indicates.
18. The RETURNCERT subsection contains the `%%return10cert%%` named field, which is defined in an INSERT. (See list item 4 on page 157.)

Examining the INSERT section

The final section of the `pkiserv.tmpl` file contains several sample INSERTS. The following example is an excerpt from the INSERT section of the `pkiserv.tmpl` file. (The vertical ellipses indicate omitted sections.)

```
# ==================================================================
#
# Sample INSERTS
#
# ==================================================================
#
<INSERT NAME=-AdditionalHeadIE>
<OBJECT
  classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
  CODEBASE="xenroll.cab#Version=5,131,3659,0"
  id="certmgr"
>
</OBJECT>
<OBJECT
 classid="clsid:884e2049-217d-11da-b2a4-000e7bbb2b09"
    id="g_objWCF"
>
</OBJECT>
</INSERT>

<INSERT NAME=-requestok> 1
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based Certificate Generation Success</TITLE>
</HEAD>
<BODY>
<H1> Request submitted Successfully</H1>
[errorinfo]
<p> Here's your transaction ID. You will need it to retrieve your
certificate.  Press 'Continue' to retrieve the certificate.
<p> <TABLE BORDER><TR><TD>[transactionid]</TD></TR></TABLE>
<FORM METHOD=GET ACTION="/[application]/ssl-cgi/caretrieve.rexx"> 2
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT NAME="TransactionId" TYPE="hidden" VALUE="[transactionid]">
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>

<INSERT NAME=-requestbad> 3
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based Certificate Generation Failure</TITLE>
</HEAD>
<BODY>
<H1> Request was not successful</H1>
<p> Please correct the problem or report the error to your Web admin
person<br>
<PRE>
[errorinfo]
</PRE>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
.
.
.
<INSERT NAME=-returnpkcs10> 4
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based Certificate Generation Application Pg 4</TITLE>
</HEAD>
<BODY>
<H1> Here's Your Certificate. Cut and Paste it to a File</H1>
<TABLE BORDER><TR><TD>
<PRE>
[base64cert] 5
</PRE>
</TD></TR></TABLE>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
.
.
.
</BODY>
```

```
</HTML>
</INSERT>
#
# ====================================================================
#
# X.509 fields (INSERTs) valid for certificate requests
#
# ====================================================================
#
:
<INSERT NAME=PublicKeyIE> 6
<SCRIPT LANGUAGE="VBScript">
<!--
:
// -->
</SCRIPT>

# ====================================================================
:
<INSERT NAME=PassPhrase> 7
<p> <LABEL for="passphrasefield">Pass phrase for securing this request. You will need to supply
this value when retrieving your certificate [optfield]</LABEL> <BR>
<INPUT NAME="PassPhrase" TYPE="password" SIZE=32 maxlength="32" id="passphrasefield"> <BR>
<p> <LABEL for="passphrase2field">Reenter your pass phrase to confirm</LABEL> <BR>
<INPUT NAME="ConfirmPassPhrase" TYPE="password" SIZE=32
 maxlength="32" id="passphrase2field">
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidPassPhrase(frm){
 if ("[optfield]" == "" && frm.PassPhrase.value == "") {
  alert("Enter required field."); frm.PassPhrase.focus();
  return false;
 }
 if ("[optfield]" == "" && frm.ConfirmPassPhrase.value == "") {
  alert("Reenter the pass phrase."); frm.ConfirmPassPhrase.focus();
  return false;
 }
 if (frm.PassPhrase.value != frm.ConfirmPassPhrase.value) {
  alert("Passwords don't match. Reenter."); frm.PassPhrase.focus();
  return false;
 }
 return true;
}
//-->
</SCRIPT>
</INSERT>
:
<INSERT NAME=-pagefooter>
<p>email: webmaster@your_company.com
</INSERT>
```

The numbers in the following list refer to the highlighted tags in the preceding excerpt of the INSERT section.

1. The `-requestok` INSERT has the logic to generate the certificate. If the certificate is successfully generated, a Web page (whose main heading is "Request submitted successfully") is displayed. This Web page includes the transaction ID.
2. The `-requestok` INSERT includes an ACTION that calls `caretrieve.rexx`, which allows the user to retrieve the certificate.
3. Alternately, if the request is not successful, the `-requestbad` INSERT gains control.
4. (The `caretrieve.rexx` CGI displays the RETRIEVECONTENT subsection (see list item 15 on page 155) HTML, which displays a Web page that prompts the user for the transaction ID associated with the certificate request. The user enters the transaction ID (and any password) and clicks the **Continue** button, which calls `cagetcert.rexx`.) The `cagetcert.rexx` CGI calls `R_PKIServ` for EXPORT of the certificate. If the export is successful, `cagetcert.rexx` displays the HTML under the RETURNCERT subsection. (See list item 18 on page 155.)
5. The base64-encoded certificate is displayed on the Web page by using the `[base64cert]` substitution variable.

6. This is a browser-qualified PublicKey INSERT for Internet Explorer.
7. Additional INSERTs are certificate field name INSERTs. These describe the fields using the HTML dialogs that are displayed on the Web pages if the user is allowed to input these fields. For example, PassPhrase is a text field with a maximum length of 32 characters. The two-year PKI browser certificate for authenticating to z/OS allows the user to fill in this field. (`%%PassPhrase%%` is listed in the input fields; see list item 8.)

Relationship between CGIs and the pkiserv.tmpl file

CGIs for the end-user Web pages are execs that gain control when the end user clicks an action button - for example, the **Request certificate** button on the PKI Services home page. The CGIs read the `pkiserv.tmpl` file to determine the action to perform. They resolve substitution variables in the `pkiserv.tmpl` file.

The CGIs for the end-user Web pages (including their directories) are:

- `/usr/lpp/pkiserv/PKIServ/public-cgi/camain.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/catmpl.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth/careq.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/caretrieve.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/cagorcvr.rexx`
- `/usr/lpp/pkiserv/PKIServ/clientauth-cgi-bin/cadisplay.rexx`
- `/usr/lpp/pkiserv/PKIServ/clientauth-cgi-bin/camodify.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/installcert.rexx`

Note: `installcert.rexx` is invoked from a link in the certificate renewal e-mail notification, not from a Web page.

The following table summarizes the actions the CGIs perform:

Table 38. CGI actions for end-user Web pages

CGI exec	Action	Sample Web page
`camain.rexx`	• When the user clicks **Request certificate**, this CGI calls `catmpl.rexx`, passing it a parameter identifying the selected template. • The user can click **Pick up certificate** to go directly to `caretrieve.rexx` (if the certificate is already requested). • The user can click **Renew or revoke certificate** to go to `cadisplay.rexx`.	See Figure 36 on page 309.
`catmpl.rexx`	• Displays Web page coded in the HTML under the CONTENT subsection (of a TEMPLATE section). • When the user clicks **Submit certificate request**, this CGI passes template and field name parameters to `careq.rexx`. • When the user clicks **Retrieve your certificate**, this CGI passes control to `caretrieve.rexx`.	See Figure 38 on page 316.

Table 38. CGI actions for end-user Web pages (continued)

CGI exec	Action	Sample Web page
`careq.rexx`	• Processes field names under the APPL subsection (of a TEMPLATE section). **Note:** Depending on the template, the field names can be: – `UserId` only – `UserId` and `HostIdMap`. • Processes hardcoded field names under the CONSTANT subsection (of a TEMPLATE section). • Depending on the results, displays Web page coded in the HTML under the SUCCESSCONTENT or FAILURECONTENT subsection (of a TEMPLATE section): – The SUCCESSCONTENT subsection includes a **Continue** button the user can click to continue to `caretrieve.rexx`.	See Figure 40 on page 319.
`caretrieve.rexx`	• Displays Web page coded in the HTML under the RETRIEVECONTENT subsection (of a TEMPLATE section). This HTML prompts the user to enter the transaction ID and a password if the user entered one when requesting the certificate. • When the user clicks **Retrieve and install certificate**, this CGI passes the transaction ID parameter to `cagetcert.rexx`.	See Figure 41 on page 320.
`cagetcert.rexx`	• Displays Web page coded in the HTML under RETURNCERT subsection (of a TEMPLATE section). This HTML determines which of the following forms to use when returning the certificate: – as a base64-encoded certificate (for server certificates) – as an ActiveX object (for Microsoft Internet Explorer browser certificates) – as an application/x-x509-user-certificate MIME type (for Mozilla-based browser certificates).	See Figure 43 on page 321.
`cagorcvr.rexx`	• Displays Web page coded in HTML under the FINDRECOVERCONTENT subsection of the APPLICATION section. This subsection displays security questions for users to answer. The answers to these questions can be used to recover a passphrase, which is used to recover a certificate.	
`cadisplay.rexx`	• Displays Web page coded in the HTML under the RECONTENT subsection (of the APPLICATION section). • For renewing a certificate, the user fills in the passphrase and clicks **Renew**. For revoking a certificate, the user clicks **Revoke**. Both actions call `camodify.rexx`.	See Figure 49 on page 326.
`camodify.rexx`	• Displays Web page coded in the HTML under the SUCCESSCONTENT subsection (of a TEMPLATE section) for a successful renewal. The SUCCESSCONTENT subsection includes a **Continue** button the user can click to call `caretrieve.rexx`. • Displays the Web page coded in HTML under the RESUCCESSCONTENT subsection (of the APPLICATION section) for a successful revocation.	See Figure 40 on page 319.
`installcert.rexx`	• When a user who is using the Internet Explorer browser clicks on a link in a notification e-mail for a renewed certificate, this CGI displays a Web page where the user can paste the renewed certificate sent in the e-mail and install it in the browser.	

Steps for performing minimal customization

You need to perform these steps only if you are customizing certificate templates for the first time. If your company used an earlier release of PKI Services, you do *not* need to do so again.

Before you begin: Review the certificate templates and decide if there are any that you want to remove from the `pkiserv.tmpl` certificates template file. If so, do this first. (To remove a certificate template, you can simply remove its name from the appropriate APPLICATION sections.)

Notes:

- Fields such as `%%Org%%` and `%%Country%%` are used to form the subject's distinguished name. Therefore, make sure that the name formed has a suffix that matches a suffix that the LDAP directory supports (that is, that it matches one of the suffix values in the LDAP server configuration file).
- The default name of the LDAP server configuration file is `ds.conf` for the IBM Tivoli Directory Server for z/OS LDAP server.

Perform the following steps to do the minimal updates on the remaining certificate templates:

1. For the SAF templates, update the following fields as needed:
   a. If present, replace the OrgUnit values in the following lines with values more appropriate to your organization:
      ```
      %%OrgUnit=Nuts and Bolts Division%%
      %%OrgUnit=SAF template certificate%%
      ```
   b. Replace `taca` in the following line with the correct label of the CERTAUTH signing certificate:
      ```
      %%SignWith=SAF:CERTAUTH/taca%%
      ```

   ---

2. For the PKI templates, replace the OrgUnit value in the following line with a value more appropriate for your organization:
   ```
   %%OrgUnit=Class 1 Internet Certificate CA%%
   ```

   ---

3. If present, replace `The Firm` with the name of your company in the following `%%Org` line:
   ```
   %%Org=The Firm%%
   ```

   ---

4. If your company location is not the United States, update the following line by specifying the correct two-letter country abbreviation:
   ```
   %%Country=US%%
   ```

   ---

5. If present, replace `host-name` with the domain name of this system in the following `%%HostIdMap` line:
   ```
   %%HostIdMap=@host-name%%
   ```

   You also need to follow the instructions in “Administering HostIdMappings extensions” on page 404.

   ---

6. For non-SAF certificates, you can notify users when certificate requests are rejected or when certificates are ready for retrieval or are expiring.
   a. If you do not want to have NotifyEmail appear as an input field for any non-SAF certificates, delete the NotifyEmail lines in the following locations in the TEMPLATE section for this certificate:
      - In the header:
        ```
        # NotifyEmail - optional
        ```
      - In the list of fields:
        ```
        %%NotifyEmail (optional)%%
        ```
   b. If you do not want to have NotifyEmail appear as an input field for renewal of any non-SAF certificates, delete the following NotifyEmail line in the APPLICATION section and in the list of fields:
      ```
      %%NotifyEmail (optional)%%
      ```

---

7. Insert the copyright statement for your company in the `-copyright` named field in the INSERT section.

---

8. Insert the e-mail address of your company's PKI Services administrator in the `-pagefooter` named field in the INSERT section.

Steps for additional first-time customization

You need to perform these steps only if you are customizing certificate templates for the first time. If your company used an earlier release of PKI Services, you do *not* need to perform these steps.

Perform the following steps if you want to perform additional customization of the end-user Web pages:

1. Review the templates and decide which one(s) you need to update.

---

2. If necessary, change the true name, alias, or nickname, as in the following lines.
   ```
   <TEMPLATE NAME=true_name>
   <TEMPLATE NAME=alias>
   <NICKNAME=nickname>
   ```

   true_name
   Is the whole and complete name of the certificate template.

   alias
   Differentiates browser from server certificates. An alias is not required. You can have more than one alias.

   nickname
   Is an 8-character name. SAF certificates do not have nicknames. If a nickname is not present, the certificate is not automatically renewable.

   Example:
   ```
   <TEMPLATE NAME=1-Year PKI SSL Browser Certificate>
   <TEMPLATE NAME=PKI Browser Certificate>
   <NICKNAME=1YBSSL>
   ```

---

3. If necessary, in the CONTENT subsection, change the certificate fields listed. The following example is from the one-year PKI SSL browser certificate template.

   Example:

   ```
   <p> Enter values for the following field(s)
    %%CommonName%%
    %%Requestor (optional)%%
    %%PassPhrase%%
    %%PublicKey2[browsertype]%%
   ```

---

4. If you add required fields in the preceding step, update the JavaScript code that is part of the embedded HTML to check for required fields that are missing.

   Example:

   ```
   ValidCommonName(frm) &&
   ValidPassPhrase(frm) &&
   ValidPublicKey2(frm) &&
   ```

---

5. If necessary, in the APPL subsection, change the list of certificate fields that the application provides. (Currently, the only supported fields are `UserId` and `HostIdMap`.) The following example is from the two-year PKI browser certificate for authenticating to z/OS:

   Example:

   ```
   <APPL>
    %%UserId%%
    %%HostIdMap=@host-name%%
   </APPL>
   ```

---

6. If necessary, in the CONSTANT subsection, update the list of certificate fields whose values are hardcoded. The following example is from the one-year PKI SSL browser certificate template:

   Example:

   ```
   <CONSTANT>
    %%NotBefore=0%%
    %%NotAfter=365%%
    %%KeyUsage=handshake%%
    %%OrgUnit=Class 1 Internet Certificate CA%%
    %%Org=The Firm%%
    %%SignWith=PKI:%%
   </CONSTANT>
   ```

   Note: If you update the CONSTANT subsection to create subject distinguished names, make sure that the names match the LDAP suffix defined for your LDAP server. Otherwise the certificates are not posted to LDAP. PKI Services constructs the subject distinguished name from the fields specified in the following order:

   - CommonName
   - Title
   - OrgUnit (if repeating, in the order that they appear in the template file)
   - Org
   - Locality
   - StateProv

- Country

---

7. If necessary, edit the ADMINAPPROVE subsection. (Certificates requiring an administrator's approval have an ADMINAPPROVE subsection. The absence of the ADMINAPPROVE subsection indicates auto-approval for requests.) Make sure the ADMINAPPROVE subsection, if present, correctly lists the minimum set of certificate fields that the administrator can change.

   Notes:

   a. There might be more fields in the ADMINAPPROVE subsection than fields that the user can complete in the certificate request (because the users do not necessarily see all fields).
   b. Do not include the `Requestor`, `Label`, `UserId`, `PublicKey`, or `SignWith` fields in the ADMINAPPROVE subsection; these fields cannot be changed and are ignored if present. (See 140 for a list of fields that can be in the ADMINAPPROVE subsection.)

   The following example of the ADMINAPPROVE subsection is from the one-year PKI SSL browser certificate template:

   Example:

```
<ADMINAPPROVE>
 %%CommonName (Optional)%%
 %%OrgUnit (Optional)%%
 %%OrgUnit (Optional)%%
 %%Org (Optional)%%
 %%NotBefore (optional)%%
 %%NotAfter (Optional)%%
 %%KeyUsage (Optional)%%
 %%HostIdMap (Optional)%%
 %%HostIdMap (Optional)%%
 %%HostIdMap (Optional)%%
 %%HostIdMap (Optional)%%
</ADMINAPPROVE>
```

   Note: The four `%%HostIdMap%%` lines in the example indicate that the approver can provide up to four `HostIdMap` entries.

---

8. If necessary, update the following:

   - The SUCCESSCONTENT subsection contains only the `%%-requestok%%` named field, which contains the HTML for the Web page whose main heading is "Request submitted successfully". To make changes to this Web page, update the `-requestok` INSERT (in the INSERT section of `pkiserv.tmpl`):

```
<INSERT NAME=-requestok>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based Certificate Generation Success</TITLE>
</HEAD>
<BODY>
<H1> Request submitted Successfully</H1>
[errorinfo]
<p> Here's your transaction ID. You will need it to retrieve your
certificate. Press 'Continue' to retrieve the certificate.
<p> <TABLE BORDER><TR><TD>[transactionid]</TD></TR></TABLE>
<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT NAME="TransactionId" TYPE="hidden" VALUE="[transactionid]">
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
```

- The FAILURECONTENT subsection contains only the %%-requestbad%% named field, which contains the HTML for the Web page whose main heading is "Request was not successful". To make changes to this Web page, update the requestbad INSERT:

```
<INSERT NAME=-requestbad>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based Certificate Generation Failure</TITLE>
</HEAD>
<BODY>
<H1> Request was not successful</H1>
<p> Please correct the problem or report the error to your Web admin
person<br>
<PRE>
[errorinfo]
</PRE>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
```

---

9. If necessary, update the RETRIEVECONTENT subsection.

   Note: See "Steps for changing the runtime user ID for retrieving certificates" on page 169 for directions for changing the runtime user ID for retrieving a certificate.

   a. The RETRIEVECONTENT subsection includes the %%-copyright%% named field. If you want to make any changes in the copyright statement, update the copyright INSERT. (The following is the copyright INSERT as it is originally provided in the pkiserv.tmpl file. You should have previously updated this INSERT by providing information tailored to your company, as described in "Steps for performing minimal customization" on page 160.)

```
<INSERT NAME=-copyright>
<!--
/*******************************************************************/
/*                                                                 */
/* LICENSED MATERIALS - PROPERTY OF IBM                            */
/* THIS SCRIPT IS "RESTRICTED MATERIALS OF IBM"                    */
/* 5647-A01 (C) COPYRIGHT IBM CORP. 2000,2001                      */
/*                                                                 */
/*******************************************************************/
-->
</INSERT>
```

   b. If necessary, update any desired Web page content (such as headers, footers, titles, background colors, frames, links, and so on) for the Web page whose main heading is "Retrieve Your (certificate template name)".

---

10. If you are updating the template for a server certificate, you can update the HTML in the RETURNCERT subsection to customize the returned Web page. (For a browser template, you cannot change the RETURNCERT subsection. It must contain the %%returnbrowsercert%% named field, which contains the [browsertype] substitution variable. The INSERT section contains browser-specific returnbrowsercert INSERTs.)

Steps for retrofitting release changes into the PKI Services certificate templates

If you used an earlier release of PKI Services, you might need to retrofit changes in the `pkiserv.tmpl` certificate templates file. (You would not want to replace the file if you customized it in the previous release.)

You can use a file comparison tool to compare the new PKI Services certificates template file (`/usr/lpp/pkiserv/samples/pkiserv.tmpl`) and your existing PKI Services certificates template file (`/etc/pkiserv/pkiserv.tmpl`).

Perform the following steps to retrofit changes into the `pkiserv.tmpl` certificate templates file so you do not lose any customization you made in a previous release.

1. Make a backup copy of your current certificate templates file. For example, enter from the UNIX command line:
   ```
   cp /etc/pkiserv/pkiserv.tmpl /etc/pkiserv/pkiserv.backup
   ```
2. Copy the new sample templates file to the runtime location. (This is the copy you will edit.)
   ```
   cp /usr/lpp/pkiserv/samples/pkiserv.tmpl /etc/pkiserv/pkiserv.tmpl
   ```
3. Using a compare program of your choice, compare the two template files:
   `/etc/pkiserv/pkiserv.tmpl`
   `/etc/pkiserv/pkiserv.backup`
4. Edit the runtime copy of the templates file (`/etc/pkiserv/pkiserv.tmpl`). Using the compare output generated in Step 3, merge the changes you made to the original template file into the runtime copy of the templates file.
5. Exit the file to save your changes.

Locating code for customizing end-user Web pages

For ongoing customization of end-user Web pages, you must know the code locations for those Web pages. The following table summarizes this information:

Table 39. Location of code for various Web pages

Main header (and sample Web page if any)	Location of code in pkiserv.tmpl certificate templates file
"1-Year PKI S/MIME Browser Certificate"	TEMPLATE section, CONTENT subsection
"1-Year PKI SSL Browser Certificate" (See Figure 38 on page 316.)	TEMPLATE section, CONTENT subsection
"2-Year PKI Browser Certificate For Authenticating To z/OS"	TEMPLATE section, CONTENT subsection
"2-Year PKI Authenticode - Code Signing Certificate"	TEMPLATE section, CONTENT subsection
"5-Year PKI Intermediate CA Certificate"	TEMPLATE section, CONTENT subsection

Table 39. Location of code for various Web pages (continued)

Main header (and sample Web page if any)	Location of code in pkiserv.tmpl certificate templates file
"5-Year PKI IPSEC Server (Firewall) Certificate"	TEMPLATE section, CONTENT subsection
"5-Year PKI SSL Server Certificate"	TEMPLATE section, CONTENT subsection
"5-Year SCEP certificate - Preregistration"	TEMPLATE section, CONTENT subsection
"*n*-Year PKI Certificate for Extensions Demonstration"	TEMPLATE section, CONTENT subsection
"Here's Your Certificate. Cut and Paste it to a File"	INSERT section, `-return10cert` INSERT. (This is referenced in the RETURNCERT subsection of the TEMPLATE section of each certificate template.)
"Internet Explorer Certificate Install" (See Figure 43 on page 321.)	INSERT section, `returnbrowsercertIE` INSERT
"Preregistration successful"	INSERT section, `-preregok` INSERT. (This is referenced in the SUCCESSCONTENT subsection of the TEMPLATE section of each certificate template.)
"PKI Services Certificate Generation Application" (See Figure 36 on page 309.)	APPLICATION section, CONTENT subsection
"Renew or Revoke a Browser Certificate" (See Figure 49 on page 326.)	APPLICATION section, RECONTENT subsection
"Internet Explorer Auto Renewed Certificate Install"	APPLICATION section, RENEWEDCERT subsection
"Request submitted successfully" (For submitting a successful certificate request or renewal, see Figure 40 on page 319.)	• For a successful certificate request or renewal: INSERT section, `-requestok` INSERT. (This is referenced in the SUCCESSCONTENT subsection of the TEMPLATE section of the appropriate certificate template.) • For a successful certificate revocation: INSERT section, `-renewrevokeok` INSERT. (This is referenced in the RESUCCESSCONTENT subsection of the APPLICATION section.)
"Request was not successful"	• For an unsuccessful certificate request: INSERT section, `-requestbad` INSERT. (This is referenced in the FAILURECONTENT subsection of the TEMPLATE section of each certificate template.) • For an unsuccessful certificate revocation request: INSERT section, `-renewrevokebad` INSERT. (This is referenced in the REFAILURECONTENT subsection of the APPLICATION section.)
"Retrieve Your 1-Year PKI S/MIME Browser Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 1-Year PKI SSL Browser Certificate" (See Figure 41 on page 320.)	TEMPLATE section, RETRIEVECONTENT subsection

Table 39. Location of code for various Web pages (continued)

Main header (and sample Web page if any)	Location of code in pkiserv.tmpl certificate templates file
"Retrieve Your 2-Year PKI Browser Certificate For Authenticating To z/OS"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 2-Year PKI Authenticode - Code Signing Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 5-Year PKI Intermediate CA Certificate"	TEMPLATE section RETRIEVECONTENT subsection
"Retrieve Your 5-Year PKI IPSEC Server (Firewall) Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 5-Year PKI SSL Server Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your [tmplname]"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your SAF Browser Certificate 1-Year"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your SAF Server Certificate 1-Year"	TEMPLATE section, RETRIEVECONTENT subsection
"SAF Browser Certificate 1-Year (Auto Approved)"	TEMPLATE section, CONTENT subsection
"SAF Server Certificate 1-Year (Auto Approved)"	TEMPLATE section, CONTENT subsection

Note: Fields (such as the Key Usage (KeyUsage) drop down or the Organizational Unit (OrgUnit) text field) are defined in the pkiserv.tmpl certificate templates file, in the INSERT section. (See Table 30 on page 127 for descriptions of the fields.)

Steps for adding a new certificate template

Perform the following steps to add a new certificate template:

1. Review the contents of the certificate templates provided with PKI Services to determine the one that most closely approximates the certificate template you want to add.

   _____

2. After you have determined the certificate template to use as a model, copy the section defining the model template in the certificate templates file.

   _____

3. Provide a new name, alias, and, if present, nickname for the certificate template.

   _____

4. Follow the steps for customizing certificate templates, starting at Step 3 on page 162.

   _____

Changing the runtime user ID

When the PKI Services CGIs are called, they are assigned a runtime user ID. This is the identity that is associated with the unit of work (task). This identity must be authorized to call the function being requested. (See Chapter 20, "RACF administration for PKI Services," on page 403 for more information.) Most of the templates run under the surrogate user ID (PKISERV) for requesting a certificate and for subsequently retrieving it.

There are two exceptions:

- The two SAF templates run under PKISERV for requesting a certificate but run under the client's user ID for certificate retrieval.
- The five-year PKI intermediate CA template runs under the client's user ID for requesting a certificate and for certificate retrieval.

The advantage of having PKISERV as the runtime user ID is that this is the only user ID that needs to be authorized for requesting certificates. The advantage of using the client's user ID is that you have greater control over who can request and retrieve certificates. For example, you can require the user to authenticate by entering user ID and password before requesting or retrieving a certificate.

You can control the user ID under which a certificate request or retrieval runs by selectively commenting and uncommenting FORM statements in the `pkiserv.tmpl` file. (For requesting a certificate, the FORM statements are in the appropriate TEMPLATE section, in the CONTENT subsection. For retrieving a certificate, the FORM statements are in the appropriate TEMPLATE section, in the RETRIEVECONTENT subsection.)

There are three levels of access control for requesting and retrieving certificates:

- Under the client's ID with user ID and password authentication
- Under the surrogate user ID with user ID and password authentication
- Under the surrogate user ID without user ID and password authentication.

Protection directives in the z/OS HTTP Server configuration file (which defaults to `/etc/httpd.conf`) enforce these three levels of access control. The default configuration for PKI Services maps the three levels of access control to the following CGI directories respectively:

- `/PKIServ/ssl-cgi-bin/auth`
- `/PKIServ/ssl-cgi-bin/surrogateauth`
- `/PKIServ/ssl-cgi-bin`

Each of the request and retrieve CGIs reside in all three directories. Thus, when you run a CGI you get the protection established for the directory from which it is called.

Each certificate template contains several FORM statements (two commented out and one uncommented, which is active) that determines which of these applies. You can change the access control by uncommenting one of the FORM statements that is commented out and commenting out the one that is active.

Steps for changing the runtime user ID for requesting certificates

Perform the following steps to change the runtime user ID for requesting a certificate.

1. In the pkiserv.tmpl file, find the CONTENT subsection of the TEMPLATE section for the template whose user ID you want to change. Locate the lines containing the FORM statements, such as those in the following example:

   Example:

```
<h3><li>Request a New Certificate
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#                "/PKIServ/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#              "/PKIServ/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
 <FORM NAME="CertReq" METHOD=POST ACTION=
               "/PKIServ/ssl-cgi-bin/careq.rexx" onSubmit=
```

   Notice that the preceding lines contain three FORM statements. The first two FORM statements are commented out, so they are not active. They are for:

   - Requesting the certificate under the client's ID and using user ID and password authentication
   - Requesting the certificate under the surrogate ID and using user ID and password authentication

   The third FORM statement is for requesting the certificate under the surrogate user ID without user ID and password authentication. This is active (it is not commented out).

---

2. To change the runtime user ID, remove the comment delimiter (#) from in front of the lines for the commented-out FORM statement you want to use and insert the comment delimiter in front of the lines for the bottom FORM statement.

Steps for changing the runtime user ID for retrieving certificates

Perform the following steps to change the runtime user ID for retrieving a certificate.

1. In the pkiserv.tmpl file, find the RETRIEVECONTENT subsection of the TEMPLATE section for the template whose user ID you want to change. Locate the lines containing the FORM statements, such as those in the following example:

   Example:

```
<H1> Retrieve Your [tmplname]
<H3>Please bookmark this page
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#      "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
```

```
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#      "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
 <FORM NAME=retrieveform METHOD=POST ACTION=
       "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
```

Notice that the preceding lines contain three FORM statements. The first two FORM statements are commented out (they are not active). These are for:

- Retrieving the certificate under the client's ID
- Retrieving it under the surrogate ID, but requiring user ID and password authentication.

The third FORM statement is for retrieving the certificate under the surrogate user ID without user ID and password authentication. This is active (it is not commented out).

---

2. To change the runtime user ID, remove the comment delimiter (#) from in front of the lines for the commented-out FORM statement you want to use and insert the comment delimiter in front of the lines for the bottom FORM statement.

Customizing the OtherName field

When you use the OtherName field, you are able to bind additional identities or owner information to the subject of the certificate using the subject alternate name extension. These identities might take different forms, such as employee numbers, customer account numbers, and other identities that you choose to use.

The OtherName value is a concatenated string that consists of one or more pairs of OIDs and their associated values. The string is saved in the subject alternate name extension in the certificate.

PKI Services implements the OtherName field as a customizable INSERT called `AltOther_<OID>`. The following certificate template in `pkiserv.tmpl` is supplied to illustrate the use of the INSERT fields.

```
Template Name - n-Year PKI Certificate for Extensions Demonstration
```

The *n*-year PKI certificate template builds a certificate using information provided primarily by users, rather than information that you control. For demonstration purposes, the template builds a certificate that contains all extensions supported by PKI Services. The template contains two sample OtherName fields:

`%%AltOther_1_2_3_4_5%%`	Builds one input field.
`%%AltOther_1_2_3_4_6%%`	Builds two input fields.

The `AltOther_1_2_3_4_5` string represents an OtherName field with OID `1.2.3.4.5`, an 11-character string that stores a customer account number. The `AltOther_1_2_3_4_6` string represents an OtherName field with OID `1.2.3.4.6`, a 17-character string that stores a 9-digit license number and an expiration date in the *yyyymmdd* format.

When you choose to use the OtherName field to build the subject alternate name extension, you might also want to customize the end-user Web pages to allow end-users to enter the required information using customized input screens that will be easier for them to use. For example, rather than asking a user to enter a string like the one shown below, you can prompt the user to enter a 9-digit license number and its expiration date.

Example of an OtherName field value:

```
1.2.3.4.6,12345678920050215
```

Steps for customizing the sample AltOther_<*OID*> INSERTs

Before you begin:

- Decide what identifiers you wish to add to the Subject Name Alternate extension.
- Select the registered OID value to use to represent your data string. Check the appropriate standards organization (ISO or ITU). If not already registered, register your own OID.
- Select which certificate templates you will update to add the Subject Name Alternate extension.
- Decide whether to use a sample INSERT for your `AltOther_<OID>` INSERT or create your own INSERT. The sample INSERT called `AltOther_1_2_3_4_5` demonstrates using one input field. The sample insert called `AltOther_1_2_3_4_6` demonstrates using two input fields.
- Determine the following values you will use to customize your INSERT.
  - The OID value for your OtherName field
  - The name and length for each input field.
- Review Figure 3 on page 173. It contains a listing of the sample INSERT called `AltOther_1_2_3_4_6` which demonstrates using two input fields. The lines you are most likely to customize are marked in Figure 3 on page 173. The following steps refer to the marked lines.

Perform the following steps to customize the `AltOther_<OID>` INSERT using Figure 3 on page 173 as a reference.

1. Change the OID value `_1_2_3_4_6` to the OID value you need in the line marked **1** and in all other lines in the sample INSERT. For example, if you chose OID `2.16.76.1.3.1` for your OtherName field, change all occurrences of `AltOther_1_2_3_4_6` to `AltOther_2_16_76_1_3_1`.

   _____

2. Customize the first input field description in the line marked **2** to prompt users of your Web page. For example, change `Customer's driver license number (9 digits)` to `Enter your member card number`.

   _____

3. Customize the first INPUT field name "`Other2a`" to your value in the line marked **3** and in all other lines in the sample INSERT. For example, change all occurrences of "`Other2a`" to "`MemNum`". Also, customize `SIZE` and `maxlength` as needed.

   _____

4. Customize the next input field description in the line marked **4** to prompt users of your Web page. For example, change `Customer's driver license expiration date (yyyymmdd)` to `Enter your date of birth (yyyymmdd)`.

   _____

5. Customize the next INPUT field name "Other2b" to your value in the line marked **5** and in all other lines in the sample INSERT. For example, change all occurrences of "Other2b" to "Birthdate". Also, customize SIZE and maxlength as needed.

---

6. Customize the starting positions and lengths for each input field value in the lines marked **6** and **7**. For example, if the member card number is an 11-digit number, change
`form.Other2a.value=form.altrawstring_1_2_3_4_6.value.substr(0,9)` to
`form.MemNum.value=form.altrawstring_2_16_76_1_3_1.value.substr(0,11)`.

---

7. Customize the validation script that begins with the line marked **8**.

---

8. Change the OID value 1.2.3.4.6 to the OID value you need in the line marked **9**. For example, if you chose OID 2.16.76.1.3.1 for your OtherName field, change 1.2.3.4.6 to 2.16.76.1.3.1.

---

9. Repeat steps 2 through 8 for each additional input field you need.

```
 ====================================================================
# Sample AltOther INSERT with two input fields
# ====================================================================
<INSERT NAME=AltOther_1_2_3_4_6> 1
<INPUT NAME="AltOther_1_2_3_4_6" TYPE="hidden" maxlength="255">

<p> Other Name for alternate name: <BR>
<p> <LABEL for="other2afield">Customer's driver license number (9 digits) [optfield]</LABEL> <BR> 2
<INPUT NAME="Other2a" TYPE="text" SIZE=9 maxlength="9" [readonly]  id="other2afield"> 3
<p> <LABEL for="other2bfield">Customer's driver license expiration date (yyyymmdd) [optfield]</LABEL> <BR> 4
<INPUT NAME="Other2b" TYPE="text" SIZE=8 maxlength="8" [readonly]  id="other2bfield"> 5

<INPUT NAME="altrawstring_1_2_3_4_6" TYPE="hidden" VALUE="[altrawvalue]">

<SCRIPT LANGUAGE="JavaScript">
<!--
//This is the script that will be called at load time.
var form=document.forms[0]
if (form.altrawstring_1_2_3_4_6.value.length > 0) {
  //The name 'Otherx' needs to match with the above INPUT NAME.
  //Substr(start position, length)
  form.Other2a.value=form.altrawstring_1_2_3_4_6.value.substr(0,9) 6
  form.Other2b.value=form.altrawstring_1_2_3_4_6.value.substr(9,8) 7
}
//-->
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript"> 8
<!--
//This is the validation script
function ValidAltOther_1_2_3_4_6(frm){
 if (("[optfield]" == "" && frm.Other2a.value.length != 9) ||
     ("[optfield]" !="" && frm.Other2a.value != "" && frm.Other2a.value.length != 9)) {
  alert("Enter 9 digit license number.");
  frm.Other2a.focus();
  return false;
 }
.
.
.
 //Build the entire AltOther field.
 if (frm.Other2a.value != "" && frm.Other2b.value != "")
  frm.AltOther_1_2_3_4_6.value = "1.2.3.4.6," + frm.Other2a.value + 9
                       frm.Other2b.value;
 else
  frm.AltOther_1_2_3_4_6.value = "";
 return true;
}
//-->
</SCRIPT>
</INSERT>
```

Figure 3. Partial listing of the AltOther_1_2_3_4_6 sample INSERT showing the lines you are most likely to customize

Chapter 12. Customizing the administration Web pages if you use REXX CGI execs

This information applies if you are using REXX CGI execs for your PKI Services Web pages. If you are using Java server pages (JSPs), see Chapter 13, "Implementing the Web application using Java server pages," on page 179.

CGIs for administration Web pages

CGIs for administration Web pages are execs that gain control when the user clicks an action button and render the Web pages dynamically. All of the administration CGIs are contained in the `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth` directory.

Table 40 (which lists the CGI execs in logical order) summarizes the actions they perform:

Table 40. CGI actions for administrative Web pages

CGI exec	Action	Sample Web page
`admmain.rexx`	This displays the administration home page. The main heading is "PKI Services Administration". This Web page lets the administrator work with a single certificate request or certificate or search for certificate requests or certificates.	See Figure 61 on page 340.
`admpend.rexx`	On the administration home page, the administrator can search for certificate requests. This displays a Web page whose main heading is one of the following: • "Certificate Requests" Web page—This lists certificate requests matching the criteria and allows the administrator to process the certificate request(s). • "Processing was not successful" Web page	For an example of the "Certificate Requests" Web page, see Figure 66 on page 345.
`admpendtid.rexx`	On the administration home page, the administrator can enter a transaction ID to work with a single certificate request. This displays a Web page whose main heading is one of the following: • "Single Request" - This lists the certificate request that matches the transaction ID and allows the administrator to process that certificate request. • "Processing was not successful"	For an example of the "Single Request" Web page, see Figure 62 on page 341.
`admmodtid.rexx`	This displays the "Modify and Approve Request" Web page that appears when the administrator decides to modify a request before approving it (on the "Single Request" Web page).	See Figure 65 on page 343.
`admicl.rexx`	On the administration home page, the administrator can search for certificates. This displays a Web page whose main heading is one of the following: • "Issued Certificates" - This lists the certificate(s) that match the search criteria and allows the administrator to revoke or delete selected certificate(s). • "Processing was not successful"	For a sample of the "Issued Certificates" Web page, see Figure 66 on page 345.

© Copyright IBM Corp. 2001, 2011

Table 40. CGI actions for administrative Web pages (continued)

CGI exec	Action	Sample Web page
admiclcert.rexx	On the administration home page, the administrator can enter a serial number to work with a single certificate. This displays a Web page whose main heading is one of the following: • "Single Issued Certificate" - This lists the certificate that matches the serial number ID and allows the administrator to revoke or delete that certificate. • "Processing was not successful"	For a sample of the "Single Issued Certificate" Web page, see Figure 70 on page 350.
admacttid.rexx	Displays a Web page after the administrator processes a single certificate request (approving it with or without modifications, rejecting, or deleting it). This Web page has one of the following as its main heading: • "Processing successful" • "Processing was not successful"	For a sample of the Web page whose main heading is "Processing successful", see Figure 63 on page 342.
admacttid2.rexx	This displays a Web page after the administrator approves a certificate request with modifications. The Web page has one of the following main headings: • "Processing successful" • "Processing was not successful"	For a sample of the Web page whose main heading is "Processing successful", see Figure 63 on page 342.
admpendall.rexx	After the administrator searches for certificate requests and admpend.rexx displays the results, the administrator clicks a button to approve, reject, or delete selected certificate requests. This calls admpendall.rexx, whose main heading is one of the following: • "Processing successful" if the action was successful • "Processing was not successful" if the action failed (for example, if the administrator tried to delete certificate requests that were already deleted) • "Processing partially successful" if not all of the selected requests are processed successfully	• For an example of the "Processing successful" Web page, see Figure 67 on page 347. • For an example of the "Processing was not successful" Web page, see Figure 68 on page 347. • For an example of the "Processing partially successful" Web page, see Figure 69 on page 348.
admactcert.rexx	Displays a Web page after the administrator tries to revoke or delete one or more selected certificates. The Web page has one of the following main headings: • "Processing successful" • "Processing was not successful"	—

Table 40. CGI actions for administrative Web pages (continued)

CGI exec	Action	Sample Web page
admiclall.rexx	After the administrator searches for certificates and admicl.rexx displays the results, the administrator clicks a button to revoke or delete selected certificates. This calls admiclall.rexx, which displays a Web page whose main heading is one of the following: • "Processing successful" if the action was successful • "Processing was not successful" if the action failed • "Processing partially successful" if not all of the selected certificates are processed successfully	—

Customizing the administration Web pages

The administration Web pages are not as customizable as the end-user Web pages. You can customize page headers, footers, frames, links, colors, and so forth, but you cannot change internal Web page content. Except for identifying the fields that an administrator can change when approving certificate requests, the administration Web page logic is fixed.

However, you can make changes in the following subsections in the PKISERV APPLICATION section of the pkiserv.tmpl certificate template file. (These subsections appear in the application section of PKISERV only.)

ADMINHEADER Contains the general installation-specific HTML content for the header of all the administration Web pages.

ADMINFOOTER Contains the general installation-specific HTML content for the footer of all the administration pages.

ADMINSCOPE This optional subsection allows the administrator to choose a different CA domain. For more information, see "Adding a new CA domain" on page 230.

Steps for customizing the administration Web pages

Perform the following steps to customize the administration Web pages:

1. Add any desired Web page header for the administration pages to the ADMINHEADER subsection of the PKISERV APPLICATION section. (The ADMINHEADER subsection is near the end of the APPLICATION section.)

   Example:

```
<ADMINHEADER>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE>Web-Based Certificate Generation Administration</TITLE></HEAD>
<BODY>
</ADMINHEADER>
```

2. Add any desired Web page footer for the administration pages to the ADMINFOOTER subsection of the APPLICATION section. (The ADMINFOOTER subsection is near the end of the APPLICATION section.)

   Example:

```
<ADMINFOOTER>
<p> email: webmaster@company.com
</BODY>
</HTML>
</ADMINFOOTER>
```

Changing the runtime behavior for accessing administration pages

When the administrator tries to access the administration pages (by clicking the **Go to administration page** button on the PKI Services home page), access to the administration pages is controlled in one of the following ways:

- A popup window appears, requiring the administrator to enter a user name and password. (See Figure 60 on page 337 for a sample of the authentication popup window.)
- Alternately, the administrator might have to authenticate by using a previously issued browser certificate. In other words, the administrator would need to have a certificate before visiting the administration Web pages.

By default, the first method is used. However, you can change the runtime behavior so that the second method is used instead. If you decide to use the second method, anyone intending to become a PKI Services administrator needs to request and retrieve a one-year PKI browser certificate for authenticating to z/OS before trying to access the administration pages.

Note: The one-year PKI browser certificate for authenticating to z/OS contains a `HostIdMappings` extension. (For more information, see Chapter 20, "RACF administration for PKI Services," on page 403.)

Steps for changing control of access to administration pages

Perform the following steps to change the access control of the administration pages to require authenticating by using a certificate:

1. Edit the `pkiserv.tmpl` certificate templates file and find the following lines in the PKISERV APPLICATION section:

   ```
   # The following action will force userid/pw authentication for administrators
   <FORM name=admform METHOD=GET ACTION="/PKIServ/ssl-cgi/auth/admmain.rexx">
   # The following action will force client certificate authentication
   # for administrators
   #<FORM name=admform METHOD=GET
   # ACTION="/PKIServ/clientauth-cgi/auth/admmain.rexx">
   <p>
   <INPUT TYPE="submit" VALUE="Go to Admin Pages">
   </FORM>
   ```

   The first FORM statement in these lines is active. (It is not commented out with # characters in front of the lines.) This requires authentication by entering the user name and password in a popup window. The second FORM statement is commented out (using # characters). This requires authentication by using a previously issued browser certificate.

   ---

2. Comment out the first FORM statement (add # characters in front of the FORM and ACTION lines) and uncomment the second FORM statement (removing the # characters in front of the FORM and ACTION lines).

Chapter 13. Implementing the Web application using Java server pages

As an alternative to the REXX CGI execs and text template file described in Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs," on page 123 and Chapter 12, "Customizing the administration Web pages if you use REXX CGI execs," on page 175, you can use Java server pages (JSPs) and an XML template file to create and customize the PKI Services Web application. This approach has several advantages over the REXX CGI approach:

- It uses Java, a popular and flexible Web application programming language.
- It uses XML, which is likely to be more familiar and intuitive to Web application programmers than the text template file format used by the REXX CGI approach.
- You can validate the syntax of your XML template file using the TemplateTool utility (see "Using the TemplateTool utility" on page 370) or Web application tools that include XML validation, such as IBM Rational® Software Development Platform.
- The XML template with modifiable JSP files is a simpler structure than the single text template file used by the REXX CGI approach.

Java server pages control the content or appearance of Web pages through the use of Java code that runs on an application server to modify a Web page before it is sent to the user who requested it. Java server pages can contain a mixture of HTML and Java code. For the PKI Services Web application, the XML template file defines the applications and the certificates that the applications provide. The Java server pages define and process the Web pages. The application server is assumed to be Websphere Application Server.

Certificate templates files used with JSPs

When you implement the Web application using JSPs, there are two versions of the certificate templates file:

- An XML templates file, `pkitmpl.xml`, used to customize the Web application
- A text CGI templates file, `pkixgen.tmpl`, used by the PKI Services daemon

Whenever you update `pkitmpl.xml`, you must use the TemplateTool utility to create an equivalent copy of `pkixgen.tmpl`. For more information, see "Using the TemplateTool utility" on page 370. If you do not create an updated `pkixgen.tmpl` file, the daemon writes the following message to the daemon log file when it determines that `pkixgen.tmpl` is not current:

```
IKYC068I The templates file used may not be current
```

The default location of the two versions of the template file is `/etc/pkiserv/`. Alternatively, if the environment variable PKISERV_CONFIG_PATH is defined, PKI JSP processing uses the value of PKISERV_CONFIG_PATH as the location of `pkitmpl.xml` and `pkixgen.tmpl` . The XML schema is defined in the file `PKIServ.xsd` in the same directory as `pkitmpl.xml`.

© Copyright IBM Corp. 2001, 2011

Examining the pkitmpl.xml file

The PKI Services XML template, `pkitmpl.xml`, defines:

- A root CA URL.

  The root CA URL is the URL of the PKI Services CA certificate.
- The PKI Services ActiveX install URL for Windows XP and earlier versions of Windows.

  This URL is the URL of the Windows installer program that installs the PKI Services ActiveX control on Microsoft Windows XP and earlier versions of Windows.
- The PKI Services ActiveX install URL for Windows Vista and later versions of Windows.

  This URL is the URL of the Windows installer program that installs the PKI Services ActiveX control on Microsoft Windows Vista and later versions of Windows.
- One or more applications.

  An application is a grouping of certificate request templates. This grouping might be done because the templates are shared among a set of end users or because the templates have a common administrator. A PKI Services installation can have one or more applications defined.
- One or more certificate request templates.

  A certificate request template is a predefined set of characteristics for certificate requests and the resulting certificates. Each certificate request template defines:
  - The type of certificate request (for example, browser or server, SCEP preregistration request, SAF or PKI certificate)
  - A name for the certificate request template, which is displayed to both the end user and the administrator
  - A nickname (maximum of 8 characters) that uniquely identifies each template, which PKI Services uses to retrieve information about the template
  - The values you want a user to input and whether those values are optional or required
  - The values you want to supply for the user
  - The values an administrator is allowed to supply
  - Whether the certificate is automatically approved
  - Whether the certificate is automatically renewed
  - A set of preregistration rules (applicable only to SCEP preregistration requests), which define whether approval is needed based on the level of authentication provided

The XML template file, `pkitmpl.xml`, begins by defining the URL of the PKI Services CA certificate.

```
<tns:CA_cert_URL>/PKIServ/cacerts/cacert.der</tns:CA_cert_URL>
```

The CA_cert_URL is the URL of the PKI Services CA certificate. The URL can be relative to the PKIServ Web root context or absolute. This URL is displayed as a link with the text “Install the CA certificate to enable SSL sessions for PKI Services” on the home page for both end users and administrators. Users need to install the PKI Services CA certificate in order to use a secure connection and retrieve certificates.

Next `pkitmpl.xml`defines the PKI Services ActiveX install URLs.

```
<tns:XEnroll_install_URL>http://hostname:port/PKIServ/PKIXEnroll/PKIXEnrollDeploy.msi<tns:XEnroll_install_URL>
<tns:CEnroll_install_URL>http://hostname:port/PKIServ/PKICEnroll/PKICEnrollDeploy.msi<tns:CEnroll_install_URL>
```

The tag for `XEnroll_install_URL` specifies the URL of the PKI Services ActiveX control installer program for Windows XP and earlier versions of Windows. The tag for `CEnroll_install_URL` specifies the URL of the PKI Services ActiveX control installer program for Windows Vista and later versions of Windows. Each URL can be relative to the PKI Services Web root context or absolute. The text "Install the PKI ActiveX Control to renew certificates" on the PKI Services home page for end users (see Figure 36 on page 309) links to the URL specified for the version of Microsoft Windows running on the system. Users need to install the PKI Services ActiveX control to install renewed certificates using the Internet Explorer browser.

The application tag (`<tns:application>`) defines a particular set of end users. The application tag consists of an application name (in this case Customers) and one or more application templates (`<tns:appltemplate>`).

```
<tns:applname>Customers</tns:applname>
<tns:appltemplate>1-Year PKI SSL Browser Certificate</tns:appltemplate>
```

The contents of the `appltemplate` tag ("1-Year PKI SSL Browser Certificate", for example) corresponds to the `certname` element of a `certreq_template` tag.

A certificate request template is defined by a `certreq_template` tag, shown in Figure 4.

```
<tns:certreq_template>
     <tns:certname>1-Year PKI SSL Browser Certificate </tns:certname>
     <tns:certtype>PKI Browser Certificate</tns:certtype>
     <tns:certtype_description>PKI Browser Certificate for Secured Connections</tns:certtype_description>

⋮
```

Figure 4. A `certreq_template` tag

The certificate type tag (`<tns:certtype>`) can have one of the following values:

- `PKI Preregistration` (for SCEP preregistration requests)
- `PKI Browser Certificate`
- `PKI Server Certificate`
- `PKI Key Certificate` (for a certificate for which PKI Services has generated the key pair)
- `SAF Browser Certificate`
- `SAF Server Certificate`

The certificate type description (`<tns:certtype_description>`) is an optional tag. Its contents are used on the Web pages wherever the certificate type is to be displayed. This tag allows administrators to use words that they feel might be more understandable to their end users than the pre-defined values for the certificate types. A common use of this tag might be to translate the certificate types to another language. If this tag is omitted the contents of the certificate type tag (`<tns:certtype>`) is used as the certificate type description.

The request authentication type tag (`<tns:request_authtype>`) and retrieve authentication type tag (`<tns:retrieve_authtype>`) >) define the type of authentication that must be used to request or retrieve a certificate. The acceptable values for these tags are:

`noAuthRunAsSurrogate`
: No authentication should be used. The task runs as a surrogate user.

zAuthRunAsSurrogate
The user will be prompted to authenticate (log in) to z/OS using a RACF user ID and password. The task runs as a surrogate user.

zAuthRunAsClient
The user will be prompted to authenticate (log in) to z/OS using a RACF user ID and password. The task runs as the client.

The next element of a certificate request template is the Auto-Approve indicator (`<tns:AutoApprove>`). A value of `Y` or `y` indicates that any certificate requests made with this template should be automatically approved (no administrator approval is required and the administrator will not have an opportunity to modify or reject certificate requests). A value of `N` or `n` indicates that certificate requests made with this template are not automatically approved and must be approved by an administrator.

The next element of a certificate request template is the Auto-Renew indicator (`<tns:AutoRenew>`). A value of `Y` or `y` indicates that any certificate created using this template will be automatically renewed. A value of `N` or `n` indicates that certificates created using this template are not automatically renewed.

The following form fields are defined with tags in the certificate request template:
- AltDomain
- AltEmail
- AltIPAddr
- AltOther
- AltURI
- ClientName
- CommonName
- Country
- CustomExt
- DomainName
- DNQualifier
- EmailAddr
- ExtKeyUsage
- HostIdMap
- KeySize
- KeyUsage
- Label
- Locality
- Mail
- NotAfter
- NotBefore
- NotifyEmail
- Org
- OrgUnit
- PassPhrase
- PostalCode
- PublicKey
- Requestor
- Security
- SerialNumber
- SignWith
- StateProv
- Street
- Title

- Uid
- UnstructAddr
- UnstructName
- UserId

A form field tag has the form:

```
<tns:Name formtype="InstallationSpecified | AdminSpecified | UserSpecified"
                 initvalue="xxxxx"
                 optional="true | false"
                 JSPfilename="xxx.jsp"/>
```

where

Name is the name of the form field, for example AltOther or Security.

formtype
can have one of the following values:

UserSpecified
The form field appears on the certificate request Web page where the user can enter data.

InstallationSpecified
The value is provided by the XML template and is not displayed on the certificate request Web page. Instead, there is a hidden form field on the certificate request Web page that specifies the value.

AdminSpecified
The administration Approve with Modification Web page should always display this form field and allow an administrator to specify a value for it.

The default for `formtype` is `UserSpecified`.

initvalue
The initial value of the form field. If `formtype` is `AdminSpecified` or `UserSpecified`, the form field is displayed with this initial value set but modifiable. If `formtype` is `Installation Specified`, this initial value is given to the hidden form field and it cannot be changed for the certificate request.

optional
Indicates whether this form field is optional. For `UserSpecified` form fields, `optional` indicates whether a value must be provided on the Certificate Request Web page. For `AdminSpecified` form fields, `optional` indicates whether a value must be provided on the administrator's Approve with Modifications Web page. For `InstallationSpecified` form fields, the optional attribute is ignored. The default value for `optional` is false, and the form field is required.

JSPfilename
The filename of a JSP file that is included to display and validate the form field. The file that is included is a modifiable include file in the `mod_inc` directory with your Web application's EAR file. If `formtype` is `Installation Specified`, `JSPfilename` is ignored.

The default value for `JSPfilename` is the name of the form field, in lowercase, combined with the `.jsp` extension. For example, the default value for `JSPfilename` for the form field tag for PassPhrase would be `passphrase.jsp`.

Rules: You can write your own JSP files to process form fields, but they must conform to the following rules:

- The HTML form field, whether it is a select field, an input field, a text area field, or a hidden form field, must have the same name as the form field tag, but in lower case. For example, for the tag

  ```
  <tns:CommonName formtype="AdminSpecified" optional="true" JSPfilename="cn.jsp"/>
  ```

  the form field must have the name `commonname` in the file `cn.jsp`.
- The HTML form field must contain a JavaScript function with the name `Valid` concatenated with the lowercase form field name. For example, for the tag

  ```
  <tns:CommonName formtype="AdminSpecified" optional="true" JSPfilename="cn.jsp"/>
  ```

  `cn.jsp` must contain a JavaScript method named `Validcommonname`. This JavaScript method should verify the form field and return `true` if it is valid and `false` if not. If there is no verification to be done, the Valid*formfieldname* method can return true in all cases.

Each of the attributes (`formtype`, `initvalue`, `optional` and `JSPfilename`) can be omitted and defaulted.

Examples of form field tags:

```
<tns:CommonName formtype="AdminSpecified" optional="true" JSPfilename="cn.jsp"/>
<tns:PassPhrase />
```

Roadmap for implementing the PKI Services Web application using JSPs

To implement the PKI Services Web application using JSPs, perform the tasks in Table 41.

Table 41. Task roadmap for implementing the PKI Services Web application using the JSPs

Subtask	Associated instructions (see . . .)
Preparation	"Steps for preparing to implement the PKI Services Web application using JSPs" on page 185
Give Websphere users authorization to use PKI Services functions	"Giving Websphere users authorization to use PKI Services functions" on page 185
Set up a Websphere SSL configuration that uses client authentication, to allow Websphere users to renew and revoke browser certificates	"Allowing Websphere users to renew and revoke browser certificates" on page 189
Customize the Web application	"Customizing the PKI Services Web application" on page 198
• Update the template file.	• "Updating the template file" on page 198
• (Optional) Modify the JSP files and update the EAR file with the modified JSP files.	• "(Optional) Modifying the JSP files and the EAR file" on page 198
• Deploy the EAR file to a Websphere application server.	• "Deploying the EAR file to a Websphere application server" on page 200

Steps for preparing to implement the PKI Services Web application using JSPs

Perform the following steps to prepare to implement the PKI Services Web application using JSPs.

1. Set the _PKISERV_ENABLE_JSP environment variable to indicate that you are using the JSPs instead of the REXX CGIs. To do this, uncomment the following line in the environment variables file `pkiserv.envars`:

   ```
   _PKISERV_ENABLE_JSP=TRUE
   ```

   (If the line is not in your copy of the environment variables file, add it.)

   ---

2. If you have not previously done so, copy the XML template file and the XML schema file from the directory in which the MVS programmer installed PKI Services to the runtime directory by entering the following commands from the UNIX command line. The default directories are `/usr/lpp/pkiserv/` and `/etc/pkiserv` respectively. The user ID you use for copying files must have superuser authority.

   ```
   cp -p /install-dir/samples/pkitmpl.xml runtime-dir
   cp -p /install-dir/samples/PKIServ.xsd runtime-dir
   ```

Note: You do not copy the file `pkixgen.tmpl`. PKI Services does not ship a copy of this file in the `samples` directory. You will generate `pkixgen.tmpl` from `pkitmpl.xml` after you customize `pkitmpl.xml`.

When you are done, you are ready to customize the PKI Services Web application.

Giving Websphere users authorization to use PKI Services functions

You need to give each Websphere user that will use the PKI Services Web application authorization to use PKI Services functions.

Steps for giving Websphere users authorization to use PKI Services functions

Before you begin: You need RACF administration skills and you must have the RACF SPECIAL attribute. You need Websphere administration skills.

Perform the following steps to give Websphere users authorization to use PKI Services functions

1. Log on to the Websphere administrative console.
2. Configure Websphere for application security using SAF authorization.
   a. In the left pane, under Security, click **Secure administration, applications, infrastructure**, and verify that the **Enable application security** checkbox is selected, as shown in Figure 5 on page 186. If it is not, select it and click **Apply**.

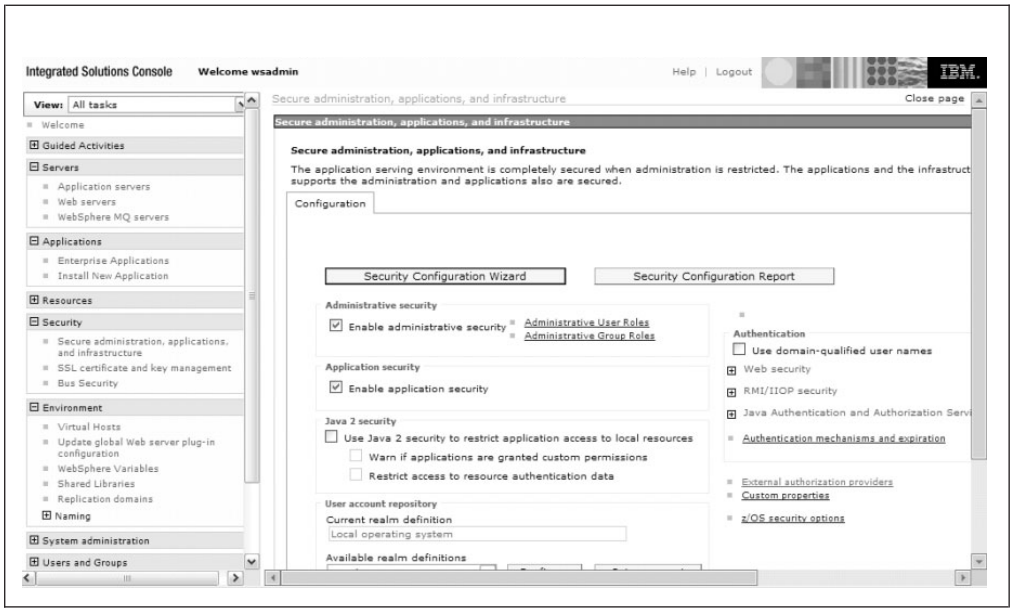


Figure 5. Configuring Websphere for application security

b. Click **External authorization providers**, and verify that **System Authorization Facility (SAF) authorization** is selected, as shown in Figure 6. If it is not, click **System Authorization Facility (SAF) authorization**, and then click **Apply**.

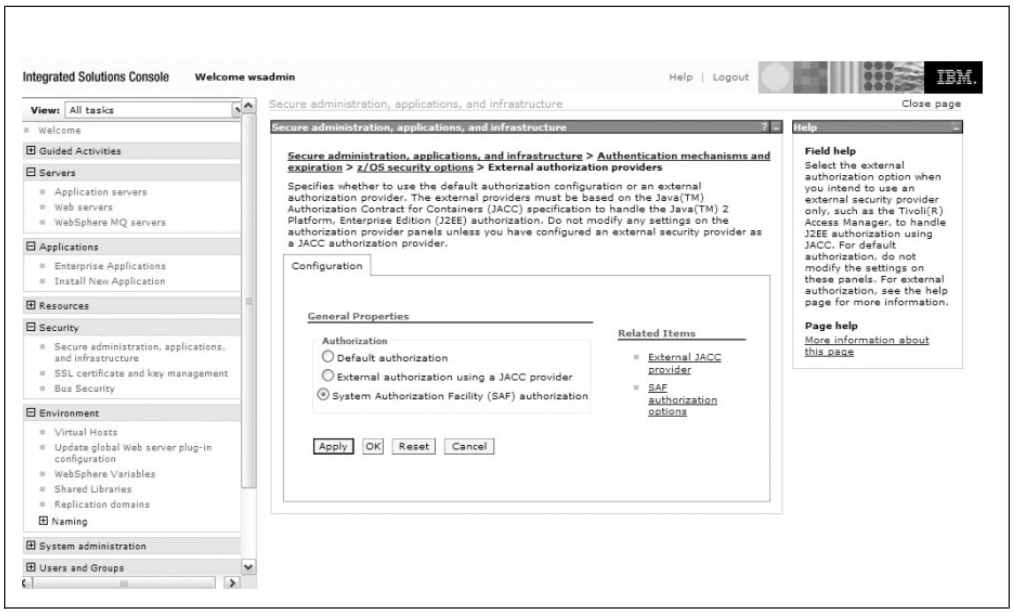


Figure 6. Configuring Websphere for application security using SAF authorization

c. From the web page shown in Figure 5, click **z/OS security options** and select the **Enable application server and z/OS thread identity synchronization** check box, then click **OK**.

d. If you changed any settings, you see the message shown in Figure 23 on page 201 indicating that changes have been made to your local configuration. Click **Save** to save the changes to the master configuration.

---

3. The web.xml file for PKIServ_Web defines two roles, SAFuser and PKIAdmin. You need the SAFuser role only if you want to specify z/OS authentication (either zAuthRunAsSurrogate or zAuthRunAsClient) for certificate request or retrieval authentication. In the sample pkitmpl.xml file, z/OS authentication is used for these certificate templates: 1-Year SAF Browser Certificate, 1-Year SAF Server Certificate, 2-Year PKI Browser Certificate For Authenticating To z/OS, 5-Year SCEP Certificate - Preregistration , and 5-Year PKI Intermediate CA Certificate. The PKIAdmin role allows users assigned it to use the PKI Services administration Web pages.

   Enter the following TSO commands to create the SAFuser and PKIAdmin roles. The first RDEFINE command creates the the SAFuser role and gives it to all users who have authenticated. The second RDEFINE command creates the PKIAdmin role.

   ```
   RDEFINE EJBROLE SAFuser UACC(READ)
   RDEFINE EJBROLE PKIAdmin UACC(NONE)
   ```

   Then enter *one* of the following commands. The first PERMIT command must be entered for *each* user who should have authorization to use the PKI Services administration Web pages.

   ```
   PERMIT PKIAdmin CLASS(EJBROLE) ID(userid) ACCESS(READ)
   ```

   or

   ```
   PERMIT PKIAdmin CLASS(EJBROLE) ID(pkigrp) ACCESS(READ)
   ```

   where *pkigrp* is the value of the variable pkigroup in the IKYSETUP exec (default value PKIGRP). For a description of the pkigroup variable, see Table 17 on page 43.

   Note: If your installation uses security domains the role name is qualified by the security domain, for example:

   ```
   RDEFINE EJBROLE securitydomain.SAFuser UACC(READ)
   RDEFINE EJBROLE securitydomain.PKIAdmin UACC(NONE)
   PERMIT securitydomain.PKIAdmin CLASS(EJBROLE) ID(PKIGRP) ACCESS(READ)
   ```

---

4. Issue TSO commands to give Websphere users authorization to use PKI Services.

   a. Give the user ID for the Websphere servant region address space READ access to the FACILITY class profile IRR.RPKISERV.*function.ca-domain*. ASSR1 is the user ID for the Websphere servant region address space. For unnamed domains issue the command:

      ```
      PERMIT IRR.RPKISERV.* CLASS(FACILITY) ID(ASSR1) ACCESS(READ)
      ```

      For named domains issue the command:

      ```
      PERMIT IRR.RPKISERV.*.DomainName CLASS(FACILITY) ID(ASSR1) ACCESS(READ)
      ```

   b. Authorize the PKI Services surrogate user to use the R_PKIServ functions REQCERT, GENCERT, ADD, VERIFY, REVOKE, RESPOND, SCEPREQ, REQRENEW, GENRENEW, EXPORT, and QRECOVER. Enter the following TSO commands. Change the value PKISERV to the user ID you specified for the surrogate user in the IKYSETUP exec, if you specified a different user ID. (See Table 17 on page 43.)

      ```
      RDEFINE FACILITY IRR.DIGTCERT.REQCERT UACC(NONE)
      PERMIT IRR.DIGTCERT.REQCERT CLASS(FACILITY) ID(PKISERV) ACCESS(READ)

      RDEFINE FACILITY IRR.DIGTCERT.GENCERT UACC(NONE)
      ```

```
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY) ID(PKISERV) ACCESS(CONTROL)

RDEFINE FACILITY IRR.DIGTCERT.ADD UACC(NONE)
PERMIT IRR.DIGTCERT.ADD CLASS(FACILITY) ID(PKISERV) ACCESS(UPDATE)

RDEFINE FACILITY IRR.DIGTCERT.VERIFY UACC(NONE)
PERMIT IRR.DIGTCERT.VERIFY CLASS(FACILITY) ID(PKISERV) ACCESS(READ)

RDEFINE FACILITY IRR.DIGTCERT.REVOKE UACC(NONE)
PERMIT IRR.DIGTCERT.REVOKE CLASS(FACILITY) ID(PKISERV) ACCESS(READ)

RDEFINE FACILITY IRR.DIGTCERT.RESPOND UACC(NONE)
PERMIT IRR.DIGTCERT.RESPOND CLASS(FACILITY) ID(PKISERV) ACCESS(READ)

RDEFINE FACILITY IRR.DIGTCERT.SCEPREQ UACC(NONE)
PERMIT IRR.DIGTCERT.SCEPREQ CLASS(FACILITY) ID(PKISERV) ACCESS(READ)

RDEFINE FACILITY IRR.DIGTCERT.REQRENEW UACC(NONE)
PERMIT IRR.DIGTCERT.REQRENEW CLASS(FACILITY) ID(PKISERV) ACCESS(READ)

RDEFINE FACILITY IRR.DIGTCERT.GENRENEW UACC(NONE)
PERMIT IRR.DIGTCERT.GENRENEW CLASS(FACILITY) ID(PKISERV) ACCESS(READ)

RDEFINE FACILITY IRR.DIGTCERT.EXPORT UACC(NONE)
PERMIT IRR.DIGTCERT.EXPORT CLASS(FACILITY) ID(PKISERV) ACCESS(CONTROL)

RDEFINE FACILITY IRR.DIGTCERT.QRECOVER UACC(NONE)
PERMIT IRR.DIGTCERT.QRECOVER CLASS(FACILITY) ID(PKISERV) ACCESS(READ)
```

c. Allow the unauthenticated user to run a task that is associated with the PKI Services surrogate user ID. Change the value `PKISERV` to the user ID you specified for the surrogate user in the IKYSETUP exec, if you specified a different user ID. (See Table 17 on page 43.)

```
RDEFINE EJBROLE PKISurrogate UACC(READ) APPLDATA('PKISERV')
SETROPTS RACLIST(EJBROLE) REFRESH
```

d. If you have modified `pkitmpl.xml` and use the zAuthRunAsClient retrieve authentication type:

```
<tns:retrieve_authtype>zAuthRunAsClient</tns:retrieve_authtype>
```

you must give any user ID that will request this kind of certificate authorization to the R_PKIServ EXPORT function:

```
PERMIT IRR.DIGTCERT.EXPORT CLASS(FACILITY) ID(userid) ACCESS(CONTROL)
```

e. Allow the Websphere user ID to run as an authenticated RACF user. To do this, the user ID for the Websphere control region must have CONTROL access to the resource BBO.SYNC.*cell_short_name.cluster_short_name*. Enter commands similar to the following:

```
RDEFINE FACILITY BBO.SYNC.DCEIMGLX.BBOC001 UACC(NONE)
PERMIT BBO.SYNC.DCEIMGLX.BBOC001 CLASS(FACILITY) ID(ASCR1) ACCESS(CONTROL)
```

f. Refresh the in-storage profiles so that the changes you have made to the FACILITY class take effect:

```
SETR RACLIST(FACILITY) REFRESH
```

When you are done, you have authorized Websphere users to use PKI Services functions.

Allowing Websphere users to renew and revoke browser certificates

To allow Websphere users to renew and revoke browser certificates (see Figure 7), you need to set up a Websphere SSL configuration that uses client authentication.

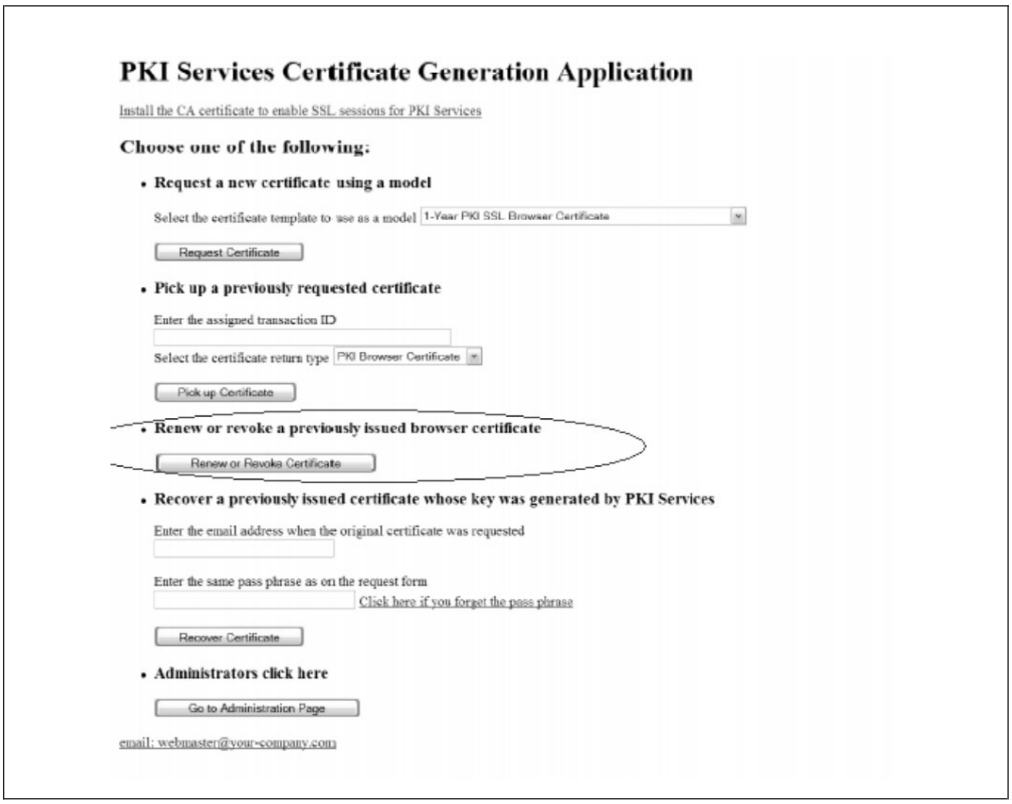


Figure 7. Renewing or revoking a browser certificate

Steps for allowing Websphere users to renew and revoke browser certificates

Before you begin: You need RACF administration skills. You must have the RACF SPECIAL attribute or sufficient authority to IRR.DIGTRING resources in the FACILITY class to issue the RACDCERT commands shown.

Perform the following steps to set up a Websphere SSL configuration that uses client authentication, so that users can renew and revoke browser certificates.

1. Create a RACF keyring that contains the PKI Services CA certificate for the Websphere Control address space user ID (ASCR1 in this example):

   ```
   RACDCERT ADDRING(WASKeyring.PKI) ID(ASCR1)
   ```

   ---

2. Add the PKI CA certificate to this keyring. The CA certificate label ("Master PKI CA") is defined in the IKYSETUP exec.

   ```
   RACDCERT ID(ASCR1) CONNECT(CERTAUTH LABEL('Master PKI CA') RING(WASKeyring.PKI))
   ```

   ---

3. Create a client certificate that is signed by the PKI CA certificate, and add this client certificate to the keyring.

```
RACDCERT GENCERT
   ID(ASCR1)
   SUBJECTSDN(CN('DCEIMGLX CLIENT CERT') OU('IBM'))
   WITHLABEL('pki ssl cert')
   SIGNWITH(CERTAUTH LABEL(Master PKI CA')) TRUST
RACDCERT ID(ASCR1)
   CONNECT(ID(ASCR1) LABEL('pki ssl cert') RING(WASKeyring.PKI) USAGE(PERSONAL))
```

Tip: The value of SUBJECTDSN does not matter here. The labels in the two commands must be the same, but otherwise do not matter.

---

4. Define the RACF key store to Websphere. On the Websphere administrator console, on the left side of the page expand **Security** and click **SSL certificate and key management**. You should see a Web page that looks like Figure 8.

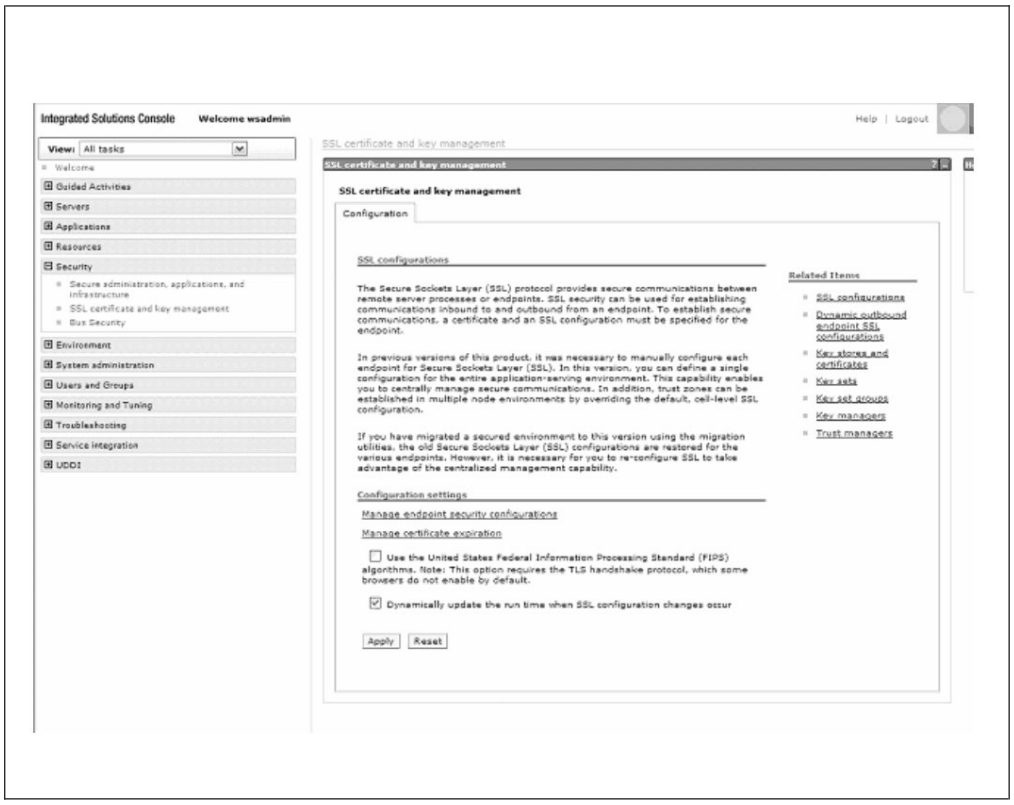


Figure 8. Websphere SSL certificate and key management page

Click **Key stores and certificates** on the right side of the page. On the next page click **New** to create a new key store. On the next page enter a name for your key store in the **Name field**. In the **Path** field enter:

```
safkeyring:///WASKeyring.PKI
```

In the **Password** and **Confirm password** fields enter a password value of `password`. Select the **Read only** check box. See Figure 9 on page 191



Figure 9. Websphere page for creating a new key store

Click **OK** and then **Save**. You should now be able to click on the name of the key store you created and view the signer certificate. See Figure 10.

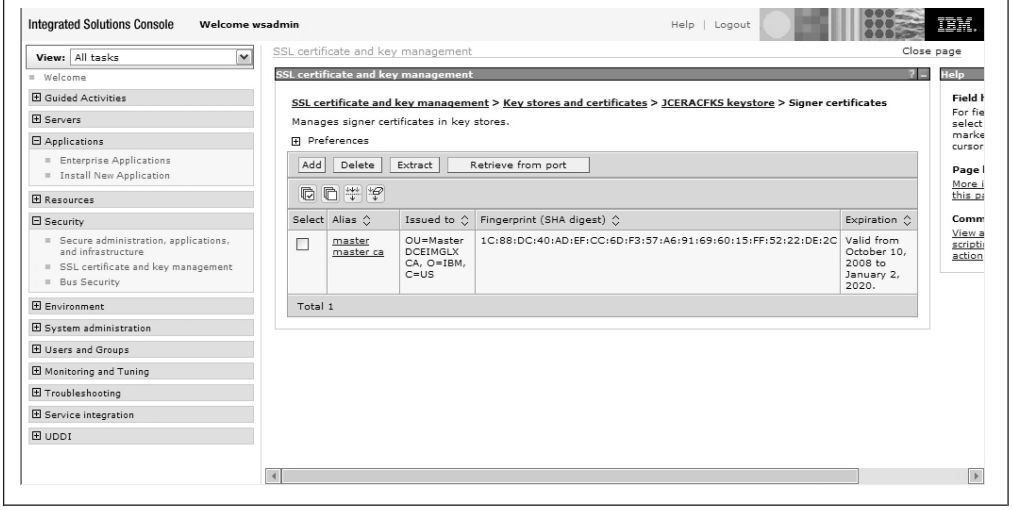


Figure 10. Viewing the signer certificate

5. Create a new JSSE type SSL configuration. On the left side of the page expand **Security** and click **SSL certificate and key management**. On the right side of the page click **SSL configurations**. (See Figure 8 on page 190.) The page shown in Figure 11 is displayed.

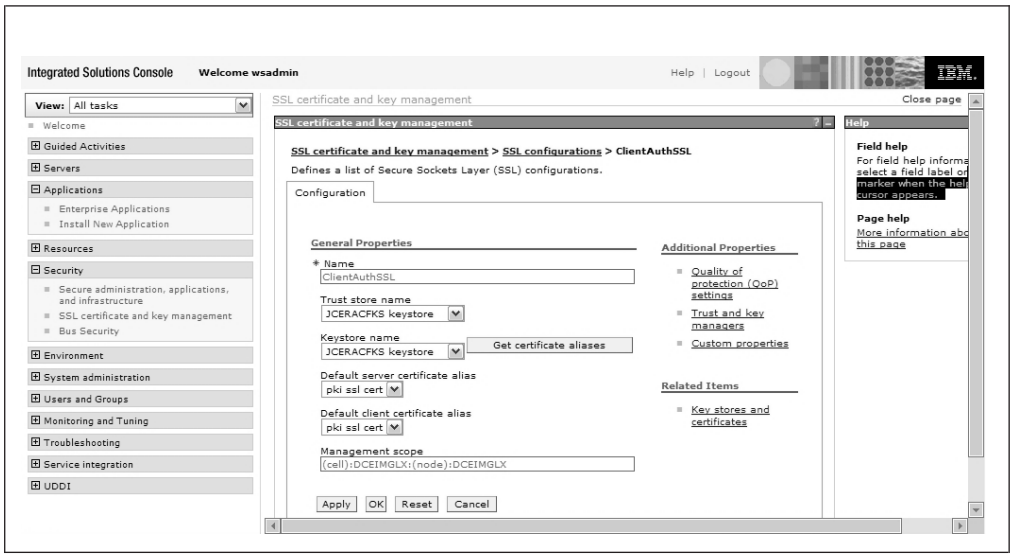


Figure 11. Websphere new SSL configuration page

In the **Name** field enter `ClientAuthSSL`. In the **Trust store name** and **Keystore name** fields, enter the name of the key store that you just defined. Click **Get certificate aliases** to get the alias (label) of the default certificate in your keyring. This will be the value for Default server certificate alias and Default client certificate alias. Click **OK**. The page shown in Figure 12 on page 193 is displayed.

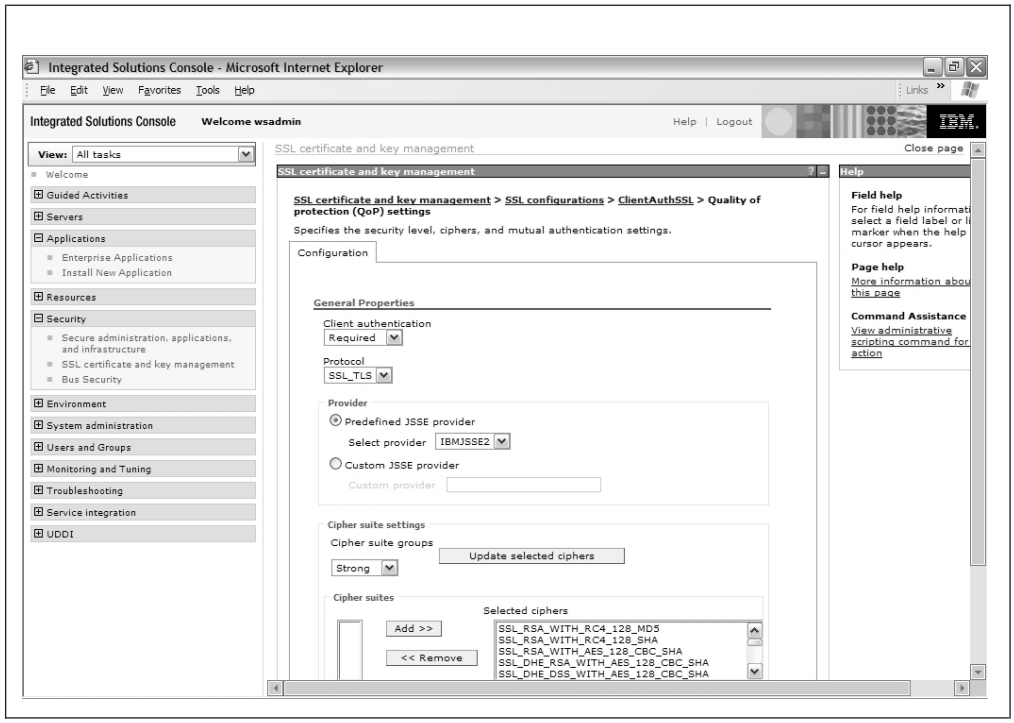


Figure 12. Websphere quality of protection settings page

In the **Client authentication** list, select **Required**, then click **OK**.

---

6. Define a new port for your default host. On the left side of the page, expand **Environment** and click **Virtual Hosts**. Then click **default host**, then click **Host Aliases**, then click **New**. On the page that results, the value 9444 in the **Port** field should match your client authorization port address in the `web.xml` file. See Figure 13 on page 194.

   If you need to change the authorization port address in the `web.xml` file, follow the instructions in "Steps for creating application domains other than Application2" on page 229, except in steps 3 and 4 you need to find the section that looks like:

```
<env-entry>
         <env-entry-name>_PKISERV_CLIENTAUTH_PORT</env-entry-name>
         <env-entry-type>java.lang.String,</env-entry-type>
         <env-entry-value>9444</env-entry-value>
</env-entry>
```

   and change the 9444 to the value that you want to use.

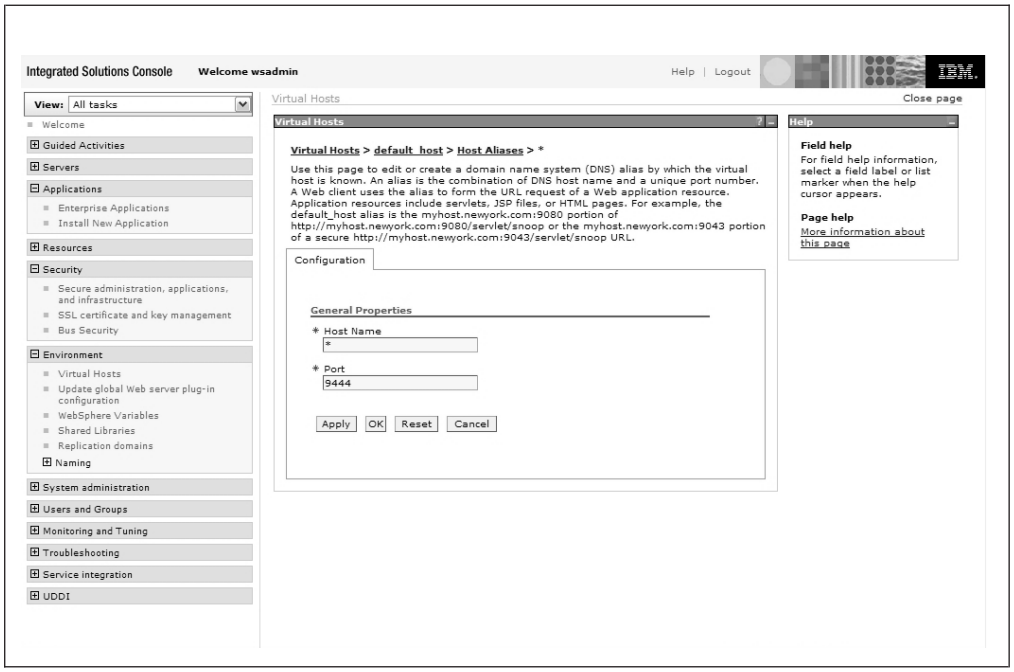


Figure 13. Defining a new port

Click **OK** and then **Save** to save this new port number.

---

7. Assign the new port number to your application server. On the left side of the page, expand **Servers** and click **Application servers**. Click the name of your application server. See Figure 14.

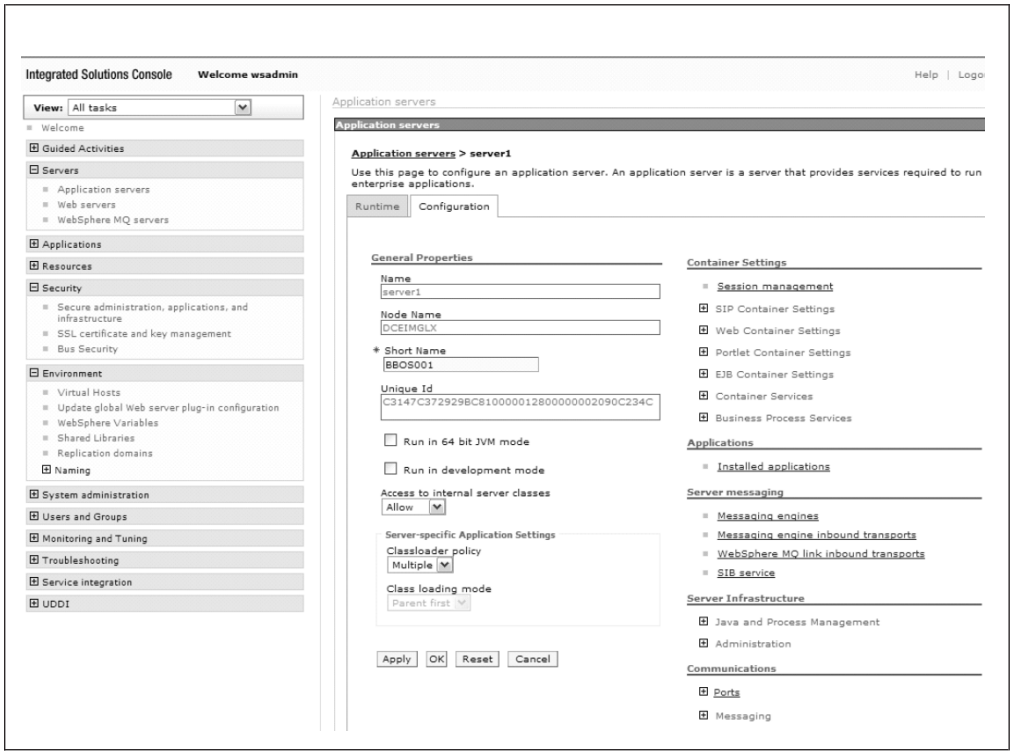


Figure 14. Assigning a port to the application server

Under Communications expand **Ports** and click **New**. Click **User-defined Port**, and in the **Specify Port name** field enter `client-authenticated SSL`. In the **Host** field enter `*`. In the **Port** field enter `9444`. See Figure 15.

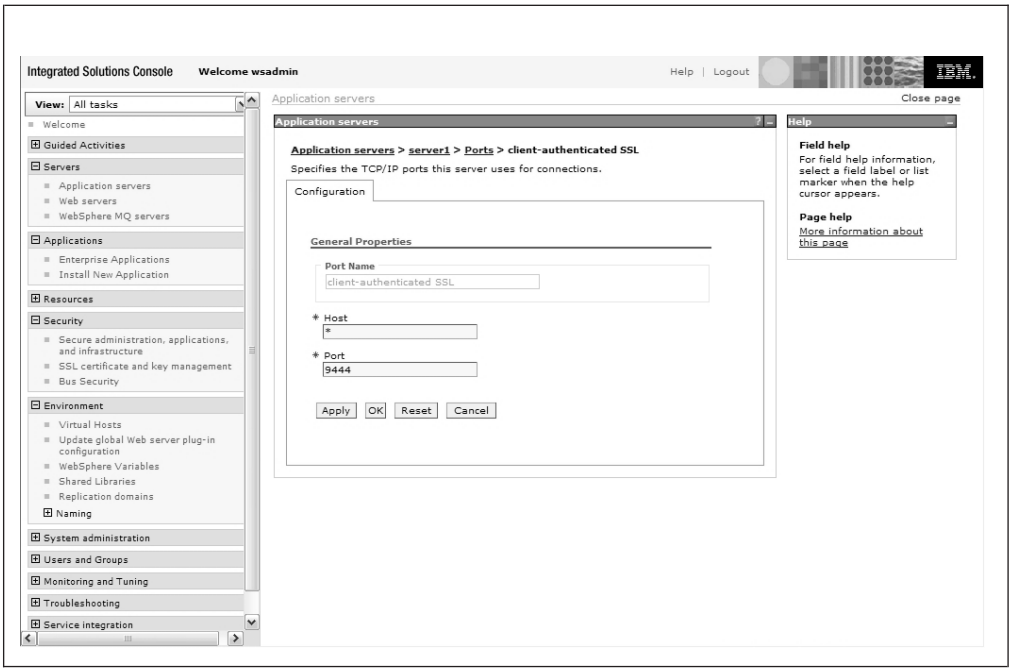


Figure 15. Assigning your new port to the application server

Click **OK** and **Save**.

---

8. Define transport chains associated with the new port.
   a. On the left side of the page expand **Servers** and click **Application servers**. Click the name of your application server. Expand **Web container settings**. See Figure 16 on page 196.

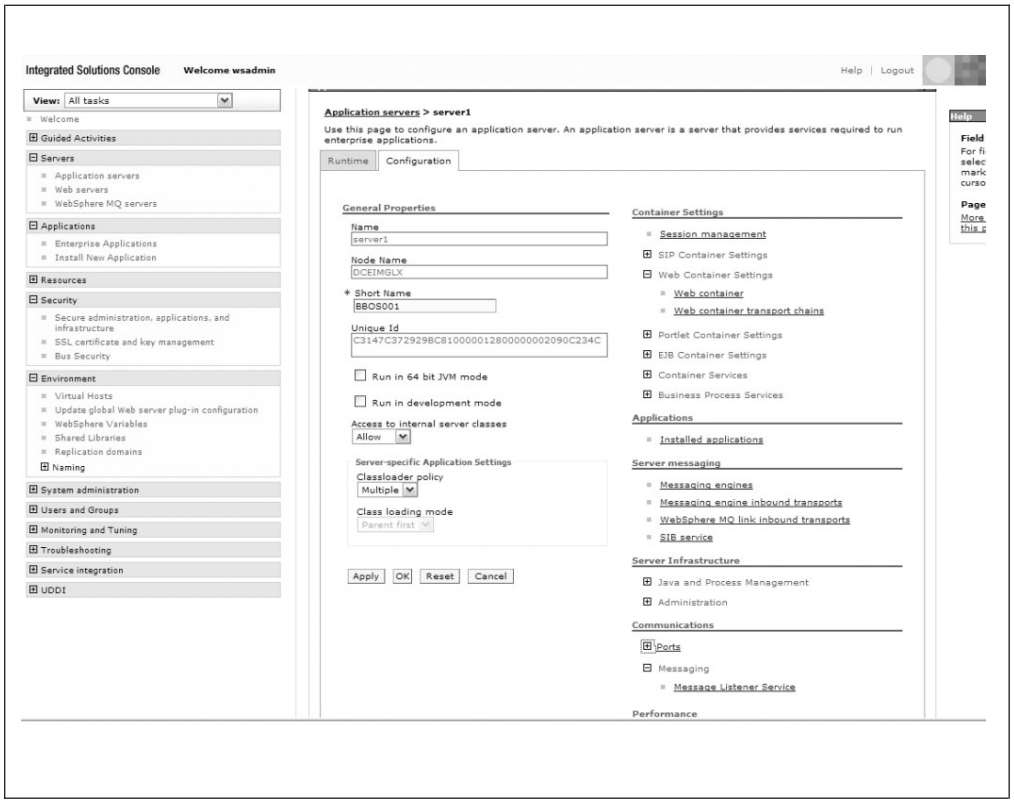


Figure 16. Selecting Web container transport chains

b. Click **Web container transport chains**, and then click **New**. You should see a page like Figure 17.

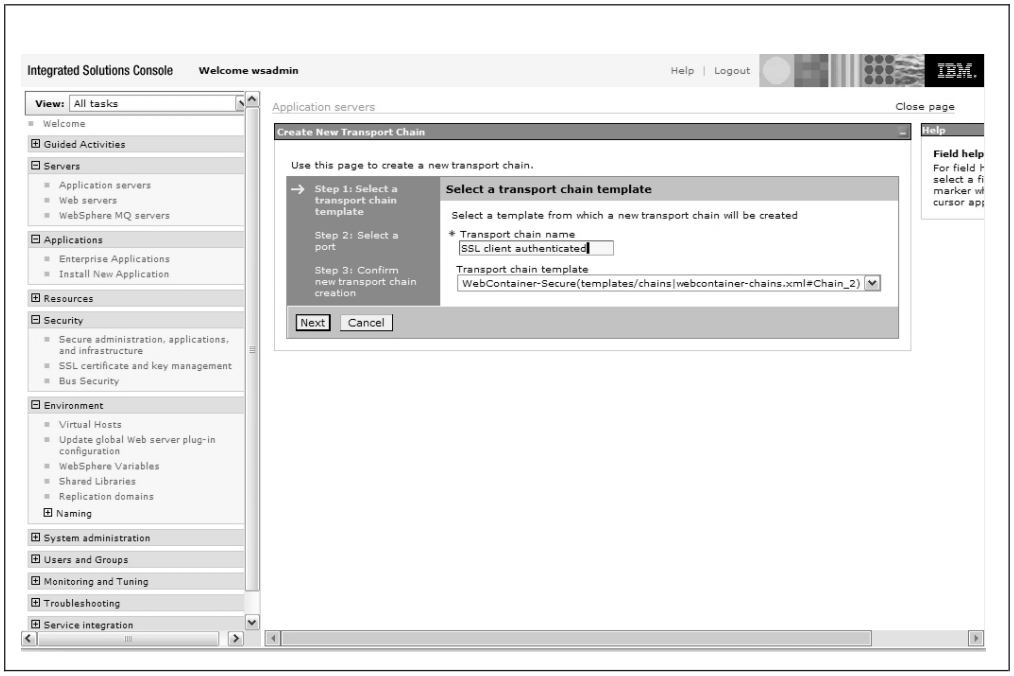


Figure 17. Selecting a transport chain template

c. Choose a secure transport chain template and click **Next**. On the next page select **Use an existing port**, then click **Confirm** and then **Save**. A page

opens that displays all existing transport chains. See Figure 18.

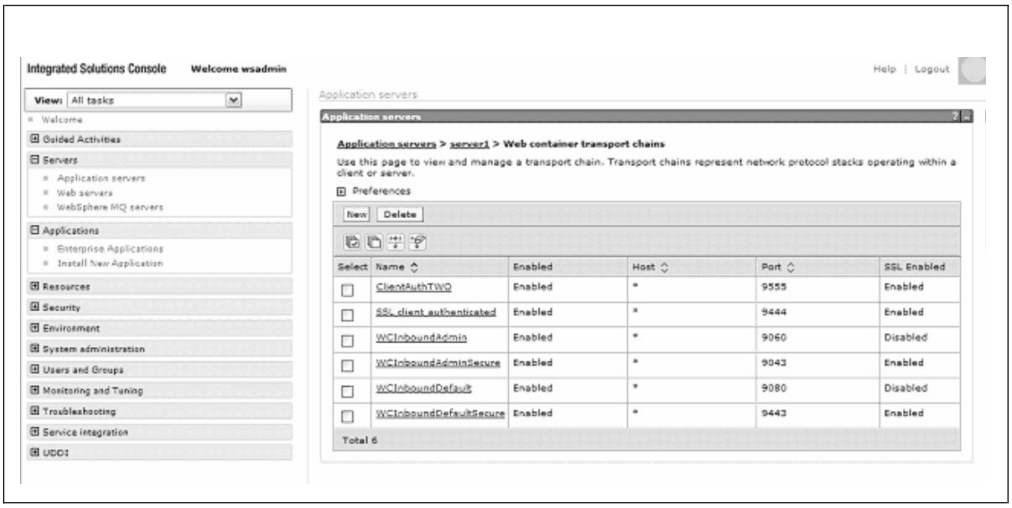


Figure 18. All existing transport chains

d. Click on the name of the chain you just defined, then click **SSL inbound channel**. You need to set the SSL inbound channel properties to use the SSL configuration you defined (which requires client authentication). See Figure 19.

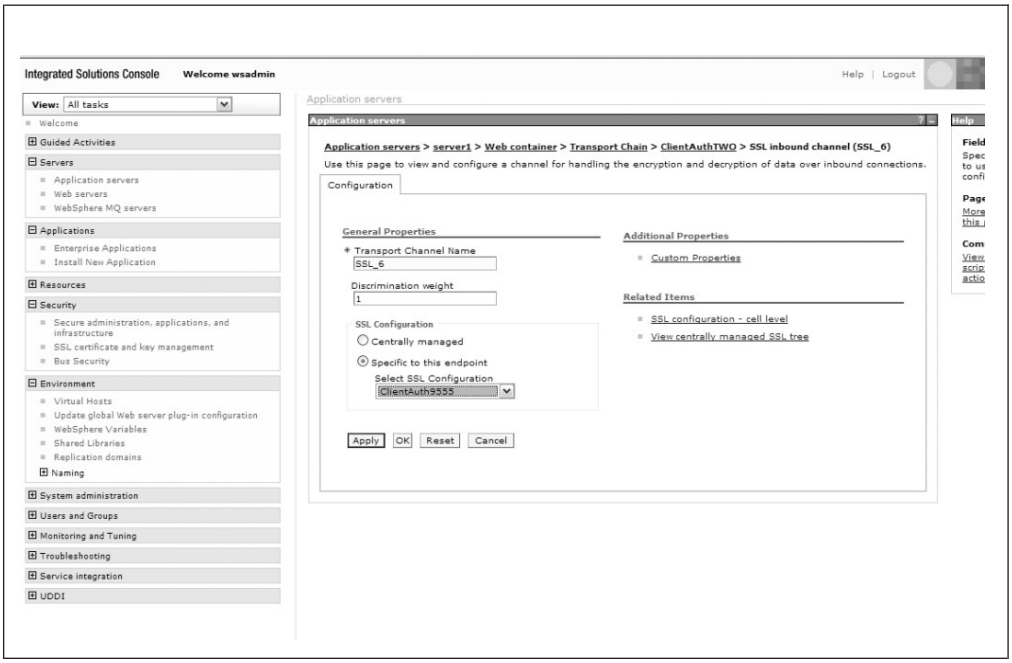


Figure 19. Setting the SSL inbound channel properties

9. Stop and restart the Websphere server so that your changes take effect.

When you are done, you have set up a Websphere SSL configuration that uses client authentication, so that users can renew and revoke browser certificates.

Customizing the PKI Services Web application

In order to customize the PKI Services Web application, you need to do the following:

1. Update the template file.
2. (Optional) Modify the JSP files and update the EAR file with the modified JSP files.
3. Deploy the EAR file to a Websphere application server

Updating the template file

You can customize the PKI Services Web application by modifying the default copies of the template file, `pkitmpl.xml`. Any time that you update `pkitmpl.xml`, you need to create an equivalent copy of `pkixgen.tmpl`. You do this using the TemplateTool utility. For more information, see "Using the TemplateTool utility" on page 370.

(Optional) Modifying the JSP files and the EAR file

After you update the template file, you can optionally perform additional customization on the PKI Services Web application by modifying the JSP files. You will probably want to modify the JSP files in the directory `mod_inc`. You can modify any line that is not marked with a comment saying that it cannot be modified. You also might want to modify JSP files in the directories `Customers` and `PKIServ`. You cannot modify JSP files in the directory `not_mod_inc`. For information about what files are contained in each directory, see "Directories for JSP files" on page 204.

Example: This example shows how you could customize a JSP file. `notbefore.jsp` is a JSP file in `mod_inc`. The portion of this file that displays the prompt and the options is shown in Figure 20, without customization.

```
⋮
Number of days after today before the certificate becomes current<<%=optional_str %>

<SELECT NAME="notbefore">
<OPTION VALUE="0"
<% if (initvalue_str.equalsIgnoreCase("0")) out.print(" SELECTED "); %>
>0
<OPTION VALUE="30"
<% if (initvalue_str.equalsIgnoreCase("30")) out.print(" SELECTED "); %>
>30
</SELECT>
```

Figure 20. A portion of the JSP file `notbefore.jsp`, without customization

You could customize this code to reword the prompt and add an option of 7 days, as shown in Figure 21 on page 199.

```
⋮
Number of days after today before the certificate is valid<%=optional_str %>

<SELECT NAME="notbefore">
<OPTION VALUE="0"
<% if (initvalue_str.equalsIgnoreCase("0")) out.print(" SELECTED "); %>
>0

<OPTION VALUE="7"
<% if (initvalue_str.equalsIgnoreCase("7")) out.print(" SELECTED "); %>
>7

<OPTION VALUE="30"
<% if (initvalue_str.equalsIgnoreCase("30")) out.print(" SELECTED "); %>
>30
</SELECT>
```

Figure 21. A portion of the JSP file notbefore.jsp, with customization

An enterprise archive (EAR) file is a specialized Java archive (JAR) file, used to deploy J2EE applications to J2EE application servers. PKI Services ships a default EAR file in the directory /usr/lpp/pkiserv/pkijsp. If you make changes to the JSP files for the PKI Services Web application, you need to update the EAR file to include your changes.

Steps for updating the EAR file: **Before you begin:** You need to have the **jar** command in your path. If you don't, define the JAVA_HOME variable (you can find its value on the Websphere administration console) and add the $JAVA_HOME/bin directory to your path. For example:

```
export JAVA_HOME=/WebSphere/V6R1/AppServer/java
export PATH=$JAVA_HOME/bin:$PATH
```

Perform the following steps to update the EAR file

1. (Optional) Set up a directory to use just for updating the EAR file and make this your working directory. For example:

   ```
   cd $HOME
   mkdir pkiear
   cd pkiear
   ```

   _____

2. Copy the EAR file that you have been using to the working directory. The default version shipped with PKI Services is in the file PKIServ.ear in the directory /usr/lpp/pkiserv/pkijsp. For example:

   ```
   cp /usr/lpp/pkiserv/pkijsp/PKIServ.ear .
   ```

   _____

3. Expand the EAR file using the **jar** command. For example:

   ```
   jar -xvf /usr/lpp/pkiserv/pkijsp/PKIServ.ear
   ```

   _____

4. Expand the file PKIServ_Web_war:

   ```
   jar -xvf PKIServ_Web.war
   ```

   _____

5. Change to the modifiable include directory:

   ```
   cd mod_inc
   ```

   _____

6. The JSP files are in ASCII (ISO8859-1 code page). To edit them on z/OS you have to convert them to EBCDIC. Use **iconv** to convert a file to EBCDIC. For example, to convert the file `footer.jsp` to EBCDIC, enter:

   ```
   iconv -f iso8859-1 -t ibm-1047 footer.jsp > $HOME/footer.jsp.edit
   ```

   _____

7. Use oedit to edit the .edit file that you created:

   ```
   oedit $HOME/footer.jsp.edit
   ```

   _____

8. Use **iconv** to converted the edited file back to ASCII. For example:

   ```
   iconv -t iso8859-1 -f ibm-1047 $HOME/footer.jsp.edit > footer.jsp
   ```

   _____

9. Go back to the directory containing the WAR and EAR files, and update the WAR file with the edited and reconverted JSP file:

   ```
   cd ..
   jar -uvf PKIServ_Web.war mod_inc/footer.jsp
   ```

   _____

10. Update the EAR file with the updated WAR file:

    ```
    jar -uvf PKIServ.ear PKIServ_Web.war
    ```

    _____

11. Make sure the `PKIServ.ear` file is publicly readable by issuing the **chmod** command.

    ```
    chmod 755 PKIServ.ear
    ```

    _____

When you are done, you have updated the JSP files, and updated the EAR file to include the updated JSP files. You can now deploy your updated EAR file to a Websphere application server.

Deploying the EAR file to a Websphere application server

In order to use the JSP files for the PKI Services Web application, a Websphere administrator must deploy them to a Websphere application server.

Steps for deploying the EAR file to a Websphere application server: Before you begin: You need to have Websphere administration skills. You might need to refer to information in the Websphere information center at http://www.ibm.com/software/webservers/appserv/was/library/.

Perform the following steps to deploy the EAR file to a Websphere application server.

1. Log on to the Websphere administrative console.

   _____

2. Create a new shared library.
   a. In the left pane, click **Environment > Shared Libraries**.
   b. Click **New** to add a shared library. The window shown in Figure 22 on page 201 opens.

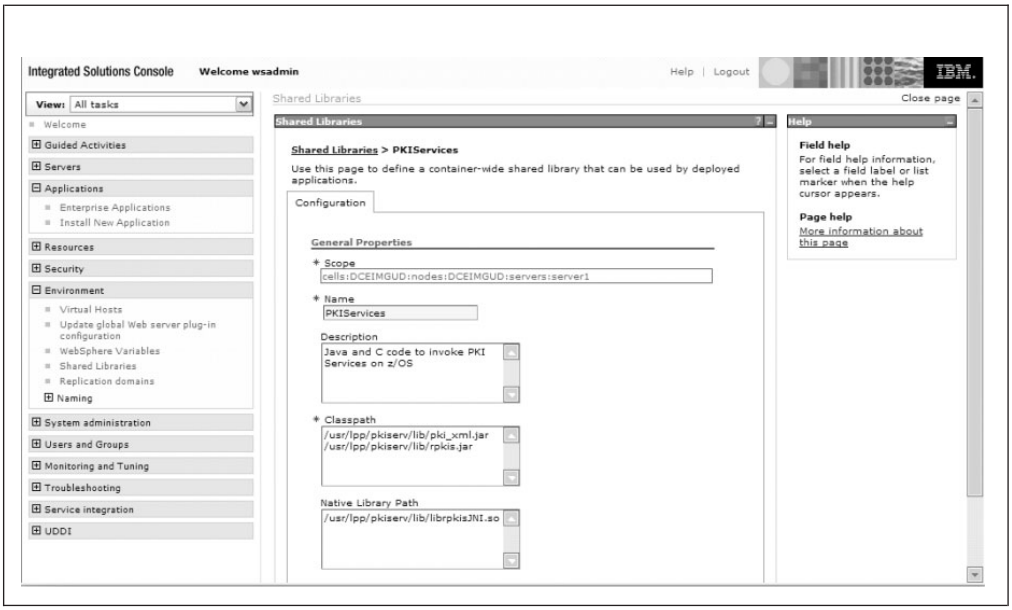


Figure 22. The Websphere Shared Libraries window

c. In the **Scope** field, select a scope that includes the server where you will deploy the PKI Services JSPs (PKIServ_EAR).

d. In the **Name** field enter `PKIServices`.

e. Optionally enter a description in the **Description** field; for example, `Java and C code to invoke PKI Services on z/OS`.

f. In the **Classpath** field, enter the following two lines:

```
/usr/lpp/pkiserv/lib/pki_xml.jar
/usr/lpp/pkiserv/lib/rpkis.jar
```

g. In the **Native Library Path** field, enter:

```
/usr/lpp/pkiserv/lib/librpkisJNI.so
```

h. Click **OK**. A message is displayed indicating that changes have been made to your local configuration. See Figure 23.

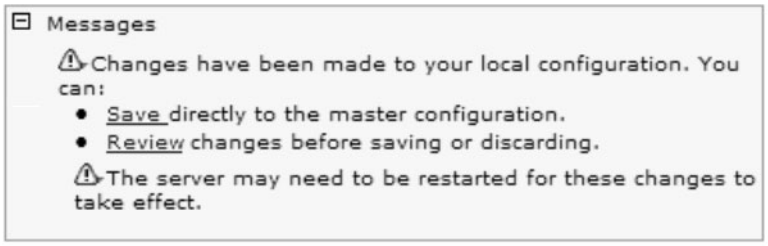


Figure 23. Message indicating that changes have been made to your local configuration.

i. Click **Save** to save the changes to the master configuration.

---

3. Deploy the enterprise archive (EAR) file.

a. In the left pane, click **Applications > Enterprise Applications**.

b. Click **Install**. The window shown in Figure 24 on page 202 opens.

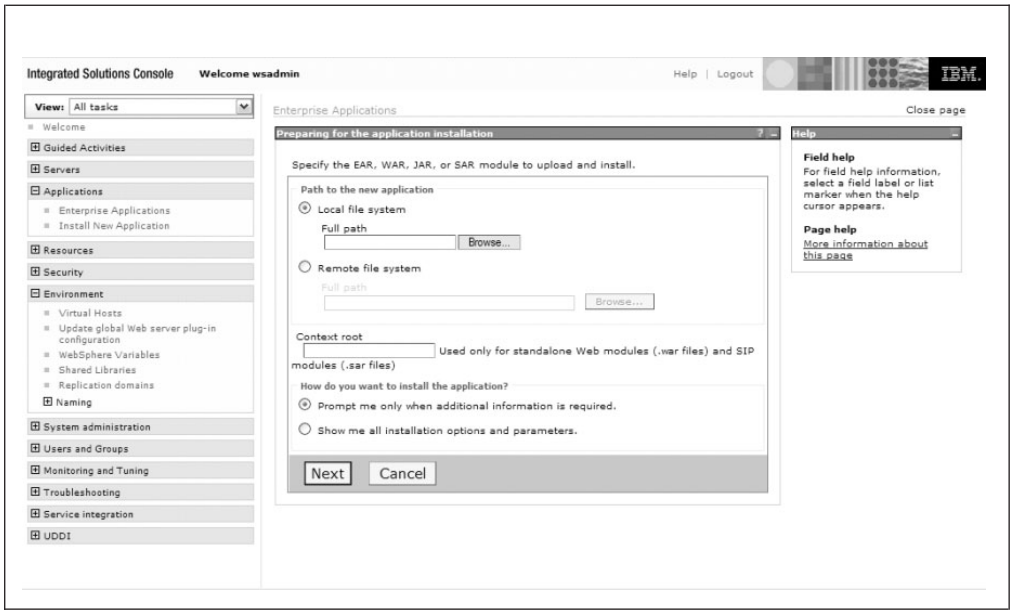


Figure 24. Window for specifying the EAR file

c. Click **Remote file system**, and click **Browse**.

d. Click the icon for the z/OS system. The root directory is displayed. Continue selecting until you reach:

   `/usr/lpp/pkiserv/pkijsp/PKIServ.ear`

e. Click **OK** and **Next**. Continue clicking **OK** and **Next** or **Finish** until you see a message saying that the EAR was installed and the message shown in Figure 23 on page 201 indicating that changes have been made to your local configuration. (If you have more than one application installed into Websphere Application server, you might see a window where you have to select a server to which you want to map modules before you click **Next**.)

f. Click **Save** to save the changes to the master configuration.

---

4. Associate the PKI Services shared library with the PKI Services application.

   a. In the left pane, click **Applications > Enterprise Applications**. Click **PKIServ_EAR**. The window shown in Figure 25 on page 203 opens.



Figure 25. Application properties page

b. Click **Shared Library References**. The window shown in Figure 26 opens.

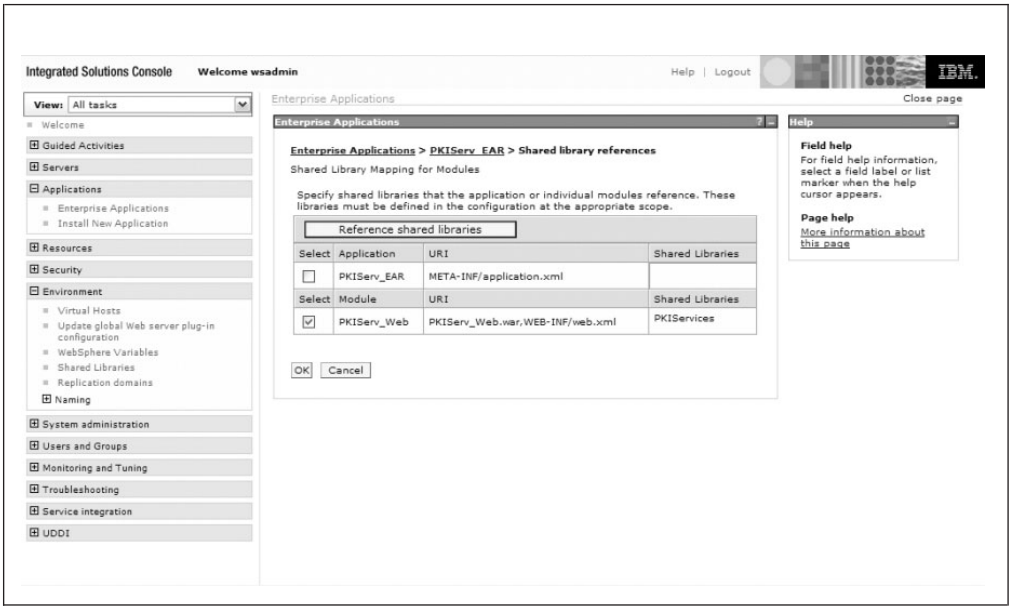


Figure 26. Shared library mapping for modules page

c. Select the **PKIServ_Web** check box and click **Reference Shared Library**.

d. Add the PKI Services shared library to the Selected list and click OK.

e. Select the **PKIServ_EAR** check box and click **Reference Shared Library**.

f. Add the PKI Services shared library to the Selected list and click OK.

g. Again you see the message shown in Figure 23 on page 201 indicating that changes have been made to your local configuration. Click **Save** to save the changes to the master configuration.

---

5. Add the PKI Services shared library's location to the server's library path.

a. In the left pane, click **Servers > Application servers >** *server1* **> (Java and Process Management) > Process Definition > Servant > Environment Entries > New** , where *server1* is the name of your server. In the **Name** field enter:

   `LIBPATH`

   In the **Value** field enter:

   `/usr/lpp/pkiserv/lib`

   In the **Description** field enter:

   `rpkiserv jni path`

   Click **Apply**. See Figure 27.

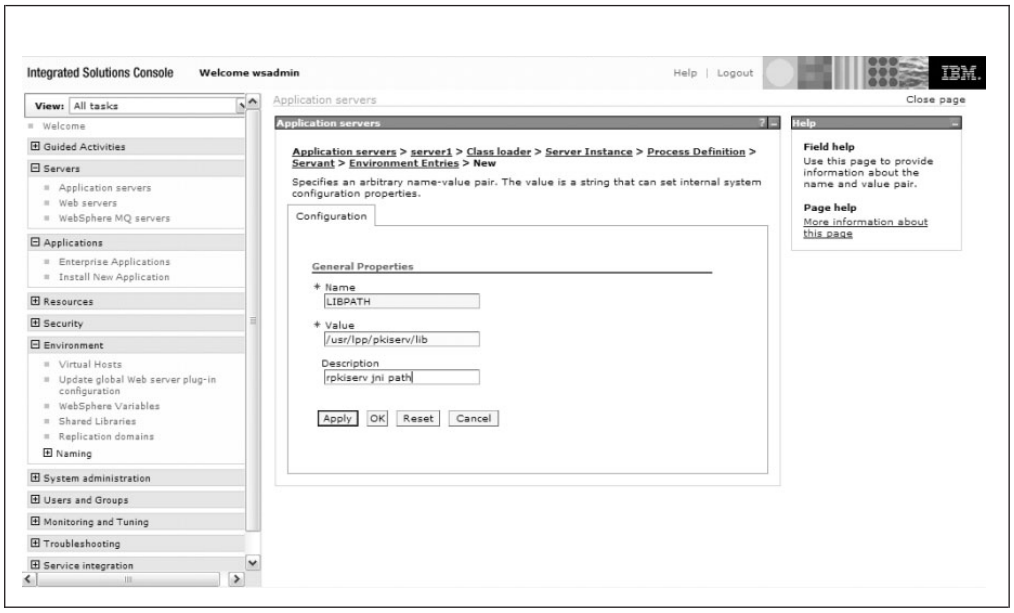


Figure 27. Adding the PKI Services shared library's location to the server's library path

b. Again you see the message shown in Figure 23 on page 201 indicating that changes have been made to your local configuration. Click **Save** to save the changes to the master configuration.

When you are done, you have deployed the PKI Services JSP files to a Websphere application server. You can now begin using the customized PKI Services Web application.

Directories for JSP files

The JSP files are shipped in `PKIServ_Web.war` in `PKIServ.ear` and are located in the following directories:

- `Customers`

  JSPs used by end users. Each JSP file in this directory defines all HTML and script for an entire end-user Web page. Each file starts with an <HTML> tag and ends with a </HTML> tag. JSP files in this directory can be modified, but

contain lines of Java processing that cannot be modified. Lines that cannot be modified are identified with comments. Table 42 lists the JSP files in this directory.

- `PKIServ`

  JSPs used by administrators. Each JSP file in this directory defines all HTML and script for an entire administrator Web page. Each file starts with an <HTML> tag and ends with a </HTML> tag. JSP files in this directory can be modified, but contain lines of Java processing that cannot be modified. Lines that cannot be modified are identified with comments. Table 43 on page 206 lists the JSP files in this directory.

- `mod_inc`

  Modifiable include files. Each JSP file in this directory defines HTML and script (JavaScript or vbscript) that might be used on multiple Web pages, such as running headers or footers for a page, or a form field element (including JavaScript to verify the input). (If you are familiar with the text template file, `pkiserv.tmpl`, used with the REXX CGIs, these modifiable include files generally correspond to an INSERT section.) Installations are likely to want to customize the code in these files. Some parts of these files cannot be modified and these parts are identified by comments in the file. Table 44 on page 207 lists the JSP files in this directory. "Customizing the PKI Services Web application" on page 198 shows an example of customizing a JSP file in this directory.

- `not_mod_inc`

  Non-modifiable include files. Each JSP file in this directory defines HTML and script that might be used on more than one web page, and should not be modified. An example is the vbscript code used to discover certificates installed in an Internet Explorer browser.

- `domain_specific`

  Contains `selectCAdomain.jsp`, which an installation would modify if it had multiple domains with a single administrator.

- `WEB-INF`

  Contains supporting Java classes, including servlets, that an installation is not expected to modify, and `web.xml`, which an installation needs to modify to add additional applications or domains. (See "Creating application domains when you use JSPs to implement the Web application" on page 229.)

Table 42. JSP files in the `Customers` directory. These files perform end-user functions.

File name	Description
`pkimain.jsp`	Starting point for all tasks available to the end user, such as requesting a new certificate, picking up a previously requested certificate, renewing or revoking a previously issued browser certificate, or recovering a certificate whose key was generated by PKI Services
`certrequest.jsp`	Web page to request a certificate.
`certretrieve.jsp`	Web page to retrieve a certificate.
`genericbad.jsp`	Displays error information when processing fails.
`genericok.jsp`	Notifies the user of successful processing when no additional information (such as a transaction ID) needs to be returned to the user.

Table 42. JSP files in the `Customers` directory (continued). These files perform end-user functions.

File name	Description
`installcert.jsp`	Contains HTML, Javascript and VBscript functions to install an automatically renewed certificate copied from an e-mail notification if using the Internet Explorer browser.
`keygenrequestok.jsp`	Notifies the user of a successful certificate request with PKI-generated keys, and includes the e-mail address where notification has been sent.
`pkcs10retrieved.jsp`	Returns a retrieved PKCS #10 certificate.
`recoverbad.jsp`	Displays information for an unsuccessful attempt to find certificates for which the private key is to be recovered.
`recoverok.jsp`	Displays results of a successful attempt to find certificates for which the private key is to be recovered.
`renew_revoke.jsp`	Displays a client-authenticated browser certificate for the user to renew, revoke or suspend.
`requestok.jsp`	Returns results (such as the transaction ID) of a successful certificate request.
`requestbad.jsp`	Returns error information for an unsuccessful certificate request.
`retrievebad.jsp`	Returns error information for an unsuccessful attempt to retrieve a certificate.

Table 43. JSP files in the `PKIServ` directory. These files perform administrator functions.

File name	Description
`actcertok.jsp`	Returns results of a successful certificate action, such as deleting, revoking, suspending, or resuming a certificate.
`actcertbad.jsp`	Returns error information on an unsuccessful certificate action, such as deleting, revoking, suspending, or resuming a certificate.
`actrequestok.jsp`	Returns results of a successful action on a certificate request, such as approving, rejecting or deleting.
`acttrequestbad.jsp`	Returns error information on an unsuccessful action on a certificate request, such as approving, rejecting or deleting.
`adminmain.jsp`	The main PKI Services administration page, from which the administrator can search certificates or certificate requests, or enter a specific transaction ID or serial number to act upon.

Table 43. JSP files in the PKIServ directory (continued). These files perform administrator functions.

File name	Description
admhome.jsp	The PKI Services administration home page, which contains links to the end-user home page for each application and to the main PKI Services administration Web page (admmain.jsp)
approvewithmods.jsp	Displays the form for an administrator to approve a certificate request with modifications.
certdetailsbad.jsp	Displays error information for a failed attempt to display the details of a certificate.
certdetailsok.jsp	Displays details of a single certificate.
genericfailure.jsp	Displays error information when the requested administration processing fails.
modifybad.jsp	Displays error information when an approve with modifications action fails.
modifyok.jsp	Displays results of successful approve with modification action.
querycertok.jsp	Displays search results when administrator queries certificates.
queryreqok.jsp	Displays search results when administrator queries certificate requests.
reqdetailsok.jsp	Displays details of a single certificate request.

Table 44. JSP files in the mod.inc directory. These files are modifiable include files.

File name	Description
adminbottomnav.jsp	Bottom of page navigation appearing on administrator's Web pages
adminfooter.jsp	Bottom footer (the last element) on administrator's Web pages
altdomain.jsp	Form field for altdomain
altemail.jsp	Form field for altemail
altipaddr.jsp	Form field for altipaddr
altother_1_2_3_4_5.jsp	altother form field sample for OID 1.2.3.4.5
altother_1_2_3_4_6.jsp	altother form field sample for OID 1.2.3.4.6
altother_1_3_6_1_4_1_311_20_2_3.jsp	altother form field sample for OID 1.3.6.1.4.1.311.20.2.3
alturi.jsp	Form field for alturl
autorenew.jsp	Form field for autorenew
bottomnav.jsp	Bottom of page navigation on end user Web pages
certrow.jsp	Displays a row in a table about certificates, format tied to PKIServ/certdetailsok.jsp

Table 44. JSP files in the `mod.inc` directory (continued). These files are modifiable include files.

File name	Description
`challengepassphrase.jsp`	Displays passphrase form field when user is prompted to match the passphrase previously entered
`clientname.jsp`	Form field for clientname
`commonfunctions.jsp`	Contains JavaScript functions that are commonly used, such as `trim()`
`commonname.jsp`	Form field for commonname
`country.jsp`	Form field for country
`CustomExt.jsp`	The HTML and JavaScript for defining custom certificate extensions.
`date.jsp`	Displays validity period form fields as two dates, the date the certificate becomes valid and the date the certificate expires, used on approve with modification processing
`dnqualifier.jsp`	Form field for distinguished name qualifier
`domainname.jsp`	Form field for domain name
`emailaddr.jsp`	Form field for emailaddr
`extkeyusage.jsp`	Form field for extended key usage
`footer.jsp`	Footer (last element) on end users pages
`hostidmap.jsp`	Form field for host id mapping
`keysize.jsp`	Form field for keysize
`keyusage.jsp`	Form field for key usage
`label.jsp`	Form field for label
`locality.jsp`	Form field for locality
`mail.jsp`	Form field for mail
`notafter.jsp`	Form field for the number of days the certificate should be valid (from the time of request)
`notbefore.jsp`	Form field for the number of days before the certificate should be valid
`notifyemail.jsp`	Form field for notifyemail
`org.jsp`	Form field for org
`orgunit.jsp`	Form field for orgunit
`passphrase.jsp`	Form field for user to enter passphrase
`postalcode.jsp`	Form field for postal code
`publickey.jsp`	"Public key" form field, text box for entering base64-encoded PKCS #10 certificate request
`publickey_smartcard.jsp`	The HTML, JavaScript and vbscript for the public key form field, when the browser provides the public key after the user selects from a list of available cryptographic providers
`recoveremail.jsp`	recoveremail form field

Table 44. JSP files in the `mod.inc` directory (continued). These files are modifiable include files.

File name	Description
`requestor.jsp`	Requestor form field (with prompt for name for tracking this request)
`requestor2.jsp`	Requestor form field (with prompt for e-mail address of requestor)
`security.jsp`	Displays a form field for entering a response to a security question
`security2.jsp`	Displays a form field for entering a response to a second security question
`serialnumber.jsp`	Form field for entering a certificate's serial number
`signwith.jsp`	Form field for selecting the signer of the requested certificate
`stateprov.jsp`	Form field for stateprov
`street.jsp`	Form field for street
`title.jsp`	Form field for title
`transactionid.jsp`	Form field for entering a certificate request's transaction ID
`uid.jsp`	Form field for uid
`unstructaddr.jsp`	Form field for unstructaddr
`unstructname.jsp`	Form field for unstructname
`userid.jsp`	Form field for userid

Chapter 14. Advanced customization

This topic describes the advanced customization procedures available for PKI Services. All are optional.

- "Scaling for high volume installations"
- "Using certificate policies" on page 212
- "Updating the signature algorithm" on page 215
- "Customizing distribution point CRLs" on page 217
- "Using the OCSP responder" on page 224
- "Creating a distribution point ARL" on page 225
- "Adding an application domain" on page 226
- "Adding a new CA domain" on page 230
- "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246
- "Customizing e-mail notifications sent to users" on page 252
- "Setting up automatic renewal of certificates" on page 258
- "Setting up PKI Services to generate keys for certificate requests" on page 259
- "Adding custom extensions to certificates" on page 261
- "Enabling support for large CRLs" on page 223

Scaling for high volume installations

Some PKI Services installations manage a large number of certificates and certificates requests. The following guidelines can help you scale your system to maintain high performance in a high volume environment.

Guidelines:

1. Use distribution point CRLs if you will average more than 500 revoked non-expired certificates at any given time. For more information, see "Customizing distribution point CRLs" on page 217.
2. If you anticipate having a large number of certificate requests pending approval at any given time, implement a PKI exit to automate the approval process. (For more information, see Chapter 15, "Customizing with installation exit routines," on page 267.) This need arises from the human limitation rather than a technical one because it becomes nearly impossible to manually approve the requests when the volume grows too high.
3. To prevent name collisions in the LDAP directory, ensure that the subject distinguished names are unique. This can either be done by implementing a PKI exit to supply a unique name, or by enforcing the use of the `MAIL=` distinguished name attribute where you require the e-mail address to be unique.
4. Queries against the request or ICL database can time out if the database contains a large number of records. The performance of the query can be vastly improved by supplying the requestor's name as additional search criteria if the saved requestor data is meaningful to your organization and it is recallable. In this case, a PKI exit can be used to supply a meaningful value, such as a Lotus® Notes® short name or customer account number.

© Copyright IBM Corp. 2001, 2011

5. Keep the size of the request and ICL databases small by quickly removing records that are no longer needed. This can be done by setting low values for the following fields in the **ObjectStore** section of the PKI Services configuration file (`pkiserv.conf`):
   - `RemoveCompletedReqs`
   - `RemoveInactiveReqs`
   - `RemoveExpiredCerts`

Using certificate policies

Certificates can contain a CertificatePolicies extension. This extension contains policy information, such as the way in which your CA operates and the intended purpose of the issued certificates. (For more information about this extension, see RFC 2459: *Certificate and CRL Profile*.)

The CertificatePolicies extension contains one or more PolicyInformation sequences. (Typical usage has just one of these.) The PolicyInformation sequence has the following format:

- Your Policy OID as registered with the appropriate standards organization (ISO or ITU)
- Zero or more PolicyQualifiers sequences, each having the following information:
  - Either a Certificate Practices Statement (CPS) URI
  - Or a UserNotice sequence, which consists of one or both of the following:
    - A notice (text string) intended to be viewed by customers using the certificate such as copyright or other legal information
    - Your organization's legal name (text string) with one or more notice numbers defined elsewhere, perhaps in your CPS.

By default, PKI Services does not include this extension in the certificates it creates. However, you can define your own CertificatePolicies extension by modifying fields in the **CertPolicy** section of the `pkiserv.conf` configuration file. You can also specify the `PolicyRequired` value to indicate whether a CertificatePolicies extension should be created for all certificate templates on a global basis or whether one is individually created based on the specifications of each certificate template.

PolicyRequired=T
: Indicates that the CertificatePolicies extension will be created on a global basis with the same values for all certificate templates based on the specifications in the **CertPolicy** section of the PKI Services configuration file. Any policies specified within the template will be ignored.

  See “Steps for creating the CertificatePolicies extension on a global basis” on page 213.

PolicyRequired=F (default)
: Indicates that the CertificatePolicies extension will be optionally created based on the specifications for each individual certificate template. If you are implementing the Web application using REXX CGI execs, the specification is done in the CONSTANT section of `pkiserv.tmpl`. If you are using Java server pages (JSPs), the specification is done with the CertPolicies tag in `pkitmpl.xml`. Policies specified within the template are used.

  See “Steps for creating the CertificatePolicies extension on a template basis” on page 214.

Note: PolicyCritical is ignored unless `PolicyRequired=T`. When `PolicyRequired=F`, setting `%%Critical=CertPolicies%%` in the CONSTANT section of the template will mark the extension critical.

Restriction: When policies are specified within an individual template, the policy data is saved with the request at the time the request is submitted or modified. Therefore, if PKI Services is stopped and restarted to make changes in the policy data before the certificate is issued, the changes will not be reflected in the issued certificate. However, the `PolicyRequired=F` setting is checked at the time the certificate is issued. Therefore, if PKI Services is stopped and restarted to make changes to the `PolicyRequired` setting before the certificate is issued, the new setting will be used to determine which policy information is used (the global policy data or the data saved with the request.)

Steps for creating the CertificatePolicies extension on a global basis

Perform the following steps to create your own CertificatePolicies extension on a global basis:

1. Edit the `pkiserv.conf` configuration file and find the **CertPolicy** section.

---

2. Change the value of `PolicyRequired` to `T` (True) as in the following line:

   ```
   PolicyRequired=T
   ```

---

3. If you want to have the extension marked critical (this is not suggested), set the `PolicyCritical` equal to `T` (True) as in the following line:

   ```
   PolicyCritical=T
   ```

---

4. Go to the **OIDs** section of the `pkiserv.conf` configuration file. By default (as shown in the following example), the name is `MyPolicy=1.2.3.4` and value is `1.2.3.4`. The value of MyPolicy should be an installation-specific (registered) Object ID identifying your organization's certificate. Replace the value of MyPolicy in the following line with your Object ID.

   Example:

   ```
   [OIDs]
   MyPolicy=1.2.3.4
   ```

   Optionally, change the parameter name `MyPolicy` to your own installation-specific name. If you change the parameter name in this step, make a note of it. You need it for the next step. You can repeat the `MyPolicy` parameter using unique names and values if you need to define multiple policies.

   Example:

   ```
   MyPolicy=1.2.3.4
   MyOtherPolicy=2.3.4.5
   ```

---

5. If you changed the parameter name `MyPolicy` in the previous step, go back to the **CertPolicy** section and update the `PolicyName1` line to change the `MyPolicy` parameter to the policy name you specified in the **OIDs** section:

   ```
   [CertPolicy]
   PolicyName1=MyPolicy
   ```

6. If you want to add qualifiers, perform the following steps:

   a. Update the Policy1Org and Policy1Notice*n* fields in the following example:

      ```
      Policy1Org=My Company,Inc
      Policy1Notice1=1
      ```

      Policy1Org Your organization's name, for example, `International Business Machines, Inc.`

      Policy1Notice1 through Policy1Notice*n*
      Your notice numbers. (You might need more than one Policy1Notice*n* line, depending on how many notice numbers you have. Repeat the line as needed, by incrementing the suffix number on the keyword, for example Policy1Notice1, Policy1Notice2, and so forth.)

   b. Change the value of the UserNoticeText1 line shown in the following. The *statement* should be your notice text string, for example, `Certificate for IBM internal use only`.

      ```
      UserNoticeText1=statement
      ```

   c. Change the value of the CPS1 line shown in the following. The value should be your CPS URI, for example, `http://www.ibm.com/cps.html`.

      ```
      CPS1=http://www.mycompany.com/cps.html
      ```

   If you do not want to add qualifiers, delete or comment out (by inserting a # character at the start of the line) the preceding lines.

7. If you need multiple qualifiers, repeat the following fields as needed, incrementing the suffix numbers, for example:

   ```
   PolicyName2=MyOtherPolicy
   Policy2Org=International Business Machines, Inc.
   Policy2Notice1=5
   Policy2Notice2=9
   UserNoticeText2=Certificate is intended for testing only
   CPS2=http://www.ibm.com/cps2.html
   ```

8. If you made any changes to the PKI Services configuration, stop and restart PKI Services to activate the changes.

Steps for creating the CertificatePolicies extension on a template basis

Perform the following steps to create your own CertificatePolicies extension on an individual template basis:

1. Edit the `pkiserv.conf` configuration file and find the **CertPolicy** section.

2. Change the value of `PolicyRequired` to `F` (False) as in the following line:

   ```
   PolicyRequired=F
   ```

3. Follow steps 4 through 7 in "Steps for creating the CertificatePolicies extension on a global basis" on page 213 to create the individual policies you need.

4. Update the certificate template to specify the CertificatePolicies extensions that are to be created for it.
   - **If you are implementing the Web application using REXX CGI execs:** Edit `pkiserv.tmpl` and customize the CONSTANT subsection under the certificate template for which you need CertificatePolicies extensions.

     For example, if you have specified values for PolicyName1, PolicyName3, and PolicyName6 in `pkiserv.conf`, then you can specify the certificate policies in `pkiserv.tmpl` in the following ways:

     ```
     %%CertPolicies=3%%
      or
     %%CertPolicies=3 6%%
      or
     %%CertPolicies=1 3 6%%
     ```

     If you want to make the CertPolicies extension critical, specify the following in the CONSTANT section:

     ```
     %%Critical=CertPolicies%%
     ```

   - **If you are implementing the Web application using Java server pages (JSPs):** Edit `pkitmpl.xml` and customize the section for the certificate template for which you need CertificatePolicies extensions.

     For example, if you have specified values for PolicyName1, PolicyName3, and Place-name in `pkiserv.conf`, then you can specify the certificate policies in `pkitmpl.xml` in the following ways:

     ```
     <tns:CertPolicies>3</tns:CertPolicies>
      or
     <tns:CertPolicies>3 6</tns:CertPolicies>
     or
     <tns:CertPolicies>1 3 6</tns:CertPolicies>
     ```

     If you want to make the CertPolicies extension critical, specify the following in the certificate template section:

     ```
     <tns:Critical>CertPolicies</tns:Critical>
     ```

   Rule: The policy numbers in the template file must exist in the `pkiserv.con` file. For each template, you can choose a different subset of these numbers.

5. If you made any changes to the PKI Services configuration, stop and restart PKI Services to activate the changes.

Updating the signature algorithm

The signature algorithm that PKI Services uses to sign certificates must be based on the key type of the CA certificate. If it is not, PKI Services is unable to start. By default, IKYSETUP creates the CA certificate with an RSA key pair. The default value of the signature algorithm in the `pkiserv.conf` file is `sha–256WithRSAEncryption`. You can change the signature algorithm by changing the `SigAlg1` value in the **CertPolicy** section of the `pkiserv.conf` configuration file. Set `SigAlg1` to one of the algorithm identifiers shown in Table 45 on page 216 for the key type of the CA certificate.

Table 45. Supported signature algorithms for each CA certificate key type

CA certificate key type	Algorithm nickname	Algorithm OID
RSA	sha-1WithRSAEncryption	1.2.840.113549.1.1.5
	sha-224WithRSAEncryption	1.2.840.113549.1.1.14
	sha-256WithRSAEncryption	1.2.840.113549.1.1.11
	sha-384WithRSAEncryption	1.2.840.113549.1.1.12
	sha-512WithRSAEncryption	1.2.840.113549.1.1.13
DSA	id-dsa-with-sha1	1.2.840.10040.4.3
ECC	ecdsa-with-sha1	1.2.840.10045.4.1
	ecdsa-with-sha224	1.2.840.10045.4.3.1
	ecdsa-with-sha256	1.2.840.10045.4.3.2
	ecdsa-with-sha384	1.2.840.10045.4.3.3
	ecdsa-with-sha512	1.2.840.10045.4.3.4

Tips: Consider these points when choosing the signature algorithm:

- In general, a SHA1 hash is more secure than an MD2 or MD5 hash. MD2 and MD5 have been found to be vulnerable to attack and should not be used.
- sha-256WithRSAEncryption is more secure than sha-1WithRSAEncryption, but some browsers cannot install certificates that use sha-256WithRSAEncryption.
- The National Institute of Standards and Technology (NIST) recommends that SHA1 not be used after the year 2010.

Steps for changing the signature algorithm

Before you begin: Change the signing algorithm *before* you create any certificate requests. If changing the signing algorithm *after* some certificates requests have been created, you must wait until all requests are approved and the certificates created, or else you must add a `SigAlg2=old-signing-algorithm` line to the **CertPolicy** section. If you take this second option, `SigAlg1` becomes the signature algorithm for new requests.

Perform the following steps to change the signature algorithm:

1. Edit the `pkiserv.conf` configuration file and find the **OIDs** section.

---

2. Ensure that the OID for the signature algorithm that you want to use is listed. (If you are using a version of `pkiserv.conf` from a previous release, OIDs added in later releases will be missing if you have not added them.) If the OID is missing, add it by adding a line with the following format:

   algorithm_nickname=algorithm_OID

   where the values of *algorithm_nickname* and the corresponding *algorithm_OID* are from Table 45. For example, if you want to use SHA512,

   `sha-512WithRSAEncryption=1.2.840.113549.1.1.13`

---

3. Find the **CertPolicy** section. If you want to use a signature algorithm other than SHA256, change `sha-256WithRSAEncryption` in the following line to the

value for the signature algorithm that you want to use. The supported values for each CA certificate key type are shown in Table 45 on page 216.

```
SigAlg1=sha-256WithRSAEncryption
```

Note: If your configuration file is from a release prior to z/OS V1R12 and contains the default value, the line for the signature algorithm is:

```
SigAlg1=sha-1WithRSAEncryption
```

Customizing distribution point CRLs

If your PKI Services installation is very active, many certificates can be in the revoked state at any one time. Therefore, the certificate revocation list (CRL) can become quite large, causing considerable network traffic and overhead to an application wishing to process it. Publishing partial CRLs to multiple distribution point (DP) CRLs is a way of keeping your CRLs small.

Guideline: Consider using distribution point CRLs if you anticipate averaging more than 500 revoked non-expired certificates at any given time.

You begin using distribution point CRLs when you accept the defaults settings contained in PKI Services configuration file (`pkiserv.conf`). You can customize those settings by specifying the number of certificates per DP CRL and by specifying the name of the DP CRL using the following two parameters in the **CertPolicy** section of the `pkiserv.conf`:

`CRLDistSize` Specifies the maximum number of certificates to be managed by a single DP. This represents the number of entries in each DP CRL if all active certificates are revoked at once.

`CRLDistName` Specifies the file name, or the constant portion of the leaf-node RDN, for the CRL distribution point.

You can choose to further customize your DP CRL processing to build the URI format name for the distribution point in the CRLDistributionPoints extension of each certificate. This allows your certificate validation programs to dynamically retrieve a CRL without being preconfigured with LDAP bind information. However, because bind credentials cannot be added to DP CRLs with URI format names, anonymous access is used to retrieve the CRL.

The URI format name is built in *addition* to the LDAP distinguished name of the DP CRL that is always added when `CRLDistSize` is greater than zero. You can add the URI format name by customizing the following two parameters in the **CertPolicy** section of the `pkiserv.conf`:

`CRLDistURI`*n* Specifies the name for the DP CRL in the form of a URI that adds the protocol type and the server domain name.

`CRLDistDirPath`
Specifies the full path for the file system directory where PKI Services will save each DP CRL.

You can also choose to have PKI Services create a CRLDistributionPoints extension for each CA certificate in addition to non-CA certificates. You choose this by customizing the `ARLDist` parameter in the **CertPolicy** section of the `pkiserv.conf`. This creates a distribution-point authority-revocation list (DP ARL) for your CA certificates. See “Creating a distribution point ARL” on page 225 for details.

Specifying the URI format

When you choose to use distribution points for CRL and ARL processing, PKI Services updates the CRLDistributionPoints extension with the distinguished name for the LDAP entry where the distribution point is posted. You can choose to add another name to the extension in the URI format which contains the protocol type and the server domain name in addition to the distinguished name. With the URI format, the location of the distribution point is self-contained in the CRLDistributionPoints extension.

The URI format contains the following information:

- the protocol type (LDAP or HTTP)
- the server domain name
- if the protocol is LDAP:
  - the distinguished name of the distribution point
  - for non-CA certificates, the attribute string `?certificateRevocationList`
  - for CA certificates, the attribute string `?authorityRevocationList`
- if the protocol is HTTP, the virtual or real pathname, ending with the file name - formed from the common name portion of the distinguished name of the distribution point with the `.crl` extension - where the distribution point CRL is stored.

Examples:

```
ldap://ldap.bankxyz.com:389/CN=CRLlist1,OU=Bank XYZ
       Authority,O=Bank XYZ,C=US?certificateRevocationList

http://www.bankxyz.com/PKIServ/cacerts/CRLlist1.crl
```

Note: This is an example of an HTTP protocol URI using a virtual pathname. When using virtual pathnames in an HTTP URI, a Pass statement will be required in the HTTP configuration file to map the virtual pathname to a real pathname. See "Determining CRLDistDirPath" on page 219 for additional information.

Restriction: Special characters, such as spaces, quotation marks, and square brackets are not considered *safe* to use in URLs and should be encoded using the appropriate *escape* sequence. For details, see RFC 1738: *Uniform Resource Locators (URL).*

Determining CRLDistURI*n*

If you are using DP CRLs (you specified a `CRLDistSize` value greater than **1** in the **CertPolicy** section of `pkiserv.conf`), you can choose to further customize your DP CRL processing to build the URI format name for the DP CRL in the CRLDistributionPoints extension of each certificate. The URI format name is built in *addition* to the LDAP distinguished name of the DP CRL, as described in "Specifying the URI format."

This is an optional parameter. If you do not specify a `CRLDistURI`*n* value, the URI format name will not be created. You can specify multiple entries for the `CRLDistURI`*n* parameter, using the parameters `CRLDistURI1`, `CRLDistURI2`, and so forth. This value is ignored if you did not specify `CRLDistSize` with a value greater than zero. The URI format will not be created if you specify `CRLDistURI`*n* with an *n* value of **0**.

There are different ways to specify the value of `CRLDistURI`*n* for different protocols. **Valid values** include *one* of the following:

- A string that begins with the characters `http://` or `ldap://`

- A string that consists of LdapServer*n*, where *n* is greater than zero.

Restriction: PKI Services provides syntax checking based only on valid values for the CRLDistURI*n* value. You must ensure that the URIs you choose can be accessed.

Specifying an HTTP URI

For HTTP, specify the complete URL but do not specify the name of a file where the CRL DP is stored. The value for CRLDistURI*n* can be specified with or without a trailing slash.

Example:

```
CRLDistURI1=http://www.bankxyz.com/PKIServ/cacerts/
```

Note: This is an example of an HTTP protocol URI using a virtual pathname. When using virtual pathnames in an HTTP URI, a Pass statement will be required in the HTTP configuration file to map the virtual pathname to a real pathname. See "Determining CRLDistDirPath" for additional information.

Specifying an LDAP URI

For LDAP, there are two ways to indicate the CRLDistURI*n* value. Choose either of the following two methods:

- Specify the protocol and the domain name (and the port, if needed). The value for CRLDistURI*n* can be specified with or without a trailing slash.

  Example:

  ```
  CRLDistURI1=ldap://ldap.bankxyz.com:389/
  ```

- Specify the keyword LdapServer*n* to have PKI Services build the CRLDistURI*n* value for you based on a server identified by the Server*n* or BindProfile*n* directives in the **LDAP** section of pkiserv.conf.

  Example:

  ```
  CRLDistURI3=LdapServer1
  ```

  This example assumes that the first server specified in the **LDAP** section was similarly defined as one of the following examples:

  Examples:

  ```
  Server1=ldap.bankxyz.com:389
            or
  BindProfile1=LOCALPKI.BINDINFO.LDAP1
  ```

  Rules for using the LdapServer*n* keyword:

  1. You must have specified a value greater than zero for NumServers in the **LDAP** section of pkiserv.conf.
  2. Each server represented by the *n* value in the LdapServer*n* keyword must be identified in *one* of the following ways:
     - The corresponding LDAP server must be identified by a Server*n* or BindProfile*n* value in the **LDAP** section of pkiserv.conf, *or*
     - The corresponding LDAP server must be identified in the default FACILITY class profile IRR.PROXY.DEFAULTS and must follow the same identification requirements for PKI Services LDAP processing. See "Using encrypted passwords for LDAP servers" on page 420.

Determining CRLDistDirPath

If the protocol for the URI you specified with CRLDistURI*n* is HTTP protocol, you need to also determine your value for the CRLDistDirPath parameter. The

CRLDistDirPath parameter specifies the full path of the var directory where PKI Services will save each DP CRL. The value can be specified with or without the trailing slash. The default value is /var/pkiserv/. If you are customizing this value for a CA Domain, it is recommended that you specify a directory name that contains the CA Domain name, for example /var/pkiserv/employees/. In a case such as this, it is also necessary to add an additional Pass statement to the HTTP configuration file that will map the virtual path name in the URI to the real path specified by this CRLDistDirPath statement.

Statements in the pkiserv.conf file:

```
CRLDistURI1 = http://www.bankxyz.com/Employees/crls/
CRLDistDirPath = /var/pkiserv/Employees/
```

Matching Pass statement in the HTTP configuration file:

```
Pass /Employees/crls/* /var/pkiserv/Employees/*
```

The default value is /var/pkiserv/. See "Specifying the URI format" on page 218. This value is ignored if you do not create a CRLDistributionPoints extension or if the URI protocol is LDAP.

Steps for customizing distribution point CRLs

Before you begin: Be aware of the following **restrictions**:

- If running PKI Services in a sysplex, all instances of PKI Services must specify the same values for the parameters CRLDistURI*n* (for all values of *n*), CRLDistDirPath, CRLDistSize, and CRLDistName.
- Once a value for CRLDistName has been set, it must not be changed or removed from the configuration file.
- Once a non-zero value has been set for CRLDistSize, it must not be changed back to zero or removed from the configuration file. Adjusting the value is acceptable.

Perform the following steps to customize distribution point CRLs:

1. Determine your value for the CRLDistSize parameter based on the following algorithm. The default value specified in pkiserv.conf is 500. Your value should be based on your desired average number of CRL entries per distribution point and your estimated revoked-certificate percentage as expressed by the following formula:

   ```
   CRLDistSize = E ⁄ P
   ```

   where:
   E is the desired average number of CRL entries per distribution point.
   P is the estimated revoked-certificate percentage.

   Example: If you estimate that 10% of the non-expired certificates will be in the revoked state at any given time and you wish the CRLs to average around 100 entries each, then:

   ```
   CRLDistSize = 100 ⁄ 0.10 = 1000
   ```

   The CRLDistSize in bytes can be roughly estimated to be 500 + (25 × *number of CRL entries*). Using the example above, the average CRL size in bytes would be 500 + (25 × 100) = 3000 bytes.

   Note: The longer the CRL, the longer it will take to process it.

   Restriction: When CRLs are posted to LDAP, a single CRL cannot exceed approximately 32K bytes in length, unless you have enabled support for

large CRLs. For more information, see "Enabling support for large CRLs" on page 223. Therefore, if a CRL is posted to LDAP and you have not enabled support for large CRLs, you must limit the length of the CRL.

Rules:

a. The value of `CRLDistSize` is a numeric value from 0 - 2147483647.
b. A non-zero value indicates that distribution point (DP) CRLs will be created.
c. A value of zero (the default) indicates that DP CRL processing will not be performed.

Guideline: If you anticipate a low revocation rate for active certificates, use a value of **0**. Your installation might not need to use distribution point CRLs and the global CRL might be sufficient.

---

2. If necessary, update the value of `CRLDistSize` in the **CertPolicy** section of `pkiserv.conf` to the customized value you determined in Step 1 on page 220.

   If you selected the **0** value for `CRLDistSize`, complete this step and then continue with Step 12 on page 222.

---

3. Determine your value for the `CRLDistName` parameter. The default value is `CRL`. The common name portion of the distinguished name of each DP CRL is formed by appending the DP number to this value. The CA's name is also appended. (See "How DP CRLs are published" on page 223.)

   Example:

   ```
   CN=CRL3,OU=My Company Certificate Authority,O=My Company,C=US
   ```

   Restrictions:

   a. The value of `CRLDistName` must contain only alphanumeric characters.
   b. The length of the entire DP distinguished name should not exceed 255 bytes. (DP distinguished names that are longer will appear truncated in the PKIDPUBR audit record.)

---

4. If necessary, update the value of `CRLDistName` in the **CertPolicy** section of `pkiserv.conf` to your customized value.

---

5. Optionally, determine your value for the `CRLDistURI`*n* parameter. Specifying this value will allow PKI Services to build a URI-formatted name for the DP CRL in each CRLDistributionPoints extension, if you also specified a `CRLDistSize` value greater than **1** in Step 2. The URI format name is built in *addition* to the LDAP distinguished name of the DP CRL in each CRLDistributionPoints extension. If you do not specify a `CRLDistURI` value, the URI format name will not be created. See "Specifying the URI format" on page 218.

   You can specify multiple entries for the `CRLDistURI`*n* parameter, using the parameters `CRLDistURI1`, `CRLDistURI2`, and so forth.

---

6. If necessary, update the value of `CRLDistURI`*n* in the **CertPolicy** section of `pkiserv.conf` to your customized value or values.

---

7. If all the protocol definitions for the URIs you specified with `CRLDistURIn` in Step 5 on page 221 are the LDAP protocol, decide whether you want to enable support for large CRLs. If you choose to enable large CRLs, follow the instructions in "Enabling support for large CRLs" on page 223. Then skip to Step 10.

8. If a protocol definition for the URI you specified with `CRLDistURIn` in Step 5 on page 221 is HTTP protocol, determine your value for the `CRLDistDirPath` parameter.

   The `CRLDistDirPath` parameter specifies the full path of the `var` directory where PKI Services will save each DP CRL. The default value is `/var/pkiserv/`. The value can be specified with or without the trailing slash. See "Determining CRLDistDirPath" on page 219.

9. If necessary, update the value of `CRLDistDirPath` in the **CertPolicy** section of `pkiserv.conf` to your customized value.

10. Optionally, determine your value for the `ARLDist` parameter. Specifying this parameter creates a distribution point ARL so you can check revocation status for CA certificates without accessing the global ARL. See "Creating a distribution point ARL" on page 225.

11. If necessary, update the value of `ARLDist` in the **CertPolicy** section of `pkiserv.conf` to your customized value.

12. If you made any updates to `pkiserv.conf`, stop and restart PKI Services to make your changes effective.

When you have finished: If you selected a `CRLDistSize` value greater than zero, you have set up distribution point CRLs. Now, created certificates will contain the CRLDistributionPoints extension indicating the location of the DP CRL that will be checked for revocation information. If you specified a URI-formatted name with `CRLDistURIn`, now your CRLDistributionPoints extensions will also contain a URI name for each DP CRL, containing the protocol type and server domain name. If you enabled the `ARLDist` option, you have set up a distribution point ARL for CA certificates.

How distribution point CRLs work

PKI Services always creates a *global* CRL regardless of whether or not you choose to use DP CRLs. The global CRL contains revocation information for certificates that have no CRLDistributionPoints extension (in other words, certificates defined with `CRLDistSize=0`). When a certificate contains a CRLDistributionPoints extension, PKI Services publishes its revocation status to the appropriate DP CRL, not in the global CRL.

The following topics will help you understand more about how DP CRLs work. This information will be useful if you write applications that process CRLs.

How DP CRLs are published

DP CRLs are published to LDAP at leaf nodes directly below the CA's entry. For example, if the CA's name is:

```
OU=My Company Certificate Authority,O=My Company,C=US
```

Then, the DP CRLs would be published to:

```
CN=DP-name,OU=My Company Certificate Authority,O=My Company,C=US
```

How DP CRLs are partitioned

The partitioning of the overall CRL into partial CRLs is based on certificate serial number and the value of CRLDistSize in `pkiserv.conf`. For example, if CRLDistSize is `100` and CRLDistName is `ABC`, then certificates with serial numbers `1–100` will appear on DP `ABC1`; `101–200` on DP `ABC2`, and so on. PKI Services dynamically creates DP CRLs as needed as a part of certificate issuance. Existing DP CRLs are refreshed along with the global CRL during CRL interval processing.

As certificates expire, they are no longer eligible for revocation and will not appear on any CRL. Therefore, over time, each distribution point will become inactive. PKI Services automatically retires DP CRLs that become inactive by no longer publishing their CRLs. However, retired DP CRLs previously published to LDAP remain in LDAP. PKI Services makes no attempt to delete these.

Even when using distribution point CRLs, the single non-DP CRL (global CRL) is still created. Revoked certificates containing the CRLDistributionPoints extension will appear only on the appropriate DP CRL, not the global CRL.

What about CA certificates?

PKI Services can be used to create other subordinate certificate-authority certificates. Since revocation activity against these CA certificates is normally low, PKI Services by default does not partition authority revocation lists (ARLs). You can choose to create a distribution point ARL in a single partition for the purpose of checking the revocation status of CA certificates. (See "Creating a distribution point ARL" on page 225.) When you choose to create a DP ARL, your CA certificates will contain a CRLDistributionPoints extension.

When you do not choose to create a DP ARL (`ARLDist=F`), applications wishing to check the revocation status of a CA certificate must check the global ARL. In addition, when `ARLDist=F`, CA certificates do not contain a CRLDistributionPoints extension, although they are treated as if they had the extension when determining the partitioning of the global CRL. For instance, with `ARLDist=F` and `CRLDistSize=10`, if you issue 10 CA certificates plus one non-CA certificate, the non-CA certificate information would be published to the second distribution point CRL. (The first DP CRL would remain empty.)

Enabling support for large CRLs

When LDAP posting of certificate revocation lists (CRLs) is enabled, by default PKI services temporarily stores CRLs in its object store for posting to LDAP. However, PKI Services imposes a limit on the size of records in the object store of approximately 32KB, which limits the size of the CRLs stored there to approximately 32KB. As certificates are revoked or suspended within the scope of a CRL, the size of the CRL increases, and can exceed the limit. If a CRL exceeds the 32KB limit, PKI Services cannot post it to the LDAP directory.

To avoid this problem, you can configure PKI Services to store CRLs for posting to LDAP in the z/OS UNIX file system instead of in the object store. When you do this, there is no limit on the size of CRLs.

Steps for enabling support for large CRLs

Before you begin: You need to be familiar with the z/OS UNIX file systems.

Perform the following steps to enable support for CRLs larger than the limit of approximately 32KB.

1. Configure a z/OS file system to hold CRLs. If you run multiple instances of PKI Services in a Parallel Sysplex (one per image), ensure that the file system is shared with each PKI Services instance in the sysplex. For information about managing the z/OS UNIX file system, see *z/OS UNIX System Services Planning*.

   _____

2. Set the value of the `LargeCRLPostPath` parameter in the **CertPolicy** section of the PKI Services configuration file, `pkiserv.conf`, to the full path of the `var` directory where PKI Services is to save each CRL for posting to LDAP. The default value is `/var/pkiserv/`. You can specify the value with or without the trailing slash. The value of `LargeCRLPostPath` can be the same as the value of `CRLDistDirPath`.

   Guideline: If you are customizing this value for a CA domain, specify a directory name that contains the CA domain name, for example `/var/pkiserv/employees/`, where `employees` is the domain name.

   _____

3. Set the value of the `EnableLargeCRLPosting` parameter in the **CertPolicy** section of the PKI Services configuration file, `pkiserv.conf`, to `T`.

   _____

4. Restart PKI Services. Your changes to the `pkiserv.conf` file do not take effect until you do this. For information about starting PKI Services see Chapter 10, “Starting and stopping PKI Services,” on page 117.

   _____

When you are done, you have enabled support for large CRLs, and CRLs are no longer limited to approximately 32KB in size.

Using the OCSP responder

As an alternative, or in addition to publishing revocation information with CRLs, you can choose to enable an Online Certificate Status Protocol (OCSP) responder. An OCSP responder is enabled when `OCSPType` is set to `basic` in the **CertPolicy** section of the PKI Services configuration file as shown in Table 19 on page 60, and when the certificate contains the necessary OCSP responder information in the AuthInfoAccess extension. (See “TEMPLATE sections” on page 136.)

In order to use an OCSP responder, you must add `/usr/lpp/pkiserv/lib` to the LIBPATH environment variable for the HTTP Server, in the `httpd.envvars` file.

Creating a distribution point ARL

You can choose to create a distribution point (DP) authority revocation list (ARL) to support revocation status checking for certificate authority (CA) certificates. You choose DP ARL processing by customizing the `ARLDist` parameter in the **CertPolicy** section of the `pkiserv.conf`. If you do not customize this parameter, PKI Services does not partition the ARL and, therefore, applications must check the global ARL to check the revocation status of a CA certificate.

ARLDist=F (default)
No distribution point ARL will be created.

ARLDist=T
When distribution point CRLs are also enabled (when `CRLDistSize` is greater than zero), you can specify **T (True)** to create a distribution point ARL.

When DP ARL processing is enabled, PKI Services provides the following support:

- Create a single distribution point (DP) for all CA certificates
- Build a CRLDistributionPoints extension containing both the distinguished name and the URI format for the DP.

DP ARL processing for CA certificates is similar to the DP CRL processing for non-CA certificates with the following differences:

- There is only one DP ARL. Its name is formed by the value specified in the `CRLDistName` parameter in the **CertPolicy** section of the `pkiserv.conf`, appended with **0** (zero). By appending a zero, the name of the DP ARL never conflicts with the name of a DP CRL. For example, if `CRLDistName=CRL`, then the DP ARL is named `CRL0`, and the DP CRLs are named `CRL1`, `CRL2`, and so forth.
- The DP ARL is a mirror copy of the global ARL. In other words, each revoked CA certificate will appear in both the DP ARL and the global ARL. By contrast, a revoked non-CA certificate is listed in the DP CRL but not in the global CRL when DP CRL processing is enabled.
- The attribute string appended to the URI format for the LDAP protocol is `?authorityRevocationList`. Otherwise, the CRLDistributionPoints extension of a CA certificate appears quite similar to that of a non-CA certificate. See Figure 28 on page 226 for a sample CRLDistributionPoints extension for a CA certificate. This sample contains several different name formats. Notice the URI format located at the end of the sample.

```
SEQUENCE {
   OBJECT IDENTIFIER cRLDistributionPoints (2 5 29 31)
   OCTET STRING, encapsulates {
       SEQUENCE {
          SEQUENCE {
              [0] {
                [0] {
                  [4] {
                    SEQUENCE {
                      SET {
                        SEQUENCE {
                          OBJECT IDENTIFIER
                            countryName (2 5 4 6)
                          UTF8String (1997) 'US'
                          }
                      }
                      SET {
                        SEQUENCE {
                          OBJECT IDENTIFIER
                            organizationName (2 5 4 10)
                          UTF8String (1997) 'Mycompany'
                          }
                      }
                      SET {
                        SEQUENCE {
                          OBJECT IDENTIFIER
                            organizationalUnitName (2 5 4 11)
                          UTF8String (1997) 'Retail'
                          }
                      }
                      SET {
                        SEQUENCE {
                          OBJECT IDENTIFIER
                            commonName (2 5 4 3)
                          UTF8String (1997) 'CRL0'
                          }
                      }
                    }
                  }
                }
              }
          }
          SEQUENCE {
              [0] {
                [0] {
                  [6]
                  'http://crl.MyCompany.de/CRL0.crl'
                }
              }
          }
          SEQUENCE {
              [0] {
                [0] {
                  [6]

'ldap://ldap.MyCompany.de/CN=CRL0,OU=Retail,O=Mycompany,C=US?authorityRevocationList'
                }
              }
          }
       }
   }
 }
```

Figure 28. A sample CRLDistributionPoints extension for a certificate authority (CA) certificate

Adding an application domain

This topic describes adding a new application domain. If you want to add a new certificate authority (CA) domain, see "Adding a new CA domain" on page 230.

By default, all CGIs or JSPs (depending on the method you are using to implement the PKI Services Web application) reside under a common URL. Based on this, all users, including PKI administrators, have the same PKI Services home page, Web content and supported certificate templates. In other words, by default, there is a single application domain: PKISERV.

The sample PKI Services template files (`pkiserv.tmpl` and `pkitmpl.xml`) contain two application sections: PKISERV and CUSTOMERS. Using these two application sections, your users can be easily divided between two subsets - customers and administrators.

You will probably want to separate your administration users and your end users. You might also need to further subset your end user population by adding different application domains for different groups of end users. Both of these objectives can be accomplished by using multiple applications. The PKI administrators and the different subsets of end users can be directed to different application domains at different URLs.

Creating application domains when you use REXX CGIs to implement the Web application

To create multiple application domains, execute the subtasks in Table 46. Both are tasks for the Web server programmer.

Table 46. Task roadmap for creating multiple application domains

Subtask	Associated procedures (See ...)
Update the PKI Services template file	"Steps for creating multiple application sections in the PKI Services template file"
Update the Web server configuration files	"Steps for adding application domains to the Web server configuration files" on page 228

Steps for creating multiple application sections in the PKI Services template file

Perform the following steps to create multiple application sections to the PKI Services template file:

1. Edit the `pkiserv.tmpl` file and find the CUSTOMERS application section.

   ---

2. Replicate the CUSTOMERS section and specify a unique name for the new application section.

   `<APPLICATION NAME=`*`appl-section-name`*`>`

   Rule: The application section name must be one word, all uppercase characters.

   ---

3. Determine which certificate types are required by this user subset. Based on these requirements, select the certificate templates that belong in the new application set by adding or removing template names from this new section as needed.

   ---

4. Customize the content of the Web pages for this application by modifying the <CONTENT>...</CONTENT> subsection. (See "TEMPLATE sections" on page 136 for a description of each subsection.)

   ---

5. Similarly, customize the original CUSTOMERS application by re-executing Steps 3 and 4, this time editing the content of the CUSTOMERS Web pages.

   ---

6. Repeat Steps 1–4 for each application section you need to add.

---

7. Optionally, rename the original CUSTOMERS application to a new section name, if desired.

   `<APPLICATION NAME=section-name>`

   Rule: The application section name must be one word, all uppercase characters.

Steps for adding application domains to the Web server configuration files

Before you begin:

- This procedure requires Web server programming skills and requires editing both the `httpd.conf` file and the `httpd1443.conf` file.
- The home page URL for the new or renamed domains would be as follows:

  `http://<webserver-fully-qualified-domain-name>/<new-appl-domain-name>/public-cgi/camain.rexx`

  where *new-appl-domain-name* corresponds to the new section name added in the template file in "Steps for creating multiple application sections in the PKI Services template file" on page 227. However, in the Web server files, the new name is case-sensitive but need not be in upper case only.
- Make note of the case you select for each character of the new *new-appl-domain-name* name. This case-sensitive value will become part of the URL for your home page and you must use it consistently in each set of `Protect`, `Redirect`, and `Exec` statements you add to your Web server configuration file.
- The administration home page URL does not change. (There is one common administration application that handles all application domains.)
- If your PKI installation has changed the name of the `Customers` domain, you must change all occurrences of `Customers` to its new value in both files. (The new value is not case-sensitive.)
- If your installation has added a new application domain, use the following procedure.

Perform the following steps to add domains or rename the `Customers` domain in Web server configuration files `httpd.conf` and `httpd1443.conf` for each new application section added to `pkiserv.tmpl`:

1. Replicate each of the following lines in both the `httpd.conf` and `httpd1443.conf` files:

   ```
   Protect /Customers/...
   Redirect /Customers/...
   Exec /Customers/...
   ```

---

2. Change each occurrence of the name `Customers` in the replicated lines in both files to the new application domain name.

   Example:

   ```
   Protect /Employees/...
   Redirect /Employees/...
   Exec /Employees/...
   ```

In contrast to the application section name, the domain name value is case-sensitive and it need not be uppercase. However, you must use it consistently in each set of `Protect`, `Redirect`, and `Exec` statements. This value will become part of the URL for your home page.

When you are done: You have defined a new PKI Services application domain at:

Example:

`http://<webserver-fully-qualified-domain-name>/employees/public-cgi/camain.rexx`

Creating application domains when you use JSPs to implement the Web application

An application, or application domain, is a grouping of certificate request templates that are typically targeted at a subset of end users. The Customers application is the default end-user application. But the `web.xml` file that ships as part of `PKIServ.ear` also defines mappings for a second application called Application2, making it easy to set up a second application domain.

To set up Application2, edit `pkitmpl.xml` and use the `<tns:application>` tag to define which certificate request templates should be part of Application2. For example:

```
<tns:application>
<tns:applname>Application2</tns:applname>
<tns:appltemplate>1-Year PKI SSL Browser Certificate</tns:appltemplate>
<tns:appltemplate>5-Year PKI SSL Server Certificate</tns:appltemplate>
<tns:appltemplate>5-Year PKI Intermediate CA Certificate</tns:appltemplate>
<tns:appltemplate>2-Year PKI Authenticode - Code Signing Certificate</tns:appltemplate>
<tns:appltemplate>n-Year PKI Certificate for Extensions Demonstration</tns:appltemplate>
<tns:appltemplate>1-Year SAF Browser Certificate</tns:appltemplate>
</tns:application>
```

While you access the Customers home page at the URL:

`http://hostname.com:9080/PKIServ_Web/Customers/pkimain.jsp`

you would access the Application2 home page at:

`http://hostname.com:9080/PKIServ_Web/Application2/pkimain.jsp`

To change the name of Application2, or to add another application, you need to edit the `web.xml` file in `PKIServ.ear`.

Steps for creating application domains other than Application2

Before you begin: You need to have the **jar** command in your path. If you don't, define the JAVA_HOME variable (you can find its value on the Websphere administration console) and add the `$JAVA_HOME/bin` directory to your path. For example:

```
export JAVA_HOME=/WebSphere/V6R1/AppServer/java
export PATH=$JAVA_HOME/bin:$PATH
```

Perform the following steps to create different application domains when you use JSPs to implement the Web application.

1. Follow steps 1 to 4 on page 199.

---

2. Edit the `web.xml` file. You can use the following command, or use your preferred editor:

```
oedit WEB-INF/web.xml
```

3. Copy the lines that begin with the comment:

   ```
   <!-- Start: For new application: Application2 -->
   ```

   and end with the comment:

   ```
   <!-- End: For new application: Application2 -->
   ```

4. Change all occurrences of "Application2" in the copied lines to the name of the application you want to create. Save the updated file.

5. Update the WAR file with the edited template file.

   ```
   jar -uvf PKIServ_Web.war WEB-INF/web.xml
   ```

6. Update the EAR file with the updated WAR file:

   ```
   jar -uvf PKIServ.ear PKIServ_Web.war
   ```

7. Make sure the `PKIServ.ear` file is publicly readable by issuing the **chmod** command.

   ```
   chmod 755 PKIServ.ear
   ```

When you are done, you have created another application domain. You can now deploy the updated files following the directions in "Deploying the EAR file to a Websphere application server" on page 200. When you edit the file `pkitmpl.xml` and add your new application, if its name is "NewApplication" your new application will be available at:

`http://`*`hostname`*`.com:9080/PKIServ_Web/NewApplication/pkimain.jsp`

Adding a new CA domain

When you want to operate more than one certificate authority (CA) on a single z/OS image, you must create a separate CA domain for each CA. Each CA domain uses its own daemon and operates as its own instance of PKI Services. This topic describes how to add a new CA domain. (If you want to add a new *application* domain, see "Adding an application domain" on page 226.)

When you add CA domains, you can create a PKI infrastructure that contains subsets of end user populations (application domains), each supported by its own unique PKI Services application (PKI Services daemon and URL) and optionally by its own dedicated set of PKI administrators. If you already use multiple application domains, the key advantage of adding multiple CA domains is that you can build a certificate hierarchy of CAs and optionally provide certificate services to multiple organizations.

By adding a new CA domain, your users still have a unique URL and set of certificate templates to choose from, but they also have the services of their own CA including the CA's certificate, signing key, object store and issued certificate list (ICL) , and LDAP repository. Enabling multiple CAs is a natural extension for multiple application domains. Each CA domain can represent one instance of a CA,

backed by a unique instance of the PKI Services daemon (and all its associated componentry), yet requiring no more than a single HTTP Server and a single set of CGIs.

Figure 29 contains an illustration showing two CA domains, one for employees and one for customers. In the illustration, a single shared administrator supports both CA domains. (You can decide to share a common administrator across multiple CA domains or have separate administrators who are each dedicated to only one CA domain.)

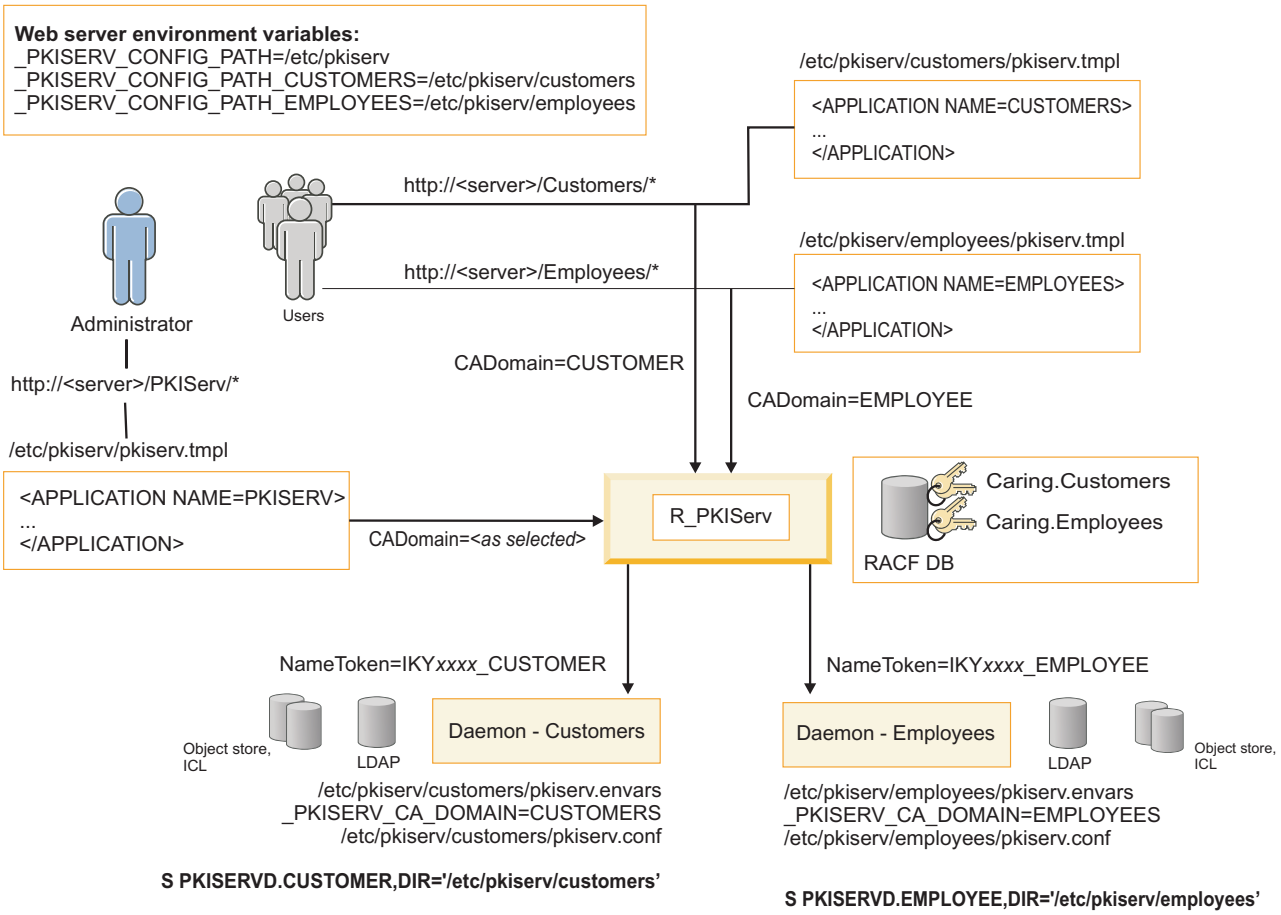


Figure 29. Illustration of two CA domains, one for employees and one for customers, administered by a single shared administrator who administers both domains. This illustration assumes that you are implementing the Web application using REXX CGI execs. If you are using JSPs, the name of the template and the tags used differ from those shown.

Task overview

This topic includes a task roadmap that you can use to add a new CA domain. The roadmap includes several subtasks, listed in Table 47 on page 233. It is intended to direct you to add a new CA domain after you have completed all required tasks in Part 2, "Configuring your system for PKI Services," on page 27. Before you begin this task, you have already implemented and tested the default setup for PKI Services and ensured that it operates properly as a single CA domain.

For each CA you add, you create a dedicated copy of the object store and issued certificate list (ICL), CA certificate, key ring, and LDAP namespace. You also create a dedicated copy of the PKI Services configuration file (`pkiserv.conf`), templates

file (pkiserv.tmpl or pkitmpl.xml), and environment variables file (pkiserv.envars), each in its own directory. You update the following CA-specific information in these files:

- pkiserv.conf - contains the CA-specific key ring name, VSAM data set names or DB2 subsystem and package name for the object store and ICL, and optionally CRLDistDirPath.
- pkiserv.envars - contains a variable _PKISERV_CA_DOMAIN to specify CA domain and the variable _PKISERV_CONFIG_PATH sets the directory for each CA domain.
- The template file that you are using:
  - pkiserv.tmpl - contains the name of the end-user application section (default is CUSTOMER) that you rename to a CA-specific name, such as <APPLICATION NAME=EMPLOYEE>. It also contains the name of the administrative application section (default is PKISERV) that you can rename to a CA-specific name, such as <APPLICATION NAME=ADMEMPLOYEES>.
  - pkitmpl.xml - defines the applications and certificate request templates that you will use for this CA domain.

If you are implementing the Web application using REXX CGI execs, you customize the HTTP configuration files and the HTTP environment variables file for the Web server to add additional application domains. For example, based on Figure 29 on page 231, you would add the following URL (to the HTTP configuration files) and the following configuration path (to the HTTP environment variables file).

New application domain (URL)	New configuration path
http://<*server-name*>/Employees/*	_PKISERV_CONFIG_PATH_EMPLOYEES=/etc/pkiserv/employees

If you are implementing the Web application using Java server pages (JSPs), you need to edit the web.xml file with the PKIServ.EAR. It helps to understand the URLs used for multiple CA domains and multiple application domains. The JSP code parses the URL to determine the CA domain and application name. The first directory after the root context of PKIServ_Web is the CA domain name if there is one and the second directory is either the application name or PKIServ (for the PKI Services administration Web pages). When the JSPs are run without a named CA domain, the first directory after the root context of PKIServ_Web is the application name or PKIServ (for the PKI Services administration pages).

Task roadmap for adding CA domains

Before you begin:

- Complete all required tasks in Part 2, "Configuring your system for PKI Services," on page 27. This task roadmap is intended to direct you to add a new CA domain *after* you have already implemented and tested the default setup for PKI Services and ensured that it operates properly as a single CA domain.
- Review Table 47 on page 233 to see the subtasks involved and the skills required for each subtask. (The team members listed are based on role definitions established in "Identifying skill requirements" on page 13.)

To create a new CA domain, complete the subtasks in Table 47 on page 233. Subtask 1 guides you through planning. Subtask 2 is a one-time setup you do when you add your first additional CA domain. Subtasks 3 - 8 are each done once for every CA domain you add. (See Part 2, "Configuring your system for PKI Services," on page 27 for additional details about these subtasks.)

Guideline: After you complete Subtasks 1 and 2 (planning and reconfiguring), perform Subtasks 3 - 8 for your *first* new CA domain and ensure that it operates properly before adding your second CA domain.

Table 47. Task roadmap for adding a new CA domain

	Subtask	Team member	Associated procedure (See...)
1.	Plan additional CA domains.	UNIX programmer	"Subtask 1: Steps for planning additional CA domains" on page 234.
2.	Reconfigure your initial CA domain to allow it to coexist with other CA domains. (This is a one-time setup.)	UNIX programmer	"Subtask 2: Steps for reconfiguring your initial CA domain to allow it to coexist with other CA domains" on page 236.
3.	Run IKYSETUP.	MVS programmer	"Subtask 3: Steps for running the IKYSETUP exec" on page 238.
4.	Configure the UNIX environment.	UNIX programmer	"Subtask 4: Steps for configuring the UNIX environment" on page 240.
5.	Update the PKI Services template file.	Web server programmer	"Subtask 5: Steps for updating the PKI Services template file" on page 241.
6.	Update the Web server configuration.	Web server programmer	"Subtask 6: Steps for updating the Web server configuration" on page 243.
7.	Set up the object store and ICL.	MVS programmer or DB2 programmer, depending on how you implement the object store and ICL	"Subtask 7: Creating the object store and ICL" on page 244.
8.	Start PKI Services.	MVS programmer	"Subtask 8: Steps for starting PKI Services" on page 245.

Recording your progress adding CA domains

As you complete each subtasks for adding a new CA domain, use Table 48 on page 234 to mark your progress. The tasks correspond to the subtasks listed in the roadmap in Table 47.

Table 48. Multiple CA domains: Worksheet #1 for recording progress adding new CA domains

	Subtask	First CA domain	Second CA domain	Third CA domain
1.	Plan additional CA domains.	☐		
2.	Reconfigure initial domain. (once only)	☐		
3.	Run IKYSETUP.	☐	☐	☐
4.	Configure the UNIX environment.	☐	☐	☐
5.	Update the PKI Services template file. (Perform this task only if you implement the Web application using REXX CGI execs.)	☐	☐	☐
6.	Update the Web server configuration.	☐	☐	☐
7.	Set up the object store and ICL.	☐	☐	☐
8.	Start PKI Services.	☐	☐	☐
	Guideline: Perform Subtasks 1 - 8 for your *first* new CA domain and ensure that it operates properly before adding your second CA domain.			

Subtask 1: Steps for planning additional CA domains

Perform the following steps to plan additional CA domains.

1. Determine how many instances of PKI Services (CA domains) you will operate in addition to the initial domain you configured when you originally customized PKI Services.

   For each CA domain, you need to pick a nickname to use as the CA domain name. The CA domain name is used to qualify the resources used by that CA domain. For example, the CA domain named `Employees` uses the following resources:

   Examples:

   - Web page URLs (in mixed case)
     - If you implement the Web application using REXX CGI execs:

       `http://webserver-domain-name/Employees/public-cgi/camain.rexx`
     - If you implement the Web application using JSPs:

       `http://webserver-domain-name/PKIServ_Web/Employee/ApplicationName/pkimain.jsp`

       Note that the CA domain name is independent of the application domain name.
   - Data set qualifiers (in upper case) - VSAM ICL data set

     `PKISRVD.EMPLOYEE.VSAM.ICL`
   - Path names (in lower case)

     `/etc/pkiserv/employees/pkiserv.conf`

   If you are implementing the object store and ICL using DB2, you need a unique DB2 package name for each CA domain. Use the CA domain name for the package name. You should also use a unique name for the DB2 plan.

   _____

2. Decide how you will administer multiple CA domains. Will you share a *common* set of administrators across all your CA domains or will you have a *dedicated* set of administrators for each CA domain?

   If you use a *dedicated* set for each CA domain, you need to pick a second nickname for each CA domain, for its administrative domain.

   _____

3. Determine your CA domain names. Unless you renamed the default domain names when you originally customized PKI Services, the initial name for the application domain is `Customers` and its administrative domain name is `PKIServ`. Your new CA domain names (nicknames) must differ from these values.

   Rules for domain names:

   - Domain names are 1 - 8 characters.
     - For REXX CGI execs, domain names can exceed 8 characters if the first 8 characters are unique from your other domain names.
     - For Java server pages (JSPs) domain names cannot exceed 8 characters.
   - The characters in the domain name are limited to the following character set: alphanumeric characters (`a - z`, `A - Z`, `0 - 9`) and the hyphen (`-`).
   - The first character must not be a number or hyphen.

   _____

4. Record information about your CA domains in Table 49 and Table 50.

   Row **1** in each table is already filled in with the defaults for an initial CA domain (`Customers`). Row **2** in each table is an example of a new CA domain managed by the same (shared) group of administrators. Row **3** in each table is an example of the same CA domain from Row **2** managed by a *dedicated* group of administrators.

   The rows in each table that are already filled in use the default values for the following variables when PKI Services was installed. (Your MVS programmer might have chosen different directories.)

Installation variable	Default directory name
install-dir	`/usr/lpp/pkiserv`
runtime-dir	`/etc/pkiserv`

   a. Fill in the values for new CA domains, administrative domains, and directories in Table 49. You can add your information in the blank lines below or you can modify or cross out the sample rows.

Table 49. Multiple CA domains: Worksheet #2 for planning your domain names. Use row 2 for shared administrators, row 3 for dedicated administrators.

	CA domain name (`runtime directory`)	Truncated CA domain name	Administrative domain name (`runtime directory`)
1.	Customers (`/etc/pkiserv`)	CUSTOMER	PKIServ (`/etc/pkiserv`)
2.	Employees (`/etc/pkiserv/employees`)	EMPLOYEE	PKIServ (`/etc/pkiserv`)
3.	Employees (`/etc/pkiserv/employees`)	EMPLOYEE	AdmEmployees (`/etc/pkiserv/employees`)
4.			
5.			

   b. Fill in your RACF user IDs, groups, and VSAM data set qualifiers or DB2 package names in Table 50 on page 236. You can add your information in the blank lines below or you can modify or cross out the sample rows.

Table 50. Multiple CA domains: Worksheet #3 for planning your RACF identifiers, z/OS UNIX identifiers, and VSAM data set names or DB2 package names. Use row 2 for shared administrators, row 3 for dedicated administrators. Row 4 is the same as row 3 but specifies a DB2 package name instead of VSAM data set qualifiers.

	Daemon user ID (UID)	Surrogate user ID (UID)	PKI administration group name (GID)	VSAM data set qualifiers or DB2 package name
1.	PKISRVD (554)	PKISERV (555)	PKIGRP (655)	PKISRVD.VSAM
2.	PKISRVD (554)	PKISERV (555)	PKIGRP (655)	PKISRVD.EMPLOYEE.VSAM
3.	PKIDEMP (556)	PKISEMP (557)	PKIGEMP (657)	PKISRVD.EMPLOYEE.VSAM
4.	PKIDEMP (556)	PKISEMP (557)	PKIGEMP (657)	MasterCA
5.				

Subtask 2: Steps for reconfiguring your initial CA domain to allow it to coexist with other CA domains

Perform the following steps to reconfigure your initial CA domain to allow it to coexist with other CA domains. (This is a one-time setup that will suffice no matter how many CA domains you add.)

1. If PKI Services is running, stop it by issuing the following MVS console command:

   ```
   P PKISERVD
   ```

   _____

2. Update the PKI Services environment variables in the `pkiserv.envars` file as follows.

   a. (Optional) If your initial CA domain does not use its own `pkiserv.envars` file, copy the default `pkiserv.envars` file from the PKI Services install directory by issuing the following command from the UNIX command line:

      ```
      cp -p /usr/lpp/pkiserv/samples/pkiserv.envars /etc/pkiserv
      ```

   b. Edit the new copy of `pkiserv.envars` file by entering the following command:

      ```
      oedit /etc/pkiserv/pkiserv.envars
      ```

   c. Add a PKI Services environment variable identifying your initial CA domain name (see Table 49 on page 235) in uppercase characters.

      Example:

      ```
      _PKISERV_CA_DOMAIN=CUSTOMERS
      ```

   _____

3. Update the HTTP server's environment variables and configuration directives as follows.

   a. Update the HTTP server's environment variables.

      1) Edit the `httpd.envvars` file by entering the following command:

         ```
         oedit /etc/httpd.envvars
         ```

      2) Add an environment variable identifying the runtime directory of your initial CA domain. (Check Table 49 on page 235.)

         Example:

         ```
         _PKISERV_CONFIG_PATH_CUSTOMERS=/etc/pkiserv
         ```

3) (Optional) If you intend to have a dedicated set of administrators for each CA domain, add an environment variable that specifies the runtime directory for the administrative domain. (Check Table 49 on page 235.)

   Example:

   ```
   _PKISERV_CONFIG_PATH_PKISERV=/etc/pkiserv
   ```

b. Update the HTTP configuration directives.

1) Edit the `httpd.conf` file by entering the following command:

   ```
   oedit /etc/httpd.conf
   ```

2) (Optional) If your HTTP configuration file includes an `InheritEnv` directive, add the following directive for each new CA domain you add. Replace the **`CUSTOMERS`** value with the `ca_domain` value you specified in Table 17 on page 43.

   This directive specifies that you want your CGI programs to inherit the PATH environment variable so that the PKI Services Web pages of each CA domain can retrieve any certificate through the Web page of any CA domain.

   Example:

   ```
   InheritEnv _PKISERV_CONFIG_PATH_CUSTOMERS=/etc/pkiserv
   ```

---

4. Update the RACF access controls for the `R_PKIServ` SAF callable service as follows. (Any change to environment variables in Step 3 requires a corresponding change to RACF access control.)

a. Determine the PKI Services surrogate user ID (default is PKISERV) and the PKI Services administrators group (default is PKIGRP). To do this, refer to the log file created when the IKYSETUP REXX exec was originally run for your initial CA domain.

b. Execute the following RACF commands from the TSO command line. Replace the highlighted values with your own, if different:

   Examples:

   ```
   RDELETE FACILITY IRR.RPKISERV.**
   RDEFINE FACILITY IRR.RPKISERV.*.CUSTOMER
   PERMIT IRR.RPKISERV.*.CUSTOMER CLASS(FACILITY) ID(PKISERV) ACCESS(CONTROL)
   RDELETE FACILITY IRR.RPKISERV.PKIADMIN
   RDEFINE FACILITY IRR.RPKISERV.PKIADMIN.CUSTOMER
   PERMIT IRR.RPKISERV.PKIADMIN.CUSTOMER CLASS(FACILITY) ID(PKIGRP)
     ACCESS(UPDATE)
   PERMIT IRR.RPKISERV.PKIADMIN.CUSTOMER CLASS(FACILITY) ID(PKISERV)
     ACCESS(NONE)
   SETROPTS RACLIST(FACILITY) REFRESH
   ```

   Restriction: If the name of your initial CA domain is longer than 8 characters, you must truncate it to exactly 8 characters when you define the resource name in the FACILITY class profiles. (In this example, the name `CUSTOMERS` was truncated to `CUSTOMER` in the second RDEFINE FACILITY command.)

---

5. (Optional) You have reconfigured your initial CA domain to allow it to coexist with other CA domains. If you want, you can test the reconfiguration now. To test it, do the following:

a. Restart PKI Services using the following MVS console command. Replace the highlighted values with your own, if different.

   Guideline: To simplify your environment, start this instance of PKI Services using a JOBNAME that matches the truncated name of the CA domain. (See your truncated value in Table 49 on page 235.) If you use the truncated values as job names, it will be easier to distinguish multiple jobs that run PKI Services after you add other CA domains.

   Example:

   ```
   S PKISERVD,JOBNAME=CUSTOMER,DIR='/etc/pkiserv/'
   ```

b. Restart the HTTP servers to enable your environment variable changes.

   ```
   F IMWEBSRV,APPL=-restart
   ```

c. Test that your PKI Services application is functioning properly. Go to your Web pages by entering the following URL from your browser:

   ```
   http://webserver-fully-qualified-domain-name/PKIServ/public-cgi/camain.rexx
   ```

   The *webserver-fully-qualified-domain-name* is the common name (CN) portion of the Web server's distinguished name; see Table 10 on page 32.

   You should be able to go through your Web pages to request, retrieve, and revoke a certificate of type "PKI browser certificate for authenticating to z/OS". Ensure you can do this before adding an additional CA domain.

When you are done: You have successfully reconfigured your initial CA domain to allow it to coexist with other CA domains. You can now perform each of the remaining subtasks once for each new CA domain.

Continue to the next subtask. **Guideline:** Complete Subtasks 3 - 8 for your *first* new CA domain and ensure that it operates properly before adding another CA domain.

Subtask 3: Steps for running the IKYSETUP exec

Before you begin: This procedure requires you to be familiar with the information in Chapter 4, "Running IKYSETUP to perform RACF administration," on page 29. You will find additional details about the following steps there.

Perform the following steps to customize a unique execution of the IKYSETUP REXX exec for this new CA domain.

1. Locate the IKYSETUP exec you originally customized for your initial CA domain and copy it to a data set member that you can edit.

---

2. Edit the new copy of IKYSETUP and set the `ca_domain` variable to the name of this new CA domain. Type the domain name preserving the case of each character as you wish it to appear in Web page URLs.

---

3. If you intend to have a dedicated set of administrators for each CA domain, customize the following variables with your values for this CA domain.

Variable name	Use your value from ...
daemon_uid	Table 10 on page 32
pki_gid	
pkigroup_mem	
surrog_uid	
daemon	Table 17 on page 43
surrog	
pkigroup	Table 49 on page 235 (Use the truncated name of the administrative domain.)

4. (Optional) If you are creating multiple CAs as part of a certificate hierarchy where a previous CA domain is to be superior (as issuer or signer) of this CA domain, set signing_ca_label to match the label of the certificate in RACF that will issue the certificate for this CA domain.

   Otherwise, skip to Step 5 and leave signing_ca_label="" (the default).

5. Update any other values, such as ca_dn and ra_dn, that you choose to differ from your initial settings or the defaults.

   You need *not* change any values in this step unless you choose to set these values to something particular for your installation. (This is because when you specify a ca_domain value, the IKYSETUP exec automatically qualifies any value that PKI Services requires to be unique by adding the CA domain name.)

6. Execute IKYSETUP by entering the following TSO command:

   ```
   EX 'data-set-name(new-member-name)' 'RUN(NO)'
   ```

7. Review the log data set to ensure that the commands created by IKYSETUP match your expectations. (For more information about these commands, see "Actions IKYSETUP performs by issuing RACF commands" on page 511.) Edit again as needed and rerun.

8. When you are satisfied with the commands and information in the log data set, rerun the IKYSETUP exec by entering the following TSO command:

   ```
   EX 'data-set-name(new-member-name)' 'RUN(YES)'
   ```

9. Check your IKYSETUP log and record the name of the SAF key ring (your ca_ring value from Table 17 on page 43).

   Table 51. IKYSETUP log information you need to record

Name of the SAF key ring:	

When you are done: You have customized and run the IKYSETUP exec for this CA domain. Record your progress in Table 48 on page 234.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

Subtask 4: Steps for configuring the UNIX environment

Before you begin: This procedure requires you to be familiar with the information in Chapter 5, "Configuring the UNIX runtime environment," on page 53. You will find additional details about the following steps there.

Perform the following steps to configure the UNIX environment for this new CA domain.

1. Set up a `var` directory for this CA domain. Perform the steps in "Steps for setting up the var directory" on page 84.
2. Locate the `pkiserv.conf`, `pkiserv.envars`, and `pkiserv.tmpl` files you originally used to create your initial CA domain. Copy them into the appropriate runtime directory for your new CA domain. (Check Table 49 on page 235.) For a new CA domain called `Employees`, run the following commands from the UNIX command line. (You might have to make the directory first.)

   Examples:

   ```
   mkdir /etc/pkiserv/employees
   chown pkisrvd /etc/pkiserv/employees
   cp -p /etc/pkiserv/* /etc/pkiserv/employees
   ```

   _____
3. Edit the new `pkiserv.conf` file by entering the following command:

   Example:

   ```
   oedit /etc/pkiserv/employees/pkiserv.conf
   ```

   _____
4. Change the following sections of `pkiserv.conf` as described for this CA domain. (Find detailed information for each variable in Table 19 on page 60.)

   ObjectStore If you are implementing the object store and ICL using VSAM, qualify each VSAM data set name with the CA domain name. **Example:** `ObjectDSN='pkisrvd.employee.vsam.ost'`

   If you are implementing the object store and ICL using DB2, set the DB2 package name to the CA domain name. **Example:** `DBPackage=employee`

   (See "Subtask 7: Creating the object store and ICL" on page 244.)

   CertPolicy If `CRLDistDirPath` is not null, modify it to reference the correct subdirectory. (You might have to create this directory.) **Example:** `CRLDistDirPath=/var/pkiserv/employees`. See "Determining CRLDistDirPath" on page 219 for more information.

   General Update each pathname to the correct subdirectory. **Example:** `ReadyMessageForm=/etc/pkiserv/employees/readymsg.form`

   SAF Update the key ring name to match the `ca_ring` value you recorded in Table 51 on page 239. **Example:** `PKISRVD/Caring.Employees`

   LDAP Do not update the **LDAP** section unless you need to change

the LDAP directory. If you need to change it, see "Steps for tailoring the LDAP section of the configuration file" on page 98.

Make sure the LDAP directory is configured with a suffix for this CA domain. (See the explanation for the `Suffix` variable in Table 20 on page 87.)

---

5. (Optional) Change other values in any section of `pkiserv.conf` as desired for this CA domain.

---

6. Edit the new `pkiserv.envars` file by entering the following command:

   Example:

   ```
   oedit /etc/pkiserv/employees/pkiserv.envars
   ```

---

7. Define the `_PKISERV_CA_DOMAIN` environment variable for this CA domain name. (For details, see "The pkiserv.envars environment variables file" on page 510.)

   Example:

   ```
   _PKISERV_CA_DOMAIN=EMPLOYEE
   ```

When you are done: You have updated the `pkiserv.conf` and `pkiserv.envars` files for this CA domain. Record your progress in Table 48 on page 234.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

Subtask 5: Steps for updating the PKI Services template file

Perform this task only if you implement the Web application using REXX CGI execs. If you implement the Web application using Java server pages (JSPs), each CA domain has a separate template file. Each template file describes the application domains for the CA domain. Each application domain is essentially a subset of the users and certificate request templates for the CA domain.

Before you begin: This procedure requires you to be familiar with the information in Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs," on page 123. You will find additional details about the following steps there.

Perform the following steps to customize the PKI Services template file (`pkiserv.tmpl`) for this new CA domain.

1. Edit the new `pkiserv.tmpl` file (you copied it in Step 2 of "Subtask 4: Steps for configuring the UNIX environment" on page 240) by entering the following command from the UNIX command line:

   Example:

   ```
   oedit /etc/pkiserv/employees/pkiserv.tmpl
   ```

---

2. Locate all occurrences of the CA domain named `Customers` (in mixed case, upper case, or lower case) and change them to the name of this new CA domain, being careful to preserve the case.

Example: If the new CA domain is `Employees` (in mixed case), change the default values to the name of your new CA domain.

Default values for the **Customers** CA domain	New values for the **Employees** CA domain
`ACTION="/Customers/..."` (in mixed case)	`ACTION="/Employees/..."` (also in mixed case)
`<APPLICATION NAME=CUSTOMERS>` (in upper case)	`<APPLICATION NAME=EMPLOYEES>` (also in upper case)

3. If you intend to have the *same* set of administrators for all your CA domains, *skip* this step and proceed to Step 4.

   If you intend to have a *dedicated* set of administrators for each CA domain, change the name of the PKISERV application section to the corresponding name of the administrative domain in Table 49 on page 235. This value must be specified in uppercase characters only.

   Example: If the new administrative domain is named `AdmEmployees`, change the default value (`PKISERV`) to the name of your new administrative CA domain.

Default value for the **PKISERV** administrative CA domain	New value for the new **AdmEmployees** administrative CA domain
`<APPLICATION NAME=PKISERV>` (in upper case)	`<APPLICATION NAME=ADMEMPLOYEES>` (also in upper case)

4. If you intend to have the same set of administrators for all your CA domains, edit the main templates file (`/etc/pkiserv/pkiserv.tmpl`) as follows:

   Do not update the PKISERV application section in this domain-specific `pkiserv.tmpl` file; it will *not* be used. The PKISERV application section in the templates file for your initial CA domain will be used instead.

   a. Replicate the following lines in the APPLICATION section of the PKISERV application:

      ```
      <h3>Go the Customers' home page </h3>
      <FORM name=admform METHOD=GET
         ACTION="/Customers/public-cgi/camain.rexx">
      <p>
      <INPUT TYPE="submit" VALUE="Customers' Home Page">
      </FORM>
      ```

   b. Change all occurrences of the string `Customers` in the replicated lines to the name of this CA domain, being careful to preserve case.

      Example: Change `ACTION="/Customers/public-cgi/camain.rexx">` to `ACTION="/Employees/public-cgi/camain.rexx">`.

   c. Uncomment the `%%SelectCADomain%%` directive in the ADMINSCOPE subsection of the APPLICATION section for the PKISERV application by removing the leading `#` character. (The `%%SelectCADomain%%` directive enables multiple CA administration.)

   d. Update the `SelectCADomain` insert to include an OPTION entry for this CA domain. If this CA domain will be used more often than any other, mark the entry `SELECTED` and remove `SELECTED` from any other entry.

Example:

```
<INSERT NAME=SelectCADomain>
<p> <LABEL for="selectcadomfield">Select the CA domain to work with </LABEL>
<SELECT NAME="domain" id="selectcadomfield">
# rename and replicate the following line for every CA domain and
# determine which one should be SELECTED by default, if any
<OPTION VALUE="Employees" SELECTED>Employees
<OPTION VALUE="Customers">Customers
</SELECT>
```

When you are done: You have customized the PKI Services template file (`pkiserv.tmpl`) for this CA domain. Record your progress in Table 48 on page 234.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

Subtask 6: Steps for updating the Web server configuration

If you are implementing the Web application using REXX CGI execs, perform the steps in "Updating the Web server configuration if you use REXX CGI execs"

If you are implementing the Web application using Java server pages (JSPs), perform the steps in "Updating the Web server configuration if you use Java server pages (JSPs)" on page 244

Updating the Web server configuration if you use REXX CGI execs

Before you begin: This procedure requires you to be familiar with the information in the following topics, where you find additional details:

- "Steps for adding application domains to the Web server configuration files" on page 228
- "Steps for updating the z/OS HTTP Server configuration files" on page 91.

Perform the following steps to customize the z/OS HTTP Server configuration files for this new CA domain.

1. Add this new CA domain (check Table 49 on page 235 for domain name and directory) following the instructions in "Steps for adding application domains to the Web server configuration files" on page 228. If you have a `CRLDistDirPath` configured in your `pkiserv.conf` file for HTTP protocol URI format CRL distribution points, add a Pass statement in the appropriate location in the HTTP configuration file to map the virtual pathname in the URI to the `CRLDistDirPath` value.

   Example:

   ```
   Pass /Employees/crls/* /var/pkiserv/employees/*
   ```

   ---

2. (Optional) If you intend to have a dedicated set of administrators for each CA domain, repeat Step 1 for the administrative domain. (Check Table 49 on page 235 for domain name and directory.) Otherwise, skip to Step 3.

   ---

3. Update the HTTP Server environment variables. Edit the `httpd.envvars` file by entering the following command from the UNIX command line:

   ```
   oedit /etc/httpd.envvars
   ```

   ---

4. Add the environment variable identifying the runtime directory of this CA domain. (Check Table 49 on page 235.)

   Example:

   ```
   _PKISERV_CONFIG_PATH_EMPLOYEES=/etc/pkiserv/employees
   ```

5. (Optional) If you intend to have a dedicated set of administrators for each CA domain, add the environment variable identifying the `pkiserv.tmpl` directory of this administrative CA domain.

   Example:

   ```
   _PKISERV_CONFIG_PATH_ADMEMPLOYEES=/etc/pkiserv/employees
   ```

When you are done: You have customized the z/OS HTTP Server configuration files for this CA domain. Record your progress in Table 48 on page 234.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

Updating the Web server configuration if you use Java server pages (JSPs)

To add a new CA domain when you implement the Web application using Java server pages (JSPs), you need to edit the `web.xml` file to define additional environment variables and URL mappings to the Websphere application, `PKIServ_EAR`. The shipped `web.xml` file contains the definitions to use either an unnamed domain or a domain named "Master". Use the definition of the CA domain Master as a model for defining additional CA domains. This section in the `web.xml` file is delimited by the two comments:

```
<!-- Start: For new domain: Master -->
⋮
<!-- End: For named domain Master with application Customers: Master/Customers -->
```

Subtask 7: Creating the object store and ICL

Before you begin: This procedure requires you to be familiar with the information in Chapter 9, "Creating the object store and ICL," on page 103. You will find additional details about the following steps there.

You can implement the object store and ICL using either VSAM data sets or DB2 tables.

If you are implementing the object store and ICL using VSAM data sets, perform the following steps to allocate the needed VSAM files for this new CA domain.

1. Locate the IKYCVSAM JCL job you originally customized for your initial CA domain and copy it to a data set that you can edit.

2. Change all occurrences of `PKISRVD.VSAM` to include the corresponding VSAM data set qualifier from Table 50 on page 236. For example, if this new CA domain is `Employees`, then the VSAM data sets might be qualified as follows:

   Example:

   ```
   PKISRVD.EMPLOYEE.VSAM.data-set-suffix
   ```

3. Submit the JCL job when your changes are complete.

When you are done: You have allocated the needed VSAM files for this CA domain. Record your progress in Table 48 on page 234.

If you are implementing the object store and ICL using DB2 tables, perform the following steps to create the needed DB2 tables, package, and plan.

1. Locate the IKYCDB2 sample you originally customized for your initial CA domain and copy it to a data set that you can edit.

   ---

2. Change all occurrences of `MASTERCA` to the package name from Table 50 on page 236. For example, if this new CA domain is `Employees`, and you are using the domain name for the package name, the table created for the object store would be named `EMPLOYEE.OST`.

   ---

3. When your changes are complete, run the modified sample using SPUFI.

   ---

4. Locate the IKYSBIND sample job you originally customized for your initial CA domain and copy it to a data set that you can edit.

   ---

5. Change all occurrences of `MASTERCA` to the package name from Table 50 on page 236.

   ---

6. Submit the JCL job when your changes are complete.

When you are done: You have create the needed DB2 tables, package, and plan for this CA domain. Record your progress in Table 48 on page 234.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

Subtask 8: Steps for starting PKI Services

Before you begin: This procedure requires you to be familiar with the information in Chapter 10, "Starting and stopping PKI Services," on page 117. You will find additional details about the following steps there.

Perform the following steps to start a separate instance of PKI Services for this new CA domain.

1. Start the PKI Services daemon for this CA domain by entering the MVS console START command qualified with the appropriate runtime directory. (Check Table 49 on page 235.)

   Example:

   ```
   S PKISERVD,JOBNAME=EMPLOYEE,DIR='/etc/pkiserv/employees'
   ```

   Guideline: To simplify your environment, give this instance of PKI Services a JOBNAME that matches or relates to this CA domain name. When you add additional CA domains, it will be easier to distinguish multiple jobs running PKI Services.

   ---

2. Restart the HTTP servers to enable the environment variables you changed for this CA domain. Optionally, you can wait to do this until after you have started all the new domain-specific daemons.

   ```
   F IMWEBSRV,APPL=-restart
   ```

   _____

3. Test that your new domain-specific PKI Services daemon is functioning properly. Go to your Web pages by entering the following URL from your browser:

   ```
   http://<webserver-fully-qualified-domain-name>/<new-admin-domain-name>/public-cgi/camain.rexx
   ```

   The *webserver-fully-qualified-domain-name* is the common name (CN) portion of the Web server's distinguished name; see Table 10 on page 32.

   You should be able to go through your Web pages to request, retrieve, and revoke an applicable certificate for this CA domain, possibly "PKI browser certificate for authenticating to z/OS". Ensure you can do this before adding new CA domains.

   _____

When you are done: You have customized the z/OS HTTP Server configuration files for this CA domain. Record your progress in Table 48 on page 234.

Once your new CA domain works properly, proceed to add another CA domain, if needed. **Guideline:** Perform Subtasks 3 - 8 for each new CA domain and ensure that the new CA domain operates properly before proceeding to add another.

Enabling Simple Certificate Enrollment Protocol (SCEP)

The Simple Certificate Enrollment Protocol (SCEP) allows you to securely issue certificates to large numbers of network devices using an automatic enrollment technique. The network devices, usually IPSEC devices such as Cisco routers, must be SCEP-enabled and preregistered (to your CA domain) before they can successfully request certificates from you. To request a certificate, the preregistered SCEP client sends a message (the certificate request) to your CA using the HTTP protocol. (The message is a PKCS #10 request enveloped in a signed PKCS #7 structure.)

You can configure PKI Services to respond automatically to some (or all) SCEP certificate requests, or to submit some (or all) SCEP certificate requests to the PKI administrator for approval or rejection. When you enable automatic enrollment, certificate requests can be automatically approved and synchronously fulfilled, based on the requestor's knowledge of a predetermined secret, the challenge passphrase.

Overview of SCEP preregistration

To request certificates using SCEP, a SCEP requestor must be *preregistered* to PKI Services, your CA. You can preregister SCEP clients in batches using the `pkiprereg` utility (see "Using the pkiprereg utility" on page 363) or the PKI administrators can preregister individual SCEP clients (one client at a time) using the end-user Web pages.

When PKI administrators preregister a SCEP client, they do so by using the end-user Web page for requesting a certificate and selecting the SCEP (preregistration) certificate template called `5-Year SCEP Certificate – Preregistration`. (See "Steps for preregistering an SCEP client" on page 331.) The PKI administrator fills out the request form by specifying the device or client name

of the SCEP client, a passphrase for client authentication, and additional (optional) subject name and alternate name information. You can customize the **<CONSTANT>** section of the SCEP (preregistration) certificate template to supply the additional optional information.

When a PKI administrator submits the form for a SCEP (preregistration) certificate request, PKI Services creates a preregistration record, not an actual certificate request, in the VSAM ObjectStore data set (request database). The client name is translated to lowercase characters, truncated to 32 characters if longer, and saved as the `Requestor` to support searching of the ObjectStore. (Each preregistration record must have a client name that is unique in the first 32 characters, regardless of upper or lower case.)

The preregistration record contains the template nickname, passphrase, and additional (optional) subject name and alternate name values. Any other information (unrelated to the subject name or alternate name) specified on the request form is ignored.

When you customize the **<CONSTANT>** section of the SCEP template to supply additional (optional) values for the following variables, those values are not saved in the preregistration record. However, those values are processed when the preregistered client subsequently requests a certificate.

- AuthInfoAcc
- CertPolicies
- Critical
- ExtKeyUsage (not typically used in a SCEP request)
- KeyUsage (not typically used in a SCEP request)
- NotAfter
- NotBefore

Overview of certificate request processing for preregistered SCEP clients

Following preregistration, when the preregistered SCEP client requests a certificate (sends a SCEP request), PKI Services searches for a preregistration record matching the client name. If found, PKI Services compares the values in the request to the challenge password and any subject name or alternate name information specified by the PKI administrator or supplied in the **<CONSTANT>** template section. (If not found, the SCEP request is automatically rejected.)

Based on the comparison of values in the request with those in the preregistration record, PKI Services considers the request to be in one of the following states:

Authenticated
When the challenge password matches and all other preregistered values are included in the request

Semiauthenticated
When the challenge password matches but some other preregistered values are missing from the request

Unauthenticated
When the challenge password does not match or is missing.

Depending on how you customize the variables in the SCEP (preregistration) certificate template, a certificate request from an `Authenticated` SCEP client is either automatically approved and fulfilled synchronously or it is queued for

administrator approval. Likewise, a certificate request from an `Unauthenticated` or `Semiauthenticated` SCEP client is either queued for administrator approval or it is automatically rejected.

Variables used in the <PREREGISTER> section

These are the valid variables you can customize in the **<PREREGISTER>** section of the `5-Year SCEP Certificate – Preregistration` template. Some variables *must* be present in your **<PREREGISTER>** section and they are labeled as *required* in the following list.

AuthenticatedClient *(required)*
Specifies which action PKI Services takes when an authenticated SCEP client submits a certificate request for the first time. Valid values are:

AutoApprove *(default)*
Automatically approves certificate requests from authenticated first-time SCEP clients and automatically creates their certificates.

AdminApprove
Submits certificate requests from authenticated first-time SCEP clients to your PKI administrator for verification and approval.

SemiauthenticatedClient *(required)*
Specifies which action PKI Services takes when a semiauthenticated SCEP client submits a certificate request for the first time. Valid values are:

AdminApprove *(default)*
Submits certificate requests from semiauthenticated first-time SCEP clients to your PKI administrator for verification and approval.

Reject Automatically rejects certificate requests from semiauthenticated first-time SCEP clients.

UnauthenticatedClient *(required)*
Specifies which action PKI Services takes when an unauthenticated SCEP client submits a certificate request for the first time. Valid values are:

AdminApprove
Submits certificate requests from unauthenticated first-time SCEP clients to your PKI administrator for verification and approval.

Reject *(default)*
Automatically rejects certificate requests from unauthenticated first-time SCEP clients.

SubsequentRequest *(optional)*
Specifies which action PKI Services takes when a previously approved SCEP client submits an additional certificate request. If not set, PKI Services uses the **AuthenticatedClient** value. Valid values are:

AutoApprove *(default)*
Automatically approves certificate requests from previously approved SCEP clients and automatically creates their certificates.

AdminApprove
Submits certificate requests from previously approved SCEP clients to your PKI administrator for verification and approval.

Reject Automatically rejects SCEP requests from previously approved clients.

RenewalRequest *(optional)*
Specifies which action PKI Services takes when a previously approved

SCEP client submits a certificate renewal request. If not set, PKI Services uses the **AuthenticatedClient** value. Valid values are:

AutoApprove *(default)*
Automatically approves certificate renewal requests from previously approved SCEP clients and automatically creates their certificates.

AdminApprove
Submits certificate renewal requests from previously approved SCEP clients to your PKI administrator for verification and approval.

Reject Automatically rejects certificate renewal requests from previously approved SCEP clients.

Checking certificate fingerprints

There are two instances when the PKI administrator checks certificate *fingerprints* (the SHA1, MD5, SHA256, and SHA512 hashes) in support of certificate request processing for SCEP clients.

- Preregistered SCEP clients who request certificates from this CA domain must download the correct PKI Services CA certificate to their workstations before they issue their certificate requests. After the download, the client can use the SCEP client software to display the fingerprints of the downloaded CA certificate and then confirm with the PKI administrator of the CA domain that it is the correct CA certificate.

  To match CA certificate fingerprints with a SCEP client, the PKI administrator can display the fingerprints of the CA certificate for this domain by issuing the following MODIFY (or **F**) console command:

  ```
  F PKISERVD,DISPLAY
  ```

  The result of this command is information message IKYP025I. **Sample output:**

```
10.37.39 STC00146: IKYP025I PKI SERVICES SETTINGS:
  CA DOMAIN NAME: Customers
  SUBCOMPONENT              MESSAGE LEVEL
     LDAP                   ERROR MESSAGES AND HIGHER
     SAF                    WARNING MESSAGES AND HIGHER
     DB                     INFORMATIONAL MESSAGES AND HIGHER
     CORE                   WARNING MESSAGES AND HIGHER
     PKID                   VERBOSE DIAGNOSTIC MESSAGES AND HIGHER
     POLICY                 WARNING MESSAGES AND HIGHER
     TPOLICY                WARNING MESSAGES AND HIGHER
  MESSAGE LOGGING SETTING: STDOUT_LOGGING
  CONFIGURATION FILE IN USE:
/etc/pkiserv/pkiserv.conf
  TEMPLATE FILE IN USE:
/etc/pkiserv/pkiserv.tmpl
  CA CERTIFICATE FINGERPRINTS:
   SHA1:   BB:B5:AF:38:BA:3B:33:61:46:F5:FE:AD:20:33:10:98:C2:D7:9A:BC
   MD5:    C6:E6:B2:F3:39:F0:7C:B5:A6:B6:F0:36:5F:2F:7D:C8
   SHA256: FC:F6:DE:AF:CF:48:15:90:0E:91:9B:8F:5C:93:9B:FF:
           1D:2D:FC:B1:10:33:2C:CB:B5:02:F4:8E:5E:41:FA:F8
   SHA512: 14:DD:45:4C:78:66:47:0D:7B:BB:BE:56:33:F0:18:52:
           F4:AD:0C:96:B9:78:5B:40:FF:AE:D5:EB:62:87:A6:22:
           48:45:37:D6:4B:3A:DD:5C:F0:7D:6F:A5:D8:6F:6E:36:
           E5:8C:77:D2:B5:BC:3E:14:E2:34:F8:A1:11:31:2B:E3
```

- When the PKI administrator receives a certificate request from a preregistered SCEP client, the PKI administrator can confirm the integrity of the certificate request by viewing its fingerprints on the "Single Request" Web page. (See Figure 62 on page 341 for a sample.)

  To ensure the integrity of the certificate request, the PKI administrator can contact the SCEP requestor to match the fingerprints in the received certificate request with the fingerprints in the original certificate request. (The certificate requestor can use the SCEP client software to view the fingerprints saved for the original request.)

Steps for enabling Simple Certificate Enrollment Protocol (SCEP)

Before you begin: The commands in the steps that follow include several variables that are described in Table 52. Determine the values for these variables and record the information in the blank boxes:

Table 52. Information you need to enable Simple Certificate Enrollment Protocol (SCEP)

Information needed	Where to find this information	Record your value here
ca_label - The label of your CA certificate in RACF.	See Table 10 on page 32.	
ra_label - The label of your RA certificate in RACF.	See Table 10 on page 32.	
ca_ring - The PKI Services SAF key ring.	See Table 17 on page 43.	
ca_expires - The date the PKI Services CA certificate expires.	See Table 17 on page 43.	
daemon - The user ID for the PKI daemon.	See Table 17 on page 43.	
ra_backup_dsn - The name of the encrypted data set containing the backup copy of your new RA certificate and private key.	See Table 17 on page 43.	
ra_dn - The RA's distinguished name.	See Table 17 on page 43.	

Perform the following steps to enable PKI Services to process Simple Certificate Enrollment Protocol (SCEP) requests:

1. (Optional) Create your PKI Services RA certificate by following these steps, if you haven't done so already. (This is optionally done by IKYSETUP.) If you already created an RA certificate, skip to Step 2.

   a. To create an RA certificate, execute the following RACF command from the TSO command line:

   ```
   RACDCERT ID(daemon) GENCERT SUBJECTSDN(ra_dn) KEYUSAGE(HANDSHAKE)
     SIGNWITH(CERTAUTH LABEL('ca_label') NOTAFTER(DATE(ca_expires))
     WITHLABEL('ra_label'))
   ```

   b. Backup the new PKI Services RA certificate and private key to a password-encrypted data set (*ra_backup_dsn*). **Important:** Remember to record and store your encryption password in case you ever need to recover the certificate or private key.

   ```
   RACDCERT ID(daemon) EXPORT(LABEL('ra_label')) DSN(ra_backup_dsn)
     FORMAT(PKCS12DER) PASSWORD('encryption-pw')
   ```

c. Add the new RA certificate to the PKI Services key ring.

```
RACDCERT ID(daemon) CONNECT(LABEL('ra_label') RING(ca_ring))
```

---

2. Edit the PKI Services configuration file (`/etc/pkiserv.conf`) and set the `RALabel` directive in the **SAF** section to specify the label (*ra_label*) of your PKI Services RA certificate. (The default in IKYSETUP is `Local PKI RA`. For details, see "(Optional) Steps for updating the configuration file" on page 60.)

   Example:

```
[SAF]
KeyRing=PKISRVD/CAring
# The label of the PKI Services RA certificate
RALabel=Local PKI RA
```

---

3. Edit the PKI Services configuration file (`/etc/pkiserv.conf`) to change the `EnableSCEP` directive in the **CertPolicy** section setting from **F**(False) to **T**(True).

```
[CertPolicy]
# Enable the Simple Certificate Enrollment Protocol, (T)rue or (F)alse
EnableSCEP=T
```

---

4. Edit the PKI Services template file (`/etc/pkiserv.tmpl` or `pkitmpl.xml`) and customize the **<PREREGISTER>** section of the `5-Year SCEP Certificate – Preregistration` template as desired or create a new preregistration template. (Refer to the list in "Variables used in the <PREREGISTER> section" on page 248 for valid variables and values.

   Example: (defaults)

```
AuthenticatedClient=AutoApprove
SemiauthenticatedClient=AdminApprove
UnauthenticatedClient=Reject
SubsequentRequest=AutoApprove
RenewalRequest=AutoApprove
```

---

5. Edit the **<CONTENT>** section of your preregistration template to allow the PKI administrator to specify subject distinguished name and alternate name fields that the SCEP client must provide to authenticate. Specify only subject distinguished name and alternate name fields here. All other fields are ignored. (For about customizing the end-user Web pages, see Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs," on page 123.)

   Example: (defaults)

```
%%SerialNumber (Optional)%%
%%UnstructAddr (Optional)%%
```

---

6. Edit the **<CONSTANT>** section of your preregistration template to supply any other desired value, such as MAIL or ORG, that must be included for every SCEP preregistration request. Any subject distinguished name and alternate name fields you specify here must match the information (in the subsequent certificate request) sent by the SCEP client to authenticate the certificate request.

   Example:

```
%%Org=The Firm%%
```

---

7. Edit the HTTP Server environment variables file, `httpd.envvars`, and update the LIBPATH variable to include `/usr/lpp/pkiserv/lib`.

   _____

8. Stop and restart PKI Services.

   _____

When you are done: You have enabled your CA domain to accept SCEP preregistration requests and process certificate requests from preregistered SCEP clients.

Customizing e-mail notifications sent to users

You can optionally notify a user by sending an e-mail message when:

- A certificate request is rejected
- A certificate is ready for retrieval
- A certificate is about to expire (unless it has already been renewed or revoked).
- A certificate has been automatically renewed.
- Requests are pending for the user's approval, when the user is the PKI administrator.

In addition, when a user requests that PKI Services recover one or more certificates for which PKI Services created the keys, PKI Services sends that user an e-mail message listing the possible certificates to be recovered.

On the days specified by the `MaintRunDays` parameter in the `pkiserv.conf` file, the PKI Services daily maintenance task checks the issued certificate list (ICL) for expiring certificates. (The ExpireWarningTime parameter (see the **CertPolicy** section in Table 19 on page 60) determines at what point before the certificate expires that it is considered to be an expiring certificate.) When PKI Services finds an expiring certificate, it does one of the following:

- If automatic renewal of certificates is in effect, PKI Services renews the certificate and sends it to the client.
- If automatic renewal of certificates is not in effect, PKI Services sends an expiration warning message to the client (unless the certificate has already been revoked). Regardless of whether sending the expiration warning message is successful, PKI Services makes only one attempt to send a notification message. If the e-mail address is incorrect or the user renews the certificate and retrieves it before the expiration message is sent, no expiration messages is sent.

You can set the AdminNotifyNew*n* keyword in the CertPolicy section of the configuration file to specify one or more e-mail addresses of PKI administrators to be notified immediately whenever there is a request pending for approval. The notification is sent only once, when the request is created. You can also set the AdminNotifyReminder*n* keyword to specify one or more e-mail addresses of PKI administrators to be reminded once a day of any requests pending for approval. An administrator receives a daily reminder of a pending request until the request is processed. To receive both the immediate notifications and the daily reminders, an administrator's e-mail must be specified on both the AdminNotifyNew and AdminNotifyReminder keywords.

If you are not sending e-mail notifications, see Step 6b on page 161 for directions.

If you are sending e-mail notifications, you need to do the following things:

- Have copies of the forms in the runtime directory. (For information about copying the message forms to the runtime directory, see Step 3 on page 56.
- Customize the forms. (For details, see "Steps for customizing e-mail notification forms" on page 257.)
- For notifications that a request has been rejected, that a certificate is ready for retrieval, that a certificate is about to expire, and that a certificate has been renewed automatically, include the `NotifyEmail` field on certificate requests. This field is already included in the `pkiserv.tmpl` certificate template file. If you are *not* sending e-mail notifications, you need to delete the `NotifyEmail` lines in the `pkiserv.tmpl` file; for details, see Step 6b on page 161.)

  For more information about the NotifyEmail field, see Table 30 on page 127. For information about fields on request forms, see Table 61 on page 311.

The following examples (of notices you can send to users) are in the sample directory:

```
From:dime-o-cert PKI
Subject:Certificate Ready For Pick Up

Attention - Please do not reply to this message as it was automatically sent by a service machine.

Dear %%requestor%%,

Thank you for choosing dime-o-cert PKI. The certificate you requested
for subject %%dn%% is now ready for pickup.

Please visit:
(edit the following link if using the REXX CGI Web Application)
https://www.dimeocert.com/Customers/ssl-cgi-bin/caretrieve.rexx?%%quicklink%%
(edit the following link if using the JSP WebSphere Application)
https://www.dimeocert.com:9080/PKIServ_Web/Customers/certretrieve.jsp?%%quicklink%%
to retrieve your certificate.

If that link does not work, try to go to
(edit the following link if using the REXX CGI Web Application)
http://www.dimeocert.com/Customers/public-cgi/camain.rexx
(edit the following link if using the JSP WebSphere Application)
http://www.dimeocert.com:9080/PKIServ_Web/Customers/pkimain.jsp
And enter the transaction ID listed below:
%%transactionid%%

You will need to input your passphrase that you entered when you submitted the request.
```

Figure 30. Sample of readymsg.form

```
From:dime-o-cert PKI
Subject:Certificate Request Rejected

Attention - Please do not reply to this message as it was automatically sent by a service machine.

Dear %%requestor%%,

Thank you for choosing dime-o-cert PKI. We are sorry to inform you that
your certificate request for subject %%dn%% has been rejected with the following explanation [if any]:
%%rejectreason%%.
Please contact the PKI Services administrator at 1-800-xxx-xxxx.
You will need the transaction ID listed below.

%%transactionid%%
```

Figure 31. Sample of rejectmsg.form

```
From:dime-o-cert PKI
Subject:Certificate Expiration

Attention - Please do not reply to this message as it was automatically sent by a service machine.

Dear %%requestor%%,

Thank you for choosing dime-o-cert PKI. The certificate your requested for
subject %%dn%% expires at %%notafter%% local time. If you wish to renew
your certificate, please visit:
(edit the following link if using the REXX CGI Web Application)
http://www.dimeocert.com/Customers/public-cgi/camain.rexx
(edit the following link if using the JSP WebSphere Application)
http://www.dimeocert.com:9080/PKIServ_Web/Customers/pkimain.jsp
If this is a browser certificate, you must use the same workstation and browser that
you used when you requested the original certificate. If this is a server
certificate, you will have to submit a #10 certificate request.
```

Figure 32. Sample of expiringmsg.form

```
From:dime-o-cert PKI
Subject:Certificate Renewed

Attention - Please do not reply to this message as it was automatically
sent by a service machine.

Dear %%requestor%%,

Your certificate with subject name %%dn%% has been automatically
renewed. Here is your new certificate in Base64 encoded format:
%%printcert%%

If your original certificate is installed in Internet Explorer,
click on this link to install the above new certificate:
(edit the following link if using the REXX CGI Web Application)
https://www.dimeocert.com/Customers/ssl-cgi-bin/installcert.rexx

(edit the following link if using the JSP WebSphere Application)
 https://www.dimeocert.com:9443/PKIServ_Web/Customers/installcert.jsp


Note: The port number 9443 should match your secure port address (_PKISERV_SECURED_PORT) in the web.xml file.
```

Figure 33. Sample of renewcertmsg.form

```
From:dime-o-cert PKI
Subject:Request(s) pending for approval

Dear %%cadomain%% administrator,
The following request(s) is/are waiting for your approval:
%%pendreqlist%%
```

Figure 34. Sample of pendingmsg.form

```
From:dime-o-cert PKI
 Subject:Certificate Recovery

Attention - Please do not reply to this message as it was automatically sent by
 a service machine.

Dear %%requestor%%,

Here is a list of certificate(s) that satisfy your searching criteria for recovery:
%%recoverylist%%

Please choose the certificate you want and visit the corresponding link to
retrieve it (you can identify the certificate by the serial number from the
part of the link between '?' and '&')

(edit the following link if using the REXX CGI Web Application)
https://www.dimeocert.com/Customers/ssl-cgi-bin/caretrieve.rexx?%%recoverylink%%
(edit the following link if using the JSP WebSphere Application)
https://www.dimeocert.com:9080/PKIServ_Web/Customers/certrecover.jsp?%%recoverylink%%

You will need to input your pass phrase that you entered when you submitted the
request.
```

Figure 35. Sample of recoverymsg.form

Notes:

1. PKI Services automatically provides the `To:` value in the forms. You can include `From:` or `Subject:` or both at the top of the file.
2. You must have a blank line between the Subject and the body of the form.

The following table summarizes the variables you can use in the forms when you customize them. At runtime, PKI Services replaces these with their actual values.

Table 53. Descriptions of variables for forms

Variable	Description
`%%cadomain%%`	The CA domain that this message comes from. It is truncated if longer than 8 characters. (This variable is valid only in the pending requests form, `pendingmsg.form`. It is ignored in the other forms.)
`%%dn%%`	The subject's distinguished name. (This variable is valid in all the forms except the pending requests form, `pendingmsg.form`.)
`%%printcert%%`	A renewed certificate in Base64-encoded format. (This variable is valid only in the renewed certificate form, `renewcertmsg.form`. It is ignored in the other forms.)
`%%notafter%%`	The certificate expiration date and time in local time in the format `YYYY/MM/DD HH:MM:SS`. (This variable is valid only in the `expiring.form`. It is ignored in the other forms.)
`%%pendreqlist%%`	A list of pending approval requests (or a single request). Each request contains the transaction ID followed by the corresponding requestor. Each request should be on a line of its own. (This variable is valid only in the pending message form, `pendingmsg.form`. It is ignored in the other forms.)

Table 53. Descriptions of variables for forms (continued)

Variable	Description
%%quicklink%%	A link to a certificate that is ready for pick up. It contains the transaction ID, the "&" character, and the string of the template name or alias, with escaped characters. The following string (broken into two lines so that it fits in the column) is an example: `TransactionId=1j86b0wokkoQ2SHV%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B&Template=PKI+Browser+Certificate` The template name or alias is located from the appldata field of the certificate from the issued certificate list (ICL). The appldata field corresponds to the value of the NICKNAME directive in the template. If PKI Services cannot determine the template name or alias, the %%quicklink%% variable is an empty string, and a warning level message IKYC056I is logged.
%%recoverylink%%	Part of the link to a certificate that can be recovered. It contains the serial number, the "&" character, and the KeyId of the recovery certificate. This part of the link is at the end of the line, appended to the URL link. The entire line is repeated for each entry in the list of certificates that can be recovered.
%%recoverylist%%	The list of certificates that meet the criteria for recovery; that is, the e-mail address for the certificate matches the e-mail address of the user requesting the recovery of the certificate, the password used for the original certificate request matches the password provided by the user, and PKI Services created the keys for the certificate. The URL for each recovered certificate is on a line by itself.
%%rejectreason%%	The reason for the rejection of a certificate request. (This variable is valid for the reject form only.)
%%requestor%%	The requestor of the certificate.
%%transactionid%%	The transaction ID (CertId) returned. (This variable is valid for the ready and reject forms only. It is ignored in the other forms.)

Table 54 summarizes which substitution variables are supported by which forms:

Table 54. Summary of substitution variables in forms. "X" indicates that the form supports the variable.

Substitution variable	readymsg	rejectmsg	expiringmsg	pendingmsg	renewcertmsg	recoverymsg
%%cadomain%%	(ignored)	(ignored)	(ignored)	X	(ignored)	(ignored)
%%dn%%	X	X	X	(ignored)	X	(ignored)
%%printcert%%	(ignored)	(ignored)	(ignored)	(ignored)	X	(ignored)
%%notafter%%	(ignored)	(ignored)	X	(ignored)	(ignored)	(ignored)
%%pendreqlist%%	(ignored)	(ignored)	(ignored)	X	(ignored)	(ignored)
%%quicklink%%	X	(ignored)	(ignored)	(ignored)	(ignored)	(ignored)
%%recoverylink%%	(ignored)	(ignored)	(ignored)	(ignored)	(ignored)	X
%%recoverylist%%	(ignored)	(ignored)	(ignored)	(ignored)	(ignored)	X
%%rejectreason%%	(ignored)	X	(ignored)	(ignored)	(ignored)	(ignored)
%%requestor%%	X	X	X	(ignored)	X	X

Table 54. Summary of substitution variables in forms (continued). "X" indicates that the form supports the variable.

Substitution variable	readymsg	rejectmsg	expiringmsg	pendingmsg	renewcertmsg	recoverymsg
%%transactionid%%	X	X	(ignored)	(ignored)	(ignored)	(ignored)

Steps for customizing e-mail notification forms

Perform the following steps to customize the forms:

1. Make sure the forms you want to use (readymsg.form, rejectmsg.form, expiringmsg.form, renewcertmsg.form, pendingmsg.form, and recoverymsg.form) are present in the runtime directory. (By default, the runtime directory is /etc/pkiserv/. For information about copying files, see Step 3 on page 56.)

---

2. Update the form. At minimum:
   a. Specify your company (instead of dime-o-cert) in the From: line and in the first line of the main paragraph
   b. If appropriate, update the subject.

      Note: There must be a blank line between the subject and the body of the note.
   c. If you are updating a ready, expiring, or recovery form, change the URL in the main paragraph to customize it for your company. The sample forms have URLs for both REXX CGI exec and Java server page (JSP) use. Modify the URL that applies for the method you are using to implement the Web application, and remove the URL that does not apply. Remove the comment lines above each URL.

      For example, if you are modifying the ready form, and you are implementing the Web application using JSPs, change:

      ```
      Please visit:
      (edit the following link if using the REXX CGI Web Application)
      https://www.dimeocert.com/Customers/ssl-cgi-bin/caretrieve.rexx?%%quicklink%%
      (edit the following link if using the JSP WebSphere Application)
      https://www.dimeocert.com:9080/PKIServ_Web/Customers/certretrieve.jsp?%%quicklink%%
      to retrieve your certificate.
      ```

      to

      ```
      Please visit:
      https://www.dimeocert.com:9080/PKIServ_Web/Customers/certretrieve.jsp?%%quicklink%%
      to retrieve your certificate.
      ```

      and modify the URL for your company.
   d. If you are updating a reject form, change the telephone number in the main paragraph to customize it for your company.

   Make any other needed changes. (You can use variables in the body of the form, but you cannot include %%transactionid%% in the expiring form or %%notafter%% in the ready or reject form.)

---

3. Save the file.

Setting up automatic renewal of certificates

You can optionally set up PKI Services to automatically renew certificates when they approach their expiration date, and e-mail the new certificates to their owners. This option is controlled on a template basis.

Steps for setting up automatic certificate renewal

Before you begin: You need to decide for which certificate templates you want to set up automatic certificate renewal. For a description of the templates provided by PKI Services, see "Supported certificate types" on page 7.

Perform the following steps to set up automatic certificate renewal.

1. In the CertPolicy section of the pkiserv.config configuration file, set the field ExpireWarningTime to specify how soon (in days or weeks) before a certificate expires to renew it and send the renewed certificate to its owner. For example, to automatically renew certificates two weeks before they expire:

   ```
   ExpireWarningTime=2w
   ```

2. Set up the renewed certificate e-mail notification form.
   a. Copy the sample renewed certificate notification form, renewcertmsg.form, from the samples directory to the runtime directory. For more information, see "Steps for copying files" on page 55.
   b. Customize the renewed certificate notification form with your company's information. For more information, see "Customizing e-mail notifications sent to users" on page 252.
   c. In the General section of the pkiserv.config configuration file, set the field RenewCertForm to indicate the file that contains the renewed certificate notification form. For example:

      ```
      RenewCertForm=/etc/pkiserv/renewcertmsg.form
      ```

3. If you are implementing the Web application using REX CGI execs, in each template for which you want certificates to be automatically renewed, insert the AUTORENEW tag immediately following the NICKNAME tag, if it is not already there, and set it to Y. For example:

   ```
   <TEMPLATE NAME=1-Year PKI SSL Browser Certificate>
   <TEMPLATE NAME=PKI Browser Certificate>
   <NICKNAME=1YBSSL>
   <AUTORENEW=Y>
   ```

   If you are implementing the Web application using Java server pages (JSPs), for each certificate request template for which you want certificates to be automatically renewed, include the tag <tns:AutoRenew>Y</tns:AutoRenew>. For example:

   ```
   <tns:certreq_template>
   <tns:certname>1-Year SAF Browser Certificate</tns:certname>
   <tns:certtype>SAF Browser Certificate</tns:certtype>
   <tns:AutoRenew>Y</tns:AutoRenew>
   ⋮
   ```

4. For each certificate type that you want to be automatically renewed, except the PKI generated key certificate, make NotifyEmail a required field.

To do this if you are implementing the Web application using REX CGI execs,, in each template for which you want certificates to be automatically renewed (except the PKI generated key certificate), remove the string `(optional)` following the NotifyEmail tag, if it is specified. For example, change

```
%%NotifyEmail (optional)%%
```

to

```
%%NotifyEmail%%
```

If you are implementing the Web application using Java server pages (JSPs), for each certificate request template for which you want certificates to be automatically renewed (except the PKI generated key certificate), remove the string `optional="true"` following the NotifyEmail tag, if it is specified. For example, change:

```
<tns:NotifyEmail optional="true" />
```

to

```
<tns:NotifyEmail />
```

Note: For a PKI generated key certificate, the requestor name is an e-mail address and overrides the NotifyEmail value if specified.

When you are done, you have set up automatic certificate renewal.

Setting up PKI Services to generate keys for certificate requests

There are two ways to generate the key pair (public key and private key) for a certificate request:

- The requestor can generate the key pair and send the public key to PKI Services with the request. In this case, PKI Services has no knowledge of the private key, and cannot recover it if the requestor loses it.
- The requestor can ask PKI Services to generate the key pair. PKI Services uses the PKCS #11 API provided by ICSF to generate the key pair and store it in the token data set (TKDS). The requestor's e-mail address is used as the requestor name. The certificate and private key are packaged in PKCS #12 format and the requestor is sent a link via e-mail to retrieve the package. In this case PKI Services can recover the certificate package if the requestor needs it.

Before PKI Services can generate key pairs for certificates, you must do some setup.

Requirement: The key generation capability requires hardware that supports the PKCS #11 CKM_RSA_PKCS_KEY_PAIR_GEN mechanism. For information about which hardware supports this mechanism, see *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*.

Steps for setting up PKI Services to generate keys for certificate requests

Before you begin: You need to know whether the ICSF token data set (TKDS) has already been set up.

Perform the following steps to set up key generation for certificate requests.

1. If the ICSF token data set (TKDS) has not already been set up, ask the ICSF programmer to set it up. (For information about the TKDS, see *z/OS*

Cryptographic Services ICSF Writing PKCS #11 Applications.) The TKDS must be set up before PKI Services starts, so if necessary stop and restart PKI Services after the TKDS is set up. (For information about stopping and restarting PKI Services, see Chapter 10, "Starting and stopping PKI Services," on page 117.

---

2. Edit the **SAF** section of the PKI Services configuration file, pkiserv.conf and verify that the `TokenName` parameter is specified. If it is not, choose a name for the token in the TKDS that PKI Services will use for storing the key pairs that it generates, and set `TokenName` to the name you choose.

   Rules: A token name must follow these rules:
   - Up to 32 characters in length
   - Permitted characters are:
     - Alphanumeric
     - National: "@" (X'5B'), "#" (X'7B'), or "$" (X'7C')
     - Period: "." (X'4B')
   - The first character must be alphabetic or national
   - Lowercase letters can be used, but are folded to uppercase
   - The IBM1047 code page is assumed

---

3. Edit the **General** section of the PKI Services configuration file, pkiserv.conf and verify that the `ReadyMessageForm` parameter is specified. If it is not:
   - Copy the ready message form from the samples directory to the runtime directory. Follow the instructions in "Steps for copying files" on page 55.
   - Update the `ReadyMessageForm` parameter to specify the full pathname or data set name of the ready message form.
   - Customize the ready message form. Follow the instructions in "Customizing e-mail notifications sent to users" on page 252.

---

4. Set up the e-mail form that is sent if a user requests that PKI Services recover a certificate for which PKI Services generated the keys. The form contains a list of certificates that can be recovered. Edit the **General** section of the PKI Services configuration file, pkiserv.conf and verify that the `RecoverForm` parameter is specified. If it is not:
   - Copy the recovery message form from the samples directory to the runtime directory. Follow the instructions in "Steps for copying files" on page 55.
   - Update the `RecoverForm` parameter to specify the full pathname or data set name of the recovery message form.
   - Customize the recovery message form. Follow the instructions in "Customizing e-mail notifications sent to users" on page 252.

---

5. If you want expired certificates whose keys were generated by PKI Services to be deleted from the ICL automatically after a certain time period, edit the **ObjectStore** section of the PKI Services configuration file, `pkiserv.conf` and update the `RemoveExpiredCertsAndKeys` parameter to specify the time period after which the expired certificates should be deleted.

---

6. The RACF administrator must give the PKI Services daemon the authorization it needs to use the PKCS #11 APIs. The following RACF commands set up the required authorization in the CRYPTOZ class. (By default the daemon user ID

is PKISERVD, but you might be using a different user ID. Check the daemon variable in Table 17 on page 43 if you're not sure what your daemon user ID is.)

```
SETROPTS CLASSACT(CRYPTOZ) GENERIC(CRYPTOZ) RACLIST(CRYPTOZ)
RDEFINE CRYPTOZ SO.daemon_id.* UACC(NONE)
RDEFINE CRYPTOZ USER.daemon_id.* UACC(NONE)
PERMIT SO.daemon_id.* CLASS(CRYPTOZ) ID(daemon_id) ACC(UPDATE)
PERMIT USER.daemon_id.* CLASS(CRYPTOZ) ID(daemon_id) ACC(CONTROL)
SETROPTS RACLIST(CRYPTOZ) REFRESH
```

Note: These commands are included in the IKYSETUP REXX exec. If you have another reason to rerun IKYSETUP, you can update the exec to set up the daemon user ID's authorization in the CRYPTOZ class at the same time. If you don't have another reason to rerun IKYSETUP, the RACF administrator can issue the commands manually.

---

7. (Optional) Because PKI Services stores certificates for which it generates the keys in the TKDS, it can recover those certificates from the TKDS. To recover a certificate, a user must provide the passphrase that was entered when the certificate was originally requested. If the user has forgotten the passphrase, you can use the PKI Services exit to allow the user to recover the passphrase by responding to security questions. For more information, see "Scenario 4: Allow users to recover a PKI generated key certificate when the passphrase is lost" on page 287. Decide whether you want to implement passphrase recovery, and if so, write exit code to implement the function.

When you are done, PKI Services can generate key pairs for certificate requests if asked to do so.

Adding custom extensions to certificates

PKI Services supports the following standard certificate extensions:

- Authority Key Identifier
- Subject Key Identifier
- Key Usage
- Extended Key Usage
- Subject Alternate Name
- BasicConstraints
- CRL Distribution Point
- HostIDMapping
- Authority Information Access
- Certificate Policies

In addition, PKI Services supports custom extensions. You can include any extension in your certificates that is in the form:

```
Extension  ::=  SEQUENCE  {
     extnID      OBJECT IDENTIFIER,
     critical    BOOLEAN DEFAULT FALSE,
     extnValue   OCTET STRING  }
```

PKI Services provides a CustomExt INSERT and a CustomExt JSP that you can use to add custom extensions to your certificate templates. The *n*-year PKI browser certificate template demonstrates the use of custom extensions.

Restriction: Do not add a critical extension to a certificate template for which PKI Services generates the keys.

Steps for adding a custom extension to a certificate template if you are using REXX CGI execs

1. Copy the CustomExt INSERT from `pkiserv.tmpl`.

   _____

2. Rename the copy of the CustomExt INSERT. The name is the string formed by concatenating the following values:
   a. The string "CustomExt".
   b. The OID. You are responsible for ensuring that you use a registered OID, PKI Services does not check this.
   c. The critical flag - `C` for a critical extension, `N` for a non-critical extension.
   d. The value type. Supported types are:
      - INT (integer in a printable hexadecimal format)
      - IA5 (IA5 string)
      - PRT (printable string)
      - BMP (BMP string)
      - OCT (Octet string)
      - UTF (UTF 8 string)

   _____

3. Customize the INSERT to contain any inputs that you want, and to have the corresponding JavaScript code manipulate and verify the inputs.

   _____

4. Add the customized INSERT to the certificate template that you want to use it in.

   _____

5. Update the CONTENT subsection of the certificate template to add new input fields.

   _____

When you are done, you have added a customized extension to a certificate template using REXX CGI execs.

Steps for adding a custom extension to a certificate template if you are using JSPs

1. Copy the CustomExt JSP from the `mod.inc` directory, and the certificate template file, `pkitmpl.xml`, by default in the `/etc/pkiserv/` directory.

   _____

2. Rename the copy of the CustomExt JSP. The name is the string formed by concatenating the following values:
   a. The string "CustomExt".
   b. The OID. You are responsible for ensuring that you use a registered OID, PKI Services does not check this.
   c. The critical flag - `C` for a critical extension, `N` for a non-critical extension.
   d. The value type. Supported types are:
      - INT (integer in a printable hexadecimal format)
      - IA5 (IA5 string)
      - PRT (printable string)
      - BMP (BMP string)

- OCT (Octet string)
- UTF (UTF 8 string)

3. Customize the certificate template file, `pkitmpl.xml` to contain any inputs that you want, and customize the JSP file to manipulate and verify the inputs.

4. Use the TemplateTool utility to validate your updated XML template file. When you have completed your updates and successfully validated them, use the TemplateTool utility to create an updated CGI template file, `pkixgen.tmpl`

5. Update the EAR file with the modified JSP file, and deploy the EAR file to a Websphere application server. For information on how to do this, see "Customizing the PKI Services Web application" on page 198.

When you are done, you have added a customized extension to a certificate template using JSPs.

Forming the CustomExt value for CertPlist for the R_PKIServ callable service

The INSERT or JSP code that you write for a custom extension has to be able to call the R_PKIServ callable service with a properly-formed `CustomExt` value for the `CertPlist` parameter list. The `CustomExt CertPlist` entry is a comma-separated 4-part string with a maximum length of 1024 bytes:

1. The first part is the OID of the extension.
2. The second part is the critical flag:
   - `C` or `c` indicates critical
   - `N` or `n` indicates non-critical
3. The third part is the encode type:
   - `INT` indicates integer
   - `IA5` indicates IA5 string
   - `PRT` indicates printable string
   - `BMP` indicates BMP string
   - `OCT` indicates octet string
   - `UTF` indicates UTF string
4. The fourth part is the value.

For example, given the following ASN.1 notation for an x.509 extension:

```
Extension :: = SEQUENCE {
 extnID OBJECT IDENTIFIER,
 critical BOOLEAN DEFAULT FALSE,
 extnValue OCTET STRING
}
```

and the following definition of an ASN.1 custom extension:

```
SEQUENCE {
. OBJECT IDENTIFIER '1 2 3 4'
. OCTET STRING, encapsulates {
. . INTEGER 240
. . }
. }
```

the `CustomExt` value to produce this would be:

```
1.2.3.4,N,INT,F0
```

Rules: The OID value must follow these rules:

- Its length must not exceed 64 characters.
- It must be comprised of decimal digits and the dot (period) character).
- It cannot start or end with a dot (period) character and cannot contain adjacent dots.
- The first integer value must be 0, 1, or 2. If the first integer is 0 or 1, the second integer must not be greater than 39.
- It must contain a minimum of 3 dot-separated segments (for example 1.2.3).
- The value of the first segment must be 0, 1, or 2. If the first integer is 0 or 1, the second integer must not be greater than 39.
- The largest integer value for all segments after the second segment is the largest 31-bit integer value: 2 147 483 647.

Rules: The critical flag must follow these rules:

- Only mark an extension critical if all applications that will use the certificate know and understand the extension. Applications that do not know and understand an extension that is marked critical are required to deny the use of the certificate.
- Do not mark an extension critical if PKI Services creates the public and private keys for the certificate.

Rules: The encoding types and values must follow these rules:

- The INT encoding type indicates that the extension value encapsulated in the extension value's octet string is a primitive `integer` type.
- Because the INTEGER type in the ASN.1 specification is not limited to fit in a 4-byte construct, the value specified for INT is a hexadecimal string rather than decimal. If the specified hexadecimal string for an INTEGER type contains an odd number of characters, the high-order bit of the first character is propagated in the encoded value. For example, if the `CustomExt` value is `12.3.4,N,INT,800`, because there are only 3 characters in 800, and the high-order of the first character is on, the encoded value is F800 in hexadecimal, or -2048 decimal.
- The PRT encoding type indicates that the extension value encapsulated in the extension value's octet string is a primitive `printableString` type. The characters specified in the `CustomExt` value must conform to the `PrintableString` character set, which is a subset of the 7-bit ASCII or IA5 character set.
- The IA5 encoding type indicates that the extension value encapsulated in the extension value's octet string is a primitive `IA5String` type. The characters specified in the `CustomExt` value must conform to the IA5String character set. Control characters that part of the IA5 character set are not allowed.
- THE UTF encoding type indicates that the extension value encapsulated in the extension value's octet string is a primitive `UTF8String` type. The characters specified in the `CustomExt` value must conform to the Basic Latin and Latin-1 characters with the exception of control characters. To encode a `UTF8String` extension value with characters that fall outside the range supported, you must use the OCT encoding type.
- The BMP encoding type indicates that the extension value encapsulated in the extension value's octet string is a primitive `BMPString` type. The characters specified in the `CustomExt` value must conform to the Basic Latin and Latin-1

characters with the exception of control characters. To encode a BMPString extension value with characters that fall outside the range supported, you must use the OCT encoding type.

- The OCT encoding type indicates that the extension value specified in the CustomExt value is the extension's octet string value. You must specify an even number of printable hexadecimal characters, and the OctetString value should not include the tag and length bytes for the OctetString.

Example: You want to build a custom extension whose extnID is 1.2.3.4, is not marked critical, and has an extnValue composed of a SEQUENCE containing an INTEGER value of 1024 (decimal) followed by a UTF-8 string with the value "AB£¬¥YZ". The ASN.1 definition is:

```
SEQUENCE {
. OBJECT IDENTIFIER '1 2 3 4'
. OCTET STRING, encapsulates {
. . SEQUENCE {
. . . INTEGER 1024
. . . UTF8String 'AB£¬¥YZ'
. . . }
. . }
. }
```

Use the following value for CustomExt:

```
CustomExt=1.2.3.4,N,OCT,3010020204000C0A4142C2A3C2ACC2A5595A
```

For the OCT value:

```
3010 = SEQUENCE(0x30) of length 0x10
02020400 = INTEGER(0x02) of length 0x02, value 0x0400 = 1024 decimal
0C0A4142C2A3C2ACC2A5595A = UTF8String(0x0C) of length 0x0A,
    value = 41(A) 42(B) C2A3(£) C2AC(¬) C2A5(¥) 59(Y) 5A(Z)
```

Chapter 15. Customizing with installation exit routines

Programming Interface information

PKI Services supports the use of installation exit routines in the following ways:

- The PKI Services daemon can call an installation-provided exit routine for automatic renewal processing.
- If you implement the PKI Services Web application using REXX CGI execs, the PKI Services Web application CGIs can call an installation-provided exit routine for end-user functions except VERIFY.
- If you implement the PKI Services Web application using Java server pages (JSPs), exit methods are called before and after end-user functions except VERIFY.

PKI Services provides a sample exit routine, `pkiexit.c`, written in the C language. It is intended to demonstrate the power of the exit routine for the daemon and REXX CGI execs, and to provide a guide for you to write your own exit routine. The main routine of the program determines which subroutine to call, based on the `R_PKIServ` function being called and whether this is a pre- or post-processing call.

No sample is provided for the exit methods used with JSPs.

You can implement the exit routines for the PKI Services daemon and the PKI Services Web application CGIs in the same program (as shown in the sample, `pkiexit.c`) or in separate programs.

PKI Services provides the following files for the daemon and CGI exit routine. Both files are, by default, located in: `/usr/lpp/pkiserv/samples/`.

Table 55. Files for the `pkiexit.c` exit routine

File name	Description
`pkiexit.c`	Code sample for the exit routine (in the C programming language). You probably need to update the exit routine code before using it.
`Makefile.pkiexit`	Makefile for `pkiexit.c`.

Exit routine processing for automatic certificate renewal

The PKI Services daemon supports an installation-provided exit routine for automatic renewal processing. An exit routine can be written to provide additional automatic renewal criteria, and to capture the renewed certificate for further processing. If you choose to implement this exit routine, it must be a UNIX executable program residing in a file system, with appropriate permission assigned. The PKI Services daemon identifies the exit routine as the program specified by the value of the `_PKISERV_EXIT` environment variable in the `pkiserv.envars` file. The value specified is limited to a maximum of 256 characters. The exit routine is invoked by the PKI Services daemon using standard UNIX parameters (that is, `argc` and `argv[]`). The exit routine communicates its results back to the PKI Services daemon by way of a return code. The exit routine is called for preprocessing and post-processing before and after automatic certificate renewal processing. Unlike the PKI Services CGI exit routines, messages written to

© Copyright IBM Corp. 2001, 2011

either STDOUT or STDERR do not appear in either the Web server or PKI Services daemon logs. If you want to write messages in the exit program, you need to open a file and write messages to that file. The sample exit routine provided in `/usr/lpp/pkiserv/samples/pkiexit.c` illustrates writing messages to a file in both the preprocessing and post-processing exit functions.

Note: This exit routine can be implemented in the same program as the exit routines for the PKI Services CGIs (as illustrated in the sample `pkiexit.c` exit program) or can be implemented as a separate program.

The `ExitTimeout` keyword in the General section of the `pkiserv.conf` file specifies the maximum time PKI Services will wait for the exit routine to return. If `ExitTimeout` is not specified, PKI Services waits at most 30 seconds for the exit routine to return. If `ExitTimeout` is specified with a value greater than 1 hour, PKI Services waits 1 hour at the most for the exit routine to return.

Steps for updating the exit routine code sample

To update the exit routine code sample, `pkiexit.c`, perform the following steps:

1. Copy the sample exit routine and makefile to the current directory by entering the following commands:

   ```
   cp /usr/lpp/pkiserv/samples/pkiexit.c pkiexit.c
   cp /usr/lpp/pkiserv/samples/Makefile.pkiexit Makefile
   ```

   ---

2. Compile and link to produce the executable program, `pkiexit`, by entering the following command:

   ```
   make
   ```

   ---

3. Move the executable program to its execution directory and set the permissions by entering the following commands:

   ```
   mv pkiexit /full-directory-name
   chmod 755 /full-directory-name/pkiexit
   ```

   ---

4. Edit the PKI Services environment variables file by entering the following command:

   ```
   oedit /etc/pkiserv/pkiserv.envars
   ```

   and add the environment variable `_PKISERV_EXIT` by adding the following line to the file:

   ```
   _PKISERV_EXIT=/full-directory-name/pkiexit
   ```

Using the exit routine for pre- and post-processing

This exit routine is called for preprocessing and postprocessing by the PKI Services daemon before and after it renews a certificate respectively. Automatic certificate renewal processing is performed by the daily maintenance task. By default this task runs when the PKI Services daemon is started and every day at midnight, but you can customize the days on which it runs and the time at which it runs. (For information about customizing when the maintenance task runs, see “Optionally updating the pkiserv.conf configuration file” on page 58.)

Table 56. Values of arguments for pre- and post-processing

Time of processing	Argument 1	Argument 2
Preprocessing	0	The function number in EBCDIC: **500** automatic renewal processing
Post-processing	1	

Automatic renewal - preprocessing

Purpose: Provide additional criteria for automatic renewal of certificates.

Arguments:

argument **3**
The Base64-encoded original certificate.

Return codes:

Return code	Meaning
0	Continue with the request for automatic renewal processing.
4	Disallow automatic renewal processing for the current request.
8	Disallow automatic renewal processing from now on.

Note: Any value other than 0, 4, and 8 is treated as 4.

STDOUT: Non-applicable.

Note: The automatic renewal exit routine cannot write to STDOUT or STDERR like other exit routines. The output has to be written to a file.

Automatic renewal - post-processing

Purpose: Capture the renewed certificate for further processing.

Arguments:

argument **3**
The Base64-encoded renewed certificate.

Return codes:

Return code	Meaning
0	Normal return.

Note: The return code is not checked from the post exit processing.

STDOUT: Non-applicable.

Note: The automatic renewal exit routine cannot write to STDOUT or STDERR like other existing exit routines. The output has to be written to a file.

Scenario for using the exit routine

This scenario disables the automatic renewal of certificates for contractors, postpones the renewal if the expiration date is more than 30 days away, and logs the subject name and serial number of certificates that are automatically renewed.

For sample code illustrating this scenario, see the sample exit routine `pkiexit.c` shipped with PKI Services.

The preprocessing exit routine for the automatic renewal function (subroutine `preProcessAutoRenewExit`) disables the automatic renewal of certificates for contractors and postpones the renewal if the expiration date is more than 30 days away. Here are the steps:

- Get the current time and format it for output.
- Call subroutines to decode the Base64 certificate data and decode the certificate.
- Get a printable version of the subject name from the certificate.
- Get a printable version of the serial number from the certificate.
- Check the subject name for an organizationalUnitName of "Contractors". If found, log a message indicating that the renewal was disabled, and return with a return code of 8 to disable the automatic renewal.
- Call a subroutine to determine how many days there are until the certificate expires.
- If there are more than 30 days before the certificate expires, log a message indicating that the renewal was postponed, and return with a return code of 4 to postpone the automatic renewal.
- If there are 30 or fewer days before the certificate expires, log a message indicating that renewal of the certificate was allowed, and return with a return code of 0 to continue with the automatic renewal.

The postprocessing exit routine for the automatic renewal function (`postProcessAutoRenewExit`) logs the subject name and serial number of certificates that have been automatically renewed. Here are the steps:

- Get the current time and format it for output.
- Call subroutines to decode the Base64 certificate data and decode the certificate.
- Get a printable version of the subject name from the certificate.
- Get a printable version of the serial number in hexadecimal from the certificate.
- Log a message indicating that the certificate was renewed, containing the subject name, serial number, and time.

Exit routine processing for the PKI Services CGIs

For the end-user functions except VERIFY, the PKI Services Web application CGIs support calling an installation-provided exit routine. The exit routine can perform tasks such as the following:

- Provide additional authorization checking
- Validate and change parameters
- Capture certificates for further processing
- Recover a passphrase used in a certificate request

If the exit routine exists, it must be a UNIX executable program residing in the file system, and it must have appropriate permission assigned. To specify the exit routine, the UNIX programmer sets the `_PKISERV_EXIT` environment variable in the

Web server's environment variables file. On input it receives standard UNIX parameters (that is, *argc* and *argv*[]). It communicates back to PKISERV through the return code and by writing to STDOUT.

Steps for updating the exit routine code sample

To update the exit routine code sample, `pkiexit.c`, perform the following steps:

1. Copy the sample exit routine and makefile to the current directory by entering the following commands:

   ```
   cp /usr/lpp/pkiserv/samples/pkiexit.c pkiexit.c
   cp /usr/lpp/pkiserv/samples/Makefile.pkiexit Makefile
   ```

2. Compile and link to produce the executable program, `pkiexit`, by entering the following command:

   ```
   make
   ```

3. Move the executable program to its execution directory and set the permissions by entering the following commands:

   ```
   mv pkiexit /full-directory-name
   chmod 755 /full-directory-name/pkiexit
   ```

4. Edit the Web server's environment variables file by entering the following command:

   ```
   oedit /etc/httpd.envvars
   ```

   and add the environment variable `_PKISERV_EXIT` by adding the following line to the file:

   ```
   _PKISERV_EXIT=/full-directory-name/pkiexit
   ```

Using the exit routine for pre- and post-processing

The exit routine is called:

- For preprocessing before calling the R_PKIServ (IRRSPX00) SAF callable service
- For post-processing after returning from the callable service.

The following table summarizes the values of the first two arguments for pre- and post-processing. (Additional arguments vary, depending on the function to perform.)

Table 57. Values of arguments for pre- and post-processing

Time of processing	Argument 1	Argument 2
Preprocessing	0	The function number from the R_PKIServ SAF callable service in EBCDIC: **1** GENCERT **2** EXPORT **9** REQCERT **11** REVOKE **12** GENRENEW **13** REQRENEW **17** QRECOVER
Post-processing	1	

Note: The parameters that are input to the CGIs and the values resolved by the CGIs (`argument 3...argument n` for all functions) will vary based on how you have customized the templates.

Return codes

The topics that follow contain tables of expected return codes. If calling the exit routine produces an unexpected return code, that is, one that is not listed, PKI Services treats it as a failure. Processing for the request stops and an error message is issued.

The return code is a one-byte value.

GENCERT and GENRENEW - preprocessing

Purpose: Provide additional authorization checking, parameter validation and modification, and a mapping of the passphrase to a set of security answers to use during QRECOVER processing, if the passphrase is forgotten.

Arguments:

argument **3...***argument n*
The parameters as input to the CGI plus values resolved by the CGI in *name=value* form, for example, "`CommonName=Sam Smith`".

Return codes:

Return code	Meaning
0	Continue with the request with possible modifications.
4	Continue with the request with possible modifications, but change it to require administrator approval.
8 - 49	Deny the request and return to the caller immediately.

STDOUT: Zero or more additional *CertPlist* parameters to add to the request in *name=value* form, one per line. For those fields defined as non-repeating (according to the documentation for the IRRSPX00 callable service, for example, *CommonName*), specifying the parameters here in effect replaces the CGI input values.

GENCERT and GENRENEW - post-processing

Purpose: Capture the TransactionId or failing return codes for further processing.

Arguments:

argument **3...***argument n***–3**
The final set of parameters as determined by the preprocessing exit in *name=value* form.

*argument n***–2**
The RACF return code from the callable service.

*argument n***–1**
The RACF reason code from the callable service.

argument n
The *TransactionId*. This is a string of undetermined value if the request was unsuccessful.

Return codes:

Return code	Meaning
0	Normal

STDOUT: Optional replacement *TransactionId*.

REQCERT and REQRENEW - preprocessing

Purpose: Provide additional authorization checking, parameter validation and modification, and a mapping of the passphrase to a set of security answers to use during QRECOVER processing, if the passphrase is forgotten.

Arguments:

argument **3...***argument n*
The parameters as input to the CGI plus values resolved by the CGI in *name=value* form, for example, "`CommonName=Sam Smith`".

Return codes:

Return code	Meaning
0	Continue with the request with possible modifications.
4	Continue with the request with possible modifications, but change it to not require administrator approval.
8 - 49	Deny the request and return to the caller immediately.

STDOUT: Zero or more additional *CertPlist* parameters to add to the request in *name=value* form, one per line. For those fields defined as non-repeating (according to the documentation for the IRRSPX00 callable service, for example, *CommonName*), specifying the parameters here in effect replaces the CGI input values.

REQCERT and REQRENEW - post-processing

Purpose: Capture the *TransactionId* or failing return codes for further processing.

Arguments:

argument **3...***argument n***–3**
The final set of parameters as determined by the preprocessing exit routine in *name=value* form.

*argument n***–2**
The RACF return code from the callable service.

*argument n***–1**
The RACF reason code from the callable service.

argument n
The *TransactionId*. This is a string of undetermined value if the request was unsuccessful.

Return codes:

Return code	Meaning
0	Normal

STDOUT: Optional replacement *TransactionId*.

EXPORT - preprocessing

Purpose: Provide additional authorization checking and parameter validation and modification.

Arguments:

argument **3...***argument n*
The parameters as input to the CGI in *name=value* form, for example, "TransactionId=12345".

Return codes:

Return code	Meaning
0	Continue with the export.
8 - 49	Deny the request and return to the caller immediately.

STDOUT: Optional replacement *TransactionId* and *ChallengePassPhrase* parameters in *name=value* form, one per line. If these values are provided, they replace the user-provided values on the call to the SAF callable service. If TransactionId is specified without *ChallengePassPhrase*, the user-provided *ChallengePassPhrase* is used. If *ChallengePassPhrase* is specified without *TransactionId*, the user-provided *TransactionId* is used.

EXPORT - post-processing

Purpose: Capture the certificate or failing return codes for further processing.

Arguments:

argument **3...***argument n***–3**
The parameters as input to the CGI in *name=value* form, followed by any modified value provided by the preprocessing exit routine, also in *name=value* form.

*argument n***–2**
The RACF return code from the callable service.

*argument n***–1**
The RACF reason code from the callable service.

argument n
The base64-encoded certificate with header and footer. This is a string of undetermined value if the request was unsuccessful.

Return codes:

Return code	Meaning
0	Normal

STDOUT: Non-applicable.

REVOKE - preprocessing

Purpose: Provide additional authorization checking and parameter validation.

Arguments:

argument **3...***argument n*
The parameters as input to the CGI in *name=value* form, for example, "`reason=1`".

Return codes:

Return code	Meaning
0	Continue with the request.
8 - 49	Deny the request and return to the caller immediately.

STDOUT: Non-applicable.

REVOKE - post-processing

Purpose: Capture the certificate or failing return codes or both for further processing.

Arguments:

argument **3...***argument n***–2**
The parameters as input to the CGI in *name=value* form, for example, "reason=1".

*argument n***–1**
The RACF return code from the callable service.

argument n
The RACF reason code from the callable service.

Return codes:

Return code	Meaning
0	Normal

STDOUT: Non-applicable.

QRECOVER - preprocessing

Purpose: Provide a mechanism to retrieve a passphrase needed to recover a certificate, in case it was forgotten. The exit routine is called when a user has entered answers to the security questions instead of a passphrase.

Arguments:

argument **3...***argument n*
The parameters as input to the CGI in *name=value* form, for example, "Security1=Brazil".

Return codes:

Return code	Meaning
0	Continue with the query.
4	Cannot determine the passphrase.
8 - 49	Deny the request and return to the caller immediately.

STDOUT: Optional replacement of `Requestor` and add the `PassPhrase` parameters in `name=value` form, one per line.

QRECOVER - post-processing

Purpose: Capture the list of the recovery certificates for further processing.

Arguments:

argument **3...***argument n***–3**
The input and output set of parameters of the preprocessing exit routine in the *name=value* form.

*argument n***–2**
The RACF return code from the callable service.

*argument n***–1**
The RACF reason code from the callable service.

argument n
The list of the recovery certificates.

Return codes:

Return code	Meaning
0	Normal

STDOUT: Non-applicable.

Scenarios for using the exit routine

The sample exit routine supplied with PKI Services, `pkiexit.c`, illustrates the following scenarios. The main routine of the program determines which subroutine to call, based on the `R_PKIServ` function being called and whether this is a pre- or post-processing call. Individual subroutines in the program handle the scenarios.

Scenario 1: Allow only selected users to request PKI browser certificates for authenticating to z/OS

This scenario is for allowing only selected local z/OS users to request PKI browser certificates for authenticating to z/OS. Additionally, this scenario is for providing a customized TITLE value for the subject's distinguished name based on the user's role in the organization. Permission and the user's role in the organization is indicated by access to the BPX.SERVER resource in the FACILITY class and by the user's level of access to FACILITY class resources called PROJ.MEMBER and PROJ.PARTNER. The access values are as follows:

NONE	No access for either resource. The user is not permitted to request this type of certificate. The certificate request is denied.
READ to PROJ.MEMBER	The user is a team member and is permitted to request the certificate. The TITLE value is set to `Team Member`. Certificate requests for team members are automatically approved. (No administrator approval is required.)
UPDATE to PROJ.MEMBER	The user is the team's leader and is permitted to request the certificate. The TITLE value is set to `Team Leader`. A certificate request by the team leader is automatically approved. (No administrator approval is required.)
READ to PROJ.PARTNER	The user is considered to be a general partner of the team, not an active team member. The user is allowed to request certificates, but the requests require administrator approval before being issued. The TITLE value is set to `Team Partner`.
UPDATE to PROJ.PARTNER	The user is considered to be a trusted partner of the team, not an active team member. The user is allowed to request certificates, and unlike requests of the general partner, the certificate request are automatically approved. The TITLE value is set to `Team Trusted Partner`.

The preprocessing exit routine call for the GENCERT and REQCERT functions (subroutine `preProcessGenReqCertExit`) handles the logic described in the preceding. Here are the steps:

- The request values are passed into the exit routine through *argv* in *field-name=field-value* pairs, and the subroutine looks for the `Template=` and `UserId=` in the input parameters.
- When the exit routine code finds a `Template=` value containing `PKI Browser Certificate For Authenticating To z/OS`, the `__check_resource_auth_np()` system function examines the user ID. This determines the user's access to the preceding profiles.
  - If the user has no authority to either of these resources, return code 8 is set. This causes the request to be denied.

– Otherwise the user's TITLE is set by writing the `TITLE=title-value` string to STDOUT.

By default, administrator approval is not required for the PKI browser certificate for authenticating to z/OS.

– When the user has only READ access to PROJ.PARTNER, the function must be changed to require administrator approval. This is done by setting return code 4.
– For all other accesses the function does not need to be changed.

Scenario 2: Maintain a customized certificate repository (database) independent of PKI Services

This scenario is for maintaining a customized certificate repository (database) that is independent of PKI Services. After a successful submission of a certificate request, PKI Services returns the transaction ID. This is saved in a new customer-provided database entry. An alias for this database entry is then returned to the end user as the transaction ID. Later, when the user wishes to pick up the certificate, the user-entered alias name is used to retrieve the actual PKI Services transaction ID. The retrieved certificate is saved in the database entry before being returned to the user.

Three different exit routine calls handle the preceding logic.

- Post-processing for the GENCERT or REQCERT functions (subroutine `postProcessGenReqCertExit`) returns a pretend alias entry name by suffixing the actual transaction ID with either `SAF` or `PKI`. This is where the database entry should be created. (Note that the exit routine performs no actual database calls because this would be too customer-specific.)
- Preprocessing for the EXPORT function (subroutine `preProcessExportExit`) reverts the transaction ID to its original value. This emulates retrieval from the database entry.
- Post-processing for the EXPORT function (subroutine `postProcessExportExit`) saves the returned certificate to a database entry. This is emulated by writing it to a file.

Scenario 3: Mandate a policy for certificate renewal only within 30 days of expiration

This scenario is for mandating a policy that allows users to renew their certificates only when certificates are within 30 days of expiring. When the condition is met, you can change the expiration date for the renew request so that the new certificate's validity period is extended by the number of days specified by the NotAfter parameter. In other words, the new certificate should expire *n* days from the current date, where *n = number of days left in the old certificate's validity period + number of days specified by NotAfter*.

The preprocessing exit routine call for GENRENEW and REQRENEW functions (subroutine `preProcessGenReqRenewExit`) handles the preceding logic. Here are the steps:

- The user's certificate is extracted from the environment variable `HTTPS_CLIENT_CERT`.
- The *NotAfter* value is extracted from the input parameters (*argv*), converted to a number, and saved in the variable *RequisitePro*.
- Subroutine `determineExpiration` is called to extract the expiration date from the user's certificate. This subroutine calls several lower subroutines to base64 decode the certificate, DER decode the binary certificate, and convert the expiration date to a seconds value.

- Upon return from `determineExpiration`, the variable *timeBeforeExp* is the number of seconds from now that the certificate expires. This is compared against the number of seconds in 30 days (86400 × 30) to see if it is greater than 30 days.
  - If it is greater than 30, the request is rejected by setting return code 8.
  - If it is not greater than 30, the new *NotAfter* value is computed as *timeBeforeExp*/86400 + *requestPeriod*.
- This new *NotAfter* value is set by writing it to STDOUT.

Scenario 4: Allow users to recover a PKI generated key certificate when the passphrase is lost

To recover a certificate for which PKI Services generated the keys, the user must provide the passphrase that was provided when the certificate was requested. This scenario illustrates how PKI Services can recover lost passphrases for PKI generated key certificates. To be able to recover a lost passphrase, the user must provide answers to security questions in addition to the passphrase when the user initially requests the PKI generated key certificate. PKI Services saves the passphrase and the answers to the security questions in a passphrase mapping database. To recover the lost passphrase, the user provides the answers to the security questions through the PKI Services Web page. PKI Services searches the passphrase mapping database, and if the security answers match those provided by the user when the certificate was requested, the passphrase is returned to the CGI. The recovered passphrase is then used to retrieve the PKI generated key certificate.

Two exit routine calls are required:

- When the user requests the PKI generated key certificate, the preprocessing exit routine for the GENCERT and REQCERT functions (subroutine `preProcessGenReqCertExit`) collects the requestor name, the passphrase, and the answers to the security questions from the exit routine's parameter list. The exit routine records the information as an entry in a passphrase mapping database.
- When the user attempts to recover the PKI generated key certificate, the preprocessing exit routine for the QRECOVER function (subroutine `preProcessQRecoverExit`) collects the requestor name and the answers to the security questions from the exit routine's parameter list. The exit routine then searches the passphrase mapping database for entries that match the requestor name and the security answers provided by the user. If a match is found, the passphrase recorded in that entry is returned to the CGI through STDOUT.

Exit routine processing for Java server pages (JSPs)

If you implement the PKI Services Web application using Java server pages (JSPs), you can use methods in the class com.ibm.pki.web.exits.UserExit to customize the Web application. These methods are called before and after each of the following `R_PKIServ` requests:

- GENCERT
- REQCERT
- EXPORT
- GENRENEW
- REQRENEW
- REVOKE
- QRECOVER

The methods are empty stub methods to which you can add code, in order to audit or modify parameters being passed to PKI Services. This topic describes these methods and related classes.

The R_PKIServ callable service is described in *z/OS Security Server RACF Callable Services*.

Table 58. Package and class summary for JSP exit processing

Package	Class
com.ibm.pki.web.exits	UserExit
	ExportCert
	QRecover
	RevokeCert
	UserExitException
com.ibm.pki.rpkiserv	CertPlist
	PkiCertificate
	QrecoverResultsList
	RpkiservException

Class UserExit

Package: com.ibm.pki.web.exits

```
public class UserExit
extends java.lang.Object
```

UserExit defines the following static constant integer values:

```
UserExit.SUCCESSFUL = 0;
UserExit.CHANGE_APPROVAL_STATUS = 4;
UserExit.QRECOVER_PASSPHRASE_NOT_FOUND = 4;
UserExit.DENY_REQUEST = 8;
```

Table 59. Methods in class UserExit

Method	Purpose
preGenReqCert	Called before GENCERT or REQCERT requests
postGenReqCert	Called after GENCERT or REQCERT requests
preGenReqRenew	Called before GENRENEW or REQRENEW requests
postGenReqRenew	Called after GENRENEW or REQRENEW requests
preExport	Called before EXPORT requests
postExport	Called after EXPORT requests
preRevoke	Called before REVOKE requests
postRevoke	Called after REVOKE requests
preQRecover	Called before QRECOVER requests
postQRecover	Called after QRECOVER requests

preGenReqCert method

```
public int preGenReqCert(java.lang.String domain,
                         com.ibm.pki.rpkiserv.CertPlist plist,
                         java.lang.String[] security)
                  throws UserExitException
```

Purpose: Called before GENCERT or REQCERT requests.

Parameters:

domain
domain name

plist
CertPlist with input parameters for GENCERT or REQCERT processing whose values can be modified by this method

security
An array of responses to security questions. These correspond to form fields with names `security1`, `security2`, and so forth, in ascending numerical order.

returns:

Value	Meaning
0 (UserExit.SUCCESSFUL)	Continue with the request.
4 (UserExit.CHANGE_APPROVAL_STATUS)	If the certificate request required administrator approval, change it to not require administrator approval (a GENCERT). If the certificate request did not require administrator approval (a GENCERT), change it to require administrator approval (a REQCERT).
8 (UserExit.DENY_REQUEST) or greater	Deny the request.

throws:

Exception	Result
UserExitException	Handled the same as a return value of 8, and exception text is displayed on the resulting web page

postGenReqCert method

```
public java.lang.String postGenReqCert(java.lang.String domain,
                                       com.ibm.pki.rpkiserv.CertPlist plist,
                                       int RACFrc,
                                       int RACFrsncode,
                                       java.lang.String transactionid)
```

Purpose: Called after GENCERT or REQCERT requests

Parameters:

domain
Domain name

plist
CertPlist that was input to R_PKIServ processing, including modifications made by the preGenReqCert method

RACFrc
RACF return code

RACFrsncode
RACF reason code

transactionid
Transaction ID, null if GENCERT processing was unsuccessful

returns: transaction ID

preGenReqRenew method

```
public int preGenReqRenew(java.lang.String domain,
                          com.ibm.pki.rpkiserv.CertPlist plist,
                          java.lang.String serialnum)
                   throws UserExitException
```

Purpose: Called before GENRENEW or REQRENEW requests

Parameters:

domain
Domain name

plist
CertPlist with input parameters for renew request, whose values can be modified by this method

Note: Most values for a certificate renewal are taken from the existing certificate and therefore are not in the certPlist. The following values can occur in the certPlist:
CertPlist.CERTPLIST_NOTIFYEMAIL
CertPlist.CERTPLIST_PASSPHRASE
CertPlist.CERTPLIST_NOTAFTER
CertPlist.CERTPLIST_CERTPOLICIES
CertPlist.CERTPLIST_AUTHINFOACC
CertPlist.CERTPLIST_CRITICAL

serialnum
Serial number of certificate being renewed

returns:

Value	Meaning
0 (UserExit.SUCCESSFUL)	Continue with the request.
4 (UserExit.CHANGE_APPROVAL_STATUS)	If the renewal request required administrator approval, change it to not require administrator approval (a GENRENEW). If the certificate request did not require administrator approval (a GENRENEW), change it to require administrator approval (a REQRENEW).
8 (UserExit.DENY_REQUEST) or greater	Deny the request.

throws:

Exception	Result
UserExitException	Handled the same as a return value of 8, and exception text is displayed on the resulting web page

postGenReqRenew method

```
public java.lang.String postGenReqRenew(java.lang.String domain,
                                        com.ibm.pki.rpkiserv.CertPlist plist,
                                        int RACFrc,
                                        int RACFrsncode,
                                        java.lang.String transactionid)
```

Purpose: Called after GENRENEW or REQRENEW requests

Parameters:

domain
Domain name

plist
The CertPlist that was input to the R_PKIserv request, including any modifications that were made by the preGenReqRenew method

RACFrc
RACF return code

RACFrsncode
RACF reason code

transactionid
Transaction ID, null if REQCERT processing was unsuccessful

returns: transactionid

preExport method

```
public int preExport(java.lang.String domain,
                     com.ibm.pki.rpkiserv.ExportCert exportobject)
```

Purpose: Called before EXPORT requests

Parameters:

domain
Domain name

exportobject
ExportCert object containing transaction ID and passphrase

returns:

Value	Meaning
0 (UserExit.SUCCESSFUL)	Continue with the EXPORT.
8 (UserExit.DENY_REQUEST) or greater	Deny the request.

throws:

Exception	Result
UserExitException	Handled the same as a return value of 8, and exception text is displayed on the resulting web page

postExport method

```
public void postExport(java.lang.String domain,
                       com.ibm.pki.rpkiserv.ExportCert exportobject,
                       int RACFrc,
                       int RACFrsncode)
```

Purpose: Called after EXPORT requests

Parameters:

domain
Domain name

exportobject
ExportCert object containing transaction ID and passphrase. The ExportCert object contains a base64-encoded certificate with header and footer if the request was successful.

preRevoke method

```
public int preRevoke(java.lang.String domain,
                     RevokeCert revokeobject)
```

Purpose: Called before REVOKE requests

Parameters:

domain
Domain name

revokeobject
RevokeCert object containing reason number and serial number

returns:

Value	Meaning
0 (UserExit.SUCCESSFUL)	Continue with the REVOKE request.
8 (UserExit.DENY_REQUEST) or greater	Deny the request, don't revoke.

throws:

Exception	Result
UserExitException	Handled the same as a return value of 8, and exception text is displayed on the resulting Web page

postRevoke method

```
public void postRevoke(java.lang.String domain,
                       RevokeCert revokeobject,
                       int RACFrc,
                       int RACFrsncode)
```

Purpose: Called after REVOKE requests

Parameters:

domain
Domain name

revokeobject
RevokeCert object containing reason number and serial number

preQRecover method

```
public int preQRecover(java.lang.String domain,
                       QRecover qrecoverobject)
```

Purpose: Called before QRECOVER requests

Parameters:

domain
Domain name

qrecoverobject
QRecover object containing recovery e-mail, passphrase and security responses

returns:

Value	Meaning
0 (UserExit.SUCCESSFUL)	Continue with the request.
4 (UserExit.QRECOVER_PASSPHRASE_NOT_FOUND)	Cannot determine the passphrase.
8 (UserExit.DENY_REQUEST) or greater	Deny the request.

postQRecover method

```
public int postQRecover(java.lang.String domain,
                        QRecover qrecoverobject,
                        int RACFrc,
                        int RACFrsncode)
```

Purpose: Called after QRECOVER requests

Parameters:

domain
Domain name

qrecoverobject
QRecover object containing recovery e-mail, passphrase, security responses, and a QrecoverResultsList array containing data for certificates that matched the search criteria

Class ExportCert

```
public class ExportCert
extends java.lang.Object
```

ExportCert contains the parameters passed on an R_PKIServ EXPORT request.

Method summary	
com.ibm.pki.rpkiserv.PkiCertificate	**getCertificate**()
java.lang.String	**getPassphrase**()
java.lang.String	**getTransactionid**()
void	**setPassphrase**(java.lang.String passphrase)
void	**setTransactionid**(java.lang.String transactionid)

Class QRecover

Package: com.ibm.pki.web.exits

```
public class QRecover
extends java.lang.Object
```

QRecover contains the parameters passed on an R_PKIServ QRECOVER request.

Method summary	
java.lang.String	**getPassphrase**()
com.ibm.pki.rpkiserv.QrecoverResultsList[]	**getQrecover_results**()
java.lang.String	**getRequestor**()
java.lang.String[]	**getSecurity_answers**()
void	(java.lang.String passphrase)**setPassphrase**
void	**setRequestor**(java.lang.String requestor)
void	**setSecurity_answers**(java.lang.String[] security_answers)

Class RevokeCert

Package: com.ibm.pki.web.exits

```
public class RevokeCert
extends java.lang.Object
```

RevokeCert contains the parameters passed on an R_PKIServ REVOKE request.

Method summary	
int	**getReason**()
java.lang.String	**getSerial_number**()
void	**setReason**(int reason)
void	**setSerial_number**(java.lang.String serial_number)

Class UserExitException

Package: com.ibm.pki.web.exits

```
public class UserExitException
extends java.lang.Exception
```

An exception thrown from the UserExit class. The PKI Services function that catches this exception treats it the same as a return code greater than or equal to 8, and stops processing. Additionally, any exception text is displayed on the Web page that reports the unsuccessful processing.

Constructors
UserExitException(java.lang.String message)
UserExitException(java.lang.Throwable cause)

Class CertPlist

Package: com.ibm.pki.rpkiserv

```
public class CertPlist
extends java.lang.Object
```

The CertPlist (certificate parameter list) is used to pass certificate information to the R_PKIServ callable service. It defines the following static constant strings, which should be used for the name parameter:

```
CERTPLIST_UNSTRUCTNAME
CERTPLIST_EMAILADDR
CERTPLIST_MAIL
CERTPLIST_TITLE
CERTPLIST_ORGUNIT
CERTPLIST_ORG
CERTPLIST_STREET
CERTPLIST_LOCALITY
CERTPLIST_STATEPROV
CERTPLIST_POSTALCODE
CERTPLIST_COUNTRY
CERTPLIST_KEYUSAGE
CERTPLIST_EXTKEYUSAGE
CERTPLIST_NOTBEFORE
CERTPLIST_NOTAFTER
CERTPLIST_ALTIPADDR
CERTPLIST_ALTURI
CERTPLIST_ALTEMAIL
CERTPLIST_ALTDOMAIN
CERTPLIST_ALTOTHER
CERTPLIST_CUSTOMEXT
CERTPLIST_NOTIFYEMAIL
CERTPLIST_PUBLICKEY
CERTPLIST_SIGNWITH
CERTPLIST_HOSTIDMAP
CERTPLIST_REQUESTOR
CERTPLIST_PASSPHRASE
CERTPLIST_USERID
CERTPLIST_LABEL
CERTPLIST_CERTPOLICIES
CERTPLIST_AUTHINFOACC
CERTPLIST_CRITICAL
CERTPLIST_SERIALNUMBER
CERTPLIST_DNQUALIFIER
CERTPLIST_UID
CERTPLIST_COMMONNAME
CERTPLIST_DOMAINNAME
CERTPLIST_EMAIL
CERTPLIST_CLIENTNAME
CERTPLIST_STARTDATE
CERTPLIST_ENDDATE
CERTPLIST_AUTORENEW
CERTPLIST_KEYSIZE
```

Method summary	
void	**addValue**(java.lang.String name, java.lang.String value) Adds a value to the certPlist.
void	**addValue**(java.lang.String name, java.lang.String[] values) Adds values to the certPlist.
java.util.Vector	**getNames**() Gets the names of all name and value pairs in the certPlist.
java.util.Vector	**getValues**(java.lang.String name) Gets the vector of string values for this name in the certPlist.
void	**removeAllValues**(java.lang.String name) Deletes all of the values associated with a name in the certPlist.
void	**removeValue**(java.lang.String name, java.lang.String value) Deletes one of the values associated with a name in the certPlist.
java.lang.String	**toString**() Returns the string representation of this object.

Class PkiCertificate

Package: com.ibm.pki.rpkiserv

```
public class PkiCertificate
extends java.lang.Object
```

Contains a certificate generated by PKI Services. The PkiCertificate class defines the following static constant integers to be used in determining the type (format) of certificate exported:

Integer	Meaning
PkiCertificate.PKIS_CERTIFICATE_TYPE_BASE64	Certificate is BASE64-encoded
PkiCertificate.PKIS_CERTIFICATE_TYPE_DER	Certificate is DER-encoded
PkiCertificate.PKIS_CERTIFICATE_TYPE_PKCS12	Certificate is DER-encoded PKCS #12
PkiCertificate.PKIS_CERTIFICATE_TYPE_PKCS7_CHAIN	Certificate is DER-encoded PKCS #7 chain

Method summary	
public java.lang.String	**getBase64Encoded**() Certificate in Base64 format. Use this method if the certificate type is PKIS_CERTIFICATE_TYPE_BASE64.
public byte[]	**getDerEncoded**() Certificate in DER format. Use this method if the certificate type is one of the DER types: PKIS_CERTIFICATE_TYPE_DER PKIS_CERTIFICATE_TYPE_DER_PKCS7_CHAIN PKIS_CERTIFICATE_TYPE_DER_PKCS12
public int	**getType**() Returns the certificate type.

Class QrecoverResultsList

Package: com.ibm.pki.rpkiserv

```
public class QrecoverResultsList
extends java.lang.Object
```

Contains QRECOVER results from R_Pkiserv.

Method summary	
java.lang.String	**getIssuerDn**() Returns the issuer's distinguished name.
java.lang.String	**getKeyId**() Returns the key Id.
java.lang.String	**getPassPhrase**() Returns the passphrase provided when the certificate request was made.
java.lang.String	**getSerialNum**() Returns the serial number.
java.lang.String	**getSubjectDn**() Returns the subject's distinguished name.
java.lang.String	**getValidityDates**() Returns the validity period in local time.

Class RpkiservException

Package: com.ibm.pki.rpkiserv

```
public class RpkiservException
extends java.lang.Exception
implements java.io.Serializable
```

Exception thrown by classes in the package com.ibm.pki.rpkiserv.

Constructor summary
RpkiservException() Constructor to create an empty RpkiservException object.
RpkiservException(java.lang.String exceptionText) Constructor to create an RpkiservException object with only exception text.

End of Programming Interface information

Part 4. Using PKI Services

This part explains how to use the PKI Services Web pages and utilities.

- Chapter 16, "Using the end-user Web pages," on page 307 shows the Web pages for the end user and explains how to perform tasks such as requesting a certificate, obtaining the certificate, and renewing or revoking a certificate.
- Chapter 17, "Using the administration Web pages," on page 335 shows the administration Web pages and explains how to process certificate requests and certificates.
- Chapter 18, "Using PKI Services utilities," on page 357 explains using the PKI Services utilities.

  createcrls A UNIX program that initiates the certificate revocation list (CRL) creation task immediately..

  iclview A UNIX program that displays the entries in the issued certificate list (ICL).

  pkiprereg A UNIX program that creates Simple Certificate Enrollment Protocol (SCEP) preregistration records

  postcerts A UNIX program that creates posting objects for existing certificates. The PKI Services daemon later posts the certificates to an LDAP server.

  TemplateTool A Java program that validates an XML certificate template file and converts it to a text CGI template file, and converts a text CGI template file to an XML template file.

  vosview A UNIX program that displays the entries contained in the object store (request database).

  vsam2db2 A UNIX program that converts data from the issued certificate list (ICL) and object store VSAM data sets into DB2 tables.
- Chapter 19, "Using the certificate management protocol (CMP) with PKI Services," on page 379 describes the support for CMP that PKI Services provides.

© Copyright IBM Corp. 2001, 2011

Chapter 16. Using the end-user Web pages

This topic describes how the end user can use the PKI Services Web pages.

Notes:

1. The PKI Services Web pages in this topic might differ slightly from those on the Web. If your installation customized the templates, the Web pages in this topic might differ greatly from those you view on the Web. Additionally, the pages might contain differences depending on the browser you are using. (This topic assumes you are using Internet Explorer.) If you need to see the exact content, view the pages on the Web.
2. If you are using Internet Explorer on a Microsoft Windows system, you might need to set up the Windows system and Internet Explorer to work with PKI Services. For information about how to do this, see "Using the PKI Services Web application with Internet Explorer on Windows systems" on page 585.

By default, the end user can perform the following tasks:

- Install a CA certificate into the browser.
- Request a new certificate.
- Pick up a previously requested certificate.
- Renew or revoke a previously issued browser certificate.
- Recover a certificate and private key, if PKI Services generated the keys for the certificate.
- Install the PKI Services ActiveX program needed to install a renewed certificate using the Internet Explorer browser.

Table 60 lists the types of certificates you can request:

Table 60. Types of certificates you can request

Type of certificate	Use
One-year PKI SSL browser certificate	End-user client authentication using SSL
One-year PKI S/MIME browser certificate	Browser-based e-mail encryption
One-year PKI generated key certificate	Generation of public and private keys by PKI Services
Two-year PKI browser certificate for authenticating to z/OS	End-user client authorization using SSL when logging onto z/OS
Two-year PKI Authenticode - code signing server certificate	Software signing
Two-year PKI Windows logon certificate	End-user client authentication for an Active Directory user logging in to a Windows desktop using a smart card
Five-year PKI SSL server certificate	SSL Web server certification
Five-year PKI IPSEC server (firewall) certificate	Firewall server identification and key exchange
Five-year PKI intermediate CA server certificate	Subordinate (non-self-signed) certificate-authority certification

© Copyright IBM Corp. 2001, 2011

Table 60. Types of certificates you can request (continued)

Type of certificate	Use
Five-year SCEP certificate	Creation of a preregistration record for certificate requestors. (Certificate requestors using Simple Certificate Enrollment Protocol (SCEP) must be preregistered.) Unlike other templates, this template is intended for administration use only.
n-year PKI browser certificate for extensions demonstration	Demonstration of all extensions supported by PKI Services
One-year SAF browser certificate	End-user client authentication where the security product (RACF, not PKI Services) is the certificate provider **Note:** The certificate generated by this template cannot be managed by the PKI Services administrator.
One-year SAF server certificate	Web server SSL certification where the security product (RACF, not PKI Services) is the certificate provider **Note:** The certificate generated by this template cannot be managed by the PKI Services administrator.

Special consideration for using SAF templates:

The templates that control processing of the SAF certificates listed in Table 60 on page 307 perform only a subset of the function available natively in RACF through the RACDCERT TSO command or the ISPF panels. They are provided to enable a Web interface for requesting certificates from RACF for browsers and off-platform servers. They are not intended to be a complete replacement for RACF certificate function.

Restriction: If you wish to generate a certificate for a server running on the local z/OS system (in other words, for a system using the RACF database where the signing certificate resides), do not use the "One-year SAF server certificate" template. Instead, use the RACDCERT TSO command or ISPF panels directly. Using the "One-year SAF server certificate" template might cause the loss of the private key if the authenticating user ID is not the same as the user ID specified when generating the certificate request in RACF.

Steps for accessing the end-user Web pages

Perform the following preliminary steps to access the PKI Services Web pages:

1. Get your organization's URL for accessing the PKI Services Web pages. Enter this URL in your browser. This takes you to the end-user "PKI Services Certificate Generation Application" Web page, shown in the following figure:

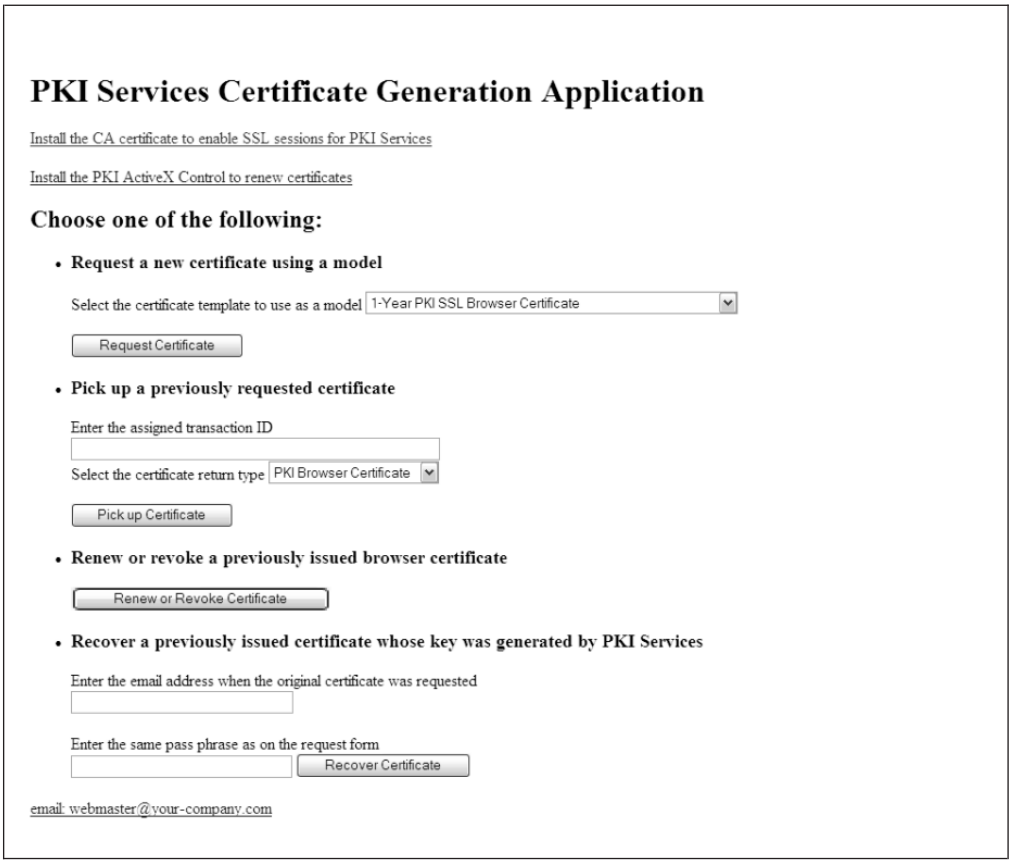


Figure 36. PKI Services end-user home page for certificate generation

---

2. If this is the first time you have accessed the forms on these Web pages, you must install the CA certificate into your browser. Click **Install the CA certificate** and follow the directions.

   The following instructions are a sample of the directions to follow for installing the CA certificate on Internet Explorer:

   a. After you click **Install the CA certificate**, a window labeled "File download" opens. Make sure that "Open this file from its current location" is selected (rather than "Save this file to disk"). Then click **OK**. Figure 37 on page 310 is an example of the window you might see, depending on the CA certificate you have installed.

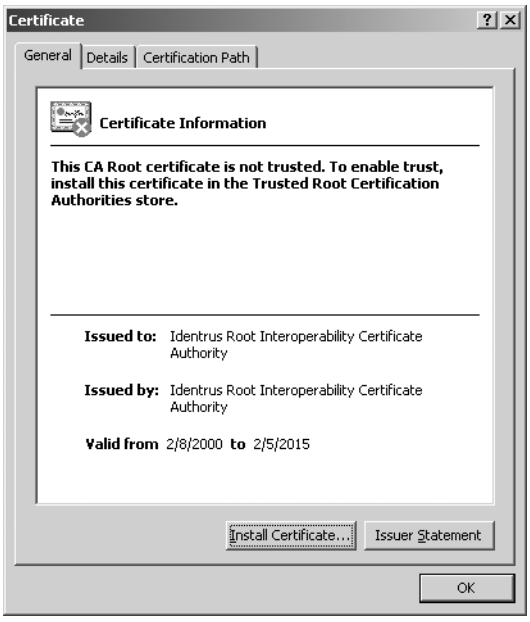


Figure 37. The certificate window for installing the CA certificate

b. Click **Install certificate**. (This initiates a series of windows in which you need to click **Next** and finally **Finish**, culminating in a window that says "The import was successful".)

Note: If you are using the Internet Explorer browser on a Microsoft Windows Vista system, you must explicitly select a store to place the certificate in. For more information, see "Installing the PKI Services CA certificate on a Microsoft Windows system" on page 592.

---

3. If you are using the Internet Explorer browser, at this time you can install the PKI Services ActiveX program that is used to install renewed certificates. For more information, see "Installing the PKI Services ActiveX program" on page 587. If you don't install it now, when you attempt to install a renewed certificate PKI Services checks whether you have either CAPICOM or the PKI Services ActiveX program installed. If neither is installed, PKI Services prompts you to install the ActiveX program.

   Note: **Install the PKI ActiveX Control to renew certificates** appears on the PKI Services home page only if you are using the Internet Explorer browser.

You are now ready to perform tasks, such as:

- Requesting a new certificate
- Picking up a previously requested certificate
- Renewing or revoking a previously issued browser certificate

Summary of fields

When you request certificates, you provide information for the fields in certificate request forms. The following table describes the fields in the end-user Web pages:

Table 61. Summary of fields in end-user Web pages

Field	Description
Certificate fields	
	Certificate fields related to Subject's Distinguished Name **Notes:** 1. The values for these fields are the relative distinguished names (RDNs) that are saved in the subject's distinguished name (DN) in the certificate. 2. For a server certificate, a base64-encoded PKCS #10 certificate request is required. If you specify one or more of these fields, the subject's distinguished name supplied in the PKCS #10 certificate request is ignored and only the fields you specify are in effect. For example, suppose that the subject's distinguished name specified in the PKCS #10 certificate request contains three RDNs - common name, organizational unit, and country. If you specify a value for organizational unit, you must also specify values for common name and country, even though you aren't changing them. If you don't, these two RDNs have no values.
Common name	Your name, such as John Smith. (You can use your first and last name, in that order.) This is a text field of up to 64 characters. For SSL servers, the common name is the server's fully qualified domain name, for example, `www.ibm.com`.
Country	The country where your organization is located. This is a 2-character text field.
Distinguished name qualifier	Specifies information to add to the subject distinguished name of an entry to make it unambiguous.
Domain component	One component of a domain name associated with the subject distinguished name. For example, the domain name www.ibm.com is represented by 3 components: www, ibm and com.
E-mail address	E-mail address with attribute EMAIL for the distinguished name. This is a text field of up to 64 characters.
Locality	The city or municipality where your organization is located, such as Pittsburgh or Paris. This is a text field of up to 64 characters.
Mail	E-mail address with attribute MAIL for the distinguished name. This is a text field of up to 64 characters. **Restriction:** If you specify a value for this parameter and for `Notification e-mail address`, the two values *must* be the same.
Organization	The legally registered name (or trademark name, for example, IBM) of your organization. This is a text field of up to 64 characters.
Organizational unit	The name of your division or department. (There can be more than one organizational unit field on a request form. For example, one could be for your department and another for your division.) This is a text field of up to 64 characters.
Postal code	Your postal code or zip code. This is a text field of up to 64 characters.
Serial number	Serial number of the subject device. This is a text field of up to 64 characters.
State or Province	The state or province where your organization is located. Your registration policies determine whether you spell out the full name of the state or province or use an abbreviation. This is a text field of up to 64 characters.
Street	Your street address. This is a text field of up to 64 characters.
Title	Your job title. This is a text field of up to 64 characters.

Table 61. Summary of fields in end-user Web pages (continued)

Field	Description
Unstructured address	The unstructured address of the subject device.
Unstructured name	The unstructured name of the subject device.
User ID	The system login name associated with the subject distinguished name.
	Certificate fields related to validity period
Not after (date)	A number of days, added to the current date, after which the certificate expires. By default, you can select either one year or two years for the time at which the certificate expires.
Not before (date)	A number of days, added to the current date (by default, you can select either `0` or `30`), before which the certificate is not valid.
	Certificate fields related to extensions
Alternate domain name	Domain name for alternate name. This is the host name of the machine where a certificate will be installed. This is a text field of up to 100 characters. **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
Alternate e-mail address	E-mail address for alternate name, including the `@` character and any periods (.). This is a text field of up to 100 characters. **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
Alternate IP address	The IP address for the alternate name. This unique IP address specifies the location of each device or workstation on the Internet. PKI Services supports both IP version 4 and IP version 6 addresses. The IP address is a text field of up to 45 characters: • For IP version 4, the IP address is in dotted decimal format; for example, `9.67.97.103`. • For IP version 6, the IP address is divided into eight 16-bit hexadecimal blocks separated by colons. Leading zeros in each 16-bit field are optional, and successive fields of zeros can be represented by double colons, but only once; for example `1:2::3:4` is equivalent to `0001:0002:0000:0000:0000:0000:0003:0004`. • In a mixed IP version 4 and IP version 6 environment, the IP address can be expressed in the format *x:x:x:x:x:x:d.d.d.d*, where the *x* values are the hexadecimal values of the six high-order 16-bit pieces of the address, and the *d* values are the decimal values of the four low-order 8-bit pieces of the address in standard IP version 4 representation; for example, `0:0:0:0:0:ABCD:1.2.3.4`, or the equivalent value `::ABCD:1.2.3.4` **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
Alternate other name	Additional identifier for the alternate name. See your PKI Services administrator for information about this field.
Alternate uniform resource identifier (URI)	Uniform resource identifier for the alternate name. This is a name or address referring to an Internet resource; a URL is one kind of uniform resource identifier. This is a text field of up to 100 characters. **Note:** The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
Extended key usage	This indicates the intended purpose of the certificate. Possible values are: **clientauth** Client side authentication **codesigning** Code signing **emailprotection** Email protection **mssmartcardlogon** Smart card logon for Microsoft Windows users **ocspsigning** OCSP response signing **serverauth** Server side authentication **timestamping** Digital timestamping

Table 61. Summary of fields in end-user Web pages (continued)

Field	Description
HostIdMapping	This is the user ID for authorization purposes in the format: subject-id@host-name **Example:** DSmith@ibm.com This is a text field of up to 100 characters.
Key usage	The intended purpose of the certificate. Each possible value is shown here with its intended purpose and possible PKIX bits:

KeyUsage value	Intended purpose	PKIX bits
certsign	Certificate and CRL signing	KeyCertSign and cRLSign
crlsign	CRL signing	cRLSign
dataencrypt, **dataencipherment**, or **dataenciph**	Data encryption	dataEncipherment
digitalsig or **digitalsignature**	Authentication	digitalSignature
docsign or **nonrepudiation**	Document signing	nonRepudiation
handshake	Protocol handshaking (for example, SSL)	digitalSignature and keyEncipherment
keyagree or **keyagreement**	Key agreement	keyAgreement
keycertsign	Certificate signing	keyCertSign
keyencrypt, **keyencipherment**, or **keyenciph**	Key transport	keyEncipherment

	Base64-encoded PKCS #10 certificate request

Base64-encoded PKCS #10 certificate request	(This is for server or device enrollment only.) You create a certificate request on behalf of another server (which could be a z/OS server or other type of server) or device for which you are requesting a certificate. You use software specific to that server to generate the PKCS #10 request before going to the PKI Services Web site. Save the request in a file. Then open the file in a text editor such as Windows Notepad and copy and paste the contents into the text box on the enrollment form. A text area of 70 columns and 12 rows is allocated for this certificate request. Here is an example of the certificate request:
	`-----BEGIN NEW CERTIFICATE REQUEST-----`
	`MIIBiDCB8gIBADAZMRcwFQYDVQQDEw5Kb2huIFEuIFB1YmxpYzCBnzANBgkqhkiG`
	`9w0BAQEFAAOBjQAwgYkCgYEAsCT1cJHAGPqi6OjAyL+xNbt8z5ngmvqO2V0O3oYu`
	`/mEnQtRM96e+2jbmDCRo5tWVklG4OYf9ZVB5biURMJFLztfa4AVdEVtun8DH2pwc`
	`wiNIZZcC1Zym5adurUmyDk64PgiiIPMQS/t0ttG4c5U8uWSK0b1J4V4f7ps+t1aG`
	`t+cCAwEAAaAwMC4GCSqGSIb3DQEJDjEhMB8wHQYDVR0OBBYEFAlKTovBBvnFqDAO`
	`1oIhtRinwRC9MA0GCSqGSIb3DQEBBQUAA4GBAIbCVpwYvppIX3HHmpKZPNY8Snsz`
	`AJrDsgAEH51WOIRGywhqKcLLxa9htoQai6cdc8RpFVTwk6UfdCOGxMn4aFb34Tk3`
	`5WYdzOiHXg8MhHiB3EruwdWs+S7Fv3JhU3FLwU6lFLfAjbVi+35iEWQymOR6mE5W`
	`CathprmGfKRsDE5E`
	`-----END NEW CERTIFICATE REQUEST-----`
	
	For a sample of the enrollment form showing the text box for a PKCS #10 request, see Figure 39 on page 318.

PKI Services internal use fields	
Challenge passphrase	This is the passphrase you entered when requesting a certificate. You type the same passphrase, exactly as you typed it on the request form. This is a case-sensitive text field of up to 32 characters.
KeySize	The size of the key pair (public key and private key) that you want PKI Services to generate for you.

Table 61. Summary of fields in end-user Web pages (continued)

Field	Description
Label	The label assigned to the requested certificate. This is a text field of up to 32 characters. This field applies only to SAF certificates.
Notification e-mail address	E-mail address for notification purposes. This is a text field of up to 64 characters. **Note:** If you specify a value for this parameter and for `Mail`, the two values **must** be the same.
Passphrase	You decide this value when requesting a certificate (and must later supply this value when retrieving the certificate). You enter and then reenter this when requesting a certificate. This is a case-sensitive text field of up to 32 characters. (There is no minimum number of characters, and you can use any characters, but alphanumeric characters (`A–Z`, `a–z`, and `0–9`) are suggested.
Requestor's name	Your name (for tracking purposes). This can be in any format, for example, John Smith or John. J. Smith. This is a text field of up to 32 characters. **Note:** For a PKI generated key certificate, the requestor name needs to be in the form of an e-mail address.
Transaction ID	This is assigned after you request your certificate. When it is displayed, you need to record this number. This is a text field of up to 56 characters.
Browser-specific fields	
Cryptographic service provider	(This is for the Internet Explorer browser only.) The cryptographic service provider to generate your public/private key pair. You select a value from the drop-down list. Larger keys are more secure, but they also increase the time that is needed for connecting to a secure session.
Key protection	(This is for the Internet Explorer browser only.) This asks if you want to enable private key protection. (The drop-down choices are **Yes** and **No**.)
Key size	(This is for Mozilla-based browsers only.) This is the key size for your public/private key pair. Select a value from the drop-down list. Larger keys are more secure, but they also increase the time needed for connecting to a secure session.

Steps for requesting a new certificate

To request a new certificate, first go to the PKI Services home page. (See Figure 36 on page 309.)

Perform the following steps to request a new certificate:

1. Click the down arrow to the right of the field beside `Request a new certificate using a model`. This displays a list of certificate templates from which you can select.

   For SCEP preregistration: Do not follow these steps to request a SCEP (preregistration) certificate template. Instead, go to "Steps for preregistering an SCEP client" on page 331.

   The following list shows the certificate templates that PKI Services provides by default. This list might differ from the certificate templates your installation provides because your installation can customize the certificate templates and Web pages.

   - One-year SAF server certificate
   - One-year SAF browser certificate
   - One-year PKI SSL browser certificate (See Figure 38 on page 316 to see a sample of this Web page.)
   - One-year PKI SSL S/MIME browser certificate
   - One-year PKI generated key certificate

- Two-year PKI browser certificate for authenticating to z/OS
- Two-year PKI Authenticode - code signing server certificate
- Two-year PKI Windows logon certificate
- Five-year PKI SSL server certificate
- *n*-year PKI browser certificate for extensions demonstration
- Five-year SCEP certificate - Preregistration
- Five-year PKI IPSEC server (firewall) certificate
- Five-year PKI intermediate CA server certificate

---

2. Click one of the items in the list. The drop-down list then collapses so that only the certificate you selected appears in the field and is highlighted.

---

3. Click **Request certificate**. A form where you fill in information is displayed.

   Note: You might need to click through some additional panels specific to your browser (for example, clicking **Next** on a Mozilla-based browser or answering `Do you want to proceed?` on Internet Explorer) before the certificate request form appears.

---

4. Fill in the necessary information in the certificate request form.

   The form that appears depends on the certificate you are requesting and, in some instances, the fields that appear on the form depend on the browser you are using. **Example:** If you request a one-year SSL browser certificate, the form shown in Figure 38 on page 316 appears.

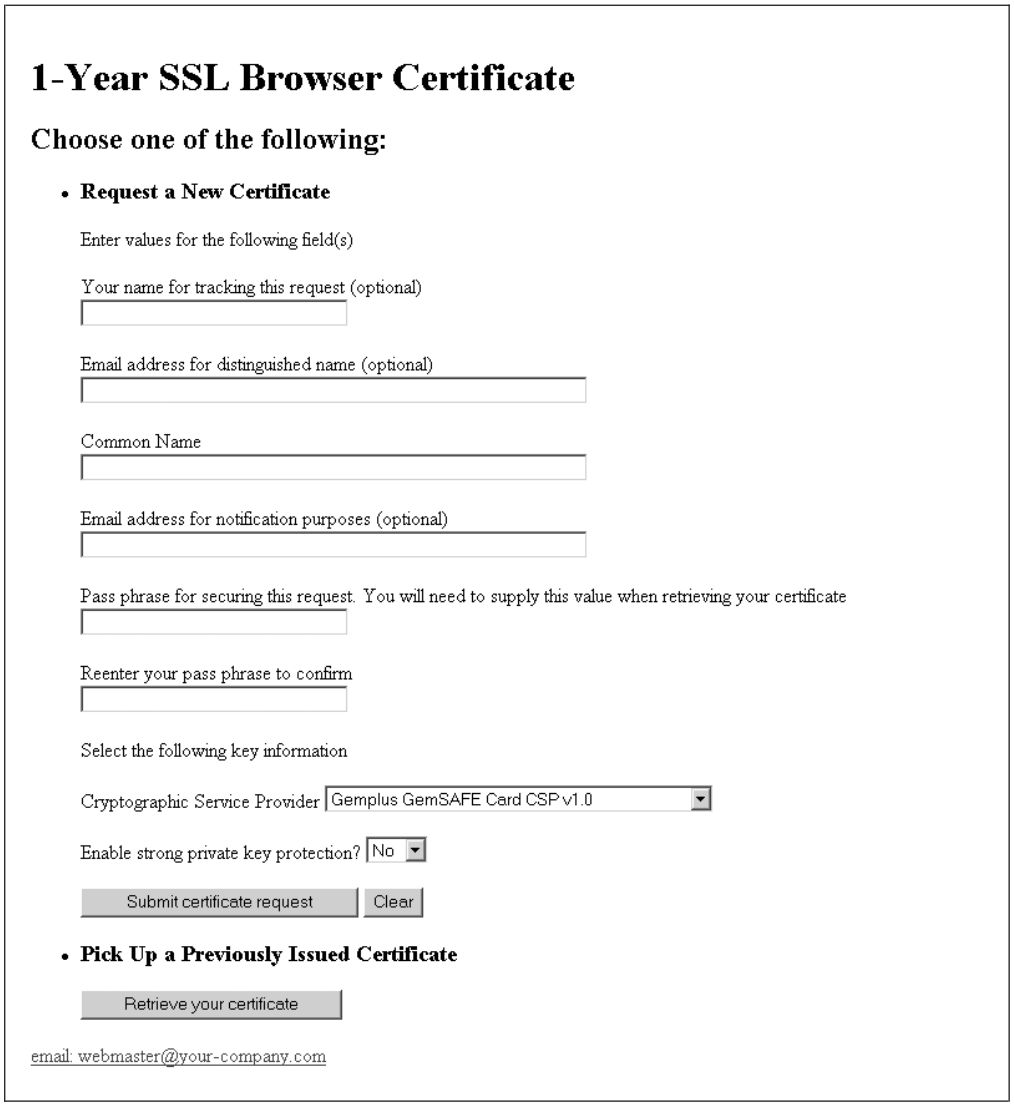


Figure 38. One-year SSL browser certificate request form

Note: In the case of the one-year SSL browser certificate, fill in your common name. (See Table 61 on page 311 for descriptions of fields.) If you are using a Mozilla-based browser, select a key size from a drop-down list. Alternately, if you are using Internet Explorer, click the drop-down lists to select your cryptographic service provider and to specify whether to use strong private key protection.

---

5. If you are requesting a server or device certificate, you need to supply a base64-encoded PKCS #10 certificate request. Use software specific to that server to generate the PKCS #10 request before going to the PKI Web site. Paste the request into the Web page as shown in Figure 39 on page 318.

   For example, you could use the RACDCERT command to generate the PKCS #10 request. Assume that the server has the distinguished name OU=Inventory,O=XYZZY,C=US and a domain name xyzzy.com. This server runs on z/OS with the user ID INVSERV. First, generate a self-signed certificate for the server and assign the label "Inventory Server" to the certificate. The certificate is associated with the user ID that is associated with the server (INVSERV).

```
RACDCERT ID(INVSERV)
         GENCERT
         SUBJECTSDN(CN('xyzzy.com')
                    OU('Inventory')
                    O('XYZZY')
                    C('US'))
         WITHLABEL('Inventory Server')
```

Next, generate a PKCS #10 Base64-encoded certificate request based on the certificate you just created, and write the request to a data set.

```
RACDCERT ID(INVSERV)
         GENREQ(LABEL('Inventory Server'))
         DSN('WAIC.INVSERV.GENREQ')
```

Copy the PKCS #10 request from the data set WAIC.INVSERV.GENREQ and paste it into the field **Base64 encoded PKCS#10 certificate request**.

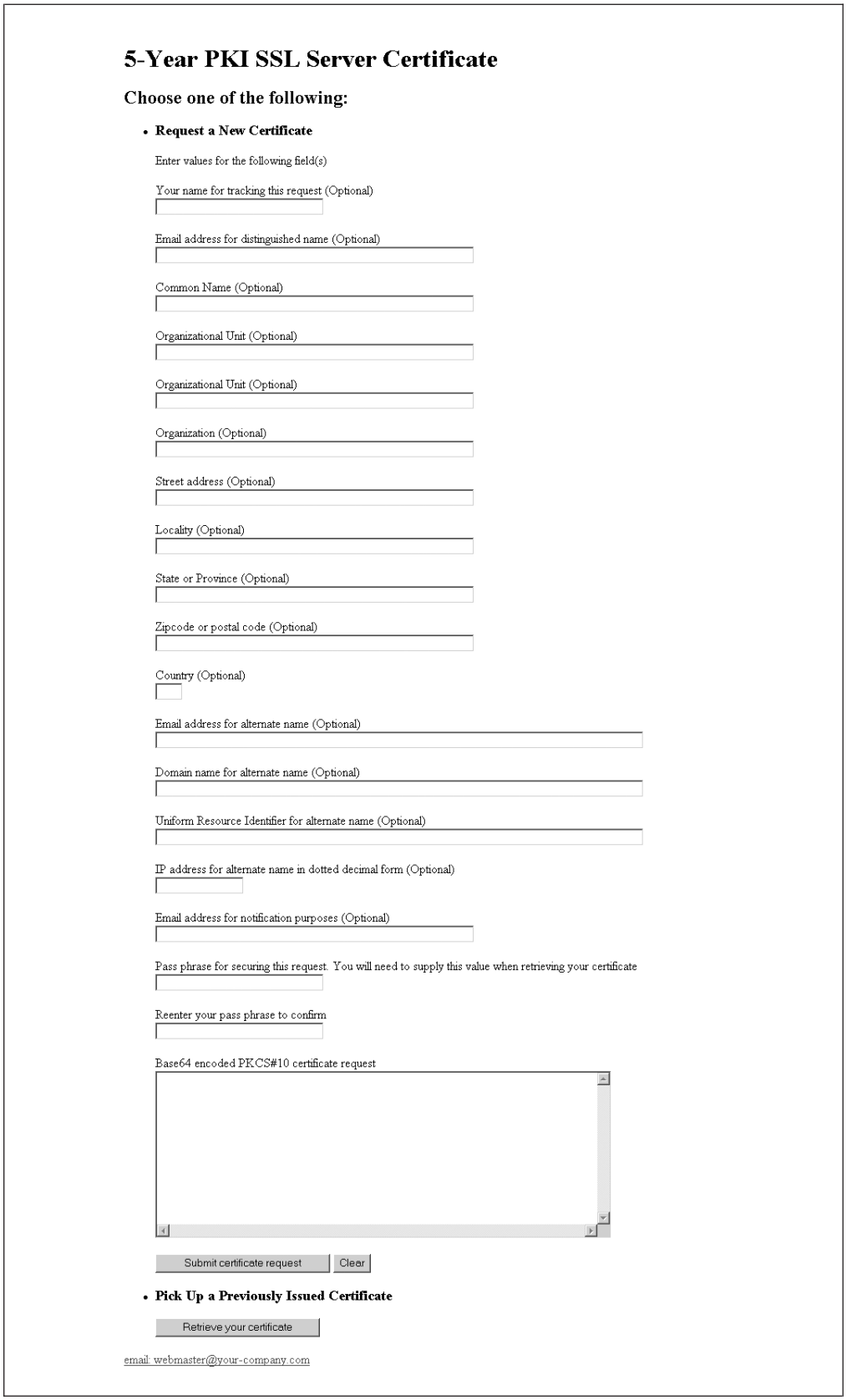


Figure 39. Supplying the PKCS #10 certificate request for a server or device certificate

For server certificates where a base64-encoded PKCS #10 certificate request is supplied, specify one or more of the fields related to the subject's distinguished name only if you wish to change the distinguished name supplied in the PKCS #10 certificate request. If you change one of these fields, the subject's distinguished name specified in the PKCS #10 certificate request is ignored and you must respecify the entire distinguished name (all fields). For a list of the fields related to the subject's distinguished name, see Table 61 on page 311.

---

6. Fill in the passphrase on the certificate request form (twice). This is a value known only to you. Pick a value that you can easily remember because you will be challenged to supply the same passphrase when you pick up your certificate. Do not use a sensitive value such as your ATM pin or login password.

---

7. Fill in any optional information as desired. When you are satisfied with the information you have entered, click **Submit certificate request**. If the request is successful, the results depend on the type of certificate you requested.
   - For all certificate types except one-year PKI generated key certificates, you see a page like the one shown in Figure 40, which tells you your transaction ID.



Figure 40. Successful request displays transaction ID

   a. Make a note of the transaction ID. (You can copy and paste the transaction ID to a file so that you have it for future reference, or you can write it in the box below. The reason for keeping a record of the transaction ID is that, depending on how you go to the Web page to retrieve your certificate (see Figure 41 on page 320), you might have to fill in the transaction ID on that Web page.)

Transaction ID:	

   b. Click **Continue**. This displays the following Web page:

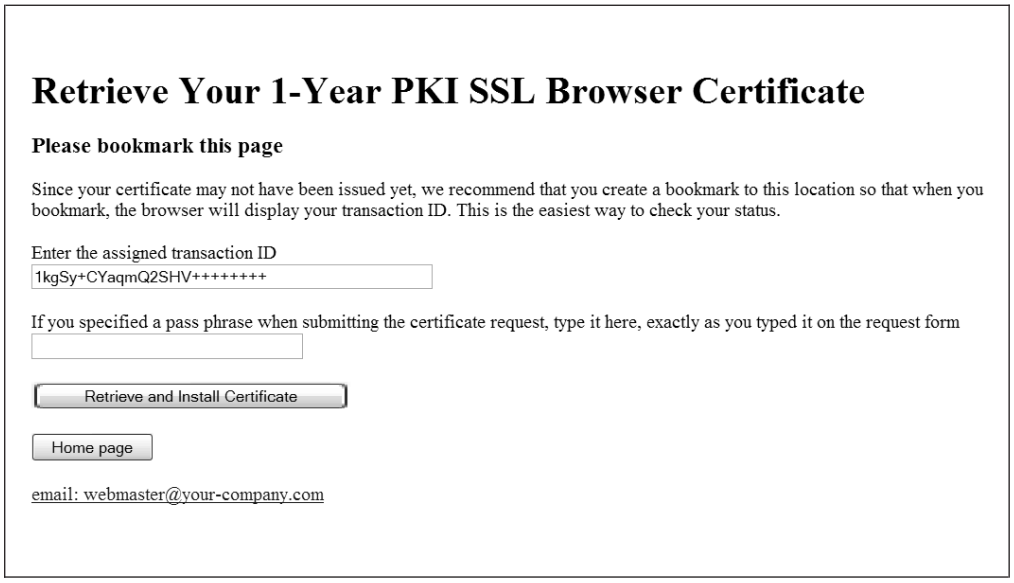


Figure 41. Web page to retrieve your certificate

c. Bookmark this Web page.

   Notes:

   1) After you submit the request for a certificate, your PKI Services administrator might need to approve the request before you can pick up your certificate. The amount of time that this takes can vary from a few minutes to a few days, depending on your installation. You bookmark this Web page so that you can return to it at a later time.
   2) If your installation has enabled e-mail notification and you supplied a valid e-mail address when submitting your certificate request, then you will receive an e-mail message when your certificate is ready for pick-up or if PKI Services rejects your certificate request.

d. From this Web page, you can start the steps to retrieve your certificate (see "Steps for retrieving your certificate from the bookmarked Web page" on page 321) or you can return to the PKI Services home page (by clicking **Home**).

- For a one-year PKI generated key certificate, you see a page like the one shown in Figure 42



Figure 42. Successful request for a one-year PKI generated key certificate

Unlike other types of certificates, this page does not show you the transaction ID for your certificate. Instead, PKI Services sends an e-mail to the address you specified in the request. The e-mail contains a link to the certificate.

Retrieving your certificate

For most certificate types, you can retrieve your certificate:

- From the Web page you bookmarked in Step 7c on page 320. (This Web page contains your transaction ID, so you do not have to enter it.) (See "Steps for retrieving your certificate from the bookmarked Web page.")
- From the PKI Services home page. (See Figure 36 on page 309 and "Steps for retrieving your certificate from the PKI Services home page" on page 322.)

For a one-year PKI generated key certificate, you receive an e-mail to notify you when your certificate is ready for retrieval. The e-mail contains a link to the certificate. (See "Steps for retrieving a PKI generated key certificate" on page 323.)

If your company has enabled e-mail notification for non-SAF certificates and you supplied a valid e-mail address when submitting your certificate request, you will receive an e-mail to notify you when your certificate is ready for retrieval (or if your certificate request has been rejected).

Note: When a certificate is retrieved, it can be of different formats:

- If the keys are not generated by PKI Services, the returned format can be a single X.509 certificate or a chain of certificates, depending on the authority of the surrogate ID that does the EXPORT.
- If the keys are generated by PKI Services, the returned format is always a single certificate and the private key in a PKCS #12 package.

Steps for retrieving your certificate from the bookmarked Web page

Perform the following steps to retrieve your certificate from the bookmarked Web page:

1. Go to the bookmarked Web page. (See Figure 41 on page 320.)

   _____

2. If you entered a passphrase when requesting your certificate, enter the passphrase.

   _____

3. Click **Retrieve and install certificate**. If you are using a Mozilla-based browser, go to Step 5. If you are using Internet Explorer and the retrieval of a certificate is successful, this displays the Web page shown in Figure 43. (This is for a browser certificate. For a server certificate, Figure 44 on page 322 shows an example of the Web page.)

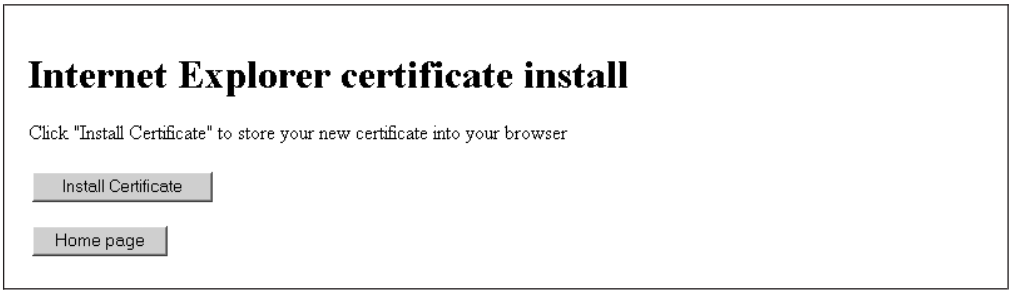


Figure 43. Browser certificate installation Web page



Figure 44. Server certificate installation Web page

---

4. Click **Install certificate**. If the certificate installs successfully, you get a popup window that says `Your new certificate installed successfully.`

---

5. Check that your certificate installed correctly:
   - For a Mozilla-based browser, click **Security**, then Certificates → Yours. Your certificate should appear in the list. Select it and click Verify.
   - For Internet Explorer, click Tools → Internet Options, then Content, **Certificates**. Your certificate should appear in the Personal list. Click Advanced to see additional information.

Steps for retrieving your certificate from the PKI Services home page

Before you begin: To retrieve your certificate from the PKI Services home page, you must first know your transaction ID. You should have recorded this when your certificate request was successful. (See Figure 40 on page 319.)

Perform the following steps to retrieve your certificate from the PKI Services home page:

1. Enter your transaction ID and select the certificate type using the drop-down. Then click **Pick up certificate** on the PKI Services home page. (See Figure 36 on page 309.) This displays the Web page that Figure 41 on page 320 shows.

---

2. Enter your passphrase (this is the challenge passphrase) if you specified one when requesting your certificate.

---

3. Click **Retrieve and install certificate**. If you are using a Mozilla-based browser, go to Step 5. If you are using Internet Explorer and the retrieval of the certificate is successful, this displays the Web page that Figure 43 on page 321 shows. (This is for a browser certificate. For a server certificate, Figure 44 shows an example of the Web page.)

---

4. Click **Install certificate**. If the certificate installs successfully, you get a popup window that says `Your new certificate installed successfully.`

5. Check that your certificate installed correctly:
   - For a Mozilla-based browser, click **Security**, then Certificates → Yours. Your certificate should appear in the list. Select it and click Verify.
   - For Internet Explorer, Click Tools → Internet Options, then Content, Certificates. Your certificate should appear in the Personal list. Click Advanced to see additional information.

Steps for retrieving a PKI generated key certificate

Before you begin: You need to have received an e-mail telling you that your certificate is ready to be picked up.

Perform the following steps to retrieve your PKI generated key certificate.

1. The e-mail telling you that your certificate is ready picked up looks similar to the one shown in Figure 45. You have two alternatives:
   - Clink the link for your certificate in the e-mail.
   - Copy the transaction ID from the e-mail. Go to the PKI Services home page (see Figure 36 on page 309). Paste the transaction ID into the field labeled "Enter the assigned transaction ID" and click **Pick up certificate**.

```
Attention - Please do not reply to this message as it was automatically
sent by a service machine.

Dear lewallen@us.ibm.com,

Thank you for choosing dime-o-cert PKI. The certificate you requested for
subject CN=Rocky,OU=Class 1 Internet Certificate CA,O=The Firm is now ready for pickup.

Please visit:
http://alps4027.pok.ibm.com/Customers/ssl-cgi-bin/caretrieve.rexx?TransactionId=1kkLpBZvA%2
Bq%2F2SHV%2B%2B%2B%2B%2B%2B%2B%2B&Template=PKI+Key+Certificate
 to retrieve your certificate.

If that link does not work, try to go to
http://alps4027.pok.ibm.com/Customers/public-cgi/camain.rexx?
And enter the transaction ID listed below:
1kkLpBZvA+q/2SHV++++++++

You will need to input your passphrase that you entered when you submitted the request.
```

Figure 45. E-mail notification that your PKI generated key certificate is ready for pickup

2. The Web page shown in Figure 46 on page 324 is displayed. Note that the transaction ID is filled in. Enter the passphrase you entered when you submitted the certificate request, and click **Retrieve Certificate**.



Figure 46. Web page for retrieving a PKI generated key certificate

3. A window opens asking whether you want to open or save the PKCS #12 package containing the certificate and private key. This window is shown in Figure 47.

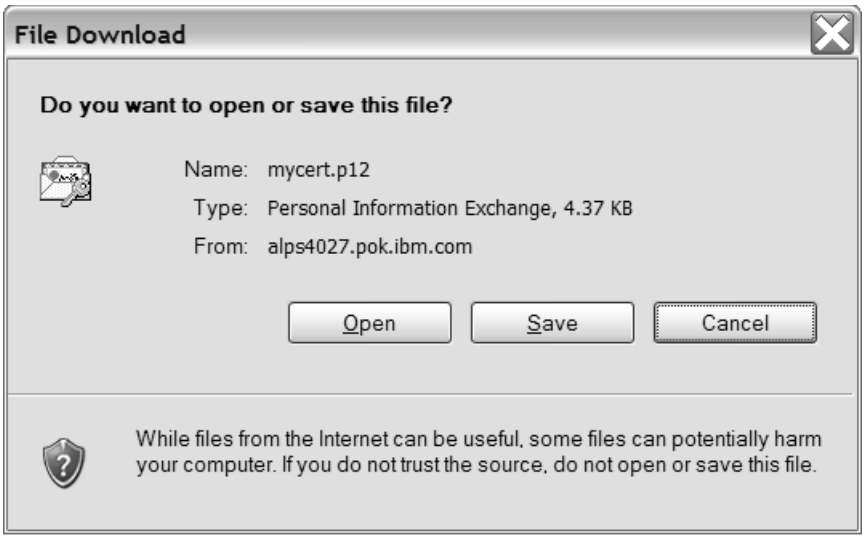


Figure 47. Window asking whether to open or save the PKCS #12 package

Click **Open** to invoke the Certificate Import Wizard to copy the certificate to a certificate store. Click **Save** to save the PKCS #12 package in a file.

When you are done, you have retrieved your PKI generated key certificate and private key.

Steps for renewing a certificate

Note: For a PKI generated key certificate, if you have changed your e-mail address since you originally requested the certificate, and the administrator has changed your e-mail in the requestor field in the certificate, you cannot renew the certificate.

Perform the following steps to renew a certificate:

1. On the PKI Services home page (see Figure 36 on page 309), click **Renew or revoke certificate**. This displays a popup window with a list of certificates, such as the following figure shows:



Figure 48. Popup window listing certificates

---

2. The popup window might list more than one certificate. The certificates are listed by nickname in the order they are installed in the browser. Therefore, you might not be able to identify the PKI Services certificate you want to renew. Highlight the entry you think is the right one and click **OK**. If the certificate you selected is one that PKI Services issued and it is not expired or revoked, Figure 49 on page 326 shows an example of the Web page you might see, depending on the certificate you are renewing:

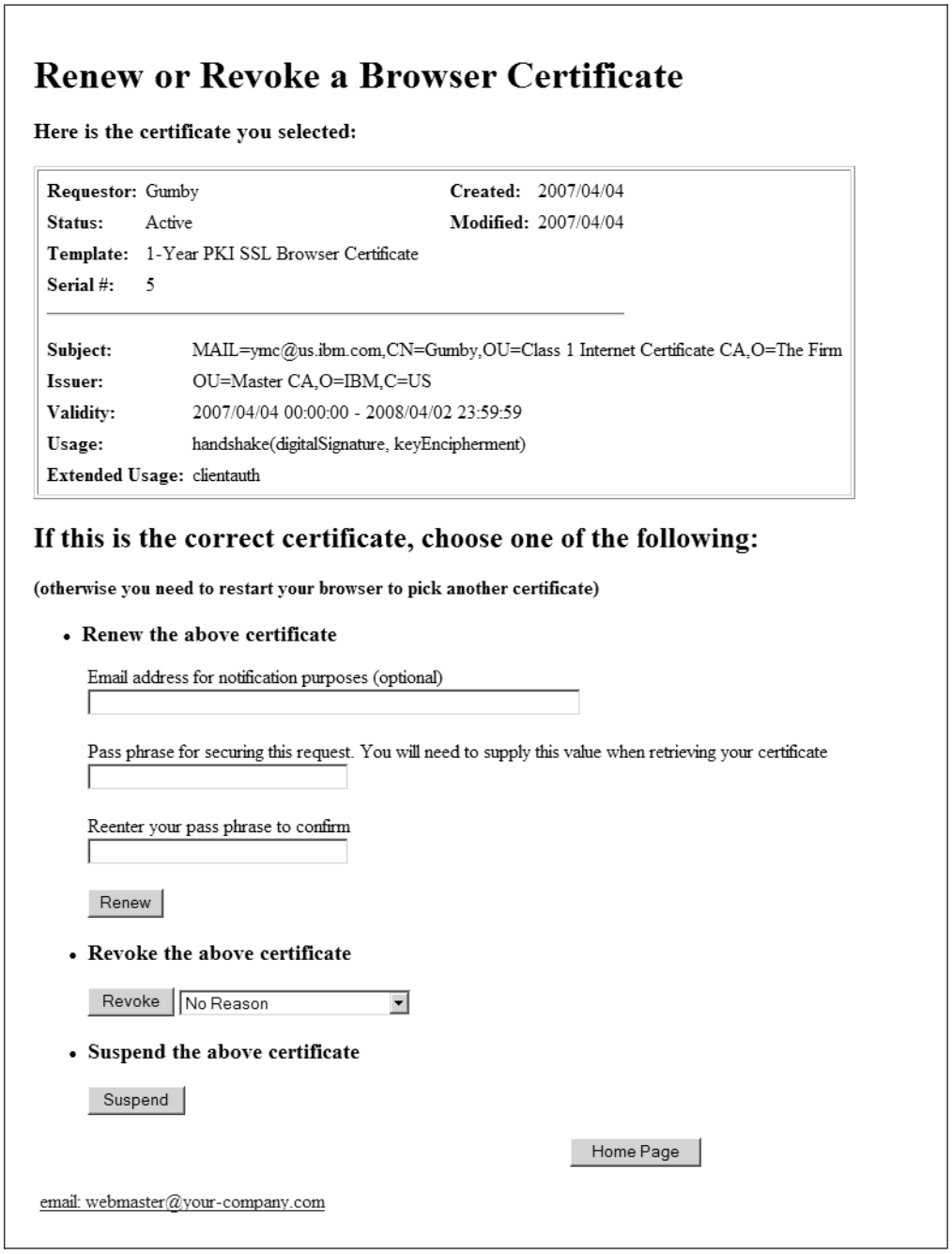


Figure 49. Renew or revoke a certificate Web page

Notes:

a. If this is not the PKI Services certificate you want to renew, you need to close your browser (because the browser caches information) before again clicking **Renew or revoke certificate** as in Step 1 on page 325.

b. If the certificate has the MAIL attribute in the subject's distinguished name, the value of NotifyEmail must match it.

---

3. Under the `Renew the above certificate` section, enter your passphrase in the two fields requesting it.

---

4. Click **Renew**.

---

5. If you are using Internet Explorer, and you don't have either CAPICOM or the appropriate PKI Services ActiveX program installed, PKI Services prompts you to install the ActiveX program. Follow the directions in the prompt. For more information, see "Installing the PKI Services ActiveX program" on page 587.
6. If the renewal request is successful, this displays a Web page that says `Request submitted successfully` and displays the transaction ID. Click **Continue** on this Web page.

---

7. This takes you the Web page from which you retrieve your certificate. (See Figure 41 on page 320 for an example of this Web page and "Steps for retrieving your certificate from the bookmarked Web page" on page 321 for the directions to follow.)

Steps for revoking or suspending a certificate

Revoking or suspending a certificate means that you cannot continue to use the certificate. You might want to permanently *revoke* your certificate if you suspect your private key has been compromised. You might want to *suspend* (temporarily revoke) your certificate if you want to discontinue using it for a period of time (known as the suspension *grace period*).

If you suspend your certificate, the PKI administrator can *resume* (reactivate) the certificate, or permanently revoke it, if the certificate has not yet expired and the grace period has not elapsed. If the grace period has elapsed, the certificate is permanently revoked the next time the certificate revocation lists (CRLs) are issued.

Perform the following steps to revoke or suspend a certificate:

1. On the PKI Services home page (see Figure 36 on page 309), click **Renew or revoke certificate**. This displays a popup window with a list of certificates, as in Figure 48 on page 325.

---

2. The popup window might list more than one certificate. The certificates are listed by nickname in the order they are installed in the browser. You might not be able to identify the PKI Services certificate you want to revoke or suspend. Highlight the entry you think is the right one and click **OK**. If the certificate you selected is one that PKI Services issued and it is not expired or revoked, this displays the "Renew or revoke a browser certificate" Web page. (See "Steps for renewing a certificate" on page 324.)

   Note: If this is not the PKI Services certificate you want to revoke or suspend, you need to close your browser before again clicking **Renew or revoke certificate** as in Step 1.

---

3. Make sure the certificate you want is the one described at the top of the Web page. Click **Revoke** or **Suspend**. Note that when you revoke a certificate, you can click the drop-down list (of reasons) to select a reason if you wish.

---

4. This displays a Web page that says `Request submitted successfully`. You can click **Home page** to return to the PKI Services home page.

When you are done: You will no longer be able to use the certificate. If you suspended the certificate, contact your PKI administrator when you wish to have it resumed.

Recovering a certificate whose keys were generated by PKI Services

If you request a PKI generated key certificate, PKI Services generates the public and private keys for the certificate and stores the certificate and keys in the ICSF token database (TKDS). If you lose the original certificate, PKI Services can recover the stored certificate and private key and return them to you in a PKCS #12 package.

Steps for recovering a certificate whose keys were generated by PKI Services

Before you begin: You need to know the e-mail address you used when you requested the certificate. You also should know the passphrase you entered on the certificate request. However, if you have forgotten the passphrase, and your company has implemented security questions, and you answered the security questions when you requested the certificate, you can provide those answers instead of the passphrase.

Perform the following steps to recover a certificate whose keys were generated by PKI Services.

1. On the PKI Services home page (see Figure 36 on page 309), take one of the following actions:
   a. If you remember the passphrase you used when you requested the certificate that you want to recover, enter the passphrase and the e-mail address you used when you requested the certificate and click **Recover Certificate**.
   b. If you have forgotten the passphrase you used, click **Click here if you forget the pass phrase**. A Web page similar to the one shown in Figure 50 on page 329 is displayed. Enter the e-mail address you used when you requested the certificate and the answers to the security questions, and click **Recover Certificate**.

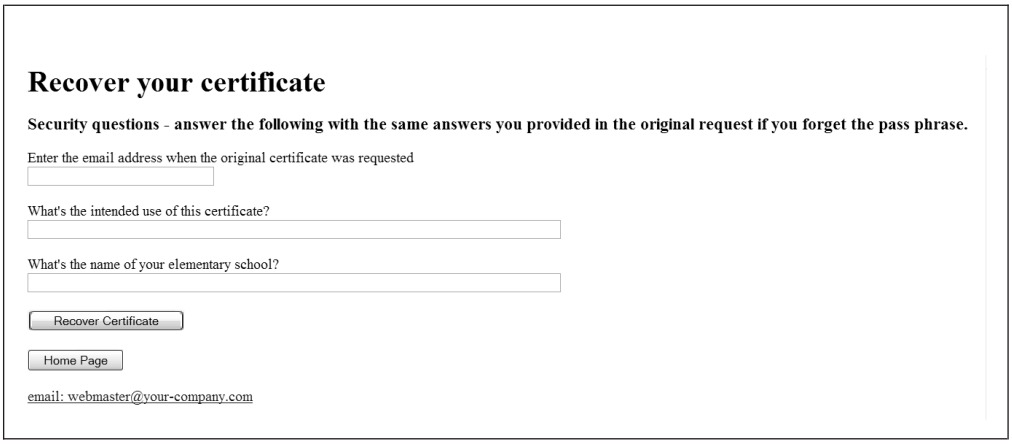


Figure 50. Web page requesting answers to security questions when you have forgotten the passphrase

2. The Web page shown in Figure 51 is displayed listing the certificates that you can recover, and an e-mail with links to those certificates is sent to your e-mail address.

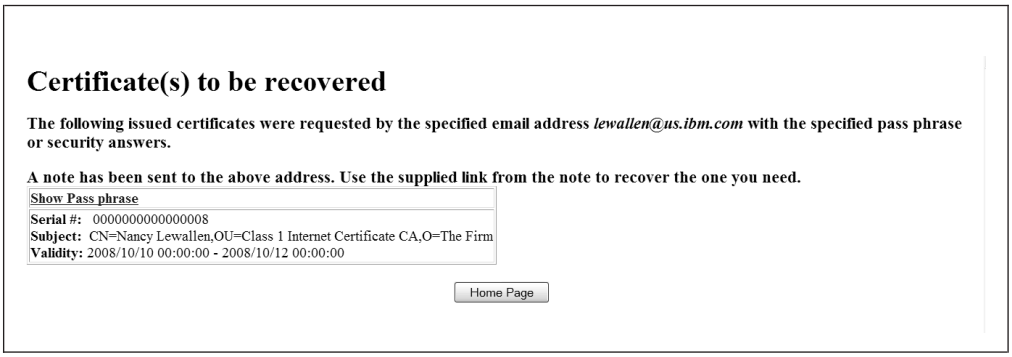


Figure 51. Web page listing certificates that can be recovered

Click **Show Pass phrase** to find out the pass phrase for the certificate you want to recover, if you have forgotten it. You will need it to recover the certificate. The passphrase is displayed as shown in Figure 52. Click **Hide Pass phrase** to hide the passphrase again.



Figure 52. Web page showing the passphrase for a certificate to be recovered

3. Open the e-mail you were sent. Figure 53 shows a sample e-mail that lists one certificate eligible for recovery. Click on the link for the certificate that you want to recover.

```
Attention - Please do not reply to this message as it was automatically sent by
a service machine.

Dear lewallen@us.ibm.com,

Here is a list of certificate(s) that satisfy your searching criteria for
recovery:
0000000000000008 : CN=Nancy Lewallen,OU=Class 1 Internet Certificate CA,O=The Firm

Please choose the certificate you want and visit the corresponding link to
retrieve it (you can identify the certificate by the serial number from the
part of the link between '?' and '&')

https://www.dimeocert.com/Customers/ssl-cgi-in/caretrieve.rexx?SerialNo=0000000000000008
&KeyID=2FBE1B1AC36F63C712AB6F5B829681549FD2095E

You will need to input your pass phrase that you entered when you submitted the
request.
```

Figure 53. Sample e-mail that lists certificates that can be recovered

4. The link takes you to the Web page shown in Figure 54.

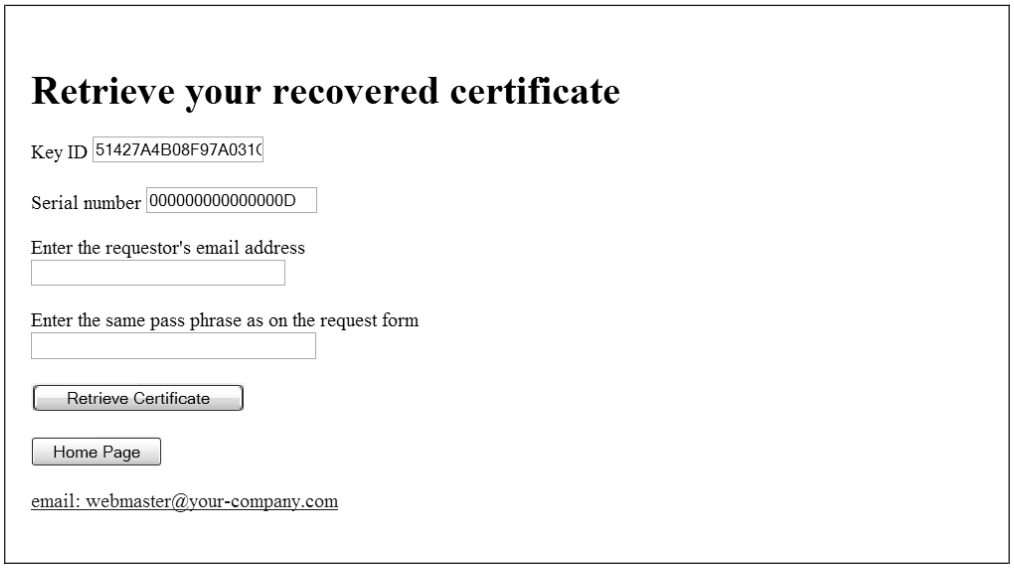


Figure 54. Web page to retrieve a recovered certificate

Fill in the e-mail address and passphrase you used on the original certificate request, and click **Retrieve Certificate**.

5. A window opens asking whether you want to open or save the PKCS #12 package containing the certificate and private key. This window is shown in Figure 55 on page 331.



Figure 55. Window asking whether to open or save the PKCS #12 package

Click **Open** to invoke the Certificate Import Wizard to copy the certificate to a certificate store. Click **Save** to save the PKCS #12 package in a file.

When you are done, you have recovered your PKI generated key certificate.

Steps for preregistering an SCEP client

These steps are performed by the PKI administrator using the end-user Web pages. To preregister an SCEP client, first go to the PKI Services home page. (See Figure 36 on page 309.)

Perform the following steps to complete a preregistration request and preregister an SCEP client:

1. Click the down arrow to the right of the field beside `Request a new certificate using a model`. (This displays a list of certificate templates from which you can select.) Select `5-Year SCEP Certificate - Preregistration` from the list. The drop-down list then collapses so that only the preregistration template appears in the field and is highlighted.

   ---

2. Click **Request certificate**. A form where you fill in information is displayed.

   ---

3. The preregistration form appears. (See Figure 56 on page 332.) Fill in the necessary information in the preregistration form for this SCEP client. The form that appears depends how this template is customized for your CA domain.

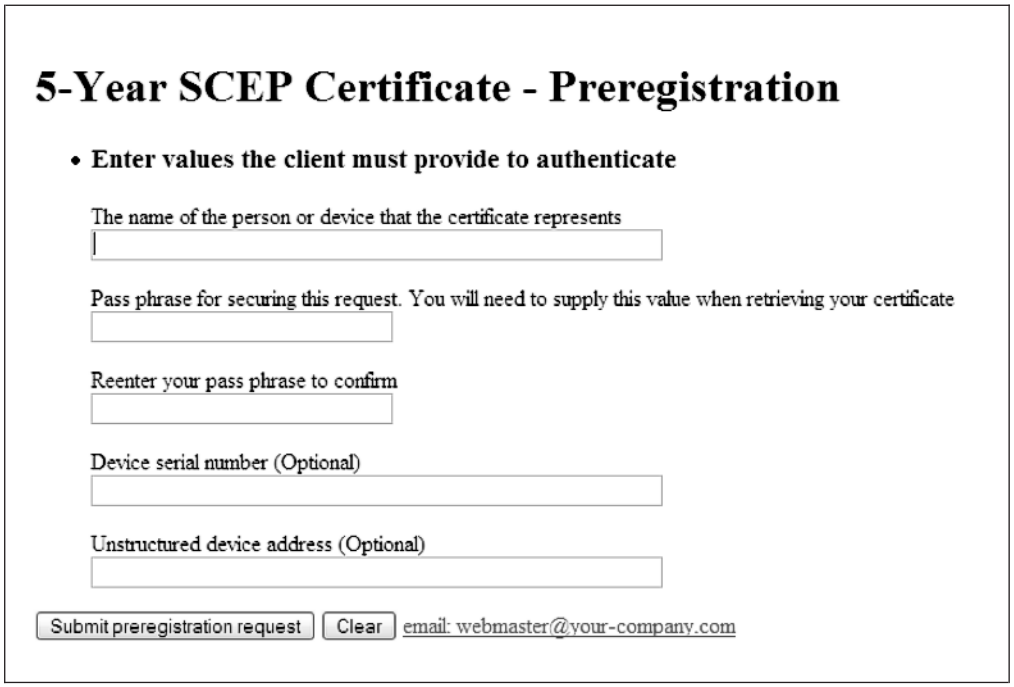


Figure 56. SCEP preregistration request form

4. Fill in the passphrase on the certificate request form (twice). This SCEP client must match this passphrase to successfully request a certificate. Do not use a sensitive value such as your ATM pin or login password.

5. Fill in any optional information as needed. This SCEP client must match all subject and alternate name information you enter to successfully request a certificate. When you are satisfied with the information you have entered, click **Submit certificate request**.

   If your preregistration request is successful, you see a page like the one shown in Figure 57, which tells you your temporary transaction ID.



Figure 57. Successful preregistration request

6. (Optional) Click **Examine Preregistration Record**.

---

7. (Optional) Click **Preregister** to preregister another client using the same template.

When you are done: You have successfully preregistered a SCEP client. Return to the PKI Services home page (by clicking **Home Page**) or return to the administration home page (by clicking **Administration Home Page**).

Chapter 17. Using the administration Web pages

This topic presents background information about certificate requests and certificates and explains how the administrator can use the administration Web pages to perform the following tasks:

- Process a certificate request
  - Approve a request without making changes
  - Approve a request with changes
  - Reject a request
  - Delete a request
- Process a preregistration record
  - Delete a certificate request from a preregistered client

    (For information about preregistering clients, see "Steps for preregistering an SCEP client" on page 331.)
- Process a certificate
  - Revoke a certificate
  - Suspend a certificate
  - Resume a certificate
  - Delete a certificate
- Perform searches for certificate requests, certificates, and preregistration records

Note: The PKI Services Web pages in this topic might differ slightly from those on the Web. If you need to see the exact content, view the pages on the Web. Additionally, the pages might contain differences depending on the browser you are using. This topic assumes you are using Internet Explorer.

Steps for accessing the administration home page

Perform the following preliminary steps to access the administration home page:

1. Get your organization's URL for accessing the PKI Services Web pages. Enter this URL in your browser. This takes you to the PKI Services administration start page.

   The administration start page contains a button for the administration home page (see the default "PKI Services Administration" home page in Figure 61 on page 340) and a button for each end-user home page (see the default "PKI Administrators Start Page" in Figure 58 on page 336).

© Copyright IBM Corp. 2001, 2011

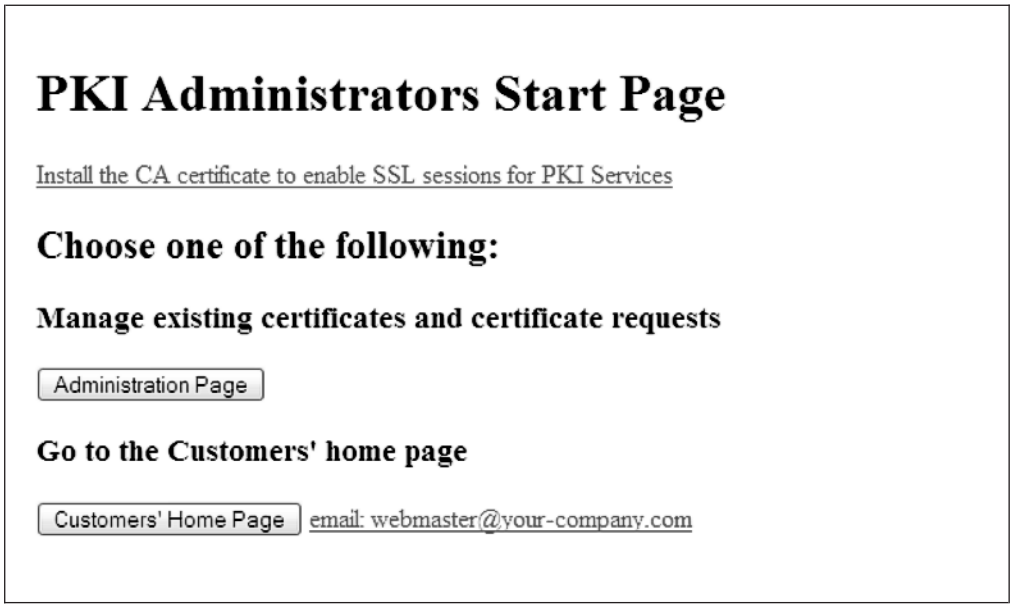


Figure 58. PKI Services administration start page

2. If this is the first time you have accessed these Web pages, you must install the CA certificate into your browser. Click the **Install the CA certificate** link and follow the directions.

   The following is a sample of the directions to follow for installing the CA certificate on Internet Explorer:

   a. After you click the **Install the CA certificate** link, a popup window called "File download" appears. Make sure the "Open this file from its current location" radio button is selected (rather than "Save this file to disk"). Then click **OK**. The following is an example of the popup window you might see, depending on the CA certificate you have installed:

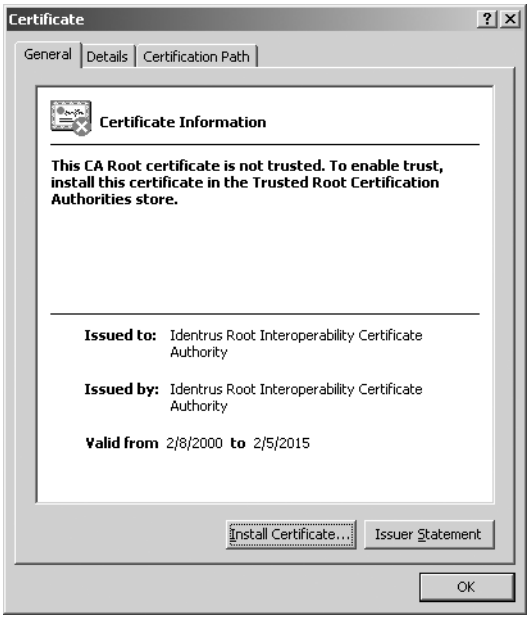


Figure 59. The certificate popup window for installing the CA certificate

b. Click **Install certificate**. (This initiates a series of pop-ups in which you need to click **Next** and finally **Finish**, culminating in a popup window that says "The import was successful").

Note: If you are using the Internet Explorer browser on a Microsoft Windows Vista system, you must explicitly select a store to place the certificate in. For more information, see "Installing the PKI Services CA certificate on a Microsoft Windows system" on page 592.

---

3. Click **Go to administration page**.

---

4. You will be prompted to authenticate, as shown in the following figure. Provide the necessary information:

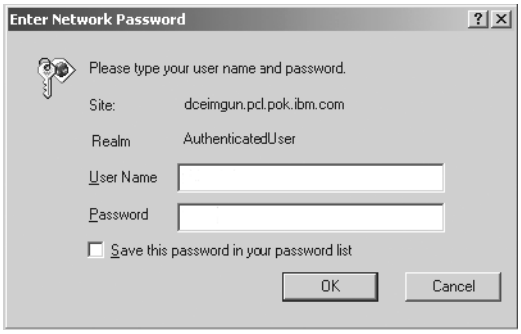


Figure 60. Entering your user ID and password

a. Fill in your z/OS user ID and password.
b. If you want to eliminate having to reenter your user ID and password each time you access the administration pages, check the check box.
c. Click **OK**.

   This calls up the "PKI Services Administration" Web page. (See Figure 61 on page 340.)

Notes:

- Your Web server programmer might provide you with an alternate URL for accessing the administration home page. You might also have to authenticate using a certificate instead of a user ID and password.
- Your browser caches the authentication information that you provide. Therefore, if you need to change this information, you first must close all instances of your browser. Then open the browser and, when the panel shown in Figure 60 appears, enter the correct information.

Fields in the administration Web pages

When you process certificates requests and certificates, you provide information for various fields in the Web pages. The following table describes the fields in the administration Web pages:

Table 62. Summary of fields in the administration pages

Field	Description
Recent activity	This specifies a time range for searches. Possible values include: • Not selected • Within the past day • Within the past week • Within the past month • Within the past six months
Requestor name	The name of the person requesting the certificate, as it appears in the common name field of the certificate request form.
Serial number	PKI Services assigns this number to a certificate when you approve it.
Transaction ID	PKI Services assigns this number to a request when a user requests it. This is a text field of up to 56 characters.

Processing certificate requests

Before you can use the Web page to process certificate requests, you need to understand the statuses of certificate requests and the actions you can perform on these certificate requests.

Status of certificate requests

Requests for certificates are kept in a request database while they are active. This is from the moment they are created until an event occurs that causes them to be deleted. The following table summarizes possible statuses. During the time period when a certificate request is active, it can have only one of the following statuses at a time:

Table 63. Statuses of certificate requests

Status	Meaning
`Pending Approval`	The request requires administrative approval. No action has been taken on the request yet.
`Approved`	The administrator explicitly approved the request or it was submitted as an auto-approved certificate request. The actual certificate might or might not have been created at this point.
`Completed`	The certificate has been issued and the requestor has retrieved it. This is a final state.
`Preregistered`	The certificate request is from a preregistered Simple Certificate Enrollment Protocol (SCEP) client.
`Rejected`	The administrator rejected the request, and the requestor has *not* been informed of this action (because the user has not tried to retrieve the certificate).
`Rejected, User notified`	The administrator rejected the request and the requestor has been informed of this action when attempting to retrieve the certificate. This is a final state.

A request is deleted from the request database when the administrator explicitly deletes it or when the request expires. This expiration time period is configurable and varies depending on whether the request was finalized or not.

Actions on certificate requests

The following table summarizes actions on certificate requests and the required status for each of these actions:

Table 64. Summary of actions to perform on requests and required status

Action	Required status of request
Approve	`Pending Approval`
Approve with modifications	`Pending Approval`
Reject	`Pending Approval`
Delete	All statuses (`Pending Approval`, `Approved`, `Completed`, `Preregistered`, `Rejected`, or `Rejected, Notified`)

Using the PKI Services administration home page

Figure 61 on page 340 shows the "PKI Services Administration" home page.

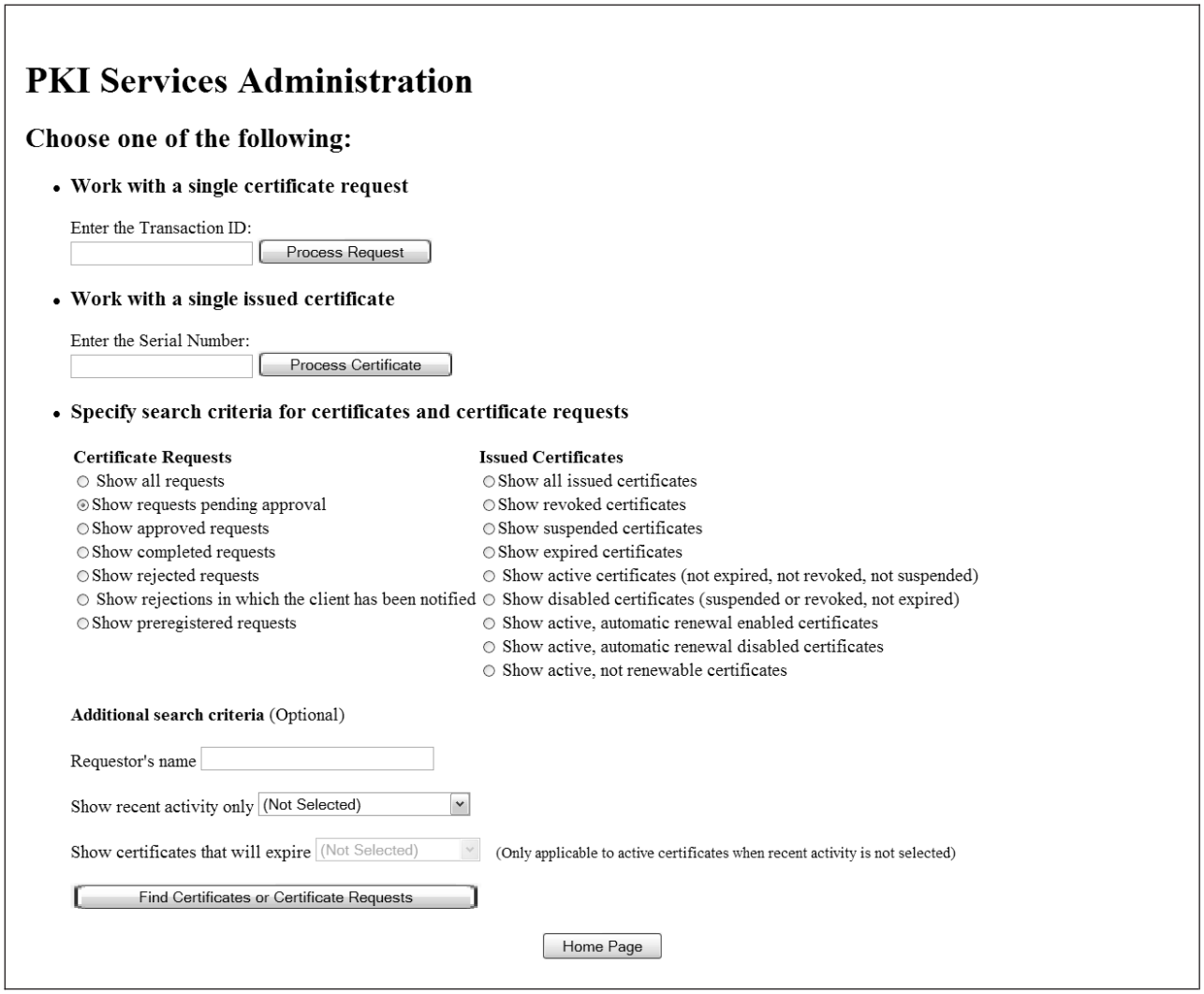


Figure 61. PKI Services administration home page

This Web page allows you to:

- Process a single certificate request (by specifying its transaction ID)
- Process a single certificate (by specifying its serial number)
- Search for groups of certificate requests or certificates by status and additional search criteria so that you can process them
- Search for certificates based on when they will expire.

You can process a single certificate request or a single preregistered (certificate) request if you know its transaction ID. Otherwise, you can perform a search to display all certificate requests of a particular status.

Steps for processing a single request

To process a single request, perform the following steps:

1. On the PKI Services administration home page (see Figure 61), enter the transaction ID in the field provided for it, and click **Process request**. This displays the "Single Request" Web page shown in Figure 62 on page 341.



Figure 62. Single request approval Web page

2. Make sure the request is the correct one by reviewing the information in the top part of the Web page.

   Guideline: If this is a preregistered certificate request, examine the *fingerprints* (the SHA1, SHA256, SHA512, and MD5 hashes) and contact the certificate requestor to confirm that the fingerprints in this received request match the fingerprints in the original request. These actions ensure the integrity of the request. (The requestor can use the SCEP client software to display the fingerprints saved for the original request.)

3. Optionally insert a comment.

4. Click one of the choices on the "Single Request" Web page to process the request.
   - **Approve the request as is**
   - **Approve the request with modifications**
   - **Reject request**

- **Delete request**

The choices on the "Single Request" Web page appear based on the status of the request. For example:

- If the status of a request is `Pending Approval`, only the first three choices in the preceding list appear.
- If the administrator has already processed the request or if the request is `Preregistered`, only **Delete request** appears.

a. When you click **Approve the request as is** and processing is successful, this displays a Web page that says you that "Processing is successful", such as the following. (Otherwise, the Web page says "Processing is not successful".)



Figure 63. Processing successful Web page

From these Web pages, you can then click **Process more request(s)** to return to the PKI Services administration home page (Figure 61 on page 340).

b. When you click **Approve the request with modifications**, this displays a "Modify and Approve Request" Web page similar to the one shown in Figure 65 on page 343.

If the subject's distinguished name contained in the current request is not in the proper format for RACF processing, you will see the note (Figure 64) on the "Modify and Approve Request" Web page.

Note - the existing subject's name is not in a format that can be recreated by PKI Services. Therefore, specifying any subject's name field below will cause the existing name to be deleted and completely replaced.

Figure 64. Restriction note on the modify and approve request Web page

Restriction: If you receive the note shown in Figure 64, you cannot change any field of the subject's distinguished name (the common name, organizational unit, or organization field) without causing PKI Services to delete the entire subject's distinguished name and replace it with your changed values. (This is because the subject's distinguished name is not in the proper format for RACF processing.)

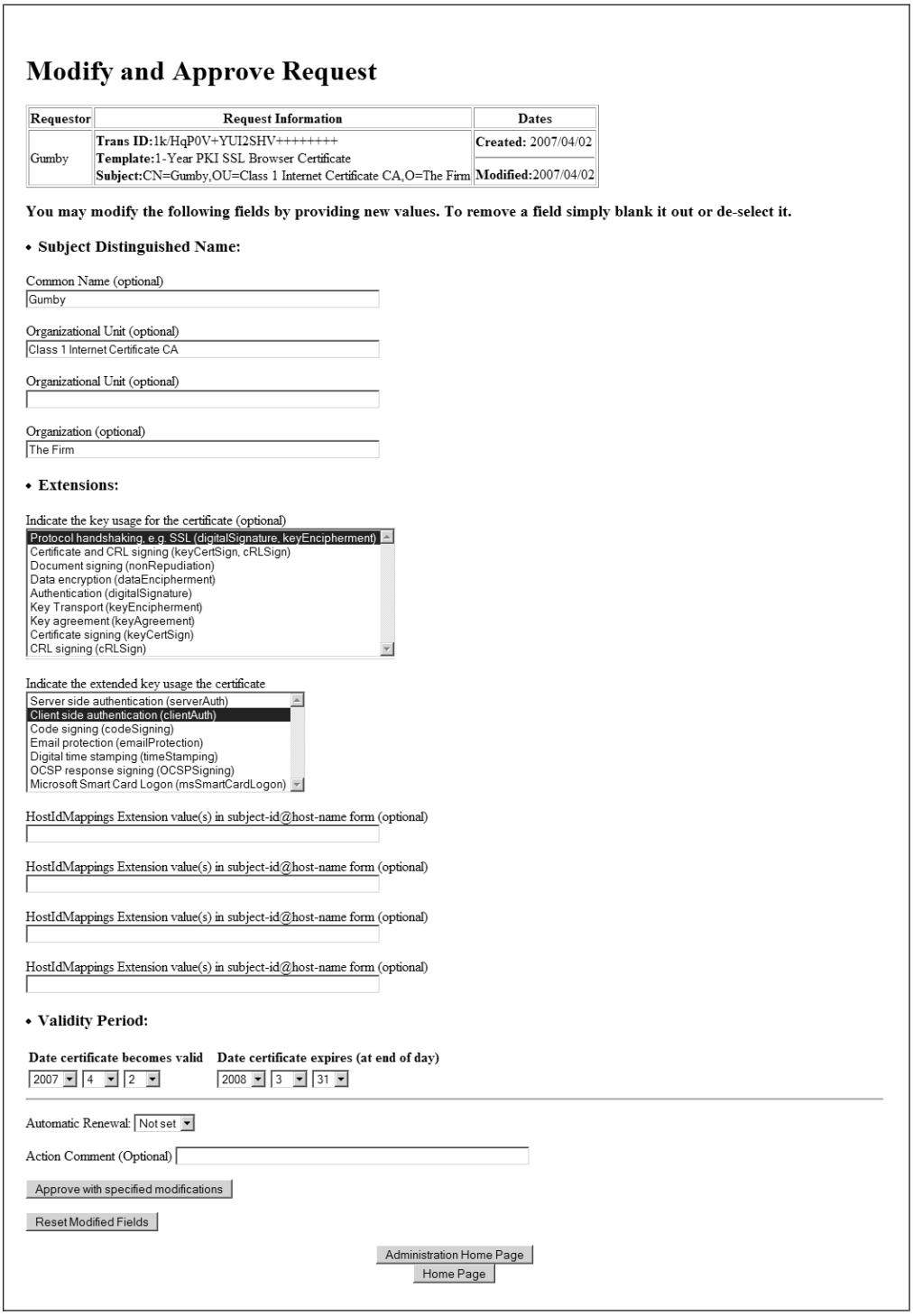


Figure 65. Modifying the request Web page

On the "Modify and Approve Request" Web page, you can change the following fields.

- Common name, unless noted as in Figure 64 on page 342
- Organizational unit (this can be multiple fields), unless noted as in Figure 64 on page 342
- Organization, unless noted as in Figure 64 on page 342
- e-mail address

Note: If you change the value of the e-mail address field (Email) and if the original request included the notification e-mail address field (NotifyEmail), the value of the latter field is changed to match the changed e-mail address value.

- Street
- Postal code
- Certificate purpose
- Date certificate becomes valid
- Date certificate expires
- `HostIdMappings` extensions (This can be multiple fields.)
- Optional comment about action you perform on the certificate.
- Automatic renewal

When you are satisfied with the changes you have made, click **Approve with specified modifications**; or, if you change your mind, you can click **Reset modified fields**. Alternately, you can click **Home page** to go to the PKI Services home page. (See Figure 36 on page 309.)

c. When you click **Reject request** , this displays a Web page that informs you that "Processing is successful" or that "Processing is not successful". From these Web pages, click **Process more request(s)** to return to the PKI Services administration home page. (See Figure 61 on page 340.)

d. When you click **Delete request**, this displays a Web page that informs you that "Processing is successful" or that "Processing is not successful". On these Web pages, click **Process more request(s)** to return to the PKI Services administration home page. (See Figure 61 on page 340.)

Steps for processing requests by performing searches

The administrator can use the Web page to search for certificate requests of various statuses. The following table summarizes the searches listed on the Web page and the certificate requests that are displayed as a result:

Table 65. Searches to display certificate requests

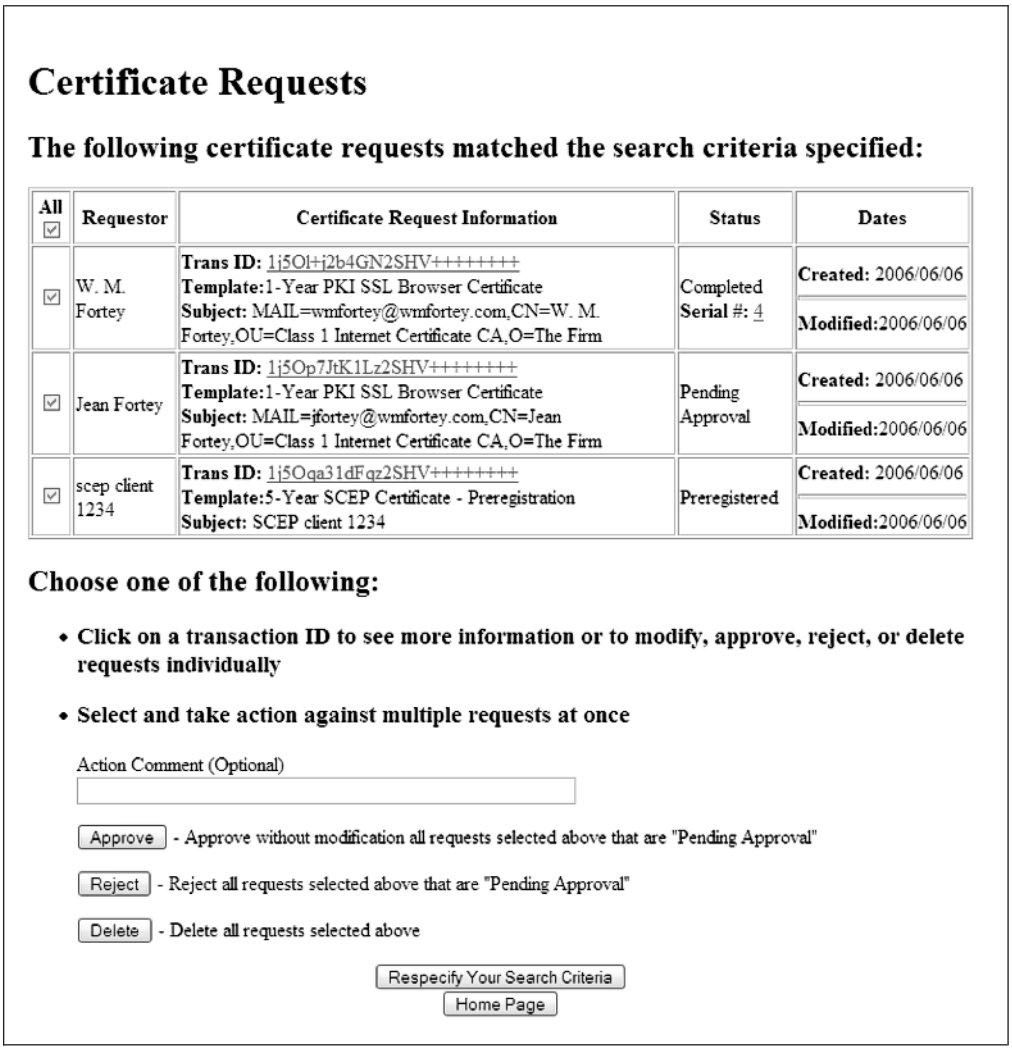
Search criteria	Results
Show all requests	Displays all certificate requests (all statuses (`Pending Approval`, `Approved`, `Completed`, `Preregistered`, `Rejected`, or `Rejected, User Notified`).
Show requests pending approval	Displays only certificate requests whose status is `Pending Approval`.
Show approved requests	Displays certificate requests whose status is `Approved` or `Completed`.
Show completed requests	Displays certificate requests whose status is `Completed`.
Show preregistered requests	Display certificate requests whose status is `Preregistered`.
Show all rejected requests	Displays certificate requests whose status is `Rejected`, or `Rejected, User Notified`.
Show rejections in which the client has been notified	Displays certificate requests whose status is `Rejected, User Notified`.

To process requests by performing a search for requests of a particular status, perform the following steps:

1. On the PKI Services administration home page (see Figure 61 on page 340), select one of the searches by clicking the appropriate choice under "Certificate Requests". (Table 65 on page 344 describes these searches.) You can optionally fill in additional search criteria ("Requestor's name" and "Show recent activity only").

   Guideline: Queries against the request database might time out if the database contains a large number of records. The performance of the query can be vastly improved by supplying "Requestor's name" as additional search criteria if the saved requestor data is meaningful to your organization and it is recallable. In this case, a PKI exit can be used to supply a meaningful value, such as a Lotus Notes short name or customer account number.

2. Click **Find certificates or certificate requests**. This displays the following Web page:



Certificate Requests

The following certificate requests matched the search criteria specified:

All ☑	Requestor	Certificate Request Information	Status	Dates
☑	W. M. Fortey	**Trans ID:** 1j5Ol+j2b4GN2SHV++++++++ **Template:**1-Year PKI SSL Browser Certificate **Subject:** MAIL=wmfortey@wmfortey.com,CN=W. M. Fortey,OU=Class 1 Internet Certificate CA,O=The Firm	Completed **Serial #:** 4	**Created:** 2006/06/06 **Modified:**2006/06/06
☑	Jean Fortey	**Trans ID:** 1j5Op7JtK1Lz2SHV++++++++ **Template:**1-Year PKI SSL Browser Certificate **Subject:** MAIL=jfortey@wmfortey.com,CN=Jean Fortey,OU=Class 1 Internet Certificate CA,O=The Firm	Pending Approval	**Created:** 2006/06/06 **Modified:**2006/06/06
☑	scep client 1234	**Trans ID:** 1j5Oqa31dFqz2SHV++++++++ **Template:**5-Year SCEP Certificate - Preregistration **Subject:** SCEP client 1234	Preregistered	**Created:** 2006/06/06 **Modified:**2006/06/06

Choose one of the following:

- **Click on a transaction ID to see more information or to modify, approve, reject, or delete requests individually**
- **Select and take action against multiple requests at once**

Action Comment (Optional)

Approve - Approve without modification all requests selected above that are "Pending Approval"

Reject - Reject all requests selected above that are "Pending Approval"

Delete - Delete all requests selected above

Respecify Your Search Criteria

Home Page

Figure 66. Processing requests after searching

Note: The table at the top of the Web page shows the certificate requests that match your search criteria. (If multiple certificates requests match the search criteria, up to ten appear on a Web page, and a button at the bottom of the Web age allows you to view the next set.)

3. You can use this Web page:
   - To process a single certificate request
   - To perform the same action on all of the certificate requests that are listed
   - To process selected requests.

   To process a single certificate request:

   a. Click on its transaction ID in the table at the top of the Web page. This transfers you to the single request Web page; see Figure 62 on page 341.
   b. From the "Single Request" Web page, you can perform the steps in the preceding section, starting with Step 2).

   To perform the same action on all the certificate requests that are listed:

   a. Optionally enter a comment.
   b. Click one of the actions below the comment field to perform that action on all listed requests:

      Approve Approves without modification all requests that are pending approval.
      Reject Rejects all requests that are pending approval.
      Delete Deletes all requests.

      Note: The **Approve** and **Reject** actions appear only if certificate requests are pending approval. Otherwise, only the **Delete** action appears.

   To process selected certificate requests:

   a. Uncheck the check box beside the **Select** column header. (When the check box beside **Select** is checked, all the individual check boxes in the body of the table are checked. This means all these certificate requests are selected. Unchecking the box in the header unchecks all the boxes in the body of the table.)
   b. Check the check boxes of all the certificate requests for which you want to perform a particular action.
   c. Optionally enter a comment.
   d. Click one of the actions below the comment field to perform that action on all listed requests. The actions include the following:

      Approve Approves without modification all requests that are pending approval.
      Reject Rejects all requests that are pending approval.
      Delete Deletes all requests.

      Note: The **Approve** and **Reject** actionss appear only if certificate requests are pending approval. Otherwise, only the **Delete** action appears.

   Tip: If you select **Show all requests** (see Figure 61 on page 340) and click **Approve** on this Web page, only the certificate requests whose status is `Pending Approval` are approved.

   Instead of processing one or more certificate requests, you can click **Respecify your search criteria Web page** to return to the PKI Services administration home page (see Figure 61 on page 340) or **Home page** to return to the PKI Services home page (see Figure 36 on page 309).

4. After you click an action, the next Web page is one of the following:
   - "Processing successful" (see Figure 67 on page 347)
   - "Processing was not successful" (see Figure 68 on page 347)

- "Processing partially successful" (see Figure 69 on page 348)

  If "Processing was not successful", you can click on the transaction ID to display the "Single Request" Web page; see Figure 62 on page 341. Processing can be unsuccessful because requests do not have the status required for the action you selected; see Table 64 on page 339.

  If you get "Processing partially successful", you can click on the transaction ID to display the "Single Request" Web page; see Figure 62 on page 341. This message can occur when your organization has more than one administrator and involves the following sequence:

  a. One administrator performs a search.
  b. Another administrator performs a search before the first administrator has approved requests displayed in the search results.
  c. One of the administrators approves only some of the requests.
  d. The other administrator tries to approve requests including at least one the preceding administrator has already approved and one that the preceding administrator has not already approved.



Figure 67. Request processing was successful Web page



Figure 68. Request processing was not successful Web page



Figure 69. Request processing was partially successful Web page

5. After approving requests as appropriate, you can:
   - Click **Process more request(s)** to return to Figure 66 on page 345.
   - Click **Administration home page** to return to Figure 61 on page 340.
   - Click **Home page** to return to Figure 36 on page 309.

Processing certificates

Before you can use the Web page to process certificates, you need to understand the statuses of certificates and actions you can perform on certificates.

Status of certificates

Certificates that have been created from requests are maintained permanently in an issued certificate database. Another name for this is the issued certificate list (ICL). Issued certificates are also published in an LDAP directory.

A certificate can have only one of the following states (statuses) at a time:

Table 66. Status of certificates

`Active`	The certificate has not yet expired, has not been revoked, and is not currently suspended.
`Active, AutoRenew`	The certificate is eligible for automatic renewal and this capability is enabled.
`Active, AutoRenewDisabled`	The certificate is eligible for automatic renewal but this capability is disabled.
`Active, NotRenewable`	The certificate cannot be renewed.
`Expired`	The certificate's validity period expired while it was active.
`Revoked`	The certificate has not expired but it has been revoked. Such certificates are published on the next certificate revocation list (CRL).
`Revoked, Expired`	The certificate was either revoked or suspended, and time has elapsed so that it is now also expired. Such certificates are not published on the next CRL.
`Suspended`	The certificate has not expired but it is currently suspended. Such certificates are published on the next certificate revocation list (CRL).

The administrator must approve a request for the certificate to have a status (as enumerated in the preceding list) or for the administrator to delete the certificate from the ICL. (An administrator can delete a certificate from the ICL, but this would not be a normal situation.) Alternately, the administrator can reject a request or delete the request from the request database (RDB). If the administrator does not approve the request, it is never listed in the ICL.

Actions for certificates

The following table summarizes actions on certificates and the required status to perform these actions:

Table 67. Summary of actions to perform and required status to do so

Action	Required status of certificate	Who performs action
Renew	`Active`	End user
Resume	`Suspended`	Administrator
Revoke	`Active` or `Suspended`	End user or administrator
Suspend	`Active`	End user or administrator
Delete	All statuses (`Active`, `Expired`, `Suspended`, `Revoked`, or `Revoked, Expired`)	Administrator
Enable automatic renewal	`Active` or `Active,AutoRenewDisabled`	Administrator
Disable automatic renewal	`Active` or `Active,AutoRenew`	Administrator
Change requestor e-mail	All statuses **Note:** This action applies only to certificates for which PKI Services generated the key pair.	Administrator

Note: You can resume (reactivate) a suspended certificate, or permanently revoke it, if the certificate has not yet expired and the suspension grace period has

not elapsed. If the grace period has elapsed, the certificate will be permanently revoked the next time certificate revocation lists (CRLs) are issued.

Steps for processing a single certificate

To process a single certificate, perform the following steps:

1. On the PKI Services administration home page (see Figure 61 on page 340), enter the serial number of the certificate you want to process in the field provided for it. The following is an example of the Web page that is displayed:

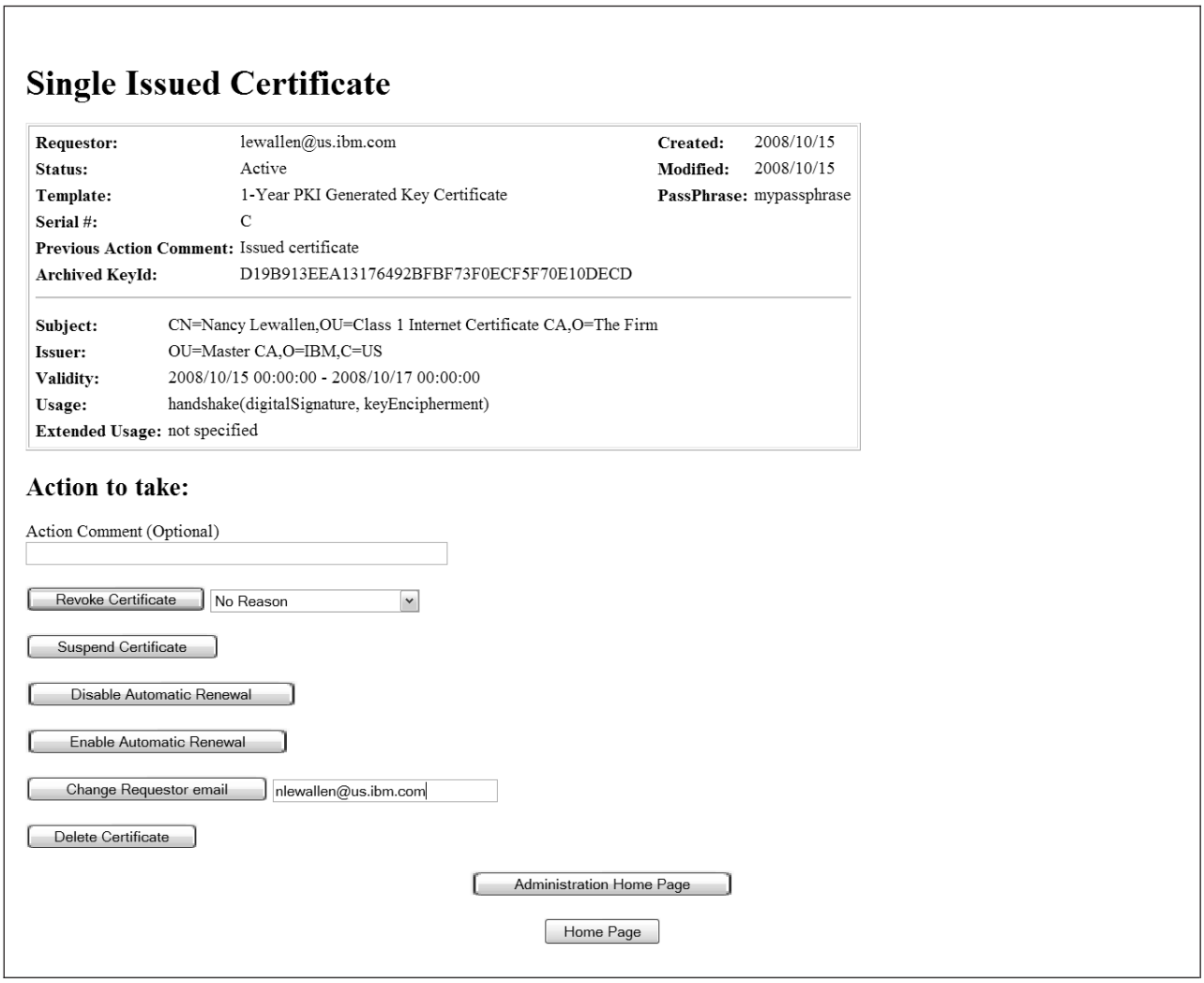


Figure 70. Processing a certificate from the single certificate Web page

2. Make sure the certificate is the correct one by reviewing the information in the top part of the Web page.

3. If you are going to process a certificate from this Web page, you can optionally insert a comment.

4. Click one of the following actions to process the certificate:

Disable automatic renewal	Disables automatic renewal for the certificate
Enable automatic renewal	Enables automatic renewal for the certificate
Revoke certificate	Revokes the certificate.
Suspend certificate	Suspends the certificate.
Resume certificate	Resumes the certificate.
Delete certificate	Deletes the certificate. (This is for cleanup purposes.)
Change Requestor email	Changes the requestor's e-mail to the value that you supply in the field next to the button. Use this function when the e-mail address of the requestor of a PKI generated key certificate has changed. If you change the requestor's e-mail, the certificate cannot be renewed.

Notes:

a. The **Suspend** and **Revoke** buttons appear only if the status of the certificate is `Active`.

b. The **Resume** button appears only if the status of the certificate is `Suspended`.

c. The **Enable Automatic Renewal** button appears only if the status of the certificate is `Active,AutoRenwDisabled`.

d. The **Disable Automatic Renewal** button appears only if the status of the certificate is `Active,AutoRenew`.

e. The **Change Requestor email** button appears only for a PKI generated key certificate.

Steps for processing certificates by performing searches

The administrator can use the Web page to search for certificates of various statuses. The following table summarizes the searches listed on the Web page and the certificates that are displayed as a result:

Table 68. Searches to display certificates

Searches	Results
Show all issued certificates.	Displays all certificates (can be any status).
Show revoked certificates.	Displays certificates whose status is `Revoked` or `Revoked, Expired`.
Show suspended certificates.	Displays certificates whose status is `Suspended`.
Show expired certificates.	Displays certificates whose status is `Expired` or `Revoked, Expired`.
Show active certificates (not expired, not revoked, not suspended).	Displays certificates whose status is `Active`.
Show disabled certificates (suspended or revoked, not expired).	Displays certificate requests whose status is `Suspended` or `Revoked`.
Show certificates with automatic renewal enabled.	Displays certificates whose status is `Active,AutoRenew`.
Show certificates with automatic renewal disabled.	Displays certificates whose status is `Active,AutoRenewDisabled`.
Show certificates that cannot be renewed.	Displays certificates whose status is `Active,NotRenewable`.

To process certificates by performing a search for certificates of a particular status, perform the following steps:

1. On the PKI Services administration home page (see Figure 61 on page 340), select one of the searches by clicking the appropriate choice under "Issued Certificates". (The preceding table describes these searches.) You can optionally fill in additional search criteria ("Requestor's name" and "Show recent activity only").

2. Click **Find certificates or certificate requests**. This displays the following Web page.

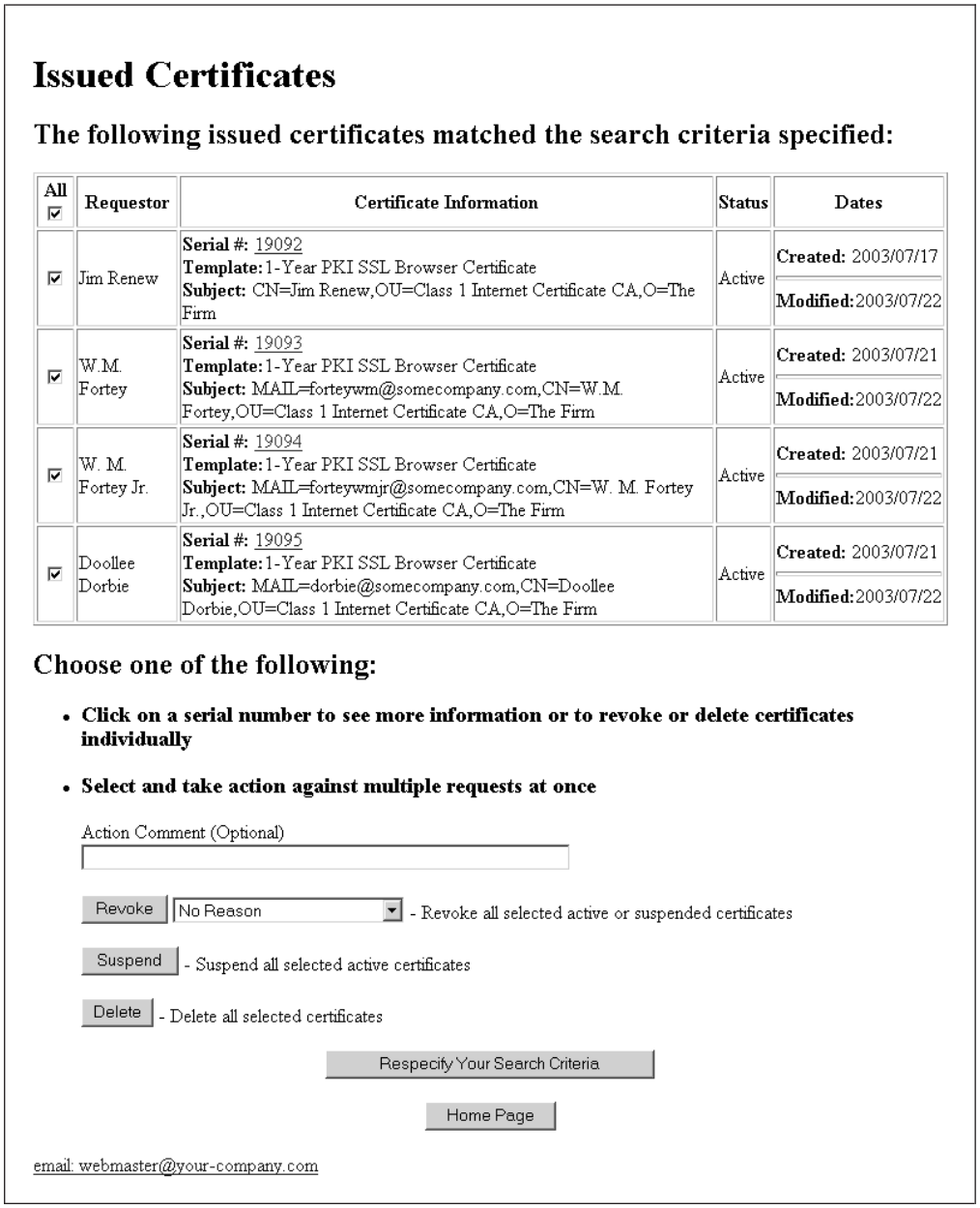


Figure 71. Processing certificates using searches

Note: The table at the top of the Web page shows the certificates that match your search criteria. (If multiple certificates match the search criteria, up

to ten appear on a Web page, and a button at the bottom of the Web page allows you to view the next set.)

---

3. You can use this Web page:
   - To process a single certificate
   - To perform the same action on all of the certificates that are listed
   - To process selected certificate

   To process a single certificate:

   a. Click on its serial number in the table at the top of the Web page. This transfers you to the single certificate Web page; see Figure 70 on page 350.
   b. From the single certificate Web page, you can perform the steps in "Steps for processing a single certificate" on page 350, starting with Step 2 on page 350).

   To perform the same action on all the certificates that are listed:

   a. Optionally enter a comment.
   b. Click one of the actions below the comment field to perform that action on all listed certificates:

      Revoke Revokes all selected active certificates.
      Suspend Suspends all selected active certificates.
      Resume Resumes all selected suspended certificates.
      Delete Deletes all selected certificates.

      Notes:

      1) **Suspend** and **Revoke** appear only when your search matches at least one certificate whose status is `Active`.
      2) **Resume** appears only when your search matches at least one certificate whose status is `Suspended`.
      3) **Enable Automatic Renewal**, **Disable Automatic Renewal**, and **Change Requestor email** are not shown on this page. They only appear on the page showing the individual certificate.

   To process selected certificates:

   a. Uncheck the check box beside the **Select** column header. (When the check box beside **Select** is checked, all the individual check boxes in the body of the table are checked. This means all these certificates are selected. Unchecking the box in the header unchecks all the boxes in the body of the table.)
   b. Check the check boxes of all the certificates for which you want to perform a particular action.
   c. Optionally enter a comment.
   d. Click one of the actions below the comment field to perform that action on all listed requests. The actions include:

      Revoke Revokes all selected active certificates.
      Suspend Suspends all selected active certificates.
      Resume Resumes all selected suspended certificates.
      Delete Deletes all selected certificates.

      Notes:

      1) The **Suspend** and **Revoke** actions appear only when your search matches at least one certificate whose status is `Active`.
      2) The **Resume** action appears only when your search matches at least one certificate whose status is `Suspended`.

Instead of processing one or more certificates, you can click **Respecify your search criteria Web page** to return to the PKI Services administration home page (see Figure 61 on page 340) or **Home page** to return to the PKI Services home page (see Figure 36 on page 309.)

---

4. After you click an action, the next Web page tells you:
   - "Processing was successful" (see Figure 72)
   - "Processing was not successful" (see Figure 73 on page 355)
   - "Processing partially successful" (see Figure 74 on page 355)

   If "Processing was not successful", you can click on a serial number to display the "Single Certificate" Web page; see Figure 70 on page 350. Processing can be unsuccessful because certificates do not have the status required for the action you selected; see Table 67 on page 349.

   If you get "Processing partially successful", you can click on the serial number to display the "Single Certificate" Web page; see Figure 70 on page 350. The "Processing partially successful" message can occur when your organization has more than one administrator and involves the following sequence:

   a. One administrator performs a search.
   b. Another administrator performs a search before the first administrator has revoked or deleted certificates displayed in the search results.
   c. One of the administrators revokes or deletes some of the certificates.
   d. The other administrator tries to revoke or delete certificates including at least one the preceding administrator has already revoked or deleted and at least one the preceding administrator has not already revoked or deleted.



Figure 72. Processing of certificate was successful Web page



Figure 73. Request processing was not successful Web page



Figure 74. Request processing was partially successful Web page

You can click **Home page** to return you to the PKI Services home page. (See Figure 36 on page 309.)

Relationship between certificate requests and matching certificates

PKI Services maintains two databases:

- The request database (RDB), also called the object store
- The issued certificate list (ICL)

RDB records are temporary in nature. They exist only to track active requests. PKI Services automatically removes these records when they are complete or go inactive. ICL records are permanent. Requests for certificates (both new and renewal) are stored in the RDB. Once approved, a matching certificate is created

from the request and stored in the ICL. (Note, the creation of the certificate might not be instantaneous.) At this point, the two database records, though related, exist independently of each other.

- After a request is approved, there is no way for you to *un*-approve a request. If you mistakenly approve a request that you meant to reject, you should immediately delete the RDB entry. This prevents the user from retrieving the certificate. You should then search the issued certificates to see if the certificate has been issued. If it has, you should revoke it in case the user has already picked it up.
- Revoking a certificate (an ICL action) has no effect on its matching RDB entry. If you revoke a certificate, you should also delete its matching RDB entry if it exists. This prevents the user from retrieving the certificate, if the user has not already done so.
- You can delete RDB entries any time after they have been completed to save space in the database if desired.
- Under normal circumstances, ICL entries should not be deleted. If you delete an ICL entry, you will no longer be able to revoke or renew the certificate.
- You can delete entries in any state in either database to clean up error conditions.

Chapter 18. Using PKI Services utilities

This topic describes the following utility programs, which are shipped with PKI Services. These programs are installed in the /*install-dir*/pkiserv/bin directory.

createcrls A UNIX program that initiates the certificate revocation list (CRL) creation task immediately..

iclview A UNIX program that displays the entries in the issued certificate list (ICL).

pkiprereg A UNIX program that creates Simple Certificate Enrollment Protocol (SCEP) preregistration records

postcerts A UNIX program that creates posting objects for existing certificates. The PKI Services daemon later posts the certificates to an LDAP server.

TemplateTool A Java program that validates an XML certificate template file and converts it to a text CGI template file, and converts a text CGI template file to an XML template file.

vosview A UNIX program that displays the entries contained in the object store (request database).

vsam2db2 A UNIX program that converts data from the issued certificate list (ICL) and object store VSAM data sets into DB2 tables.

© Copyright IBM Corp. 2001, 2011

Using the createcrls utility

Purpose

The `createcrls` program initiates the task that creates certificate revocation lists (CRLs). Later, depending on how PKI Services is configured, the PKI Services daemon either posts the CRLs to an LDAP server (for LDAP) or saves them in the HFS (for the URI format). (The `PostInterval` parameter in the **LDAP** section of the configuration file determines when the posting to LDAP occurs.) You can use this program to create a CRL immediately, instead of waiting for PKI Services to do it automatically based on the `TimeBetweenCRLs` parameter in the configuration file.

Path setup

Update your PATH, LIBPATH, and NLSPATH environment variables with the appropriate `pkiserv` directory before you run `createcrls`. (Note that you are updating the environment variables for the user running the utility, not updating values in the PKI Services environment variables file, `pkiserv.envars`.) Once you have updated these variables, you can run `createcrls` from the UNIX command line.

Variable name	You must add ...
PATH	*/install-dir*/pkiserv/bin
LIBPATH	*/install-dir*/pkiserv/lib
NLSPATH	*/install-dir*/pkiserv/lib/nls/msg/%L/%N

The default directory for *install-dir* is `/usr/lpp`.

Format

```
createcrls [-D CA-Domain-name]
```

Parameters

-D *CA-Domain-name*
: Specifies the 1-8 character name of the CA domain for which CRLs are to be created. The name can be entered using uppercase or lowercase letters. This option is required only if PKI Services is running with multiple CA domains.

Examples

To create a CRL for the domain `mydomain` and post the CRL to LDAP, enter the command:

```
createcrls –D mydomain
```

To create a CRL and post it to LDAP if you are not running PKI Services with multiple CA domains, enter the command:

```
createcrls
```

Using the iclview utility

Purpose

The iclview program displays the entries contained in an issued certificate list (ICL). Depending on how you have configured PKI Services, the ICL can be in a VSAM data set or in a DB2 table. Each ICL record consists of a fixed header followed by a variable-length section containing the BER-encoded certificate. For each entry iclview displays the header information and optionally calls a user-provided program to process the BER-encoded certificate.

Path setup

Update your PATH, LIBPATH, and NLSPATH environment variables with the appropriate pkiserv directory before you run iclview. (Note that you are updating the environment variables for the user running the utility, not updating values in the PKI Services environment variables file, pkiserv.envars.) Once you have updated these variables, you can run iclview from the UNIX command line.

Variable name	You must add ...
PATH	/*install-dir*/pkiserv/bin
LIBPATH	/*install-dir*/pkiserv/lib
NLSPATH	/*install-dir*/pkiserv/lib/nls/msg/%L/%N

The default directory for *install-dir* is /usr/lpp.

Format

```
iclview {-d vsam-dataset-name [-r] | -b db2-subsystem-name -k db2-package-name
        | -c [-p path]}
        [-D CA-domain-name] [-s decode-command-string]
```

Parameters

You can display usage information about the iclview command format and parameters when you issue the iclview utility command with no parameters.

-d *vsam-dataset-name*
: Specifies the MVS data set name of the VSAM issued certificate list (ICL).

  If you specify -d, do not specify -b, -k or -c. If you do, the utility issues an error message and the command fails.

  Note: If the data set name has no quotes, the program uses the invoker of the command as the first qualifier. If you specify the fully-qualified data set name, use quotes, and make sure to include the *escape* character, which is a backslash (\), before the quotation marks enclosing the data set name. For example, see \'pkisrvd.vsam.ost\' in "Examples" on page 373.

-r
: Indicates to open in VSAM record-level sharing (RLS) mode the VSAM data set specified with the -d option.

  -r is ignored and the utility issues a warning message if -b and -k are specified, or if -c is specified.

-b *db2-subsystem-name*
: Specifies the name of the DB2 subsystem where the ICL is located.

If you specify -b and you do not specify -k, or you specify -d or -c, the utility issues an error message and the command fails.

-k *db2-package-name*
Specifies the DB2 package name of the ICL.

If you specify -k and do not specify -b, or you specify -d or -c, the utility issues an error message and the command fails.

-c Indicates to retrieve the location of the ICL from the pkiserv.conf configuration file. Either the VSAM data set name is retrieved, or the DB2 subsystem name and package name are retrieved, depending which you are using. For VSAM, the SharedPlex value determines whether the VSAM data set is opened in record-level sharing (RLS) mode. (For DB2, the SharedPlex value has no effect on this utility.)

If you specify -c, and you also specify -d, -b, or -k, the utility issues an error message and the command fails.

Note: When you also specify the -D option, you must use the -p option to specify the CA domain configuration directory if it is not /etc/pkiserv.

-p *path*
Specifies the directory where the pkiserv.conf configuration file resides. If not specified, the directory defaults to /etc/pkiserv. This option is only valid when specified with the -c option. If specified with the -b, -k, or -d options, the utility issues a warning message, and the -p option is ignored.

-D *CA-domain-name*
Specifies the CA domain name where this utility command is directed.

Notes:
1. The -D option is required only if PKI Services is running in multiple-CA mode.
2. The *CA-domain-name* value can be entered using uppercase or lowercase letters.
3. When you also specify the -c option, you must use the -p option to specify the CA domain configuration directory if not /etc/pkiserv.

-s *decode-command-string*
Specifies an optional command to call for decoding the ASN.1-encoded data. The command must be able to read and decode binary (BER) data from STDIN.

Examples

To view the records in the ICL in the VSAM data set 'PKISRVD.VSAM.ICL', passing the certificate to a utility called dumpasn1, use the following command:

```
iclview -d \'pkisrvd.vsam.icl\' -s 'dumpasn1 -'
```

To view the records in the ICL for the CA domain MasterCA using the information from the pkiserv.conf file located in the directory /etc/pkiserv/MasterCA, use the following command:

```
iclview -c -p /etc/pkiserv/MasterCA -D MasterCA
```

To view the records in the ICL in the DB2 subsystem DSN9 with a package name of MasterCA, passing the certificate to a utility called dumpasn1 and directing the output to the file icl.out, use the following command:

```
iclview -b DSN9 -k MASTERCA -s 'dumpasn1 -' >icl.out
```

Note: A `dumpasn1` utility is not shipped with PKI Services.

Output

The fixed header data that is displayed for each record would look like the following sample:

```
Cert 8: John Q. Public@someWebProvider.com
     ISSUED (Issued certificate)
     Issued at 2001-12-19 17:27:41
     Last changed 2001-12-19 17:42:30
     Subject: CN=John Smith,OU=Class 1 Internet Certificate CA,O=The Firm
     Issuer: OU=PKI Services CA,O=IBM,C=US
     Requester: John Smith
     ApplData: 1YBSSL
     Serial Number: CCD
     Email flag: Off
     AutoRenew flag: Enabled
     Additional flags Set: NotRenewable
     KeyID: 12FD68977EE1F987DC9CA1440B62CCCD1CD0A9BB
```

An explanation of the lines in the fixed header follows:

The first line specifies certificate's sequential position within the ICL, relative to the other certificates, and requestor's name.

The second line specifies the certificate state, which is one of the following, and a comment (if any):
- ISSUED
- REVOKED, not posted
- REVOKED, awaiting CRL post
- REVOKED, on posted CRL

Issued at Indicates when the certificate was issued.

Last changed Indicates when the administrator last changed the certificate.

Subject Indicates the name of the person who owns the certificate.

Issuer Indicates the name of the certificate authority that issued the certificate.

Requestor Indicates the requestor's name.

Appldata Indicates the 8-character string identifying to the application the short name or nickname of the certificate template. (PKI Services provides sample certificate templates but it is RACF, or an equivalent security product, rather than PKI Services that handles the SAF templates.) The following table shows the nicknames for each certificate template. (These nicknames are supplied in the `pkiserv.tmpl` certificate templates file as defaults but your installation might have changed them or added others during customization. See "TEMPLATE sections" on page 136 for more information.)

Table 69. Names, aliases, and nicknames of certificate templates

True name	Alias	Nickname
1-Year PKI SSL Browser Certificate	PKI Browser Certificate	1YBSSL
1-Year PKI S/MIME Browser Certificate	PKI Browser Certificate	1YBSM
1-Year PKI Generated Key Certificate	PKI Key Certificate	1YKRC

Table 69. Names, aliases, and nicknames of certificate templates (continued)

True name	Alias	Nickname
2-Year PKI Browser Certificate For Authenticating To z/OS	PKI Browser Certificate	2YBZOS
2-Year PKI Authenticode - Code Signing Certificate	PKI Server Certificate	2YIACS
2-Year PKI Windows Logon Certificate	PKI Browser Certificate	2YBWL
5-Year PKI SSL Server Certificate	PKI Server Certificate	5YSSSL
5-Year PKI IPSEC Server (Firewall) Certificate	PKI Server Certificate	5YSIPS
5-Year PKI Intermediate CA Certificate	PKI Server Certificate	5YSCA
5-Year SCEP Certificate - Preregistration	—	5YSCEPP
n-Year PKI Certificate for Extensions Demonstration	PKI Browser Certificate	SAMPLB
1-Year SAF Browser Certificate	SAF Browser Certificate	—
1-Year SAF Server Certificate	SAF Server Certificate	—

Serial Number
Indicates the serial number of the certificate as a hexadecimal number.

Email flag Indicates whether or not to send an expiration warning message. The possible values are `On` or `Off`.

AutoRenew flag
Indicates whether a certificate is AutoRenew enabled, disabled, or not set up for automatic renewal.

Revoked at Indicates the date and time the certificate was revoked or suspended.

Revocation Reason
Indicates one of the following reasons:
- No reason.
- User key was compromised.
- Original use no longer valid.
- CA key was compromised.
- User changed affiliation.
- Certificate was superseded.
- Temporarily suspended.

Additional flags Set
The NotRenewable flag indicates that the certificate cannot be renewed. This state occurs when the keys for the certificate were generated by PKI Services, the certificate has a Mail RDN, and the user's e-mail address has changed since the certificate was created.

KeyID The SHA1 hash of the public key in EBCDIC format, 40 bytes in length. This line is only displayed if the key was generated by PKI Services.

Using the pkiprereg utility

Purpose

The pkiprereg program creates Simple Certificate Enrollment Protocol (SCEP) preregistration records in batch.

Path setup

Update your PATH, LIBPATH, and NLSPATH environment variables with the appropriate pkiserv directory before you run pkiprereg. (Note that you are updating the environment variables for the user running the utility, not updating values in the PKI Services environment variables file, pkiserv.envars.) Once you have updated these variables, you can run pkiprereg from the UNIX command line.

Variable name	You must add ...
PATH	/*install-dir*/pkiserv/bin
LIBPATH	/*install-dir*/pkiserv/lib
NLSPATH	/*install-dir*/pkiserv/lib/nls/msg/%L/%N

The default directory for *install-dir* is /usr/lpp.

Format

```
pkiprereg {-m mode [-t tmpl-file] -s SCEP-file [-o out-file] [-d domain]
          | [-h | -?]}
```

Parameters

You can display usage information about the pkiprereg command format and parameters when you execute the pkiprereg utility command with the -h or -? option or when you enter an incorrect parameter.

-m *mode*

Indicates one of the following modes of operation for this pkiprereg utility execution.

Mode name	Function
verify	Checks the data file for format errors but does *not* load the records into PKI Services.
generate	Generates a random 8-character passphrase whenever it finds a passphrase *placeholder* (the * character) in the file. It does *not* load the records into PKI Services.
load	Calls PKI Services to load the preregistration records.
remove	Calls PKI Services to remove the preregistration records.

You can specify the mode value as the mode name or any number of initial characters of the mode name. **Example:** You can specify **verify** mode using any of the following options:

```
-m verify
-m ve
-m v
```

-t *tmpl-file*

When used in **verify** or **load** mode, specifies the pathname of the PKI Services certificate templates file (pkiserv.tmpl if you are implementing the Web application using REXX CGI scripts, or pkixgen.tmpl if you are

implementing the Web application using Java server pages), which is used as input. When used in other modes, the `-t` option is ignored.

`-s SCEP-file`
Specifies the pathname of the file containing the preregistration data that is input for all modes.

`-o out-file`
When used in **generate** mode, specifies the pathname of the output file for the preregistration data. When used in other modes, the `-o` option is ignored.

`-d domain`
When used in **load** or **remove** mode, specifies the PKI Services CA domain name where this utility command is directed.

Notes:
1. When used in other modes, the `-d` option is ignored.
2. The `-d` option is required only if PKI Services is running in multiple-CA mode.
3. The *domain* value can be entered using uppercase or lowercase letters.

`-h` or `-?`
Displays the syntax of the `pkiprereg` command.

Input

The input for the `pkiprereg` utility is a data file (*SCEP-file*) containing preregistration records. The rules regarding format of the preregistration records are as follows:

Rules:

- Each preregistration record consists of multiple consecutive *name=value* pairs, terminated by a blank line or an end-of-file indicator.
- Each line consists of single field name and its value, separated by a `=` character, forming one *name=value* pair per line.
- Each *name=value* pair (except the Template *name=value* pair) represents data that the client must supply at enrollment time for authentication purposes.
- In `pkiprereg` **verify** and **load** modes, the Template and ClientName *name=value* pairs are required. All other *name=value* pairs are optional.
- In `pkiprereg` **remove** mode, only the ClientName *name=value* pair is required.
- Any line beginning with a `#` character is considered a comment.
- All characters before the `=` character, except any leading and trailing white space, are considered the *name*.
- All characters after the `=` character, except any leading and trailing white space, are considered the *value*.
- The field name supplied as a *name* is case-sensitive and must match one of the field names listed in Table 70 on page 365.
- Line length is limited to 300 characters. After 300 characters, any additional characters are truncated.

The following field names are supported in the preregistration record.

Table 70. List of valid field names for use in the preregistration record as input to the pkiprereg utility

Field name	Maximum length	Description
Template	8 characters	The nickname of the preregistration template to be used.
ClientName	64 characters	The name of the person or device being preregistered. The first 32 characters are case-insensitive and must be unique for each user preregistered in PKI Services.
PassPhrase	32 characters	The password to be communicated to the requestor. The * value can be used as a *placeholder* when running in **generate** mode.
SerialNumber	64 characters	Serial number of the subject device.
UnstructAddr	64 characters	Unstructured address of the subject device.
EmailAddr	64 characters	E-mail address for the EMAIL= attribute for subject distinguished name.
Mail	64 characters	E-mail address for the MAIL= attribute for subject distinguished name.
DNQualifier	64 characters	Specifies information to add to the subject's distinguished name to make it unambiguous.
Uid	64 characters	The system login name associated with the subject.
Title	64 characters	Title for subject distinguished name.
DomainName	64 characters	One component of a domain name. For example, domain name www.ibm.com is represented by 3 components: www, ibm and com.
OrgUnit	64 characters	Organizational unit for subject distinguished name.
Org	64 characters	Organization for subject distinguished name.
Street	64 characters	Street for subject distinguished name.
Locality	64 characters	Locality for subject distinguished name.
StateProv	64 characters	State or province for subject distinguished name.
PostalCode	64 characters	Postal code for subject distinguished name.
Country	2 characters	Country abbreviation for subject distinguished name.

Table 70. List of valid field names for use in the preregistration record as input to the pkiprereg utility (continued)

Field name	Maximum length	Description
AltIPAddr	45 characters	The IP address for the subject alternate name extension. PKI Services supports IP version 4 and IP version 6 addresses. • For IP version 4, the IP address is in dotted decimal format; for example, `9.67.97.103`. • For IP version 6, the IP address is divided into eight 16-bit hexadecimal blocks separated by colons. Leading zeros in each 16-bit field are optional, and successive fields of zeros can be represented by double colons, but only once; for example `1:2::3:4` is equivalent to `0001:0002:0000:0000:0000:0000:0003:0004`. • In a mixed IP version 4 and IP version 6 environment, the IP address can be expressed in the format *x:x:x:x:x:x:d.d.d.d*, where the *x* values are the hexadecimal values of the six high-order 16-bit pieces of the address, and the *d* values are the decimal values of the four low-order 8-bit pieces of the address in standard IP version 4 representation; for example, `0:0:0:0:0:ABCD:1.2.3.4`, or the equivalent value `::ABCD:1.2.3.4`
AltURI	255 characters	URI for subject alternate name extension.
AltEmail	100 characters	E-mail address for subject alternate name extension.
AltDomain	100 characters	Domain name for subject alternate name extension.
AltOther	255 characters	Other Name for subject alternate name extension.

For information on the format required for these values, see the `R_PKIServ` (IRRSPX00) section of *z/OS Security Server RACF Callable Services*.

Examples

The following example shows two preregistration records in the `/etc/scep.txt` file.

Examples:

```
Template=5YSCEPP
ClientName=www.ibm.com
PassPhrase=gumby
Org=IBM
Template=5YSCEPP
ClientName=scep.company.com
PassPhrase=*
```

The following command examples produce the actions listed:

Example:

```
pkiprereg -m v -t /etc/pkiserv/pkiserv.tmpl -s /etc/scep.txt
```

Action: Verifies the `/etc/scep.txt` file, indicating that one PassPhrase has an incorrect value, the placeholder (`*`).

Example:

```
pkiprereg -m g -s /etc/scep.txt -o /etc/scepfinal.txt
```

Action: Generates a passphrase for the placeholder and saves the records in the `/etc/scepfinal.txt` output file.

Example:

```
pkiprereg -m l -t /etc/pkiserv/pkiserv.tmpl -s /etc/scepfinal.txt
```

Action: Loads the preregistration records into PKI Services. If the template identified by the nickname `5YSCEPP` contains any subject or alternate name information in the **<CONSTANT>** section, that information overrides any matching information specified in the `/etc/scepfinal.txt` file.

Using the postcerts utility

Purpose

The `postcerts` program creates LDAP posting objects for certificates, which the PKI Services daemon later posts to an LDAP directory. (The `PostInterval` parameter in the **LDAP** section of the configuration file determines when the posting occurs.) You can use this utility if you have created certificates that PKI Services did not automatically post to an LDAP directory; for example if you created certificates before you configured PKI Services to automatically post them.

Path setup

Update your PATH, LIBPATH, and NLSPATH environment variables with the appropriate `pkiserv` directory before you run `postcerts`. (Note that you are updating the environment variables for the user running the utility, not updating values in the PKI Services environment variables file, `pkiserv.envars`.) Once you have updated these variables, you can run `postcerts` from the UNIX command line.

Variable name	You must add ...
PATH	*/install-dir*/pkiserv/bin
LIBPATH	*/install-dir*/pkiserv/lib
NLSPATH	*/install-dir*/pkiserv/lib/nls/msg/%L/%N

The default directory for *install-dir* is `/usr/lpp`.

Format

```
postcerts –s serial-numbers [-D CA-domain-name] [-c comment]
```

Parameters

–s *serial-numbers*
: The serial numbers of the certificates to be posted. This parameter contains one or more serial number specifiers, separated by commas, where a serial number specifier can be:
  - A single serial number
  - Two serial numbers separated by a dash, indicating a range of serial numbers

-D *CA-domain-name*
: The 1-8 character name of the CA domain for which certificates are to be posted. The name can be entered using uppercase or lowercase letters. This option is required only if PKI Services is running with multiple CA domains.

-c *comment*
: A comment to be stored in the issued certificate list (ICL).

Examples

To post a single certificate with serial number X'17':

```
postcerts –s 17
```

To post multiple certificates with a comment:

```
postcerts –s 17,18,8A –c 'Posting Certificates for Coop'
```

To post certificates whose serial numbers are in the range X'20' to X'3B':

```
postcerts –s 20-3B
```

To post certificates whose serial numbers are in the ranges X'20' to X'3B' and to also post a single certificate with serial number X'17':

```
postcerts –s 20-3B,17
```

Using the TemplateTool utility

Purpose

The TemplateTool utility works with certificate template files. It performs functions you need if you implement the Web application using Java server pages (JSPs). There are three certificate template files used by PKI Services:

- A text CGI template file, `pkiserv.tmpl`, used to implement the Web application using REXX CGI execs.
- An XML template file, `pkitmpl.xml`, used to implement the Web application using Java server pages (JSPs).
- A text CGI template file, `pkixgen.tmpl`, generated from the XML template file, and used by the PKI Services daemon if you implement the Web application using JSPs. Any time you update `pkitmpl.xml`, you must use TemplateTool to create an equivalent copy of `pkixgen.tmpl`. If you do not do this, the daemon writes the following message to the daemon log file when it determines that `pkixgen.tmpl` is not current:

  ```
  IKYC068I The templates file used may not be current
  ```

The TemplateTool utility performs the following functions:

- validateXML

  The validateXML function validates an XML template file against the XML schema file. It can optionally convert the XML template file to a text CGI template file. Use this function after you make changes to your XML schema file, `pkiserv.tmpl`, to validate the changes and create a CGI template file, `pkixgen.tmpl`, for use by the PKI Services daemon.
- convertCGI

  The convertCGI function converts a text CGI template file to an XML template file. This function can assist PKI Services administrators in converting the Web application from the REXX CGI execs to the JSPs. The conversion process might not convert all tags, so you should always verify the contents of the resulting XML template file.
- help

  The help function prints the format of the utility.

Requirement: You must have Java V5.0 or higher installed to run the TemplateTool utility.

Path setup

Update your PATH and CLASSPATH environment variables with the appropriate directories before you run `TemplateTool`. (Note that you are updating the environment variables for the user running the utility, not updating values in the PKI Services environment variables file, `pkiserv.envars`.) Once you have updated these variables, you can run `TemplateTool` from the shell script or from Java.

Variable name	You must add ...
PATH	*/install-dir*/pkiserv/bin
	The directory containing the `java` command.
CLASSPATH	*/install-dir*/pkiserv/lib/pki_xml.jar

The default directory for *install-dir* is `/usr/lpp`.

Format

```
TemplateTool -help |

             -validateXML [-XMLTemplate XML_template_file]
                          [-CGITemplate CGI_template_file]  |

             -convertCGI [-CGITemplate CGI_template_file]
                         [-XMLTemplate XML_template_file]
```

To run the utility from Java:

```
java com.ibm.pki.template.TemplateTool function parameters
```

To run the utility from the shell script:

```
TemplateTool function parameters
```

Parameters

-XMLTemplate *XML_template_file*
: An XML template file. Specify the name of the XML template file if it is in the current directory, or the full path name.

  For the validateXML function, this parameter is required. The specified XML template file is validated against the XML schema file `PKIServ.xsd`, which must be in the same directory.

  For the convertCGI function, this parameter is required. The CGI template file specified as input is converted to an XML template file using the XML schema file `PKIServ.xsd`, and output to the file specified by -XMLTemplate.

-CGITemplate *CGI_template_file*
: A CGI template file. Specify the name of the CGI template file if it is in the current directory, or the full path name.

  For the validateXML function, this parameter is optional. If specified, the XML template file specified as input is converted into CGI format and output to the file specified by -CGITemplate. If the CGI template file is to be used by the PKI Services daemon, it must be named `pkixgen.tmpl`.

  For the convertCGI function, this parameter is required. The CGI template file specified is converted to an XML template file using the XML schema file `PKIServ.xsd`, which must be in the same directory, and output to the file specified by -XMLTemplate.

Examples

Example:

```
TemplateTool -validateXML -XMLTemplate pkitmpl.xml -CGITemplate pkixgen.tmpl
```

Action: Invokes TemplateTool from the shell script. Validates that the XML template file `pkitmpl.xml` is valid, and generates a text CGI template file from it named `pkixgen.tmpl`, for use by the PKI Services daemon.

Example:

```
java com.ibm.pki.template.TemplateTool -convertCGI -CGITemplate pkiserv.tmpl -XMLtemplate pkitmpl.xml
```

Action: Invokes TemplateTool from Java. Converts the text CGI template file named `pkiserv.tmpl` to an XML template file named `pkitmpl.xml`.

Using the vosview utility

Purpose

The vosview program displays the data in the PKI Services object store (the request database). Depending on how you have configured PKI Services, the object store can be in a VSAM data set or in a DB2 table. Each request record consists of a fixed header, followed by a variable-length section. For each entry vosview displays the header information and optionally calls a user-provided program to process the BER-encoded request.

Path setup

Update your PATH, LIBPATH, and NLSPATH environment variables with the appropriate pkiserv directory before you run vosview. (Note that you are updating the environment variables for the user running the utility, not updating values in the PKI Services environment variables file, pkiserv.envars.) Once you have updated these variables, you can run vosview from the UNIX command line.

Variable name	You must add ...
PATH	/*install-dir*/pkiserv/bin
LIBPATH	/*install-dir*/pkiserv/lib
NLSPATH	/*install-dir*/pkiserv/lib/nls/msg/%L/%N

The default directory for *install-dir* is /usr/lpp.

Format

```
vosview {-d vsam-dataset-name [-r] | -b db2-subsystem-name -k db2-package-name
        | -c [-p path]}
        [-D CA-domain-name] [-s decode-command-string]
```

Parameters

You can display usage information about the vosview command format and parameters when you issue the vosview utility command with no parameters.

-d *vsam-dataset-name*
: Specifies the MVS data set name of the VSAM object store.

  If you specify -d, do not specify -b, -k or -c. If you do, the utility issues an error message and the command fails.

  Note: If the data set name has no quotes, the program uses the invoker of the command as the first qualifier. If you specify the fully qualified data set name, use quotes, and make sure to include the *escape* character, which is a backslash (\), before the quotation marks enclosing the data set name. For example, see \'pkisrvd.vsam.ost\' in "Examples" on page 373.

-r
: Indicates to open in record-level sharing (RLS) mode the VSAM data set specified with the -d option.

  -r is ignored and the utility issues a warning message if -b and -k are specified, or if -c is specified.

-b *db2-subsystem-name*
: Specifies the name of the DB2 subsystem where the object store is located.

If you specify -b and you do not specify -k, or you specify -d or -c, the utility issues an error message and the command fails.

-k *db2-package-name*
Specifies the DB2 package name of the object store.

If you specify -k and do not specify -b, or you specify -d or -c, the utility issues an error message and the command fails.

-c Indicates to retrieve the location of the object store from the pkiserv.conf configuration file. Either the VSAM data set name is retrieved, or the DB2 subsystem name and package name are retrieved, depending which you are using. For VSAM, the SharedPlex value determines whether the VSAM data set is opened in record-level sharing (RLS) mode. (For DB2, the SharedPlex value has no effect on this utility.)

If you specify -c, and you also specify -d, -b, or -k, the utility issues an error message and the command fails.

Note: When you also specify the -D option, you must use the -p option to specify the CA domain configuration directory if it is not /etc/pkiserv.

-p *path*
Specifies the directory where the pkiserv.conf configuration file resides. If not specified, the directory defaults to /etc/pkiserv. This option is only valid when specified with the -c option. If specified with the -b, -k, or -d options, the utility issues a warning message, and the -p option is ignored.

-D *CA-domain-name*
Specifies the CA domain name where this utility command is directed.

Notes:
1. The -D option is required only if PKI Services is running in multiple-CA mode.
2. The *CA-domain-name* value can be entered using uppercase or lowercase letters.
3. When you also specify the -c option, you must use the -p option to specify the CA domain configuration directory if not /etc/pkiserv.

-s *decode-command-string*
Specifies an optional command to call for decoding the ASN.1-encoded data. (The command must be able to read and decode binary (BER) data from STDIN.)

Examples

To view the records in the VSAM object store data set 'PKISRVD.VSAM.OST', passing the request data to a utility called dumpasn1, use the following command:

```
vosview -d \'pkisrvd.vsam.ost\' -s 'dumpasn1 -'
```

To view the records in the object store for the CA domain MasterCA using the information from the pkiserv.conf file located in the directory /etc/pkiserv/MasterCA, use the following command:

```
vosview -c -p /etc/pkiserv/MasterCA -D MasterCA
```

To view the records in the object store in the DB2 subsystem DSN9 with a package name of MasterCA, passing the request data to a utility called dumpasn1, and redirecting the output to the file vos.out, use the following command:

```
vosview -b DSN9 -k MASTERCA -s 'dumpasn1 -' >vos.out
```

Note: A `dumpasn1` utility is not shipped with PKI Services.

Output

The fixed header data is displayed for each record. Records 1 and 2 contain system data only. They do not represent certificate requests. Record 2 contains internal system data only.

Sample record 1

```
Object key = 1
Last used key = 726, CRL serial number = 518, ARL serial number =  518,
High DP = 24, Low DP = 7
name = ""
tid = ????????????????????????
appldata =
comment =
data len = 20
flags = 0 - Type = ???? ObjSt???????
Creation time is: 2002/04/22 17:29:48
Last modified time is: 2002/04/29 18:23:49
```

Last used key
The primary index for the last record in the data set.

CRL serial number
The number to be used for the next CRL.

ARL serial number
The number to be used for the next ARL.

High DP The number of the highest distribution point CRL issued by PKI Services.

Low DP The number of the lowest currently active distribution point CRL.

The remaining fields contain no meaningful data.

Sample certificate request record

```
Object key = 105
name = "John Q. Public"
tid = 1F45AEF2D3729FA35156BC47
appldata = "1YBSSL"
comment = ""
data len = 570
flags = 1020111 - Type = Cert State = RA CertReqActive [State Flag]
```

Object key The index into the VSAM data set name.

name The requestor's name.

tid The transaction ID data.

appldata Indicates the 8-character string identifying to the application the short name or nickname of the certificate template. (PKI Services provides sample certificate templates but it is RACF, or an equivalent security product, rather than PKI Services, that handles the SAF templates.) The following table shows the nicknames for each certificate template. (These nicknames are supplied in the `pkiserv.tmpl` certificate templates file as defaults but your

installation might have changed them or added others during customization. See "TEMPLATE sections" on page 136 for more information.)

Table 71. Names, aliases, and nicknames of certificate templates

True name	Alias	Nickname
1-Year PKI SSL Browser Certificate	PKI Browser Certificate	1YBSSL
1-Year PKI S/MIME Browser Certificate	PKI Browser Certificate	1YBSM
1-Year PKI Generated Key Certificate	PKI Key Certificate	1YKRC
2-Year PKI Browser Certificate For Authenticating To z/OS	PKI Browser Certificate	2YBZOS
2-Year PKI Authenticode - Code Signing Certificate	PKI Server Certificate	2YIACS
2-Year PKI Windows Logon Certificate	PKI Browser Certificate	2YBWL
5-Year PKI SSL Server Certificate	PKI Server Certificate	5YSSSL
5-Year PKI IPSEC Server (Firewall) Certificate	PKI Server Certificate	5YSIPS
5-Year PKI Intermediate CA Certificate	PKI Server Certificate	5YSCA
5-Year SCEP Certificate - Preregistration	—	5YSCEPP
n-Year PKI Certificate for Extensions Demonstration	PKI Browser Certificate	SAMPLB
1-Year SAF Browser Certificate	SAF Browser Certificate	—
1-Year SAF Server Certificate	SAF Server Certificate	—

comment A comment the administrator supplied the last time the request was updated.

data len The length of the variable data portion (that is, the BER-encoded request).

flags Represent the current state of the request:

Type

Cert Certificate request (new or renewal).

CRL Certificate revocation list (CRL).

Rev Revocation request.

Post Certificate waiting to be posted to LDAP.

State The prefix (RA or CA) and one of the following:

CertPreregistered Certificate preregistration record.

CertReqActive Certificate request in some state of being completed.

CertSigned Certificate request where the certificate has been created.

CertReqRejected	Certificate request that has been rejected.
RevReqActive	Revocation request in some state of being completed.
CRLWaitingForRA	CRL to be posted to LDAP.
CertPostPending	Certificate to be posted to LDAP.
CaInfoPostPending	PKI Services' CA certificate to be posted to LDAP.

State Flag Optional. If present, is one of the following:

Complete	Request is complete. For approved requests, the end user has retrieved the certificate.
Error	The certificate could not be posted to LDAP.
NeedsConfirm	Approved or rejected. End user has yet to be notified of the final outcome.
AutoRenewEnabled	The certificate returned automatic renewal and this capability is enabled.
AutoRenewCapable	The certificate returned active certificates capable for auto renewal but disabled.
Synchronous	Request is an in-progress synchronous certificate request.

Using the vsam2db2 utility

Purpose

The vsam2db2 program copies data from the issued certificate list (ICL) and object store VSAM data sets into DB2 tables. Use this utility if you have been using VSAM data sets for the ICL and object store and want to use DB2 tables instead. Run vsam2db2 when PKI Services is not running. You must create the DB2 tables before you run vsam2db2. For more information, see "Converting the object store and ICL from VSAM to DB2" on page 113.

Path setup

Update your PATH, LIBPATH, and NLSPATH environment variables with the appropriate pkiserv directory before you run vsam2db2. (Note that you are updating the environment variables for the user running the utility, not updating values in the PKI Services environment variables file, pkiserv.envars.) Once you have updated these variables, you can run vsam2db2 from the UNIX command line.

Variable name	You must add ...
PATH	/*install-dir*/pkiserv/bin
LIBPATH	/*install-dir*/pkiserv/lib
NLSPATH	/*install-dir*/pkiserv/lib/nls/msg/%L/%N

The default directory for *install-dir* is /usr/lpp.

Format

```
vsam2db2 -o ostvsam-dataset-name -i iclvsam-dataset-name [-r]
         -b db2-subsystem-name -k db2-package-name
         [-D CA-domain-name] [-a]
```

Parameters

-o *ostvsam-dataset-name*
: Specifies the MVS data set name of the VSAM object store.

-i *iclvsam-dataset-name*
: Specifies the MVS data set name of the VSAM issued certificate list (ICL).

-r
: Specifies that the VSAM data set is to be opened in record-level sharing (RLS) mode.

-b *db2-subsystem-name*
: Specifies the DB2 subsystem name of the object store and issued certificate list (ICL)

-k *db2-package-name*
: Specifies the DB2 package name of the object store and issued certificate list (ICL)

-D *CA-domain-name*
: Specifies the CA domain name

-a
: If specified, all records in the object store and issued certificate list (ICL) are copied from the VSAM data sets to the DB2 tables. If not specified:
  - For the object store, all objects except completed certificate request objects are copied.

- For the ICL, all certificate objects except expired certificates for which the user provided the public key are copied.

Note: It will take longer to run the utility with the `-a` option if you have not removed the completed certificate requests and the expired certificates for which the user provided the public key from the VSAM data sets.

Examples

In this sample script, the object store for the domain MasterCA is in the VSAM file `pkisrvd.vsam.ost`, and the ICL is in the VSAM file `pkisrvd.vsam.icl`. They are to be copied to DB2 tables in the DB2 subsystem DSN9 and the DB2 package MasterCA. The VSAM data sets are to be opened in RLS mode. All records in the VSAM data sets are to be copied, including those for completed requests and expired certificates.

```
vsam2db2 -o \'pkisrvd.vsam.ost\' \
         -i \'pkisrvd.vsam.icl\' \
         -b DSN9 \
         -k MasterCA \
         -r \
         -D MasterCA \
         -a
```

Output

As the `vsam2db2` utility runs, it reports its progress. For every 2000 records copied from VSAM data sets to DB2 tables, the utility reports the total number of records copied.

If an error occurs, `vsam2db2` displays a count of processed records and a count of copied records. (If `-a` is specified, these counts should be the same.)

Chapter 19. Using the certificate management protocol (CMP) with PKI Services

Certificate management protocol (CMP) is an internet protocol used to manage X.509 digital certificates within a PKI. It is described in RFC 4210 and uses the certificate request message format (CRMF) described in RFC 4211. A certificate request message object is used within the protocol to convey a request for a certificate to a certificate authority. CMP messages are ASN.1-encoded. PKI Services allows a CMP client to communicate with it to request, revoke, suspend and resume certificates.

Restrictions: The following restrictions apply to the PKI Services support for CMP:

1. PKI Services supports only a subset of the CMP messages, and only some fields in those messages. See "Support for CMP messages" on page 380 for a description of the support.
2. PKI Services supports only the HTTP protocol for CMP messages.

PKI Services implements CMP through a CGI program. The tcp-message is sent to the PKI CMP CGI program by HTTPS POST, as specified in *Internet X.509 Public Key Infrastructure -- Transport Protocols for CMP* . The entire POST body is the message and the mime-type for both requester and responder (client and server) is `application/pkixcmp`.

Note: The `application/pkixcmp` mime-type requires that the entire tcp-message be Base64-encoded.

When a CMP client sends a request to the HTTP Server, it must send the request directly to the HTTP Server (and port number) that handles the client authentication requests. The request cannot be handled by a redirect statement.

Table 72 shows the format of version 10 tcp-messages (the only existing version):

Table 72. Format of tcp-messages

Field	Length	Contents
Length	32 bits	The length of the rest of the tcp-message (the length of the CMP message + 3)
Version	8 bits	10
Flags	8 bits	The least significant bit indicates a closed connection, the other bits are unused.
Message type	8 bits	Supported types are: • `pkiReq`, value X'00', indicating a synchronous request • `pkiRep`, value X'05', indicating a synchronous response • `errorMsgRep`, value X'06', indicating an error

© Copyright IBM Corp. 2001, 2011

Table 72. Format of tcp-messages (continued)

Field	Length	Contents
Value	Variable, based on the length of the CMP message	The ASN.1 encoded CMP message

The communication between the CMP client and the CGI program is over HTTPS only. Client authentication is required. The client (the CMP requester) needs to have a certificate installed in RACF under the client's ID. This certificate is used by the requester to authenticate itself, and its owner ID is used to access the PKI Services functions.

Support for CMP messages

PKI Services supports the following request message types from the client:
- Certificate request message (type `cr`)
- Revocation request message (type `rr`)
- PKCS #10 certificate request message (type `p10cr`)

and responds with one of the following response messages:
- Certificate response message (type `cp`)
- Revocation response message (type `rp`)
- Error message (type `error`)

Each message supported by PKI Services contains the following parts:
- The header, containing information common to many messages
- The body, containing information specific to the message
- Optionally, certificates that might be useful to the recipient

Table 73 identifies the fields in the `PKIMessage` structure defined in RFC 4210 that PKI Services supports.

Table 73. Supported fields in the `PKIMessage` structure

Field name	Notes
`header`	See Table 74 on page 381.
`body`	See Table 75 on page 381.
`extraCerts`	This field can be used by the client on a certificate request message (cr) when PKI Services is generating the public and private key for the requested certificate. This field can contain a list of x.509 certificates to be used as recipients of the private key to be returned by PKI Services. If this field is present, the _PKISERV_CMP_HONOR_CLIENT_CERTS environment variable determines whether extra certificates are allowed, and how many are allowed. If allowed, and if PKI Services generates the public and private key pair for the request, each certificate has a recipientInfo structure added to the returned encrypted private key (PKCS #7 EnvelopedData structure). For information about the _PKISERV_CMP_HONOR_CLIENT_CERTS environment variable, seeTable 84 on page 396.

Table 74 on page 381 identifies the fields in the `PKIHeader` structure defined in RFC 4210 that PKI Services supports.

Table 74. Supported fields in the PKIHeader structure

Field name	Notes
pvno	
sender	
recipient	If this GeneralName field is in the form of a directoryName in a cr, p10cr, or rr message, it can be used to determine the PKI Services CA domain to which the request is directed. For information on how PKI Services determines the CA domain, see "Determining the CA domain to which a request is routed" on page 385.
transactionID	
generalInfo	The only InfoTypeAndValue recognized by PKI Services is ImplicitConfirm, which is required for cr and p10cr messages. This field is ignored if present on an rr request message.

Table 75 identifies the values in the PKIBody structure defined in RFC 4210 that PKI Services supports. These are the CMP message types.

Table 75. Supported values in the PKIBody structure. These are the CMP message types that PKI Services supports.

Value	Description	Notes
cr	Certificate request	See Table 76.
cp	Certificate response	See Table 78 on page 383.
p10cr	PKCS #10 certificate request	See Table 77 on page 383.
rr	Revocation request	See Table 79 on page 384.
rp	Revocation response	See Table 80 on page 384.
error	Error message	See Table 81 on page 385.

Support for the CMP certificate request message (type cr)

Table 76 identifies the fields that PKI Services supports in the data structure defined in RFC 4211 for the CMP certificate request message (type cr).

Table 76. Supported fields in the CMP certificate request message (type cr)

Field name		Notes
In the CertReqMsg structure:		PKI Services supports a single CertReqMsg in the CertReqMessages field, and rejects a cr message with more than one CertReqMsg.
	certReq	
	popo	
In the ProofOfPossession structure:		

Table 76. Supported fields in the CMP certificate request message (type cr) (continued)

Field name		Notes
	signature	signature is the only supported choice. It should only be present if the CMP client has supplied publicKey in the CertTemplate structure. The POPOSigningKey structure must not contain a poposkInput field.
In the CertRequest structure:		
	certReqId	
	certTemplate	
In the CertTemplate structure:		
	version	
	serialNumber	
	signingAlg	
	issuer	If supplied, this field is used in conjunction with the _PKISERV_CMP_DOMAIN_ISSUER*n* environment variables to determine to which PKI Services CA domain to route the request. For information about the _PKISERV_CMP_DOMAIN_ISSUER*n* environment variables, see Table 82 on page 392. For information about how PKI Services determines the CA domain, see "Determining the CA domain to which a request is routed" on page 385.
	validity	If supplied, the _PKISERV_CMP_HONOR_CLIENT_DATES environment variable must set to **1**; otherwise the cr message is rejected. For information about the _PKISERV_CMP_HONOR_CLIENT_DATES environment variable, see Table 83 on page 393.
	subject	If omitted, the cr message is rejected.
	publicKey	Optional; if omitted PKI Services generates the public and private keys for the certificate request using environment variables to determine the key type and size.
	extensions	If the _PKI_CMP_HONOR_CLIENT_EXTS environment variable is not set to **1** and extensions is specified, the message is rejected. If the environment variable is set to **1**, extensions is honored if present, but is not required.

Support for the CMP PKCS #10 certificate request message (type p10cr)

Table 77 identifies the fields that PKI Services supports in the data structure defined in RFC 4210 for the CMP PKCS #10 certificate request message (type p10cr)).

Table 77. Supported fields in the CMP PKCS #10 certificate request message (type p10cr)

Field name		Notes
In the CertificationRequest structure:		When this message type is received by PKI Services, the recipient field is used in conjunction with the _PKISERV_CMP_DOMAIN_ISSUER*n* environment variables to determine to which PKI Services CA domain to route the request, provided the recipient is encoded as a directoryName. For information about the _PKISERV_CMP_DOMAIN_ISSUER*n* environment variables, see Table 82 on page 392.
	certificationRequestInfo	
	signatureAlgorithm	
	signature	
In the CertificationRequestInfo structure:		
	version	
	subject	
	subjectPKInfo	
	attributes	If the _PKI_CMP_HONOR_CLIENT_EXTS environment variable is not set to **1** and the extensionReq attribute is specified, the message is rejected. If the environment variable is set to **1,** the reqExtensions attribute is honored if present, but is not required.

Support for the CMP certificate response message (type cp)

Table 78 identifies the fields that PKI Services supports in the data structure defined in RFC 4210 for the CMP certificate response message (type cp). The cp message is returned to the CMP client for a successful cr or p10cr request.

Table 78. Support for fields in the CMP certificate response message (type cp)

Field name		Notes
In the CertRepMessage structure:		privateKey is only returned in response to a cr message that does not specify publicKey.
	caPubs	PKI Services does not use this element.
	response	
In the CertResponse structure:		PKI Services returns only one CertResponse for a certificate request.
	certReqId	

Table 78. Support for fields in the CMP certificate response message (type cp) (continued)

Field name		Notes
	status	
	certifiedKeyPair	
	rspInfo	PKI Services does not use this element.
In the CertifiedKeyPair structure:		
	certOrEncCert	
	privateKey	PKI Services sets the object ID for intendedAlg to the PKCS #7 OID, which is 1.2.840.113549.1.7. encValue is a bit string encapsulation of a PKCS #7 EnvelopedData structure whose encrypted content is the DER-encoded private key for the issued certificate.

Support for the CMP revocation request message (type rr)

Table 79 identifies the fields that PKI Services supports in the data structure defined in RFC 4210 for the CMP revocation request message (typerr).

Table 79. Supported fields in the CMP revocation request message (type rr)

Field name		Notes
In the RevDetails structure:		PKI Services supports a single RevDetails in the RevReqContent message, and rejects an rr message with more than one RevDetails sequence.
	certDetails	PKI Services requires that certDetails contains the serial number of the certificate to be revoked, suspended, or resumed.
	crlEntryDetails	PKI Services recognizes only the cRLReason extension and ignores all other non-critical extensions in crlEntryDetails. If an extension other than the cRLReason extension is present and marked critical, PKI Services rejects the rr message.

Support for the CMP revocation response message (type rp)

Table 80 identifies the fields that PKI Services supports in the data structure defined in RFC 4210 for the CMP revocation response message (typerp). PKI Services returns the rp message to the CMP client for a successful rr request.

If PKI Services receives a type rp message from a CMP client, it returns an errorMessage response.

Table 80. Supported fields in the CMP revocation response message (type rp)

Field name		Notes
In the RevRepContent structure:		
	status	

Table 80. Supported fields in the CMP revocation response message (type rp) (continued)

Field name		Notes
	revCerts	PKI Services returns revCerts in the rp message when the corresponding rr message contains an issuer in certDetails or specifies a recipient in the form of a directoryName.
In the PKIStatusInfo structure:		
	status	
	statusString	
	failInfo	

Support for the CMP error message (type error)

Table 81 identifies the fields that PKI Services supports in the data structure defined in RFC 4210 for the CMP revocation request message (type error). The error codes and details that can be returned are described in "Messages and codes returned from the CMP functions" on page 397.

If PKI Services receives a type error message from a CMP client, it returns an errorMessage response.

Table 81. Supported fields in the CMP error message (type error)

Field name		Notes
In the ErrorMsgContent structure:		
	pKIStatusInfo	
	errorCode	This field is present in all type error messages that PKI Services returns. For the error codes and corresponding details that can be returned, see "Messages and codes returned from the CMP functions" on page 397.
	errorDetails	

Determining the CA domain to which a request is routed

To determine the CA domain to which it will route a request, the CMP CGI program first tries to determine the issuer distinguished name:

- If the message type is cr (certificate request), the issuer field of the CertTemplate structure is used as the issuer distinguished name if it is present. If it is not, the recipient field in the message header is used as the issuer distinguished name if it is in the form of a Directory Name (distinguished name). If the recipient field is not in the form of a Directory Name, an issuer distinguished name is not used to determine the CA domain name; instead, the URL to which the CMP request was sent is used to determine the CA domain.
- If the message type is p10cr (PKCS #10 certificate request message), the recipient field in the message header is used as the issuer distinguished name if it is in the form of a Directory Name (distinguished name). If the recipient field is not in the form of a Directory Name, an issuer distinguished name is not used to determine the CA domain name; instead, the URL to which the CMP request was sent is used to determine the CA domain.

- If the message type is `rr` (revoke request), the issuer and serial number fields of the `CertDetails` field are used as the issuer distinguished name and certificate serial number to be revoked or suspended. If the serial number is not present, the request is rejected. If the issuer field is not present, the `recipient` field in the message header is used as the issuer distinguished name if it is in the form of a Directory Name (distinguished name). If the `recipient` field is not in the form of a Directory Name, an issuer distinguished name is not used to determine the CA domain name; instead, the URL to which the CMP request was sent is used to determine the CA domain.

If the CMP CGI program was able to determine the issuer distinguished name, and the request is a certificate request (type `cr` or `p10cr`), the CMP CGI program does the following processing to determine to which CA domain it routes the request:

1. The CMP CGI program compares the issuer distinguished name extracted from the request in string format to the values defined in the _PKISERV_CMP_DOMAIN_ISSUER*i* environment variables (where *i* is 1 through the number of CA domains). The comparison is made by comparing the relative distinguished names in order of specification (first from most specific to least specific, then least specific to most specific). For example, if the issuer distinguished name in a request message is OU=STG,O=IBM,C=US, it would match a _PKISERV_CMP_DOMAIN_ISSUER*i* environment variable whose value was set to either of the following values:
   - OU=STG, O=IBM, C=US (most specific first)
   - c=us,o=ibm,ou=stg (least specific first)

   The comparison is made without regard to the character case (case-insensitive). Some differences in spacing are allowed. For example, “O=IBM” matches “O = IBM”, but not “O=I B M”

   If a match is found for the issuer distinguished name, the CMP CGI continues to step 2. If no match is found, it uses the URL to which the CMP request was sent to determine the CA domain.
2. The number (*i*) of the matching _PKISERV_CMP_DOMAIN_ISSUER*i* environment variable is used to read the domain name environment variable _PKISERV_CMP_DOMAIN_NAME*i_n*. The _PKISERV_CMP_DOMAIN_NAME*i_n* with the highest value of *n* is used because it represents the current domain name for the CA for accepting new certificate requests. (When the value of *n* is greater than 1, at least one CA rollover has occurred.) For example, if the issuer distinguished name in a `cr` message matched the value in _PKISERV_CMP_DOMAIN_ISSUER3, and one CA rollover has occurred for that CA, the CA domain name is retrieved from the _PKISERV_CMP_DOMAIN_NAME3_2 environment variable. If the _PKISERV_CMP_DOMAIN_NAME*i_n* environment variable cannot be read, the CMP request is rejected.

If the CMP CGI program was able to determine the issuer distinguished name, and the request is a revoke request (type `rr`), and a serial number was present in the request, the CMP CGI program does the following processing to determine to which CA domain it routes the request:

1. The CMP CGI program compares the issuer distinguished name extracted from the request in string format to the values defined in the _PKISERV_CMP_DOMAIN_ISSUER*i* environment variables (where *i* is 1 through the number of CA domains). The comparison is made in the same manner as described in step 1 for `cr` and `p10cr` requests. If a match is found for

the issuer distinguished name, the CMP CGI continues to step 2. If no match is found, it uses the URL to which the CMP request was sent to determine the CA domain.

2. The number (*i*) of the matching _PKISERV_CMP_DOMAIN_ISSUER*i* environment variable is then used to read the serial number domain name environment variables _PKISERV_CMP_DOMAIN_FSTSN*i_n*. Each _PKISERV_CMP_DOMAIN_FSTSN*i_n* environment variable is read, starting with an *n* value of 1, and compared to the serial number retrieved from the `rr` request.
   - If there are no _PKISERV_CMP_DOMAIN_FSTSN*i_n* environment variables defined, the domain name is read from the _PKISERV_CMP_DOMAIN_NAME*i*_1 environment variable.
   - If the serial number is greater than the environment variable value and less than the next environment variable value (or is the last environment variable), the domain name is read from the corresponding _PKISERV_CMP_DOMAIN_NAME*i_n* environment variable.

     Example: An `rr` request is made by a CMP client specifying an issuer distinguished name of OU=STG, O=IBM, C=US, and the following is an excerpt from the defined environment variables:

     ```
     _PKISERV_CMP_DOMAIN_ISSUER3= OU=STG, O=IBM, C=US
     _PKISERV_CMP_DOMAIN_NAME3_1=STG_CA
     _PKISERV_CMP_DOMAIN_NAME3_2=STG_CA2
     _PKISERV_CMP_DOMAIN_NAME3_3=STG_CA3
     _PKISERV_CMP_DOMAIN_FSTSN3_1=3
     _PKISERV_CMP_DOMAIN_FSTSN3_2=12500
     _PKISERV_CMP_DOMAIN_FSTSN3_3=25000
     ```

     If the serial number specified in the `rr` request is 20000 decimal, the revocation request is routed to the STG_CA2 domain because STG_CA2's first serial number is less than 20000, and the first serial number issued by STG_CA3 is greater than 20000.

How PKI Services interprets distinguished names (DNs) on CMP requests

The subject distinguished name is encoded in a CMP certificate request from the CMP client. The order in which the relative distinguished names (RDNs) are placed in the subject field by the client is the order in which the RDNs appear in the issued certificate. PKI Services interprets the order to be least significant RDN first and most significant RDN last. This becomes important when PKI Services posts the issued certificate to an LDAP server, because the LDAP server defines a root suffix that it will allow objects and attributes to stored under. If the LDAP server has a defined root suffix of "C=US", and PKI Services attempts to post a certificate with a subject name "CN=Gumby,O=IBM,C=US", the request will likely succeed because the string format of the subject name has the C=US as the right-most RDN, and that is the defined LDAP root suffix. If however the CMP client encoded the subject name in the reverse order, the subject name string that PKI uses to post the certificate would be "C=US,O=IBM,CN=Gumby". This post request will likely fail because the interpreted root suffix of CN=Gumby would not exist in LDAP. **Guideline:** When encoding a subject distinguished name in a certificate request, clients should place the least significant RDN first and the most significant RDN last.

Example: This example shows the encoded form of `CN=Gumby,O=IBM,C=US`:

```
SEQUENCE {
. SET {
. . SEQUENCE {
. . . OBJECT IDENTIFIER countryName (2 5 4 6)
. . . PrintableString 'US'
. . . }
. . }
. SET {
. . SEQUENCE {
. . . OBJECT IDENTIFIER organizationName (2 5 4 10)
. . . PrintableString 'IBM'
. . . }
. . }
. SET {
. . SEQUENCE {
. . . OBJECT IDENTIFIER commonName (2 5 4 3)
. . . PrintableString 'Gumby'
. . . }
. . }
. }
```

When the PKI Services CMP CGI program receives a CMP request from a CMP client, it attempts to determine the target CA domain to route the request to using the supplied issuer DN or the recipient (if it is in the form of a DN). The CMP CGI program interprets the encoded DN values to be the least significant RDN first and most significant RDN last. The program builds a string representation of the issuer or recipient DN from left to right starting with the last RDN in the sequence and ending with the first RDN. The CMP CGI program then compares the string it built to the values of the _PKISERV_CMP_DOMAIN_ISSUER*x* environment variables. During the comparison of the strings, the program tries the comparison in both ways, both right to left and left to right, so if the distinguished name in the request is OU=Master CA,O=IBM,C=US it matches a _PKISERV_CMP_DOMAIN_ISSUER*x* value of either OU=Master CA,O=IBM,C=US, or C=US,O=IBM,OU=Master CA.

Setting up a client to make CMP requests to PKI Services

Before a client (the CMP requester) can make requests to PKI Services through CMP, the client must have a certificate installed in the RACF database under the client's RACF user ID. The certificate is used by the requester to authenticate the client, and the client's user ID is used to access the PKI Services functions. It must be signed by a CA certificate that is connected to the HTTP Server's key ring.

There are several ways to set up the client certificate. For example, you can have it generated elsewhere (not by PKI Services) and add it to the RACF database using the RACDCERT ADD command. Alternatively, you can generate the certificate yourself and add it to the RACF database using RACDCERT commands. The following instructions illustrate the latter approach.

Steps for setting up a certificate for a CMP requester

Before you begin: You need to have RACF SPECIAL authorization, or authorization to the RACDCERT commands shown. For more information, see *z/OS Security Server RACF Command Language Reference*.

Perform the following steps to set up a certificate in the RACF database for a CMP requester.

1. Generate a certificate for the CMP requester in the RACF database, signed by a certificate that is in the HTTP Server's key ring.

Example:

```
RACDCERT ID(User123) GENCERT
SUBJECT(CN('Messager') OU('OrgUnitA') O('OrgA') C('AU'))
WITHLABEL('client') SIGNWITH(CERTAUTH LABEL('Master PKI CA'))
```

---

2. Export the certificate and its private key to a data set.

   Example:

   ```
   RACDCERT ID(User123) EXPORT(LABEL('client'))
   DSN('User123.private.eecert') FORMAT(PKCS12B64)
   PASSWORD('secret')
   ```

   Tip: You could use `FORMAT(PKCS12DER)` if you don't plan to use copy and paste operations to transport the certificate and private key to the client system.

---

3. Transport the certificate to the system where the CMP client will run. You can do this using FTP, or perhaps by copy and paste operations, depending on the CMP client software.

When you are done, the client can make CMP requests to PKI Services.

Setting up PKI Services to process CMP requests

Before PKI Services can process CMP requests, you must do the following tasks:

1. Enable the CMP support by setting the `EnableCMP` configuration parameter.
2. If you want PKI Services to create private keys for CMP clients, you must set up ICSF and set up PKI Services to encrypt the private keys.

Enabling the CMP support

The PKI Services configuration file, `pkiserv.conf`, contains a parameter in the **CertPolicy** section that enables the CMP support. By default this parameter, `EnableCMP`, is set to false. If the parameter is set to false or omitted from the configuration file, PKI Services rejects all CMP requests.

To enable the CMP support, find the following lines in the configuration file:

```
# Enable the Certificate Management Protocol (CMP)
# T = True, CMP is enabled
# F = False, CMP is disabled (default if not specified)
#
EnableCMP=F
```

Change `EnableCMP=F` to `EnableCMP=T`, and restart PKI Services.

Setting up PKI Services to create private keys for CMP clients

PKI Services can create private keys for CMP clients and return a private key with a certificate. It uses the PKCS #11 API provided by ICSF to create private keys. Note, however, that PKI Services does not archive the private keys in the ICSF token data set (TKDS), as it does for private keys that it creates for certificate requests it receives from the end-user Web application. To allow PKI Services to create private keys, you must ensure that the ICSF programmer has installed and configured ICSF, and has set up the TKDS. For more information, see “Installing and configuring ICSF (optional)” on page 23.

Note: You do not need to perform any of the other tasks described in "Steps for setting up PKI Services to generate keys for certificate requests" on page 259, such as setting the `TokenName` parameter in the configuration file, to allow the PKI Services CMP CGI program to generate private keys for CMP clients. Those tasks apply only to private key generation done by the PKI Services daemon, for certificates requested via the PKI Services Web application.

Determining the source of certificates used to encrypt the returned private key

If a CMP client requests a certificate for which PKI Services creates the keys, PKI Services must encrypt the private key before returning it with the certificate to the recipient. There are three ways that the encryption of the private key can be done, and the method used is determined in the following order:

1. If the client provides an `extraCerts` field in the `PKIMessage` structure and the number of certificates in the `extraCerts` field does not exceed the value of the _PKISERV_CMP_HONOR_CLIENT_CERTS_*domain_name* environment variable, the CMP CGI program encrypts the private key in a manner in which it can be decrypted by each of the certificates in the `extraCerts` field.
2. If the `extraCerts` field is not specified in the request message, and the _PKISERV_CMP_KEYRING_*domain_name* environment variable is set, the private key is encrypted in a manner in which it can be decrypted by each of the certificates in the specified key ring.
3. If the `extraCerts` field is not specified in the request message, and no key ring is defined, the private key is encrypted with the public key of the certificate used to establish the secure client authentication session to the CMP CGI program.

Steps for setting up PKI Services to encrypt returned private keys with certificates in a key ring

For PKI Services to encrypt private keys using certificates in a key ring, you must set up a key ring containing a digital certificate for each recipient.

Before you begin:

- You need to have RACF SPECIAL authorization, or sufficient authority to the following resources in the FACILITY class:
  - IRR.DIGTCERT.ADDRING
  - IRR.DIGTCERT.ADD
  - IRR.DIGTCERT.CONNECT
- You must have a certificate for each certificate recipient, in a data set.

Note: This example assumes that you are using certificates that were created somewhere else. Alternatively, you could create the certificates using the RACF command RACDCERT GENCERT. If you take this approach, you need authorization to the resource IRR.DIGTCERT.GENCERT in the FACILITY class.

Perform the following steps to set up PKI Services to encrypt returned private keys from certificates in a key ring.

1. Set the HTTP Server environment variable _PKISERV_CMP_KEYRING:

   ```
   _PKISERV_CMP_KEYRING_domain_name=RACF_userID/ring_name
   ```

RACF_userID can be any RACF user ID; for example, the PKI Services daemon user ID, or the CMP requester user ID. *ring_name* is a name that you choose, and use when you create the key ring.

---

2. Create a RACF key ring, specifying the RACF user ID and the ring name that you specified in step 1 on page 390.

```
RACDCERT ID(RACF_userID) ADDRING(ring_name)
```

---

3. Add certificates for each recipient to the RACF database, using the RACF user ID that you specified in step 1 on page 390.

```
RACDCERT ID(RACF_userID) ADD(dataset_1) WITHLABEL('label_1')TRUST
RACDCERT ID(RACF_userID) ADD(dataset_2) WITHLABEL('label_2')TRUST
⋮
RACDCERT ID(RACF_userID) ADD(dataset_n) WITHLABEL('label_n')TRUST
```

dataset_n is the name of the data set containing the certificate for recipient *n*. *label_n* is the label to be associated with the certificate for recipient *n*.

---

4. Add the digital certificates to the key ring that you created in step 2.

```
RACDCERT ID(RACF_userID) CONNECT(LABEL('label_1') RING(ring_name))
RACDCERT ID(RACF_userID) CONNECT(LABEL('label_2') RING(ring_name))
⋮
RACDCERT ID(RACF_userID) CONNECT(LABEL('label_n') RING(ring_name))
```

label_n is the label you associated with the certificate for recipient *n* in step 3.

---

5. Authorize the PKI Services CMP CGI program to access the key ring. This program runs with the RACF user ID that the client-supplied certificate maps to, so you must give that RACF user ID access to the key ring. You can use one of two methods:

   - (Preferred) Define a profile in the RDATALIB class for the key ring and give each CMP client user ID READ access:

     ```
     RDEFINE RDATALIB ring_owner.ring_name.LST UACC(NONE)
     PERMIT ring_owner.ring_name.LST CLASS(RDATALIB) ID(client_user_id) ACCESS(READ)
     ```

     ring_owner is the RACF user ID you that specified in step 2, and *ring_name* is the ring name that you specified.

   - (Alternative) Define a profile in the FACILITY class for IRR.DIGTCERT.LISTRING. Give the ring owner READ access and the client user IDs UPDATE access:

     ```
     RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
     PERMIT IRR.DIGTCERT.LISTRING ID(ring_owner) ACCESS(READ)
     PERMIT IRR.DIGTCERT.LISTRING ID(client_user_id) ACCESS(UPDATE)
     ```

---

6. If either the DIGTCERT or DIGTRING class is RACLISTed, refresh the RACLISTed classes to activate your changes.

```
SETROPTS RACLIST(DIGTCERT, DIGTRING) REFRESH
```

When you are done, you have set up PKI Services to encrypt returned private keys with certificates in a RACF key ring.

Setting up the HTTP Server for CMP

You can pass information about CMP requests through HTTP Server environment variables. The default name for the HTTP Server environment variables file is `httpd.envvars`. (The actual name might be different and is typically defined in the local version of the IMWEBSRV started procedure.) The variables that you can set are described in Table 82, Table 83 on page 393, and Table 84 on page 396. For information about HTTP Server environment variables, see *z/OS HTTP Server Planning, Installing, and Using*.

Table 82. HTTP Server environment variables used to determine the CA

Environment variable name	Description
_PKISERV_CMP_DOMAIN_ISSUER*x*	CA's subject distinguished name with comma-separated RDNs **Example:** `_PKISERV_CMP_DOMAIN_ISSUER1=CN=Issuer CA,OU=Lab,O=IBM,C=AU`

_PKISERV_CMP_DOMAIN_NAME*x_y*	Issuer's PKI CA domain name. The domain name variable is limited to 8 characters beginning with an alphabetic character or an underscore "_" and no imbedded spaces. For an unnamed domain, this variable must be set to `<NONE>`. A rollover of an unnamed domain requires the new domain to be named.
	
	Example: If there are, or have been, 3 domains of certificates for the same issuer, there would be 3 entries in the environment variables file:
	`_PKISERV_CMP_DOMAIN_NAME1_1=CardCA1`
	`_PKISERV_CMP_DOMAIN_NAME1_2=CardCA2`
	`_PKISERV_CMP_DOMAIN_NAME1_3=CardCA3`
	
	Example: The domain is unnamed.
	`PKISERV_CMP_DOMAIN_NAME1_1=<NONE>`

_PKISERV_CMP_DOMAIN_FSTSN*x_y*	Starting serial number, in decimal, for the first certificate issued by _PKISERV_CMP_DOMAIN_NAME*x_y*. **Example:** `PKISERV_CMP_DOMAIN_FSTSN1_2=1001`

Notes:

1. *x* represents any of the available certificate issuers from 1 to the maximum number of issuers. If there are 2 issuers of certificates, there would be 2 entries in the configuration file. The values of *x* are consecutive integers starting with 1.
2. *y* represents any of the available CA domains that issue or have issued certificates on behalf of _PKISERV_CMP_DOMAIN_ISSUER*x*. The values of *y* are consecutive integers starting with 1.
3. Environment variable values must not be enclosed in quotes even if they include whitespace.
4. Any comma in a relative distinguished name (RDN) value must be escaped by placing a backslash immediately before the comma in the _PKISERV_CMP_DOMAIN_ISSUER*x* variable. For example, if your Organization RDN value is "Widgets, Inc.", the RDN must be specified as "O=Widgets\, Inc.". The commas that separate the RDNs must not be escaped.

   Example:

   ```
   CN=Issuer CA,OU=Lab,O=Widgets\, Inc.,C=AU
   ```

Table 83. HTTP Server environment variables used to control the content of the certificate within a CA

Environment variable name	Description

_PKISERV_CMP_KEYTYPE_*domain*	Specifies the key type or encryption algorithm to be used to generate the key pair. PKI Services supports `RSA`, `NISTECC` and `BPECC`.
	
	Examples:
	`_PKISERV_CMP_KEYTYPE_CardCA1=RSA`
	`_PKISERV_CMP_KEYTYPE=BPECC`

_PKISERV_CMP_KEYSIZE_*domain*	A three- or four-digit number that specifies the length of the key.
	• For RSA, minimum size is 512, maximum size is 4096, and the default is 1024. The value must be a multiple of two.
	• For NISTECC, valid sizes are 192, 224, 256, 384 and 521, and the default is 192.
	• For BPECC, valid sizes are 160, 192, 224, 256, 320, 384, 512, and the default is 192.
	
	Examples:
	`_PKISERV_CMP_KEYSIZE_CardCA1=2048`
	`_PKISERV_CMP_KEYSIZE=512`

_PKISERV_CMP_HONOR_CLIENT_DATES_*dom*	Specifies whether to honor client-specified certificate validity dates. Valid values are 0 and 1.
	
	The value 1 specifies that PKI Services uses start and end dates specified in the validity field of the certificate request. If validity dates are not specified, PKI Services uses the values in the environment variables _PKISERV_CMP_NOTBEFORE_*domain* and _PKISERV_CMP_NOTAFTER_*domain*. If either the start date or the end date is specified and the other is not specified, PKI Services uses the value set by the corresponding environment variable for the unspecified value.
	
	The value 0, or the absence of this variable, indicates that PKI Services is to use the values in the environment variables _PKISERV_CMP_NOTBEFORE_*domain* and _PKISERV_CMP_NOTAFTER_*domain*. If the client specifies either a start or end date in the validity field of a `cr` message when the value of this variable is 0 or absent, PKI Services rejects the request.
	
	Examples:
	`_PKISERV_CMP_HONOR_CLIENT_DATES_CardCA1=1`
	`_PKISERV_CMP_HONOR_CLIENT_DATES=0`

_PKISERV_CMP_NOTBEFORE_*domain*	Specifies the number of days from day of issue to when the certificate becomes valid. Valid values are 0 - 30 days. The default is 0, specifying that the certificate is valid from the start of the current day.
	
	Examples:
	`_PKISERV_CMP_NOTBEFORE_CardCA1=7`
	`_PKISERV_CMP_NOTBEFORE=5`

Table 83. HTTP Server environment variables used to control the content of the certificate within a CA (continued)

Environment variable name	Description

_PKISERV_CMP_NOTAFTER_*domain*	Specifies the number of days from today's date to when the certificate expires. Valid values are 1 - 9999. The default is 365. The value 1 specifies that the certificate is valid until the end of the current day.
	
	Examples:
	`_PKISERV_CMP_NOTAFTER_CardCA1=1825`
	`_PKISERV_CMP_NOTAFTER=1580`

_PKISERV_CMP_HONOR_CLIENT_EXTS_*domain*	Specifies whether to honor the client-specified extensions Subject Alternate Name, Keyusage and Extended keyusage. Valid values are 0 and 1.
	
	The value 1 specifies that PKI Services uses the extensions Subject Alternate Name, Keyusage and Extended keyusage from the request.
	
	The value 0, or the absence of this variable, specifies that PKI Services is to use the extensions Keyusage and Extended keyusage from the environment variables _PKISERV_CMP_KEYUSAGE_*domain* and _PKISERV_CMP_EXTKEYUSAGE_*domain*. If the client specifies the extensions field of a `cr` message or the attributes field of a `p10cr` message when the value of this variable is 0 or absent, PKI Services rejects the request.
	
	Examples:
	`_PKISERV_CMP_HONOR_CLIENT_EXTS_CardCA1=1`
	`_PKISERV_CMP_HONOR_CLIENT_EXTS=0`

_PKISERV_CMP_KEYUSAGE_*domain*	Blank-separated character strings defining the key usage to be added to requested certificates. Valid values are: `handshake`, `dataencrypt`, `certsign`, `docsign`, `digitalsignature`, `digitalsig`, `nonrepudiation`, `keyencipherment`, `keyenciph`, `keyencrypt`, `dataencipherment`, `dataenciph`, `keyagreement`, `keyagree`, `keycertsign`, and `crlsign`. The values are not case-sensitive. No default value.
	
	Examples:
	`_PKISERV_CMP_KEYUSAGE_CardCA1=digitalsig keyencrypt`
	`_PKISERV_CMP_KEYUSAGE=handshake`

_PKISERV_CMP_EXTKEYUSAGE_*domain*	Blank-separated character strings defining the extended key usage to be added to requested certificates. Valid values are: `serverauth`, `clientauth`, `codesigning`, `emailprotection`, `timestamping`, `ocspsigning`, and `mssmartcardlogon`. The values are not case-sensitive. No default value.
	
	Examples:
	`_PKISERV_CMP_EXTKEYUSAGE_CardCA1=clientauth`
	`_PKISERV_CMP_EXTKEYUSAGE=serverauth`

Table 83. HTTP Server environment variables used to control the content of the certificate within a CA (continued)

Environment variable name	Description

_PKISERV_CMP_AUTHINFOACC_*domain*	Deprecated. Use _PKISERV_CMP_AUTHINFOACC*n_domain* instead. Ignored if you specify _PKISERV_CMP_AUTHINFOACC*n_domain*.
	
	A comma-separated two-part string specifying information used to create the `AuthorityInfoAccess` extension. The two-part string identifies the `accessMethod` and `accessLocation`. The `accessMethod` is either `OCSP` or `IdentrusOCSP` (case-insensitive). The `accessLocation` is a URI in the form `URI=access-url` or `URL=access-url`. The `access-url` must be an HTTP protocol. No default value.
	
	Examples:
	`_PKISERV_CMP_AUTHINFOACC_CardCA1=OCSP,`
	`  URL=http://www.widgets.com/CardCA1/public-cgi/caocsp`
	`_PKISERV_CMP_AUTHINFOACC=OCSP,`
	`  URI=http://www.widgets.com/PKIServ/public-cgi/caocsp`

_PKISERV_CMP_AUTHINFOACC*n_domain*	Specifies one or more `AuthorityInfoAccess` extensions in the form of a comma-separated two-part string. The two-part string identifies the `accessMethod` and `accessLocation`. The `accessMethod` is either `OCSP` or `IdentrusOCSP` (case-insensitive). The `accessLocation` is a URI in the form `URI=access-url` or `URL=access-url`. The `access-url` must be an HTTP protocol. No default value. There can be multiple entries, where *n* is 1 for the first `AuthorityInfoAccess` extension and increases sequentially for additional entries.
	
	Examples:
	`_PKISERV_CMP_AUTHINFOACC1_CardCA1=OCSP,`
	`  URL=http://www.widgets.com/CardCA1/public-cgi/caocsp`
	`_PKISERV_CMP_AUTHINFOACC2_CardCA1=OCSP,`
	`  URI=http://www.widgets.com/PKIServ/public-cgi/caocsp`

_PKISERV_CMP_CERTPOLICIES_*domain*	Specifies the certificate policies that are to be included in the issued certificates. The value is a set of numbers, separated by blanks, each representing one of the `PolicyName` values specified in the **CertPolicy** section of the PKI Services configuration file. Valid values are 1 - 99. No default value.
	
	Examples:
	`_PKISERV_CMP_CERTPOLICIES_CardCA1=1 2 3`
	`_PKISERV_CMP_CERTPOLICIES=1 4`

_PKISERV_CMP_CUSTOMEXT*n_domain*	Specifies one or more custom extensions in the form of a comma-separated four-part string as indicated in the CustomExt field in the GENCERT CertPlist. There can be multiple entries, where *n* is 1 for the first custom extension and increases sequentially for additional entries.
	
	Example:
	`_PKISERV_CMP_CUSTOMEXT1_CardCA1=1.3.6.1.4.1.311.20.2,`
	`  N,BMP,myDomainController`

Notes:

1. *_domain* or *_dom* represents the domain name contained in the variable _PKISERV_CMP_DOMAIN_NAME*x_y*. For an unnamed domain, omit *_domain* or *_dom*.
2. Environment variable values must not be enclosed in quotes even if they include whitespace.

Table 84. HTTP Server environment variables used to configure the certificate recipients

Environment variable name	Description
_PKISERV_CMP_HONOR_CLIENT_CERTS_*domain*	Specifies the maximum number of extra input recipient certificates that can be supplied in the input message, via the `extraCerts` construct in the PKIMessage structure. Valid values are 0 - 5. If omitted or set to a value of 0, certificate requests that contain `extraCerts` are rejected. **Example:** `_PKISERV_CMP_HONOR_CLIENT_CERTS_CardCA1=3` For more information, see "Determining the source of certificates used to encrypt the returned private key" on page 390.
_PKISERV_CMP_KEYRING_*domain*	Specifies the RACF user ID and the name of the key ring associated with that user ID that contains the certificates that are to be used to encrypt the private key for certificate request messages. **Example:** `_PKISERV_CMP_KEYRING_CardCA1=CMPCLNT/CardKeyRing` For more information, see "Determining the source of certificates used to encrypt the returned private key" on page 390.

Table 85. HTTP Server environment variables used to control tracing

Environment variable name	Description
_PKISERV_CMP_TRACE	Specifies a bit mask enabling CMP trace options. No trace option is enabled if the bit mask is 0 and all trace options are enabled if the bit mask is 0xff. The bit mask can be specified as a decimal (*nnn*), octal (0*nnnn*) or hexadecimal (0x*hh*) value. These trace options are available: **0x01** CMP error messages **0x02** CMP informational messages **0x04** R_PKIServ callable service parameter list traces on entry and exit **0x08** Elapse time messages of events within the CMP program **0x10** CMP program function entry and exit trace messages **0x20** DER buffer display messages **0x40** Displays environment variables that are set at CMP program startup

Table 85. HTTP Server environment variables used to control tracing (continued)

Environment variable name	Description
_PKISERV_CMP_TRACE_FILE	Specifies the name of the trace file. Defaults to `/tmp/pkicmp.%.trc`. The trace file is not used if the _PKISERV_CMP_TRACE environment variable is not defined or is set to 0. The current process identifier is included as part of the trace file name when the name contains a percent sign (%). For example, if _PKISERV_CMP_TRACE_FILE is set to `/tmp/pkicmp.%.trc` and the current process identifier is 247, the trace file name is `/tmp/pkicmp.247.trc`. **Guideline:** Because multiple copies of the CMP CGI program can run concurrently for multiple CMP clients, the value of _PKISERV_CMP_TRACE_FILE should include the percent sign (%) to prevent multiple copies of the CMP CGI program from writing to the same file.

Tracing the PKI CMP CGI program

You can enable tracing of the PKI Services CMP CGI program using the environment variable _PKISERV_CMP_TRACE to set the trace options, and the environment variable _PKISERV_CMP_TRACE_FILE to specify the name of the trace file. A single trace file is created for each invocation of the CMP CGI program. For information about these environment variables, see Table 85 on page 396.

Messages and codes returned from the CMP functions

Most messages are returned to the caller as a CMP error response. In addition, all messages are echoed to the CGI program error log in the format:

date time pkicmp-> *function*: (*error code*) *message text*

For example:

```
Wed Oct 28 09:34:12 2009 pkicmp->processNotDays: (577) NotBefore date supplied is invalid(date is before today).
```

Table 86. CMP error codes

Error code	Explanation
Note: Error codes 06 - 99 are the reason codes from the RACF IRRSPX00 callable service.	
06	Request queue serialization timeout occurred.
08	Request denied, not authorized.
12	An internal error has occurred during RACF processing.
20	Function code specified is not defined.
28	Certificate generation provider not available for specified CA domain
32	Incorrect value specified for CA domain.
40	Incorrect Reason Specified.
52	Parameter has an incorrect value.
56	Required field is missing from request.
60	Certificate generation provider error.
64	SerialNum has an incorrect length.
72	The status of the certificate has been changed by another process.

Table 86. CMP error codes (continued)

Error code	Explanation
76	Conflicting fields names in CertPlist.
99	General error for other RACF callable service IRRSPX00 errors.
510	(510) Base64 decode of input message failed, error=*error-code*
511	(511) Base64 encode of output message failed, error=0x*hex-error code*
512	(512) Storage allocation failed (client certificate storage:*error-code*)
513	(513) Error occurred, HTTP access is forbidden.
514	(514) Error occurred, HTTP method was *HTTP method name* instead of POST.
515	(515) Failed to create CMP response message.
516	(516) Unsupported TCP Message protocol version.
517	(517) Unsupported TCP Message message type.
518	(518) Error occurred attempting to read the HTTP input message
519	(519) Unsupported CMP message type: *CMP message type specified*
520	(520) Key size (*envar name* envar) of *envar value* is not a multiple of 2.
521	(521) Key size (*envar name* envar) of *envar value* is not valid for {RSA | NISTECC | BPECC} keys, must be {between 512-4096 | 192, 224, 256, 384, or 521 | 160, 192, 224, 256, 320, 384, or 512}.
522	(522) Storage allocation failed *element:size*
523	(523) HonorClientDates (*envar name* envar) value of *envar value* is not numeric.
524	(524) HonorClientDates (*envar name* envar) value of *envar value* is not valid, expected 0 or 1.
525	(525) notBefore (*envar name* envar) value of *envar value* is not numeric.
526	(526) notAfter (*envar name* envar) value of *envar value* is not numeric.
527	(527) HonorClientCerts (*envar name* envar) value of *envar value* is not numeric.
528	(528) HonorClientCerts (*envar name* envar) value of *envar value* invalid, should be 0-5.
529	(529) HonorClientExts (*envar name* envar) value of *envar value* is not numeric.
530	(530) HonorClientExts (*envar name* envar) value of *envar value* invalid, should be 0 or 1.
533	(533) CMP Envar *envar name* missing in config file.
534	(534) CMP Envar *envar name* value of *envar value* is not numeric.
535	(535) CMP Envar *envar1 name* with value of *envar1 value* <= *envar2 name* with value of *envar2 value*.
536	(536) Key size (*envar name* envar)not specified, defaulting to *default key size value*.
537	(537) Key size (*envar name* envar) of *Key size value specified* is not numeric.
538	(538) KeyType (*envar name* envar) of *KeyType value specified* is not valid.
539	(539) *envar name* envar value length is greater than the maximum length of *maximum length*.
543	(543) request.extraCerts[*index value*].write() failed, status=*error code*

Table 86. CMP error codes (continued)

Error code	Explanation
544	(544) gsk_decode_certificate failed, error code=0x*System SSL error code - System SSL brief error description*
545	(545) gsk_decode_base64 failed, error code=0x*System SSL error code - System SSL brief error description*
547	(547) gsk_open_keyring() failed: Error 0x*System SSL error code - System SSL brief error description*
548	(548) gsk_decode_certificate failed, error code=0x*System SSL error code - System SSL brief error description*
549	(549) gsk_get_record_by_index() failed: Error 0x*System SSL error code - System SSL brief error description*
550	(550) Specified Keyring *Keyring name* contains no certificates
553	(553) gsk_decode_import_certificate failed, error code=0x*System SSL error code - System SSL brief error description*
554	(554) gsk_encode_private_key, error code=0x*System SSL error code - System SSL brief error description*
555	(555) gsk_make_enveloped_data_msg_extended failed, error code=0x*System SSL error code - System SSL brief error description*
556	(556) gsk_encode_export_certificate failed, error code=0x*System SSL error code - System SSL brief error description*
562	(562) Triple Des Algorithm not available, using Single Des.
563	(563) Gencert succeeded, But no Transaction ID returned.
573	(573) Could not decode CMP message.
576	(576) Missing ImplicitConfirm in PKIHeader.
577	(577) {NotBefore | NotAfter} date supplied is invalid({cannot compute seconds since epoch | date is before today}).
581	(581) Validity supplied when not configured to honor client dates.
582	(582) No CA domain found for issuer *Issuer Distinguished name*.
584	(584) Number of extraCerts > HonorClientCerts (*envar name* envar) value of *envar value*.
587	(587) Critical crl extension oid=*Extension OID value* is not supported.
588	(588) crlReason extension value is not valid; decode error *error-code*.
589	(589) Serial number required in certTemplate.
590	(590) {Revoke/Suspend | Resume} of serial number *decimal-serial-number*(0x*hex-serial-number*) failed for CA Domain *Domain name*.
600	(600) Error encountered while encoding response body(*failing element*[:*error code*])
601	(601) Error encountered while encoding response header(*failing element*[:*error code*])
602	(602) {Attributes | Extensions} supplied when not configured to honor client extensions.
603	(603) Base64 encode of CertificationRequest failed, error=*error-code*.
604	(604) CertReqMsg with publicKey has missing or unsupported ProofOfPossesion

Table 86. CMP error codes (continued)

Error code	Explanation
606	(606) Error encountered while encoding CertReqMsg(*failing element*[:*error code*]).
607	(607) Error obtaining the current time of day(*failing step:error code*).
608	(608) Error retrieving information from the CMP request (*failing element*[:*error code*])
609	(609) {cr | rr} message does not contain only one {CertReqMsg | RevDetails}.
610	(610) Subject name absent from CertTemplate for a cr message
611	(611) Unsupported CMP message version (*version specified* not equal 2)
612	(612) Error initializing PKI Services configuration file (*configuration-file-name*)
613	(613) CA domain *domain-name* does not have CMP support enabled
614	(614) Error retrieving CMP environment variables
620	(620) Key type {[null] | *specified KEYTYPE value*} is not valid.
621	(621) Cannot initialize ICSF PKCS#11 interfaces (C_Initialize return code 0x*hex-return-code*)
622	(622) Error encountered while destroying a {Publik | Private} key object (return code 0x*hex-return-code*)
623	(623) Internal PKCS#11 API failure (*PKCS #11 API*) return code 0x*hex-return-code*
624	(624) {RSA | ECC} key generation failure (C_GenerateKeyPair return code 0x*hex-return-code*)
464453637	VSAM contention caused the request to fail. Retry the request.
464453634	VSAM contention caused the request to fail. Retry the request.

Part 5. Administering security for PKI Services

This part explains how to administer security for PKI Services.

- Chapter 20, "RACF administration for PKI Services," on page 403 describes how to use RACF to administer security for PKI Services.

  The following tasks are covered:

  - "Authorizing users for the PKI Services administration group" on page 403
  - "Authorizing users for inquiry access" on page 403
  - "Administering HostIdMappings extensions" on page 404
  - "Locating your PKI Services certificates and key ring" on page 406
  - "Establishing PKI Services as an intermediate CA" on page 408
  - "Renewing your PKI Services CA and RA certificates" on page 410
  - "Recovering a CA certificate profile" on page 413
  - "Retiring and replacing the PKI Services CA private key" on page 414
  - "R_PKIServ (IRRSPX00) callable service" on page 417
  - "Using encrypted passwords for LDAP servers" on page 420.

© Copyright IBM Corp. 2001, 2011

Chapter 20. RACF administration for PKI Services

This topic describes the tasks that the RACF administrator performs after PKI Services has been set up and customized.

The following tasks are covered:

- "Authorizing users for the PKI Services administration group"
- "Authorizing users for inquiry access"
- "Administering HostIdMappings extensions" on page 404
- "Locating your PKI Services certificates and key ring" on page 406
- "Establishing PKI Services as an intermediate CA" on page 408
- "Renewing your PKI Services CA and RA certificates" on page 410
- "Recovering a CA certificate profile" on page 413
- "Retiring and replacing the PKI Services CA private key" on page 414
- "R_PKIServ (IRRSPX00) callable service" on page 417
- "Using encrypted passwords for LDAP servers" on page 420.

For more information about the RACF commands shown in this topic, see *z/OS Security Server RACF Command Language Reference*.

Authorizing users for the PKI Services administration group

You need to know how to add and delete members from the PKI Services administration group (by default, PKIGRP).

Connecting members to the group

The PKI Services administration group is a RACF group containing the list of user IDs that are authorized to use PKI Services administration functions. To connect a member to the group, execute the following command, replacing *pkigroup_mem* with the member's user ID and *pkigroup* with the name of the PKI Services administration group (PKIGRP by default). (See Table 17 on page 43 for more information.)

```
CONNECT pkigroup_mem GROUP(pkigroup)
```

Note: You need to enter this command for each user ID in turn.

Deleting members from groups

To remove a user from a group, execute the following command, replacing *pkigroup_mem* with the user ID of the member you want to delete and *pkigroup* with the name of the PKI Services administration group (PKIGRP by default).

```
REMOVE pkigroup_mem GROUP(pkigroup)
```

Authorizing users for inquiry access

You can add groups of users who do not need the full administrative authority of users in the PKIGRP group. You can use the following procedure to authorize a new group for inquiry abilities, such as a help desk might require. The commands shown include variables whose names are appropriate for this scenario.

© Copyright IBM Corp. 2001, 2011

Steps for authorizing users for inquiry access

Before you begin: You need to know the high-level VSAM data set qualifier used for the IKYSETUP variable *vsamhlq* value, in case your installation did not use the `PKISRVD` default. (See Table 17 on page 43.)

Perform the following steps to add and administer a group that needs authority to query PKI Services information.

1. Add the new group.

   Example:

   ```
   ADDGROUP HELPDESK OMVS(GID(197312))
   ```

   ---

2. Connect each member to the new group. Repeat for each user ID you need to connect.

   Example:

   ```
   CONNECT OPER17 GROUP(HELPDESK)
   ```

   ---

3. Authorize the new group for READ access to the resources of PKI Services. Replace your installation's value for the data set's high-level qualifier if your installation did not use the `PKISRVD` default.

   Example:

   ```
   PERMIT 'PKISRVD.**' ID(HELPDESK) ACCESS(READ)
   PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY)
      ID(HELPDESK) ACCESS(READ)
   SETROPTS GENERIC(DATASET) REFRESH
   SETROPTS RACLIST(FACILITY) REFRESH
   ```

   The SETROPTS commands activate the profiles that authorize READ access.

   ---

4. If necessary, you can remove a user from the group. The following example removes the user you connected in Step 2.

   Example:

   ```
   REMOVE OPER17 GROUP(HELPDESK)
   ```

   ---

5. If necessary, you can delete the group. The following example deletes the group you created in Step 1.

   Example:

   ```
   DELGROUP(HELPDESK)
   ```

Administering HostIdMappings extensions

You can add a `HostIdMappings` extension to certificates you create for certain users, allowing you to specify the user IDs that each user will be able to use for login to particular servers (or hosts). Controlling an identity used for login purposes is a very important security objective. Therefore, you must exercise administrative control in the following areas by authorizing:

- PKI Services as a highly trusted certificate authority whose certificates will be honored when they contain `HostIdMappings` extensions

- Particular servers to accept logins from clients whose certificates contain HostIdMappings extensions

Steps for administering HostIdMappings extensions

Perform the following steps to allow the Web server to accept logins from clients who have been issued PKI Services certificates with HostIdMappings extensions:

1. Determine if PKI Services is defined as a highly trusted certificate authority on your system by listing its certificate authority definition by using the RACDCERT CERTAUTH LIST command.

   Example:

   ```
   RACDCERT CERTAUTH LIST(LABEL('Local PKI CA'))
   ```

   Check the Status information near the top of the output listing for the HIGHTRUST attribute.

   ---

2. If not already defined, add the HIGHTRUST attribute to the certificate authority definition for PKI Services.

   Example:

   ```
   RACDCERT CERTAUTH ALTER(LABEL('Local PKI CA')) HIGHTRUST
   ```

   ---

3. Define a resource in the SERVAUTH class for each server (host) name you want your Web server to honor when accepting logins for certificates containing HostIdMappings extensions. The resource name follows the format: IRR.HOST.*hostname*. The *hostname* is the value of the HostIdMappings extension entry pertaining to the z/OS host system you are administering (without the subject ID portion). This is usually a domain name, such as plpsc.pok.ibm.com. The following example shows defining a resource.

   Example:

   ```
   RDEFINE SERVAUTH IRR.HOST.PLPSC.POK.IBM.COM UACC(NONE)
   ```

   ---

4. Permit your Web server to access this resource with READ authority. Be sure the Web server is defined as a RACF user.

   Example:

   ```
   PERMIT IRR.HOST.PLPSC.POK.IBM.COM CLASS(SERVAUTH) ID(WEBSRV) ACCESS(READ)
   ```

   ---

5. Activate the SERVAUTH class, if not already active.

   Example:

   ```
   SETROPTS CLASSACT(SERVAUTH)
   ```

   If already active, refresh the SERVAUTH class.

   Example:

   ```
   SETROPTS CLASSACT(SERVAUTH) REFRESH
   ```

Note: On a z/OS system, a HostIdMappings extension is not honored if the target user ID was created after the start of the validity period for the certificate containing the HostIdMappings extension. Therefore, if you are creating user IDs specifically for certificates with HostIdMappings extensions, make sure that you create the user IDs before the certificate requests are submitted.

Alternately, when approving the certificate, you can modify the date the certificate becomes valid so that it is not earlier than the date the user ID was created. For renewed certificates, all the original information is replicated in the new certificate, including the date the certificate becomes valid and any `HostIdMappings`. If you want to change a `HostIdMappings` extension when approving the renewed certificate, you must also modify the date the certificate becomes valid so that it is not earlier than the date the user ID was created.

See *z/OS Security Server RACF Command Language Reference* for details about syntax and authorization required for using the RACDCERT command.

Locating your PKI Services certificates and key ring

The IKYSETUP exec sets up the RACF environment for PKI Services. After the setup is complete, you might need to go back and locate the PKI Services CA certificate, key ring, or the optional RA certificate, possibly to diagnose error conditions. You can do this by using various RACF TSO commands.

Before you begin: You need to determine the following setup information:

Table 87. Information you need for locating your PKI Services certificates and key ring

Information needed	Where to find this information	Record your value here
ca_label - The label of your CA certificate in RACF	See Table 10 on page 32.	
ra_label - The label of your RA certificate in RACF	See Table 10 on page 32.	
ca_ring - The PKI Services SAF key ring	See Table 17 on page 43.	
daemon - The user ID for the PKI Services daemon	See Table 17 on page 43.	
log_dsn - The data set name of the IKYSETUP log	See Table 17 on page 43.	
cacert_dsn - The data set name of your CA certificate as exported from RACF	See Table 17 on page 43.	

Steps for locating the PKI Services certificates and key ring

Perform the following steps to locate the PKI Services CA certificate, key ring, and the optional RA certificate:

1. Locate the CA certificate using one of the following two methods (Step 1a or Step 1b) and examine its information.
   a. Locate the CA certificate using the name of its export data set. (Get the export data set name from *cacert_dsn* in Table 87.) Display its information by executing the following RACF command from a TSO command prompt:

      ```
      RACDCERT CHECKCERT(cacert_dsn)
      ```

      Sample output:

      ```
      Digital certificate information for CERTAUTH:
        Label: Local PKI CA
        Certificate ID: 2QiJmZmDhZmjgdOWg4GTQNfSyUDDwUBA
        Status: HIGHTRUST
        Start Date: 2001/06/04 23:00:00
        End Date:   2020/01/01 22:59:59
      ```

```
Serial Number:
     >00<
Issuer's Name:
     >OU=Human Resources Certificate Authority.O=IBM.C=US<
Subject's Name:
     >OU=Human Resources Certificate Authority.O=IBM.C=US<
Key Usage: CERTSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
```

b. Alternately, locate the CA certificate using its certificate label. (Get the label name from *ca_label* in Table 87 on page 406.) Display its information by entering the following RACF command from a TSO command prompt.

```
RACDCERT CERTAUTH LIST(LABEL('ca_label'))
```

The RACDCERT CERTAUTH LIST command produces the same output as the RACDCERT CHECKCERT (shown in Step 1a) with the addition of information about any ring associations. For example:

Sample output:

```
Ring Associations:
  Ring Owner: PKISRVD
  Ring:
     >CAring<
```

c. Examine the CA certificate information. If you are diagnosing errors, note the following:

- The first line must indicate that this is a CERTAUTH certificate.
- `Label` must match your `ca_label` value (as in the preceding table).
- If `Serial Number` is not equal to `00`, this indicates that the certificate has been renewed or was issued by another certificate authority.
- If `Issuer's Name` differs from `Subject's Name`, this indicates that the certificate was issued by another certificate authority.
- `Subject's Name` must match the original value recorded for the PKI Services SUBJECTSDN in the IKYSETUP log.
- `Private Key Type` and `Private Key Size` must be present.
- `Private Key Type` indicates whether the key is a non-ICSF key (software key) or an ICSF key (ICSF or PCICC key).
- If `Ring Associations` are listed, ensure that an association is displayed for the daemon user ID as ring owner and your `ca_ring` value (from Table 87 on page 406) as ring name.

---

2. Locate the CA key ring and examine its information.

a. Get the ring name from *ca_ring* in Table 87 on page 406 and display its information by executing the following RACF command from a TSO command prompt:

```
RACDCERT ID(daemon) LISTRING(ca_ring)
```

Sample output:

```
Digital ring information for user PKISRVD:

  Ring:
       >CAring<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  --------------------------------    ------------   --------   -------
  Local PKI CA                        CERTAUTH       PERSONAL     YES
  Local PKI RA                        PKISRVD        PERSONAL     NO
```

b. Examine the key ring information. If you are diagnosing errors, note the following:
   - The entry for the PKI Services CA certificate must have USAGE PERSONAL and DEFAULT YES.
   - If you use an optional RA certificate, you will see the second line. If present, the entry for the PKI Services RA certificate must have USAGE PERSONAL and DEFAULT NO.

---

3. If you use an optional RA certificate, locate it and examine its information.
   a. Locate the RA certificate using its certificate label. (Get the RA's certificate label from *ra_label* in Table 87 on page 406 or from the RACDCERT LISTRING output shown in Step 2a.) Display the RA certificate information by executing the following RACF command from a TSO command prompt:

      ```
      RACDCERT ID(certificate-owner) LIST(LABEL('certificate-label-name'))
      ```

      Sample output:

      ```
      Digital certificate information for PKISRVD:
        Label: Local PKI RA
        Certificate ID: 2QiJmZmDhZmjgdOWg4GTQNfSyUDDwUBA
        Status: TRUST
        Start Date: 2001/06/04 23:00:00
        End Date:   2020/01/01 22:59:59
        Serial Number:
             >01<
        Issuer's Name:
             >OU=Human Resources Certificate Authority.O=IBM.C=US<
        Subject's Name:
             >CN=Registration Authority.OU=Human Resources Certif<
             >icate Authority.O=IBM.C=US<
        Key Usage: HANDSHAKE
        Private Key Type: Non-ICSF
        Private Key Size: 1024
      ```

   b. Examine the RA certificate information. If you are diagnosing errors, note the following:
      - The user ID of the certificate owner (indicated in the first line) must match the user ID of the PKI Services daemon.
      - `Issuer's Name` must match the `Subject's Name` of the CA certificate.
      - `Private Key Type` and `Private Key Size` must be present.
      - `Private Key Type` indicates whether the key is a non-ICSF key (software key) or an ICSF key (ICSF or PCICC key).

Establishing PKI Services as an intermediate CA

The default setup for PKI Services establishes the PKI Services certificate authority as a root CA, also known as a self-signed CA. Because there is no established trust hierarchy leading to a self-signed certificate, it is impossible to verify that a self-signed certificate is genuine. Accordingly, any person or application that wishes to process certificates issued by a root authority must explicitly trust the authenticity of the self-signed CA certificate.

Alternately, you can establish the PKI Services certificate authority as an intermediate (subordinate) certificate authority. An intermediate certificate authority is one whose certificate is signed by another higher certificate authority. This higher certificate authority can be a root CA or another intermediate CA. If

the root CA certificate has previously been trusted, you can verify any lower intermediate CA certificate using the higher certificate.

In the following steps, you will be replacing the self-signed CA certificate created by IKYSETUP with one signed by another authority.

Steps for changing PKI Services from a self-signed CA to an intermediate CA

Before you begin:

1. This procedure assumes that the PKI Services CA certificate is issued by a root, or self-signed, CA.
2. The commands in the steps that follow include several variables. The following table describes these variables. Determine the values for these variables and record the information in the blank boxes:

Table 88. Information you need for establishing PKI Services as an intermediate CA

Information needed	Where to find this information	Record your value here
cacert_dsn - The data set name of the new PKI Services CA certificate.		
ca_label - The label of your CA certificate in RACF	See Table 10 on page 32.	
export_dsn - The data set name of the root CA certificate as exported from RACF.		
temp_dsn - The name of the temporary data set to contain your new certificate request and returned certificate.	You decide this based on local data set naming conventions.	

Perform the following steps to change PKI Services from a self-signed certificate authority to an intermediate certificate authority:

1. Determine what certificate authority will be acting as a higher authority for PKI Services. (This could be a public certificate authority, such as VeriSign, or a local, internal certificate authority, perhaps even another instance of PKI Services.)

   _____

2. Create a new certificate request from your existing self-signed CA certificate by entering the following RACF command from a TSO command prompt:

   ```
   RACDCERT CERTAUTH GENREQ(LABEL('ca_label')) DSN(temp_dsn)
   ```

   _____

3. Send the certificate request to the higher certificate authority, following the procedures that the higher authority requires.

   _____

4. If the root CA is not one that is already know by RACF, then add the root CA to RACF as a certificate authority. To do this:
   a. Receive the root CA certificate and place it into the certificate data set (*temp_dsn*).

      Note: The procedure for doing this can vary greatly depending on how the higher certificate authority delivered the certificate:

- If the certificate is delivered as base64 encoded text, the easiest way to deposit the certificate into the data set is to edit the certificate data set:
  1) Delete all existing lines in *temp_dsn*.
  2) Copy the base64 encoded text.
  3) Paste the copied text into the ISPF edit window.
  4) Save.
- If the certificate is delivered as binary data (also called DER encoded), the easiest way to deposit the certificate into the data set is to use binary FTP.

b. Add the new root CA certificate into the RACF database by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH ADD(temp_dsn) WITHLABEL('label-for-root-CA')
```

---

5. Add the new PKI Services CA to RACF as a certificate authority:

   a. Receive the PKI Services CA certificate and place it into the certificate data set (*cacert_dsn*). This step is similar to step 4, except that it uses *cacert_dsn* as the data set name instead of *temp_dsn*, because you want to keep the PKI Services CA certificate permanently in the data set *cacert_dsn*.

   b. Add the new PKI Services CA certificate back into the RACF database by entering the following RACF command from a TSO command prompt:

   ```
   RACDCERT CERTAUTH ADD(cacert_dsn)
   ```

   Guideline: Do not specify a label on this command.

---

6. Export the root CA certificate in DER format to the export data set by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH EXPORT(LABEL('label-for-root-CA')) DSN(export_dsn)
   FORMAT(CERTDER)
```

---

7. Make your new root CA certificate available to your clients, because it becomes the Web server's root CA certificate too. To do this, set up the `var` directory by performing Step 2 through Step 4 in “Steps for setting up the var directory” on page 84.

   Note: Make sure that the root CA certificate, not your intermediate CA certificate, is stored in `/var/pkiserv/cacert.der`.

Renewing your PKI Services CA and RA certificates

Eventually, your PKI Services CA and RA certificates will expire. To avoid complications related to an expired CA or RA certificate, renew those certificates before they expire. (You will receive MVS console message IKYP026E as the expiration date approaches.)

This topic contains these procedures:

- “Steps for renewing your PKI Services CA certificate” on page 411
- “Steps for renewing your PKI Services RA certificate” on page 412

Steps for renewing your PKI Services CA certificate

Before you begin: The commands in the steps that follow include several variables. The following table describes these variables. Determine the values for these variables and record the information in the blank boxes:

Table 89. Information you need for renewing your PKI Services certificate authority certificate

Information needed	Where to find this information	Record your value here
cacert_dsn - The data set name of your renewed CA certificate as exported from RACF. (This data set is needed for recovery.)		
ca_label - The label of your CA certificate in RACF	See Table 10 on page 32.	
temp_dsn - The temporary data set to contain your new certificate request and returned certificate.	You decide this based on local data set naming conventions.	

Perform the following steps to renew your PKI Services CA certificate:

1. Create a new certificate request from your current CA certificate by entering the following RACF command from a TSO command prompt:

   ```
   RACDCERT CERTAUTH GENREQ(LABEL('ca_label')) DSN(temp_dsn)
   ```

   _____

2. If your PKI Services certificate authority is a root CA (that is, it has a self-signed certificate, which is the default), generate the self-signed renewal certificate by entering the following RACF command from a TSO command prompt. The *ca_expires* variable indicates the new expiration date.

   ```
   RACDCERT CERTAUTH GENCERT(temp_dsn) NOTAFTER(DATE(ca_expires))
     SIGNWITH(CERTAUTH LABEL('ca_label'))
   ```

   _____

3. Alternately, if your PKI Services certificate authority is an intermediate certificate authority, perform the following steps:

   a. Send the certificate request to the higher (external) CA, following the procedures that the higher authority requires. If your CA retains the original certificate signing requests (CSR), you might not need to create and store a new request based on the expiring certificate. You might be able to request a renewal using the original CSR.

   b. After the certificate has been issued, receive the certificate back into the certificate data set (*temp_dsn*).

      Note: The procedure for doing this can vary greatly depending on how the higher certificate authority delivers the new certificate:

      - If the certificate is delivered as base64 encoded text, the easiest way to deposit the certificate into the data set is to edit the certificate data set:
        1) Delete all existing lines in *temp_dsn*.
        2) Copy the base64 encoded text.
        3) Paste the copied text into the ISPF edit window.
        4) Save.

- If the certificate is delivered as binary data (also called DER encoded), the easiest way to deposit the certificate into the data set is to use binary FTP.

c. Add the renewed certificate back into the RACF database by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH ADD(temp_dsn)
```

Guideline: Do not specify a label on this command.

---

4. Export the certificate in DER format to the CA certificate data set by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH EXPORT(LABEL('ca_label')) DSN(cacert_dsn) FORMAT(CERTDER)
```

Save this data set for recovery if needed later.

---

5. If your PKI Services certificate authority is a root CA, *and* it is also the Web server's root certificate, the renewed root needs to be accessible to the clients. To make your new certificate available to your clients, set up the `/var/pkiserv` directory by performing Step 2 through Step 4 in "Steps for setting up the var directory" on page 84.

---

6. Stop and restart PKI Services.

Steps for renewing your PKI Services RA certificate

Before you begin: The commands in the steps that follow include several variables. The following table describes these variables. Determine the values for these variables and record the information in the blank boxes:

Table 90. Information you need for renewing your PKI Services RA certificate

Information needed	Where to find this information	Record your value here
ca_label - The label of your CA certificate in RACF	See Table 10 on page 32.	
daemon - The user ID of the PKI Services daemon.	See Table 17 on page 43.	
racert_dsn - The name of the data set to contain your new certificate request.	You decide this based on local data set naming conventions.	
ra_label - The label of your RA certificate in RACF.	See Table 10 on page 32.	

Perform the following steps to renew your PKI Services RA certificate:

1. Create a new certificate request from your existing RA certificate by entering the following RACF command from a TSO command prompt:

```
RACDCERT ID(daemon) GENREQ(LABEL('ra_label')) DSN(racert_dsn)
```

---

2. Create the renewed PKI Services certified RA certificate by entering the following RACF command from a TSO command prompt. The *ra_expires* variable indicates the new expiration date.

```
RACDCERT ID(daemon) GENCERT(racert_dsn) NOTAFTER(DATE(ra_expires))
  SIGNWITH(CERTAUTH LABEL('ca_label'))
```

3. Stop and restart PKI Services.

Recovering a CA certificate profile

Unless you change the IKYSETUP REXX exec to disable the function, IKYSETUP automatically backs up the PKI Services CA certificate and private key to a passphrase-encrypted data set that has PKCS #12 format, if the certificate and private key were created by software. (If the certificate was created by hardware, for example by RACDCERT with the PCICC keyword, it is not backed up.) If the CA certificate profile in the RACF database is accidentally deleted, you can recover it by adding the certificate and private key back to the RACF database from the backup PKCS #12 data set .

Steps for recovering a CA certificate profile

Before you begin: The commands in the steps that follow include several variables. Table 91 describes these variables. Determine the values for these variables and record the information in the blank boxes:

Table 91. Information you need for recovering a CA certificate profile

Information needed	Where to find this information	Record your value here
backup_dsn - The name of the data set containing the backup copy of your original CA certificate and its private key	See Table 17 on page 43.	
cacert_dsn - The data set name of your CA certificate as exported from RACF		
ca_label - The label of your CA certificate in RACF	See Table 10 on page 32.	
ca_ring - The PKI Services SAF key ring	See Table 17 on page 43.	
daemon - The user ID for the PKI Services daemon	See Table 17 on page 43.	
your-passphrase - The passphrase you used when backing up the private key	You specified this when running IKYSETUP.	

Perform the following steps to recover a CA certificate profile:

1. Issue the following TSO commands:

   Notes:

   a. If you are not using ICSF, omit the ICSF keyword on the first ADD command.

   b. If your CA certificate has been renewed, the second ADD command recovers the most current version using the saved CA certificate. If your certificate has not been renewed, you can omit the second ADD command. For information about renewing your CA certificate, see “Renewing your PKI Services CA and RA certificates” on page 410.

```
RACDCERT CERTAUTH ADD(backup_dsn) PASSWORD(your-passphrase)
     WITHLABEL('ca_label') ICSF
RACDCERT CERTAUTH ADD(cacert_dsn)
RACDCERT ID(daemon) CONNECT(CERTAUTH LABEL('ca_label')
     RING(ca_ring) USAGE(PERSONAL) DEFAULT)
```

Retiring and replacing the PKI Services CA private key

For certificates that are associated with private keys, such as the PKI Services CA certificate, you should periodically retire the private keys and replace them with new ones. This process is commonly called *certificate rekeying* or *key rollover*. Do this to prevent private keys from being overused. (The more a key is used, the more susceptible it is to being broken and recovered by an unintended party.)

To rekey and rollover the PKI Services private key, use the REKEY and ROLLOVER operands of the RACF RACDCERT command. The REKEY operand makes a self-signed copy of the original certificate with a new public-private key pair. The ROLLOVER operand finalizes the rekey operation by replacing the use of the original certificate with the new certificate in every key ring to which the original certificate is connected. It also destroys the original private key and copies over information about its serial number base so the new certificate can be used to sign new certificates.

A retired CA certificate can not be used to sign new certificates. However, until it expires, it can be used to verify previously signed certificates. If you have an RA certificate for SCEP processing that was not replaced when the CA certificate that signed it was retired, you need to reconnect the CA certificate to the CA key ring.

Steps to retire and replace the PKI Services CA private key for the PKI templates

The commands used in this procedure are examples based on the following scenario:

Assumptions:

- The certificate you are rekeying is a CERTAUTH certificate with label `'Local PKI CA'`. It was issued by a commercial CA and is being used by PKI Services for the PKI templates as a certificate authority (CA) certificate, making the PKI Services CA a subordinate CA.
- The PCI cryptographic coprocessor will to be used to generate the new key-pair.
- The size of the new private key will be 1024 bits (RACF default size).

Perform the following procedure to rekey and replace the private key.

1. Initiate the rekeying by executing the following RACF command:

   ```
   RACDCERT CERTAUTH REKEY(LABEL('Local PKI CA')) WITHLABEL('Local PKI CA-2') PCICC
   ```

2. Create a request for a commercial CA to sign the new public key and reissue the certificate. To create a certificate request for the new key and store it in MVS data set `'SYSADM.CERT.REQ'`, execute the following command:

   ```
   RACDCERT CERTAUTH GENREQ(LABEL('Local PKI CA-2')) DSN('SYSADM.CERT.REQ')
   ```

Restriction: The certificate request data contained in the data set must be sent to, and received from, the commercial CA using the process defined by the CA. Those steps are not included.

---

3. Receive the newly signed and reissued certificate back from the commercial CA into MVS data set `'SYSADM.CERT.B64'`.

---

4. Add the newly signed certificate into RACF and replace the self-signed rekeyed one by executing the following command:

```
RACDCERT CERTAUTH ADD('SYSADM.CERT.B64')
```

---

5. You are now ready to retire the original certificate and must stop all use of the original private key. Stop the PKI Services daemon.

   Note: At this point, the original certificate and its private key exist in RACF with label `'Local PKI CA'`. The new certificate and its private key exist in a separate entry in RACF with label `'Local PKI CA-2'`. You can proceed to rollover the key.

---

6. Finalize the rollover by entering the following command:

```
RACDCERT CERTAUTH ROLLOVER(LABEL('Local PKI CA')) NEWLABEL('Local PKI CA-2')
```

---

7. If an RA certificate is in use that was signed by the retired CA certificate, connect the retired CA certificate to the key ring.

```
RACDCERT ID(daemon) CONNECT(CERTAUTH LABEL('Local PKI CA') RING(ringname) USAGE(CERTAUTH))
```

8. Restart the PKI Services daemon.

When you are done: You have retired and replaced the old PKI Services CA certificate. All the information for the original certificate is updated to reflect the new certificate, including the key ring connections. You can now begin to use the new certificate and its private key. You can continue to use the old certificate for signature verification purposes until it expires. However, you cannot use the old certificate to sign new certificates. Additionally, do not connect the old certificate to any key rings as the default certificate.

Steps to retire and replace the PKI Services CA private key for the SAF templates: Scenario 1

The commands used in this procedure are examples based on the following scenario:

Assumptions:

- The certificate you are rekeying is a CERTAUTH certificate with label `'taca'`.
- It was issued by a local CA certificate labeled `'Local RACF CA'` that was generated by RACF and is being used by PKI Services for the SAF templates as a certificate authority (CA) certificate.

Perform the following procedure to rekey and replace the private key.

1. Initiate the rekeying by executing the following RACF command:

```
RACDCERT CERTAUTH REKEY(LABEL('taca')) WITHLABEL('taca-2')
```

2. Generate a certificate request based on the new self-signed certificate and store it in MVS data set 'SYSADM.CERT.REQ' by executing the following command:

```
RACDCERT CERTAUTH GENREQ(LABEL('taca-2')) DSN('SYSADM.CERT.REQ')
```

3. Execute the following command to sign the new certificate:

```
RACDCERT CERTAUTH GENCERT('SYSADM.CERT.REQ')
   SIGNWITH(CERTAUTH LABEL('Local RACF CA'))
```

   At this point, the original certificate and its private key exist in RACF with the label 'taca'. The new certificate and its private key exist in a separate entry in RACF with the label 'taca-2'. You can proceed to rollover the key.

4. Finalize the rollover by entering the following command:

```
RACDCERT CERTAUTH ROLLOVER(LABEL('taca')) NEWLABEL('taca-2')
```

5. Change the certificate label used in the SIGNWITH field in the SAF templates to the new label name.

When you are done: You have retired and replaced the old certificate. All the information for the original certificate is updated to reflect the new certificate, including the key ring connections. You can now begin to use the new certificate and its private key. You can continue to use the old certificate for signature verification purposes until it expires. However, you cannot use the old certificate to sign new certificates. Additionally, do not connect the old certificate to any key rings as the default certificate.

Steps to retire and replace the PKI Services CA private key for the SAF templates: Scenario 2

The commands used in this procedure are examples based on the following scenario:

Assumptions:

- The certificate you are rekeying is a CERTAUTH certificate with label 'taca'.
- It was a self-signed certificate in RACF and is being used by PKI Services for the SAF templates as a certificate authority (CA) certificate.

Perform the following procedure to rekey and replace the private key.

1. Initiate the rekeying by executing the following RACF command:

```
RACDCERT CERTAUTH REKEY(LABEL('taca'))
   WITHLABEL('taca-2')
```

   At this point, the original certificate and its private key exist in RACF with the label 'taca'. The new certificate and its private key exist in a separate entry in RACF with the label 'taca-2'. You can proceed to rollover the key.

2. Finalize the rollover by entering the following command:

```
RACDCERT CERTAUTH ROLLOVER(LABEL('taca')) NEWLABEL('taca-2')
```

3. Change the certificate label used in the SIGNWITH field in the SAF templates to the new label name.

When you are done: You have retired and replaced the old certificate. All the information for the original certificate is updated to reflect the new certificate, including the key ring connections. You can now begin to use the new certificate and its private key. You can continue to use the old certificate for signature verification purposes until it expires. However, you cannot use the old certificate to sign new certificates. Additionally, do not connect the old certificate to any key rings as the default certificate.

R_PKIServ (IRRSPX00) callable service

Authorized applications, such as servers, that invoke the `R_PKIServ` callable service (IRRSPX00) can request the generation, retrieval, and administration of PKIX-compliant X.509 Version 3 certificates and certificate requests. Applications can request end-user functions or administrative functions related to these requests. You authorize these applications by administering RACF resources in the FACILITY class, based on whether the application requests end-user functions or administrative functions.

See *z/OS Security Server RACF Callable Services* for the details of invoking IRRSPX00.

Authorizing end-user functions

The end-user functions are:

EXPORT Retrieves (exports) a previously requested certificate, or retrieves (exports) the PKI Services registration authority (RA) certificate or the certificate authority (CA) certificate.

GENCERT Generates an auto-approved certificate.

GENRENEW Generates an auto-approved renewal certificate. (The request submitted is automatically approved.)

QRECOVER Lists certificates whose key pairs were generated by PKI Services under a requestor's e-mail address and passphrase.

REQCERT Requests a certificate that an administrator must approve before it is created.

REQRENEW Requests certificate renewal. The administrator needs to approve the request before the certificate is renewed.

RESPOND Invokes the PKI OCSP responder.

REVOKE Revokes a certificate that was previously issued.

SCEPREQ Generates a certificate request using Simple Certificate Enrollment Protocol (SCEP).

VERIFY Confirms that a given user certificate was issued by this certificate authority and, if so, returns the certificate fields.

For end-user functions, FACILITY class resources protect this interface. Access authority is based on the user ID for the application (the user ID from the ACEE associated with the address space). To determine the user ID for the application, the current TCB is checked for an ACEE. If one is found, the authority of that user

is checked. If there is no ACEE associated with the current TCB, the ACEE associated with the address space is used to locate the user ID.

The form for the FACILITY class resources is:

`IRR.RPKISERV.`*`function`*`[.`*`ca_domain`*`]`

function
Specifies one of the end-user function names in the preceding list.

ca_domain
Optionally specifies the PKI Services certificate authority (CA) domain name. Use this when your installation has established multiple PKI Services CAs and the `CA_domain` parameter is provided with IRRSPX00.

Restriction: If the name of your initial CA domain is longer than 8 characters, you must truncate it to exactly 8 characters when you define the resource name in the FACILITY class.

Example: For the GENCERT function, when the *ca_domain* is named `Customers` and the `CA_domain` parameter is provided with IRRSPX00, then the FACILITY class resource controlling the function is IRR.RPKISERV.GENCERT.CUSTOMER. (The name `Customers` was truncated to CUSTOMER. See the restriction for the *ca_domain* parameter.) When the `CA_domain` parameter is not provided with IRRSPX00, the FACILITY class resource is IRR.RPKISERV.GENCERT.

The access authorities you can assign for these FACILITY class resources have the following effects:

NONE Access is denied.

READ Access is permitted based on subsequent access checks against the caller's user ID.

UPDATE Access is permitted based on subsequent access checks against the application's user ID.

CONTROL (or user ID has RACF SPECIAL)
Access is permitted, and no subsequent access checks are made.

Example: If you defined the FACILITY class profile IRR.RPKISERV.GENCERT.CUSTOMER to control access to the GENCERT function on the CA domain named `Customers`, you can prevent the user ID MYAPP from using the GENCERT function on that CA domain by issuing the command:

```
PERMIT IRR.RPKISERV.GENCERT.CUSTOMER CLASS(FACILITY) ID(MYAPP) ACCESS(NONE)
```

For SAF GENCERT and EXPORT requests where the application has READ and UPDATE access, subsequent access checks are performed against the `IRR.DIGTCERT.`*`function`* FACILITY resources. These are identical to the checks the RACDCERT TSO command makes. See *z/OS Security Server RACF Command Language Reference* for more information.

For PKI Services EXPORT, GENCERT, GENRENEW, QRECOVER, REQCERT, REQRENEW, RESPOND, REVOKE, SCEPREQ, and VERIFY requests in which the application has READ and UPDATE access, subsequent access checks are performed against the `IRR.DIGTCERT.`*`function`* FACILITY resources.

The following table summarizes the access requirements for the user ID whose access is checked.

Table 92. Summary of access authorities required for PKI Services requests

Request	Access
EXPORT	• IRR.DIGTCERT.EXPORT – READ access if `PassPhrase` is specified or if `CertID` is specified as `PKICACERT`. – UPDATE access if the `PassPhrase` parameter is not specified with IRRSPX00. – CONTROL access if you want to export a PKCS #7 certificate.
GENCERT	• IRR.DIGTCERT.GENCERT — CONTROL access • IRR.DIGTCERT.ADD – UPDATE access if any hostIdMappings information is specified in the certificate request parameter list or the `UserId` field in the certificate request parameter list indicates the certificate is being requested for another user other than the caller – READ access otherwise
GENRENEW	• IRR.DIGTCERT.GENRENEW — READ access • IRR.DIGTCERT.GENCERT — CONTROL access **Note:** It is assumed that the calling application has already verified the input certificate using the VERIFY function.
QRECOVER	• IRR.DIGTCERT.QRECOVER — READ access
REQCERT	• IRR.DIGTCERT.REQCERT — READ access
REQRENEW	• IRR.DIGTCERT.REQRENEW — READ access **Note:** It is assumed that the calling application has already verified the input certificate using the VERIFY function.
RESPOND	• IRR.DIGTCERT.RESPOND — READ access
REVOKE	• IRR.DIGTCERT.REVOKE — READ access **Note:** It is assumed that the calling application has already verified the target certificate using the VERIFY function.
SCEPREQ	• IRR.DIGTCERT.SCEPREQ — READ access
VERIFY	• IRR.DIGTCERT.VERIFY — READ access **Note:** It is assumed that the calling application has already verified that the end user possesses the private key that correlates to the input certificate.

Authorizing administrative functions

The administrative functions are:

CERTDETAILS Get detailed information about one PKI Services issued certificate.

MODIFYCERTS Change PKI Services issued certificates.

MODIFYREQS Change PKI Services certificate requests.

QUERYCERTS Query PKI Services issued certificates.

QUERYREQS Query PKI Services about certificate requests.

PREREGISTER	Preregister clients who use Simple Certificate Enrollment Protocol (SCEP).
REQDETAILS	Get detailed information about one PKI Services certificate request.

For the all administrative functions, the following single FACILITY class resource protects this interface.

`IRR.RPKISERV.PKIADMIN[.`*`ca_domain`*`]`

ca_domain
Optionally specifies the PKI Services certificate authority (CA) domain name. Use this when your installation has established multiple PKI Services CAs and the `CA_domain` parameter is provided with IRRSPX00.

Restriction: If the name of your initial CA domain is longer than 8 characters, you must truncate it to exactly 8 characters when you define the resource name in the FACILITY class.

- If the caller is RACF SPECIAL, no further access is necessary.
- Otherwise, the caller needs:
  - READ access to perform read operations (QUERYREQS, QUERYCERTS, REQDETAILS, and CERTDETAILS)
  - UPDATE access for the action operations (PREREGISTER, MODIFYREQS and MODIFYCERTS).

Example: For administrative functions, when the *ca_domain* is named `Customers` and the `CA_domain` parameter is provided with IRRSPX00, the FACILITY class resource controlling this interface is IRR.RPKISERV.PKIADMIN.CUSTOMER. (The name `Customers` was truncated to CUSTOMER. See the restriction for the *`ca_domain`* value.) When the `CA_domain` parameter is not provided with IRRSPX00, IRR.RPKISERV.PKIADMIN is the name of the FACILITY class resource.

To determine the appropriate access level of the caller, the current TCB is checked for an ACEE. If one is found, the authority of that user is checked. If there is no ACEE associated with the current TCB, the ACEE associated with the address space is used to locate the user ID.

Attention: UPDATE access to the IRR.RPKISERV.PKIADMIN[.*ca_domain*] resource also controls who can act as PKI Services administrators. PKI Services administrators play a very powerful role in your organization. The decisions they make when managing certificates and certificate requests determine who will access your computer systems and what privileges they will have when doing so.

Guideline: Give UPDATE authority to only highly trusted individuals, but avoid allowing these same individuals to have direct access to the end-user functions of the `R_PKIServ` callable service described in "Authorizing end-user functions" on page 417. This helps to maintain a secure separation of duties.

Using encrypted passwords for LDAP servers

PKI Services uses an LDAP directory to store certificates. LDAP requires authenticating (binding) to the directory. You can do this by using a distinguished name and passwords. Passwords for binding (to multiple LDAP directories) can be encrypted or in clear text. The UNIX programmer or LDAP programmer or both determine whether or not to use encrypted LDAP bind passwords. You store information about passwords in the PKI Services configuration file, `pkiserv.conf`.

If you do not need the bind password for the LDAP server to be encrypted, you specify the values for Server1, AuthName1 and AuthPwd1 in the `pkiserv.conf` configuration file. If you want the bind password for the LDAP server to be encrypted, you can use of either one of the following profiles:

- A profile named IRR.PROXY.DEFAULTS in the FACILITY class (This profile stores default binding information. It is the profile where PKI Services looks when there is no binding information.)
- A profile (you select the name) in the LDAPBIND class. (You can name this profile whatever you want as long as it matches the BindProfile1 value specified in the `pkiserv.conf` configuration file. (See Step 3 on page 101.)

Before creating either of the preceding profiles, the RACF administrator defines the LDAP.BINDPW.KEY profile in the KEYSMSTR class. This profile contains a SSIGNON segment, which holds either the masked or encrypted value for the key that encrypts passwords stored in the RACF database. Then the RACF administrator creates either of the preceding profiles with a PROXY segment that stores the binding information—the server name, bind distinguished name, and password.

Steps for using encrypted passwords

Perform the following steps to use encrypted LDAP bind passwords:

1. Define a RACF KEYSMSTR class profile by entering the following command, replacing the highlighted value with your own key:

   Example:

   ```
   RDEFINE KEYSMSTR LDAP.BINDPW.KEY SSIGNON(KEYENCRYPTED(0023528875DECFAC))
   ```

   In this example:

   - LDAP BIND passwords are masked by using a key saved in the KEYSMSTR class, LDAP.BINDPW.KEY.
   - The key is **0023528875DECFAC**. (Replace this with your own key.)
   - `KEYENCRYPTED` is specified (rather than `KEYMASKED`) because ICSF is active. (If ICSF is not active, replace `KEYENCRYPTED` with `KEYMASKED`.)

   ---

2. Activate the KEYSMSTR class by entering the following command:

   ```
   SETROPTS CLASSACT(KEYSMSTR)
   ```

   ---

3. If you intend to use the LDAPBIND class, for each LDAP directory, create a RACF LDAPBIND class profile by entering the following command:

   ```
   RDEFINE LDAPBIND MY.LDAP.SERVER1
      PROXY(LDAPHOST(ldap://some.ldap.host:389)
      BINDDN('CN=JOE USER,OU=POUGHKEEPSIE,O=IBM,C=US') BINDPW('MYPASS1')
   ```

   Replace the highlighted parameters as follows:

   a. Optionally, replace *MY.LDAP.SERVER1* with the profile name you want to use.
   b. Replace *ldap://some.ldap.host:389* with your LDAP server URL. You can specify the URL with or without the preceding string “ldap:” or “ldaps:”.
   c. Replace *CN=JOE USER,OU=POUGHKEEPSIE,O=IBM,C=US* with the bind DN.
   d. Replace *MYPASS1* with the bind password.

      Note: All bind DN qualifiers and the bind password are case-sensitive.

4. If you intend to use IRR.PROXY.DEFAULTS instead of the LDAPBIND class for encrypted LDAP bind passwords, execute the following command to create the profile:

```
RDEFINE FACILITY IRR.PROXY.DEFAULTS
   PROXY(LDAPHOST(ldap://some.ldap.host:389)
   BINDDN('CN=JOE USER,OU=POUGHKEEPSIE,O=IBM,C=US') BINDPW('MYPASS1')
```

   Replace the highlighted parameters as follows:

   a. Replace *ldap://some.ldap.host:389* with your LDAP server URL. You can specify the URL with or without the preceding string "ldap:" or "ldaps:".
   b. Replace *CN=JOE USER,OU=POUGHKEEPSIE,O=IBM,C=US* with the bind DN.
   c. Replace *MYPASS1* with the bind password.

      Note: All bind DN qualifiers and the bind password are case-sensitive.

5. Optionally, check your work by listing the segment with the RLIST command. If you are using the LDAPBIND class, execute the following:

```
RLIST LDAPBIND MY.LDAP.SERVER1 PROXY NORACF
```

   Replace `MY.LDAP.SERVER1` with the profile name you used.

   Results: This command displays information like the following:

```
CLASS        NAME
LDAPBIND     MY.LDAP.SERVER1

PROXY INFORMATION
LDAPHOST= LDAP://SOME.LDAP.HOST:389
BINDDN= CN=LDAP ADMINISTRATOR,OU=POUGHKEEPSIE,O=IBM,C=US
BINDPW= YES
```

   If you are using the IRR.PROXY.DEFAULTS profile of the FACILITY class, execute the following command:

```
RLIST FACILITY IRR.PROXY.DEFAULTS PROXY NORACF
```

   Results: This command displays information like the following:

```
CLASS        NAME
FACILITY     IRR.PROXY.DEFAULTS

PROXY INFORMATION
LDAPHOST= LDAP://SOME.LDAP.HOST:389
BINDDN= CN=LDAP ADMINISTRATOR,OU=POUGHKEEPSIE,O=IBM,C=US
BINDPW= YES
```

Part 6. Using the certificate validation service

This part explains how to implement the PKI Services Trust Policy (PKITP) plug-in for OCSF.

- Chapter 21, "PKI Services Trust Policy (PKITP)," on page 425 describes the certificate validation service. It gives an overview of the OCSF plug-in PKITP, describes certificate policies and extensions, and explains additional configuration needed for PKITP and using the Trust Policy API, `CSSM_TP_PassThrough`.

© Copyright IBM Corp. 2001, 2011

Chapter 21. PKI Services Trust Policy (PKITP)

This topic:

- Provides an overview of PKITP, the PKI Services Trust Policy plug-in for OCSF
- Describes:
  - Certificate policies
  - Revoke status checking
  - Certificate extensions
  - CRL extensions and CRL entry extensions
- Explains how to perform additional OCEP configuration needed for PKITP
- Describes `CSSM_TP_PassThrough` (the Trust Policy API).

Overview of PKITP

The PKI Services Trust Policy (PKITP) is an OCSF plug-in to perform certificate validation against a SAF keyring that contains a trusted CA or site certificate (called an *anchor certificate*) or a virtual key ring of either CERTAUTH or SITE certificates. For information about creating a SAF key ring using the RACDCERT ADDRING command, see *z/OS Security Server RACF Command Language Reference*. For information about using a virtual key ring with the `R_datalib` callable service, see *z/OS Security Server RACF Callable Services*.

PKITP supports the following two functions through the implementation of `CSSM_TP_PassThrough`:

- **CertGroupVerify**
- **FreeEvidence**

Server applications running on z/OS can use this function to verify certificates that other network entities (for example, users and other servers) present. PKI Services or other certificate authorities might have issued these certificates.

The server application must attach to and open the key ring using the OCEP DL plug-in. (For more information about OCEP and the use of SAF key rings, see *z/OS Integrated Security Services Open Cryptographic Enhanced Plug-ins Application Programming* .) The server application must also bind to any needed LDAP directories by attaching to and opening these directories using the OCSF LDAPDL plug-in. These LDAP directories can be internal corporate directories, directories of extranet business partners, directories of public certificate authorities, or combinations of these.

The following figure illustrates this diversity. The uppercase letter boxes are certificate authorities, and the lowercase letter boxes are end-entity certificates.

© Copyright IBM Corp. 2001, 2011

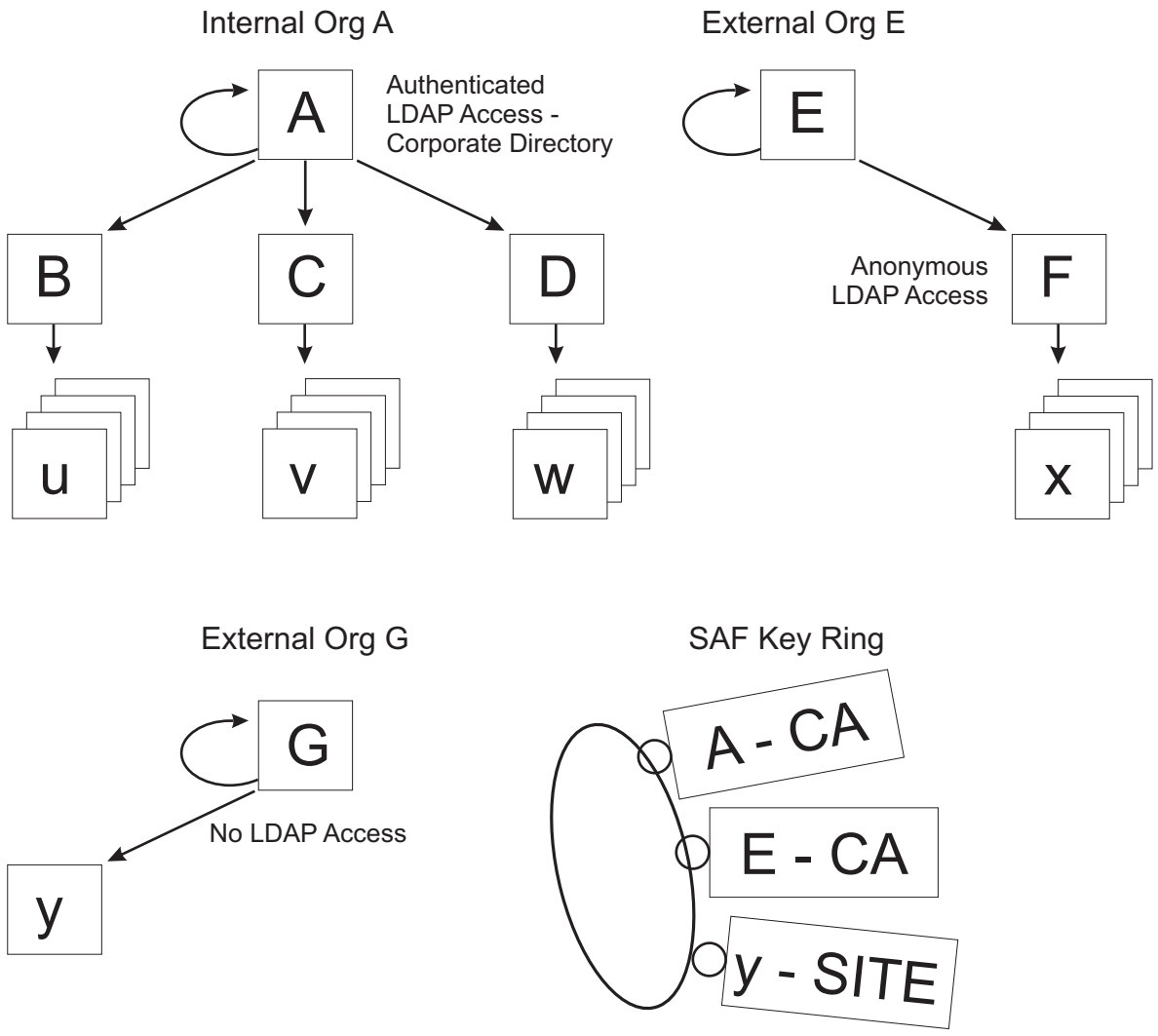


Figure 75. Examples of organizations, certificates, and chains

Organization A represents the local (corporate) certificate hierarchy. It contains one self-signed root certificate "A". Perhaps RACF or the Tivoli PKI created this. "B", "C", and "D" are intermediate CAs. They could be separate instances of PKI Services. Certificates issued within this hierarchy are stored in an LDAP directory accessible to corporate server applications.

Organization E represents a public or business partner's certificate hierarchy with an LDAP directory that allows anonymous access. Organization G represents some other certificate hierarchy, in which either the directory does not exist or it is not accessible. The key ring contains three anchor certificates. Certificates "A" and "E" are trusted CAs, and there is a business need to trust end-entity certificate "y", even though it cannot be verified.

If each of these CAs has posted current CRLs to either their default LDAP locations or to distribution point CRLs in LDAP and all certificate chains to be verified are genuine, the PKITP **CertGroupVerify** function can validate the following input chains:

- Single certificates `x`, `u`, `v`, or `w` (PKITP can extract the missing links from the directories.)

- Chains u-B, v-C, w-D, u–B-A, v-C-A, w-D-A, x-F, or x-F-E (These chains have no missing links.)
- Any chain beginning with certificate "y" (As Figure 75 on page 426 shows, "y" is in the key ring as a SITE. Site certificates are trusted regardless.)

Note that, as with the OCEP Trust Policy, non-self-signed (intermediate) CA certificates can be connected to the key ring to shorten the validation path. Doing so has the following consequences:

- Certificate revocation list (CRL) checking is not performed for the anchor certificate in the chain, even if this happens to be an intermediate CA certificate. If the intermediate CA certificate is revoked, PKITP does not detect it.
- A chain containing the parent chain of the intermediate CA cannot be verified.

  Guideline: When an intermediate CA certificate is connected to the key ring, the certificates that make up its parent chain should be connected as well. This ensures that all chains originating from the intermediate CA or higher can be verified.

Certificate policies

PKITP supports CA and server application-defined certificate policies. CAs can and, in most cases, do establish their own policies for issuing certificates. These policies are declared within issued certificates through the CertificatePolicies extension. When this extension exists and is *not* marked critical, the extension is for informational purposes only - for example, specifying the URL for locating the CA's certificate practice statement (CPS). When this extension exists and *is* marked critical, the policies identified in the extension restrict the use of the certificate. These restrictions apply to subordinate CA certificates and to end-entity certificates. (For information about how PKI Services support the CertificatePolicies extension, see “Using certificate policies” on page 212.)

Similarly, a server application can be a general application that wishes to verify certificates for no specific policy or can be an application that was written for a specific purpose and wishes to verify certificates issued for that purpose (policy).

If the server application specifies an explicit set of policies, then at least one of these policies must be present in each certificate of the certification path (chain). Additionally, PKITP extracts the certificate polices marked critical from each certificate in the chain to determine the intersection - that is, only policies listed in every critically marked CertificatePolicies extension are retained. The server application must indicate that it supports at least one of these polices. If any of these tests is unsuccessful, certificate validation fails.

Checking certificate status with PKITP

PKITP checks the revocation status of a certificate by retrieving certificate revocation lists (CRLs) or, when specified in the certificate, by invoking an online validation service that uses the online certificate status protocol (OCSP).

PKITP certificate revocation checking is performed when useCRLS is set higher than 0. It follows the sequence of validation stages shown in Table 93 on page 428.

Table 93. Sequence of validation stages for PKITP certificate revocation checking

Validation stage	Description
OCSP responder	The trust policy invokes the OCSP responder specified in the AuthInfoAccess extension. If none is specified or if the trust policy fails to receive certificate status from the OCSP responder, it proceeds to the next stage.
DP CRL, using the URI format	The trust policy searches for the DP CRL using the directories, if any, listed in URI format in the CRLDistributionPoints extension in the order they appear. If the DP CRL is found, it is used to determine if the certificate is revoked. If the trust policy fails to find the DP CRL using the URI formats, it proceeds to the next stage.
DP CRL, using the distinguished-name format	The trust policy searches for the DP CRL in the LDAP directories attached through the distinguished name specified, if any, in the CRLDistributionPoints extension. If the trust policy fails to find the DP CRL using the distinguished name and the extension is *not* marked critical, it proceeds to the next stage. If the trust policy fails to find the DP CRL and the extension is marked critical, the validation fails and error code `8029` (CRL not found) is returned. If DP CRL processing is not to be performed (`useCRLS` is set to `0`) and the target certificate contains a CRLDistributionPoints extension marked critical, validation fails and error code `8029` is returned. No attempt is made to locate the DP CRL.
Global revocation list	The trust policy uses the global CRL to find revocation status information for the certificate.

Certificate extensions

PKITP supports the following certificate extensions:

AuthorityInformationAccess Checked for form only.

AuthorityKeyIdentifier Checked for form only.

BasicConstraints For CA certificate, cA flag must be on. Also checked for certification path length.

CertificatePolicies See “Certificate policies” on page 427.

CRLDistributionPoints See “Checking certificate status with PKITP” on page 427.

HostIdMappings Checked for form only.

IssuerAltName Checked for form only. Must be marked critical if the issuer DN is empty.

KeyUsage For CA certificates, the key CertSign flag must be on.

SubjectAltName Checked for form only. Must be marked critical if the subject DN is empty.

SubjectKeyIdentifier Checked for form only.

All other extensions are ignored if they are not marked critical. Unsupported critical extensions prevent certificate validation.

CRL extensions and CRL entry extensions

PKITP supports the following CRL and CRL entry extensions, which are checked for form only:

CRL extensions:

- AuthorityKeyIdentifier
- CRLNumber
- IssuerAltName
- IssuingDistributionPoint

CRL entry extensions:

- CertificateIssuer
- CRLReason
- HoldInstructionCode
- InvalidityDate

All other extensions are ignored if they are not marked critical. Unsupported critical extensions prevent certificate validation.

Files for PKITP

The following table lists files for PKITP:

Table 94. Summary of information about important files for PKITP

File	Description	Source location (default)
`Makefile.pkitpsamp`	Makefile for `pkitpsamp.c`.	`/usr/lpp/pkiserv/samples/`
`install_pkitp`	Program that registers the PKI Services Trust Policy plug-in with OCSF.	`/usr/lpp/pkiserv/bin`
`pkitp_ivp`	This program verifies that the plug-in installed successfully.	`/usr/lpp/pkiserv/bin`
`pkitp.h`	Contains `#define` definitions for applications calling the PKI Services OCSF Trust Policy.	`/usr/lpp/pkiserv/include/`
`pkitp.so`	This is the OCSF Trust Policy plug-in for PKI Services.	`/usr/lpp/pkiserv/lib`
`pkitpsamp.c`	Sample application program (in the C language) to call the PKI Trust Policy plug-in.	`/usr/lpp/pkiserv/samples`

Configuring and getting started with PKITP

If you have not already installed and configured OCSF and OCEP, you need to do so now. Follow the instructions in "Tasks to perform before configuring PKITP" on page 25, and then perform the following post-installation instructions.

The PKITP must be registered with OCSF before being used.

Steps for configuring PKITP

Before you begin: If you have not already done so, run the OCSF and OCEP install and verification scripts.

Perform the following steps to install and configure PKITP:

1. Run the PKITP post installation script by entering the following command:

   ```
   /usr/lpp/pkiserv/bin/install_pkitp
   ```

   The program prompts you for certain information. Assuming PKI Services has been installed in its default location, answer the prompts as follows:

Prompt	Response
addin directory?	`/usr/lpp/pkiserv/lib`
addin filename?	`pkitp.so`
action? [install | uninstall]	`install`

   You know you are done and that the installation was successful when you see the following:

   Result:

   ```
   Installing IBMPKITP...
   Addin successfully installed.
   ```

2. Update your C/C++ environment variable `_CEE_RUNOPTS` to include `XPLINK(ON)` if it does not already include it. For example, execute the following command from a UNIX shell.

   Example:

   ```
   export _CEE_RUNOPTS=$_CEE_RUNOPTS' XPLINK(ON)'
   ```

3. To verify that the installation was successful, run the verification program (`/usr/lpp/pkiserv/bin/pkitp_ivp`).

   You know you are done and that the verification program ran successfully when you see the following:

   ```
   Starting pkitp IVP
   Initializing CSSM
   CSSM Initialized
   Attaching pkitp
   Attach successful, Detaching pkitp
   Detach of pkitp successful
   Completed pkitp IVP
   ```

Trust Policy API

Programming Interface information

PKITP supports only one API, `CSSM_TP_PassThrough`. The globally unique identifier (GUID) for this plug-in is: `{01EBC8AC-CC6F-450c-83B4-F0BE0FBE78F9}`. (Before an application can use a module, an installation application must register the module's name, location, and description with OCSF. The name given to a module includes both a logical name and a GUID. The logical name is a string the module developer chooses to describe the module. The GUID is a structure used to differentiate between service provider modules in the OCSF registry.)

CSSM_TP_PassThrough

Purpose

This function lets applications call TP module-specific operations that have been exported. For PKITP, the module-specific operations support certificate chain validation, based on the CA and SITE certificates that are contained within a key ring.

Format

```
void * CSSMAPI CSSM_TP_PassThrough
      (CSSM_TP_HANDLE TPHandle,
      CSSM_CL_HANDLE CLHandle,
      CSSM_DL_HANDLE DLHandle,
      CSSM_DB_HANDLE DBHandle,
      CSSM_CC_HANDLE CCHandle,
      uint32 PassThroughId,
      const void *InputParams)
```

Parameters

TPHandle
Handle to this Trust Policy module (PKITP).

CLHandle
Not used. PKITP ignores this.

DLHandle
Not used. PKITP ignores this.

DBHandle
Not used. PKITP ignores this.

CCHandle
Not used. PKITP ignores this.

PassThroughId
Used to indicate the pass-through service requested. Two services are provided:

- Service 1 **CertGroupVerify** (`TP_VERIFY_PASSTHROUGH`)
- Service 2 **FreeEvidence** (`TP_FREE_EVIDENCE_PASSTHROUGH`)

InputParams
Pointer to the API-caller-provided input parameter structure. The same structure is used for both pass-through functions. It is declared in `pkitp.h` as follows:

```
typedef struct tp_verify_extra {

   /* similar parameters as TP_CertGroupVerify */
   CSSM_CL_HANDLE CLHandle;
   CSSM_DL_DB_LIST_PTR DBList;
   unsigned int reserved; //@L1C
   CSSM_TP_STOP_ON VerificationAbortOn;
   CSSM_CERTGROUP_PTR CertToBeVerified;

   /* extra parameters: input */
   TP_INITIALPOLICY_PTR InitialPolicy;
   time_t CurrentTime;
   time_t ValidationTime;

   /* extra parameters: output */
   CSSM_BOOL result;
   uint32 DLStatusCode;                  // Status code from DL failures
```

```
    uint32 DLindex;                       // Index (from 0) into DBList
    TP_EVIDENCE_PTR Evidence;

} TP_VERIFY_EXTRA, *TP_VERIFY_EXTRA_PTR;
```

The DB list

This DBList contains one or more handles to open DB stores. The last entry in this list must be a handle to an OCEPDL DB (a real or *virtual* SAF key ring). The key ring is used to declare the list of trusted CA and SITE certificates. Like the OCEP Trust Policy, certificate chains to verify must originate from one of these trusted CAs (anchors) or the end-entity certificate must be one of the SITE certificates. Also like the OCEP Trust Policy, if the security product (SAF) marks any certificate in the candidate chain NOTRUST, the certificate chain fails validation.

The other entries in the list are used for LDAPDL DB stores. PKITP runs through these to locate CRLs and intermediate CA certificates. For each item PKITP requests, the LDAPDLs are queried in the order in which they appear in the list. The search stops the first time an LDAPDL returns an item or when the OCEPDL is reached. No query is made to the OCEPDL to locate CRLs or intermediate CA certificates.

The initial policy

The following optional, caller-provided and initialized structure defines InitialPolicy. PKITP uses the default values if the structure is not provided:

```
typedef struct tp_initialpolicy {

   /* initial-policy-set */
   uint32 NumberofPolicyIdentifiers; // number of application specific
                                     // policy OIDs (defaults to 0)
   CSSM_OID_PTR PolicyIdentifiers;   // Address of array of policy OIDs
                                     // or 0
   uint32 useCRLs;                   // 0 - no CRL processing
                                     // 1 - Check CRLs only if current CRLs found
                                     // 2 - Strong CRL checking (default)
   /* initial-explicit-policy indicator */
   CSSM_BOOL initialExplicitPolicy; // If true, indicates PKITP should
                                    // consider policy set critical
                                    // defaults to false
   /* initial-policy-mapping-inhibit indicator */
   CSSM_BOOL initialPolicyMappingInhibit; // not used, ignored

} TP_INITIALPOLICY, *TP_INITIALPOLICY_PTR;
```

The evidence

The following optional, caller-provided structure defines the evidence. This structure is used to return information relative to the validation decision PKITP makes. The caller must free the data areas returned. (The **FreeEvidence** pass-through function is provided for this.)

```
typedef struct tp_evidence {

   /* valid certification path if validation succeeds */
   CSSM_CERTGROUP_PTR CompleteCertGroup;

   /* relevant CRL if validation fails */
   CSSM_DATA_PTR CRL;

   /* relevant certificate if validation fails */
   CSSM_DATA_PTR Cert;

   /* authority-constrained-policy */
   CSSM_BOOL authAnyPolicy;
   uint32 NumberofAuthCertPolicyIdentifiers;
   CSSM_OID_PTR AuthCertPolicyIdentifiers;
```

```
   /* list of policy mappings that occurred */
   uint32 NumberOfMappedPolicies;
   TP_CSSM_OID_PAIR_PTR mappedPolicies;

} TP_EVIDENCE, *TP_EVIDENCE_PTR;
```

Error codes

Table 95 lists the error codes that are unique to PKI Services OCSF Trust Policy (PKITP).

Table 95. PKI Services OCSF Trust Policy (PKITP) error codes

Decimal value	Error description
8001	Certificate encoding error. Incorrect CertificatePolicies extension.
8002	Certificate policies violation.
8003	Incorrect certificate distinguished name chaining.
8004	Certificate encoding error. Subject name missing.
8006	Incorrect certificate BasicConstraints extension - cA flag off in signing certificate.
8008	Incorrect certificate KeyUsage extension - keyCertSign flag off in signing certificate.
8010	Unsupported AltName form in certificate.
8013	Certificate or CRL encoding error. Signature algorithm mismatch.
8014	Certificate encoding error. Incorrect version.
8015	CRL encoding error. Incorrect version.
8016	Unsupported critical extension in certificate.
8017	Unsupported critical extension in CRL.
8018	Unsupported critical entry extension in CRL.
8019	Certificate encoding error. Duplicate extension.
8020	CRL encoding error. Duplicate extension.
8021	Certificate signature failed verification.
8022	CRL signature failed verification.
8023	Incorrect date range in certificate or CRL. NotAfter earlier than NotBefore.
8024	Certificate's date range is in the future.
8025	Certificate has expired.
8026	CRL's date range is in the future.
8027	CRL has expired.
8028	DBList incorrect, no LDAPDL DBs or non-LDAPDL specified.
8029	CRL not found.
8030	Certificate is revoked.
8031	Unable to build certificate chain.
8033	Certificate not trusted.
8034	Incorrect CRLDistributionPoints extension in certificate.
8501	Unexpected status code returned from accessing LDAPDL.
8502	Unexpected status code returned from accessing OCEPDL.
8503	DBList incorrect, no OCEPDL DB or DB empty.

Building the sample application to invoke the certificate validation service

To perform certificate validation, your server application calls the CSSM_TP_PassThrough API (see CSSM_TP_PassThrough on page 431), passing it the certificate chain to verify. The API returns a Boolean value indicating success or failure, along with additional information about the certificate chain. The pkitpsamp.c code sample that follows at "Code sample of the PKITP program (pkitpsamp.c)" on page 435 is provided as an aid for developing your own server application. By default, you can find this file in the /usr/lpp/pkiserv/samples directory.

Steps for building the sample application

Perform the following steps to build the sample application:

1. Copy the pkitpsamp.c program and Makefile.pkitpsamp to the current directory by entering the following commands:

   ```
   cp /usr/lpp/pkiserv/samples/pkitpsamp.c pkitpsamp.c
   cp /usr/lpp/pkiserv/samples/Makefile.pkitpsamp Makefile
   ```

   _____

2. Before compiling pkitpsamp.c, you need to edit some data (for example, information about how you want the Trust Policy to operate and where your LDAP is located). In the pkitpsamp.c code (see "Code sample of the PKITP program (pkitpsamp.c)" on page 435), find the section that begins with a block comment that says // Start of application specific options. Update the code as necessary up to the block comment that says // End of application specific options:

   a. If the number of LDAP servers is not 1, change NUM_LDAPS.

   b. Update ldap_info by specifying your LDAP server and port (*myldap.mycompany*.com:389 in the sample program). If you have more than one LDAP server, you need to provide this information for each LDAP server.

   c. Replace the "@USERID@/@KEYRINGNAME@" default value for the char ringname[ ] variable in the code sample. Specify either the name of the real SAF key ring containing your trusted CA or site certificates, or the name of the *virtual* key ring that points to all your trusted CA or site certificates.

      - If using a real SAF key ring, specify the owning user ID and ring name of the real SAF key ring. **Example:** patelusr/ring01
      - If using a virtual key ring, replace the default value with either *AUTH*/* or *SITE*/* to point to all your trusted CA or site certificates, respectively. (The name of a virtual key ring is always an asterisk.)

   d. If necessary, change the value of useCRLS:

      0 This means using no CRL processing. (You must specify 0 if you have no LDAP servers.)

      1 This means querying LDAP for CRLs and processing those found. This is the value in the sample.

      2 This means using strong CRL checking. (With strong CRL checking, a valid CRL must be found for each CA certificate in the chain.)

   e. If necessary, change NUM_POLICIES, the policies that the application calling PKITP uses. In the sample, this is 2. For each policy, specify the DER-encoded policy information.

f. If necessary, change `INITIALExplicitPolicy` from the default of `FALSE` to `TRUE` if you want PKITP to require all certificates in the chain to have at least one `policydata` in the preceding list.

---

3. Compile and link to produce the executable, pkitpsamp, by entering the following command:

   ```
   make
   ```

---

4. Export LIBPATH to include `/usr/lpp/pkiserv/lib`.

   Example:

   ```
   export LIBPATH=$LIBPATH:/usr/lpp/pkiserv/lib
   ```

---

5. Enable program control by setting the extended attribute for pkitpsamp.

   Example:

   ```
   extattr +p pkitpsamp
   ```

   Restriction: To execute the **extattr** command with the **+p** option, you must have at least READ access to the BPX.FILEATTR.PROGCTL resource in FACILITY class.

---

6. Update your C/C++ environment variable `_CEE_RUNOPTS` to include `XPLINK(ON)` if it does not already include it. For example, execute the following command from a UNIX shell.

   Example:

   ```
   export _CEE_RUNOPTS=$_CEE_RUNOPTS' XPLINK(ON)'
   ```

---

7. Run the `pkitpsamp.c` in your own directory by entering the following command:

   ```
   pkitpsamp
   ```

Code sample of the PKITP program (pkitpsamp.c)

Note: The following listing might not be identical to the code sample shipped with the product. For the most current sample, see the `/usr/lpp/pkiserv/samples` directory.

```
/*********************************************************************/
/* This file contains sample code. IBM provides this code on an      */
/* 'as is' basis without warranty of any kind, either express or     */
/* implied, including but not limited to, the implied warranties     */
/* of merchantability or fitness for a particular purpose.           */
/*********************************************************************/
/*********************************************************************/
/*                                                                   */
/*    Licensed Materials - Property of IBM                           */
/*    5694-A01                                                       */
/*    (C) Copyright IBM Corp. 2001, 2005                             */
/*    Status = HKY7720                                               */
/*                                                                   */
/*********************************************************************/
/*    Sample use of IBM PKITP program                                */
/*                                                                   */
/*    Purpose: Program attaches needed CSSM modules, then prompts    */
/*             the user for filename(s) containing DER encoded       */
/*             certificates.  The certificate(s) are read from the   */
/*             file, then passed to PKITP for verification.          */
/*             A summary of the results are printed to stdout.       */
/*                                                                   */
```

```
/* Caution: In order to run this sample program, modification MUST   */
/*          BE MADE to several values assigned to the following      */
/*          variables that are defined between the block comment     */
/*          containing the text "Start of application specific       */
/*          options" and the block comment containing the text       */
/*          "End of application specific options"(without the        */
/*          quotation marks):                                        */
/*                                                                   */
/*   #define NUM_LDAPS 1                                             */
/*           Define the number of LDAP servers that PKITP should     */
/*           query for certificates, CRLs and ARLs.  This can be 0,  */
/*           if entire certificate chain will be passed as input to  */
/*           PKITP AND caller requests to NOT process CRLs/ARLs (see */
/*           useCRLs option below).                                  */
/*                                                                   */
/*   struct ldap_info ldapserver[NUM_LDAPS] =                        */
/*                    { "@LDAPSERVERNAME:PORTNUMBER@",               */
/*                      "@LDAPUSER@",                                */
/*                      "@LDAPUSERPASSWORD@"};                       */
/*           If NUM_LDAPS > 0, then ldapserver array should define   */
/*           the LDAP server:port, user and password for each LDAP   */
/*           server. Replace @LDAPSERVERNAME:PORTNUMBER@ with the    */
/*           appropriate ldap server name and port number (e.g       */
/*           myldap.mycompany.com:389 ). Replace @LDAPUSER@ with the */
/*           appropriate ldap admin user name (e.g cn=root) and      */
/*           @LDAPUSERPASSWORD@ with the password for the specified  */
/*           ldap user name (e.g rootpw)                             */
/*                                                                   */
/*   char keyring[] = "@USERID@/@KEYRINGNAME@";                      */
/*           Define the SAF keyring containing trusted CA and/or     */
/*           site certificates. Format is "USERID/keyname". Replace  */
/*           @USERID@ with the userid of the keyring owner and       */
/*           @KEYRINGNAME@ with the name of the keyring. (e.g        */
/*           IBMUSER/CAring) Note that the userid and the keyring    */
/*           names are case sensitive so the userid is all           */
/*           uppercase and the keyring name is mixed case in this    */
/*           example.                                                */
/*                                                                   */
/*   #define USECRLS 1                                               */
/*           Define how the useCRLs option should be set.            */
/*             Set to 0 if no CRL processing is to be performed      */
/*             Set to 1, if LDAP is to be queried for CRLs and       */
/*             process the CRLs found.                               */
/*             Set to 2, for strong CRL checking (With strong CRL    */
/*             checking, a valid CRL must be found for each CA       */
/*             certificate in the chain.)                            */
/*                                                                   */
/*   #define NUM_POLICIES 2                                          */
/*   static unsigned char my_policy1[5] =                            */
/*          {0x06,0x03,0x2a,0x03,0x04};       // DER encoded 2.3.4   */
/*   static unsigned char my_policy2[7] =                            */
/*          {0x06,0x05,0x2a,0x03,0x03,0x02,0x01}; // DER 2.3.3.2.1   */
/*   CSSM_DATA policydata[NUM_POLICIES] =                            */
/*          {{sizeof(my_policy1),(unsigned char *)my_policy1},       */
/*           {sizeof(my_policy2),(unsigned char *)my_policy2}};      */
/*           Define the policies that the application calling PKITP  */
/*           uses. These become important if a certificate in the    */
/*           certificate chain has a critically marked policy        */
/*           extension.  At least one policy that is listed in such  */
/*           a critically marked policy extension, must appear in    */
/*           the list defined here or PKITP will return certificate  */
/*           policy error.                                           */
/*                                                                   */
/*   #define INITIALExplicitPolicy FALSE                             */
/*           Set to true if you want PKITP to require that all       */
/*           certificates in chain to have at least one policy       */
/*           listed by the policydata defined above.                 */
/*                                                                   */
/*******************************************************************/
/*                                                                   */
/*Change-Activity:                                                   */
/* $D0=MG00545, HKY7708, 020306, BRW: Initialize                @D0A*/
/*              TP_EVIDENCE fields for printEvidence            @D0A*/
/*                                                                  */
/* $D1=MG00547, HKY7708, 020306, BRW: Change to allow compile   @D1A*/
/*              if NUM_LDAPS or NUM_POLICIES is zero            @D1A*/
/* $D2=MG01368, HKY7708, 021030, BRW: Delete attaching of CSP   @D2A*/
/* $D3=MG04177, HKY7720, 040614, TCG: Fix memory/init errors    @D3A*/
/*******************************************************************/
#pragma runopts("XPLINK(ON)")
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <cssm.h>
#include <ibmocepdl.h>
#include <ibmswcsp.h>
#include <cssmapi.h>
#include <cssmtype.h>
#include <pkitp.h>
#include <ldapdl.h>

struct ldap_info
```

```
 {
 char * ldapserver;
 char * ldapauthuser;
 char * ldapauthpass;
 };

//------------------------------------------------------------
// storage function definitions needed to talk to CSSM
//------------------------------------------------------------

#ifdef __cplusplus
extern "C"
#endif
void * OurMalloc(size_t size, void * allocRef)
{
   return malloc(size);
}

#ifdef __cplusplus
extern "C"
#endif
void OurFree(void* memPtr, void * allocRef)
{
   free(memPtr);
}

#ifdef __cplusplus
extern "C"
#endif
void * OurRealloc(void * memPtr,
                  size_t size,
                  void * allocRef)
{
   return realloc(memPtr, size);
}

#ifdef __cplusplus
extern "C"
#endif
void * OurCalloc(size_t num,
              size_t size,
              void* allocRef)
{
   return calloc(num, size);
}

static CSSM_API_MEMORY_FUNCS memoryFuncs; // used to pass function addresses to CSSM

//------------------------------------------------------------
// internal function declarations
//------------------------------------------------------------
int connectTP(char * ringname,
              int number_ldap,
              CSSM_DL_DB_LIST *,
              CSSM_TP_HANDLE *);  //@D1C

void disconnectTP(CSSM_DL_DB_LIST *, CSSM_TP_HANDLE);

int buildCertGroup(CSSM_CERTGROUP *, char * [], uint32);
void verifyCertGroup(CSSM_CERTGROUP certgroup,
                     CSSM_DL_DB_LIST * datasources_ptr,
                     CSSM_TP_HANDLE tphandle);

void
reportCertGroupVerify
   (TP_VERIFY_EXTRA extraVerifyInfo);

void printEvidence(TP_EVIDENCE_PTR evidence_ptr);

void freeCertGroup(CSSM_CERTGROUP * certGroupPtr);
////////////////////////////////////////////////////////////////////////////
//
// Start of application specific options
//
// The defines and declarations that follow should be altered to fit the
// particular application calling PKITP.
//
////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////
// Define the number of LDAP servers that PKITP should query for certificates,
// CRLs and ARLs.  This can be 0, if entire certificate chain will be passed as
// input to PKITP AND caller requests to NOT process CRLs/ARLs (see useCRLs
// option below).
//
// If NUM_LDAPS > 0, then ldapserver array should define the LDAP server:port,
// user and password for each LDAP server, as this example shows.
////////////////////////////////////////////////////////////////////////////
#define NUM_LDAPS 1

#if NUM_LDAPS != 0   //@D1A
struct ldap_info ldapserver[NUM_LDAPS] =
```

```
                                  { "@LDAPSERVERNAME:PORTNUMBER@",  // LDAP server:port
                                    "@LDAPUSER@",                   // user
                                    "@LDAPUSERPASSWORD@"};          // password
#endif  //@D1A

////////////////////////////////////////////////////////////////////////////
// Define the SAF keyring containing trusted CA and/or site certificates.
// Format is "USERID/keyname"
////////////////////////////////////////////////////////////////////////////
char keyring[] = "@USERID@/@KEYRINGNAME@";

////////////////////////////////////////////////////////////////////////////
// Define how the useCRLs option should be set.
// Set to 0 if no CRL processing to be done
// Set to 1, if we are to query LDAP for CRLs and process those found
// Set to 2, for strong CRL checking -- must find CRLs in LDAP.
////////////////////////////////////////////////////////////////////////////

#define USECRLS 1

////////////////////////////////////////////////////////////////////////////
// Define the policies that the application calling PKITP uses.
//
// These become important if a certificate in the certificate chain has a
// critically marked policy extension.  At least one policy
// that is listed in such a critically marked policy extension, must appear
// in the list defined here or PKITP will return certificate policy error.
////////////////////////////////////////////////////////////////////////////

#define NUM_POLICIES 2

#if NUM_POLICIES != 0   //@D1A
static unsigned char my_policy1[5] = {0x06,0x03,0x2a,0x03,0x04};  // DER encoded 2.3.4
static unsigned char my_policy2[7] = {0x06,0x05,0x2a,0x03,0x03,0x02,0x01};  // DER 2.3.3.2.1

CSSM_DATA policydata[NUM_POLICIES] = {{sizeof(my_policy1),(unsigned char *)my_policy1},
                                      {sizeof(my_policy2),(unsigned char *)my_policy2}};
#endif   //@D1A


#define INITIALExplicitPolicy FALSE  // Set to true if you want PKITP to require that all
                                     // certificates in chain have at least one policy
                                     // listed by our policydata defined above

////////////////////////////////////////////////////////////////////////////
//
// End of application specific options
//
////////////////////// ///////////////////////////////////////////////////

//------------------------------------------------------------
// main
//------------------------------------------------------------
int
main(int argc, char* argv[])
{

 CSSM_DL_DB_LIST datasources;
 CSSM_TP_HANDLE  tphandle = 0;
 CSSM_CERTGROUP certGroup;
 int repeating = 1;
 char buffer[1024];
 int num_certs = 0;
 char * cert_files[25];
 char * next_file;
 char * input;

 int rc;

 rc = connectTP(keyring,NUM_LDAPS, &datasources, &tphandle); //@D1C
 if (rc == 0)
  {
  ////////////////////////////////////////////////////////////////////////////
  // prompt for certificates to verify
  ////////////////////////////////////////////////////////////////////////////
  do
    {
    num_certs = 0;
    printf("Enter filename(s) of certificate(s).  (List EE first).  ");
    printf("Blank line to quit.\n");

    if ((input = gets(buffer)) != NULL)      // get input line
      {
      next_file = strtok(input," ");
      while ((next_file != NULL) && (num_certs < 25)) // tokenize it
        {
        cert_files[num_certs] = next_file;
        num_certs++;
        next_file = strtok(NULL," ");
        }
      }
```

```
    ////////////////////////////////////////////////////////////////
    // If we were given a list of files containing certificates, input them to TP
    ////////////////////////////////////////////////////////////////
    if (num_certs > 0)
      {
      rc = buildCertGroup(&certGroup, cert_files, num_certs);
      if (rc == 0)
        {
        verifyCertGroup(certGroup, &datasources, tphandle);
        freeCertGroup(&certGroup);
        }
      }
    } while (num_certs > 0);
  }
 disconnectTP(&datasources, tphandle);
}

//-------------------------------------------------------------
// connectTP
//
// Purpose:  connect to the datasources PKITP needs
// then connect to the PKITP
//
// Input:  ringname - string containing "USERID/ringname" of SAF
//                   keyring containing trusted CA and/or SITE certificates
//         number_ldap - number of ldap servers
//         ldapserver - array of ldap_info structures
//
// Output: The CSSM_DL_DB_LIST structure addressed by datasources will have
//         been initialized with the various handles that CSSM_ModuleAttach
//         and CSSM_DL_DbOpen calls have returned
//         The CSSM_TP_HANDLE addressed by tphandle_ptr will have been initialied.
//         int returned will be 0 if successful, -1 if not successful.
//-------------------------------------------------------------

int connectTP(char * ringname,
              int number_ldap,
              CSSM_DL_DB_LIST * datasources_ptr,
              CSSM_TP_HANDLE * tphandle_ptr)        //@D1C
  {
   uint32 status = 0;
   int z;
   CSSM_VERSION cssm_version = {CSSM_MAJOR, CSSM_MINOR};
   CSSM_DB_ACCESS_TYPE access = { CSSM_TRUE,
                                  CSSM_FALSE,
                                  CSSM_FALSE,
                                  CSSM_FALSE};
   CSSM_VERSION DL_version;
   CSSM_DL_HANDLE LDAP_dlhandle;
   CSSM_MODULE_INFO* moduleInfoPtr;
   void * voidptr;
   CSSM_DB_ACCESS_TYPE accessRequest = { CSSM_TRUE,     // ReadAccess
                                        CSSM_TRUE,     // WriteAccess
                                        CSSM_FALSE,    // PrivilegedMode
                                        CSSM_FALSE  }; // Asynchronous


   memoryFuncs.malloc_func = OurMalloc;
   memoryFuncs.free_func = OurFree;
   memoryFuncs.realloc_func = OurRealloc;
   memoryFuncs.calloc_func = OurCalloc;
   memoryFuncs.AllocRef = NULL;

   DL_version.Major = IBMOCEPDL_MAJOR_VERSION;
   DL_version.Minor = IBMOCEPDL_MINOR_VERSION;

   datasources_ptr->NumHandles = number_ldap + 1;
   voidptr = malloc(sizeof(CSSM_DL_DB_HANDLE)*(number_ldap +1));   // get storage for DBlist
   if (voidptr == NULL) {                                          // @D3A
      printf("connectTP unable to obtain memory: line %d\n",__LINE__); // @D3A
      return -1;                                                   // @D3A
   }                                                               // @D3A

   memset(voidptr,0,(sizeof(CSSM_DL_DB_HANDLE)*(number_ldap +1))); // zero it
   datasources_ptr->DLDBHandle = (CSSM_DL_DB_HANDLE *)voidptr;

   if (CSSM_Init(&cssm_version, &memoryFuncs, NULL) != CSSM_OK)
        {
        printf("Failed CSSM_Init: %d, line %d\n",CSSM_GetError()->error,__LINE__);
        return -1;
        }

   ////////////////////////////////////////////////////////////////////////////
   // attach to LDAP and open each LDAP DB
   ////////////////////////////////////////////////////////////////////////////
#if NUM_LDAPS != 0   //@D1A
   if (number_ldap > 0)              // if we have any LDAP sources
     {
     moduleInfoPtr = CSSM_GetModuleInfo((CSSM_GUID*)&LDAPDL_GUID,
                                        CSSM_SERVICE_DL,
                                        CSSM_ALL_SUBSERVICES,
                                        CSSM_INFO_LEVEL_ALL_ATTR);
```

```
    if (!moduleInfoPtr)
      {
      printf("Failed CSSM_GetModduleInfo: %d, line %d\n",CSSM_GetError()->error,__LINE__);
      return -1;
      }

    LDAP_dlhandle = CSSM_ModuleAttach((CSSM_GUID*)&LDAPDL_GUID,
                                &moduleInfoPtr->Version,
                                &memoryFuncs,
                                0,
                                0,
                                0,
                                NULL,
                                NULL);
    if (!LDAP_dlhandle)
      {
      printf("Failed CSSM_ModuleAttach: %d, line %d\n",CSSM_GetError()->error,__LINE__);
      return -1;
      }

    // connect to multiple database instances

    //-------------------------------------------------------------
    // fill in LDAP DL authentication information:
    // necessary only if user is supplying a name and password
    //-------------------------------------------------------------
    for (z = 0; z < number_ldap; z++)             // for each LDAP source
      {
      LDAP_BIND_PARMS bindParms;
      CSSM_USER_AUTHENTICATION userAuthentication = {0,0};
      CSSM_DATA userCredential = {0,0};
      CSSM_USER_AUTHENTICATION_PTR userAuthenticationPtr = 0;

      datasources_ptr->DLDBHandle[z].DLHandle = LDAP_dlhandle;
      if (ldapserver[z].ldapauthuser && ldapserver[z].ldapauthpass)   //@D1C
        {
        //-------------------------------------------------------------
        // fill in LDAP DL specific data structure: LDAP_BIND_PARMS
        //-------------------------------------------------------------
        bindParms.DN = ldapserver[z].ldapauthuser;                    //@D1C
        bindParms.SASL = 0;
        bindParms.credentials.Data = (uint8 *)ldapserver[z].ldapauthpass;  //@D1C
        bindParms.credentials.Length = strlen(ldapserver[z].ldapauthpass)+1;  //@D1C
        userCredential.Length = sizeof(LDAP_BIND_PARMS);
        userCredential.Data   = (unsigned char*)&bindParms
        userAuthentication.Credential = &userCredential
        userAuthenticationPtr = &userAuthentication
        }

     //-------------------------------------------------------------
     // Open LDAP DL Database
     //-------------------------------------------------------------
     datasources_ptr->DLDBHandle[z].DBHandle = CSSM_DL_DbOpen(LDAP_dlhandle,
                                     ldapserver[z].ldapserver,            //@D1C
                                     &accessRequest,
                                     userAuthenticationPtr,
                                     (void *)0);
     if (!datasources_ptr->DLDBHandle[z].DBHandle)
       {
       printf("Failed CSSM_DL_DbOpen %d, line %d\n", CSSM_GetError()->error,__LINE__);
       return -1;
       }

    }                             // end of for each each LDAP source

    if (CSSM_FreeModuleInfo(moduleInfoPtr) == CSSM_FAIL)
    {
    printf("Failed CSSM_FreeModuleInfo, line %d, error %d\n",__LINE__,
           CSSM_GetError()->error);
    // This is not a catastrophic error, we'll continue
    }
  }                               // end if we have any LDAP sources
#endif   //@D1A

  ////////////////////////////////////////////////////////////////////////
  // Attach to OCEP DL (to access RACF keyring)
  ////////////////////////////////////////////////////////////////////////

  datasources_ptr->DLDBHandle[number_ldap].DLHandle =
                                CSSM_ModuleAttach(&IBMOCEPDL_GUID,
                                &DL_version,
                                &memoryFuncs,
                                0,
                                0,
                                0,
                                NULL,
                                NULL);


  if (!(datasources_ptr->DLDBHandle[number_ldap].DLHandle))
    {
    printf("Failed CSSM_ModuleAttach: %d, line %d\n",CSSM_GetError()->error,__LINE__);
```

```
    return -1;
    }

  datasources_ptr->DLDBHandle[number_ldap].DBHandle =
              CSSM_DL_DbOpen(datasources_ptr->DLDBHandle[number_ldap].DLHandle,
                             ringname,
                             &access,
                             NULL,
                             NULL);

  if (!(datasources_ptr->DLDBHandle[number_ldap].DBHandle))
    {
    printf("Failed CSSM_DL_DbOpen %d, line %d\n", CSSM_GetError()->error,__LINE__);
    return -1;
    }


  ////////////////////////////////////////////////////////////////////
  // Attach to PKITP
  ////////////////////////////////////////////////////////////////////
  moduleInfoPtr = CSSM_GetModuleInfo((CSSM_GUID*)&PKITP_GUID,
                                     CSSM_SERVICE_TP,
                                     CSSM_ALL_SUBSERVICES,
                                     CSSM_INFO_LEVEL_ALL_ATTR);
  if (!moduleInfoPtr)
    {
    printf("Failed CSSM_GetModduleInfo: %d, line %d\n",CSSM_GetError()->error,__LINE__);
    return -1;
    }

  *(tphandle_ptr) = CSSM_ModuleAttach((CSSM_GUID*)&PKITP_GUID,
                                &moduleInfoPtr->Version,
                                &memoryFuncs,
                                0,
                                0,
                                0,
                                NULL,
                                NULL);
  if (!(*tphandle_ptr))
    {
    printf("Failed CSSM_ModuleAttach: %d, line %d\n",CSSM_GetError()->error,__LINE__);
    return -1;
    }

  if (CSSM_FreeModuleInfo(moduleInfoPtr) == CSSM_FAIL)
     {
     printf("Failed CSSM_FreeModuleInfo, line %d, error %d\n",__LINE__,
           CSSM_GetError()->error);
     // This is not a catastrophic error, we'll continue
     }

 return 0;
}

//------------------------------------------------------------
// disconnectTP
//
// Purpose:  to close any open databases and detach any CSSM modules
//           that connectTP attached
//
// Input:  The CSSM_DL_DB_LIST structure, CSSM_TP_HANDLE,
//         that were initialized by connectTP.
//
// Output: None
//------------------------------------------------------------

void disconnectTP(CSSM_DL_DB_LIST * datasources_ptr, CSSM_TP_HANDLE tphandle)
 {
 int x;
 int status;

 ////////////////////////////////////////////////////////////////
 // Sever ties to LDAP
 // For each LDAP database opened -- call CSSM_DL_DbClose
 ////////////////////////////////////////////////////////////////

#if NUM_LDAPS != 0   //@D1A
 for (x = 0; x < datasources_ptr->NumHandles - 1; x++)
    {
     // we close each ldap database separately
     if (datasources_ptr->DLDBHandle[x].DBHandle)        // if we opened database
       {
       status = CSSM_DL_DbClose(datasources_ptr->DLDBHandle[x]);
       if (status != 0)
         {
         printf("Failed CSSM_DL_DbClose %d, line %d\n", CSSM_GetError()->error,__LINE__);
         // we continue trying to close other stuff
         }
       }
    }

  ////////////////////////////////////////////////////////////////
```

```
   // Now detach the LDAP module
   ////////////////////////////////////////////////////////////////
   if (datasources_ptr->DLDBHandle[0].DLHandle)
     {
     if ((status = CSSM_ModuleDetach(datasources_ptr->DLDBHandle[0].DLHandle)) != 0)
       {
       printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error,__LINE__);
       // we continue trying to close other stuff
       }
     datasources_ptr->DLDBHandle[0].DLHandle = 0;   // clear handle
     }
#endif  //@D1A

  ////////////////////////////////////////////////////////////////
  // Say goodbye to OCEP
  ////////////////////////////////////////////////////////////////
   status = CSSM_DL_DbClose(datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1]);
   if (status != 0)
      {
      printf("Failed CSSM_DL_DbClose %d, line %d\n", CSSM_GetError()->error,__LINE__);
      // we continue trying to close other stuff
      }

   if (datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle)
     {
     if ((status = CSSM_ModuleDetach(datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle)) != 0)
       {
       printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error,__LINE__);
       // we continue trying to close other stuff
       }
     datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle = 0;
     }

  ////////////////////////////////////////////////////////////////
  // Farewell PKITP
  ////////////////////////////////////////////////////////////////
  if (tphandle)
    {
    if ((status = CSSM_ModuleDetach(tphandle)) != 0)
      {
      printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error,__LINE__);
      // we continue trying to close other stuff
      }
    }

 return;
}

 /*********************************************************************
  *  name: buildCertGroup - read certificates from files, set up
  *        CSSM_CERTGROUP to reference input certificates
  *
  *  input: CSSM_CERTGROUP * -- addresses unintialized CSSM_CERTGROUP
  *         certFile - array of strings containing names of files that
  *             have DER encoded certificates to be verified by PKITP
  *         certCount - number of elements (strings) in certFile
  *
  *  output: returns CSSM_OK if all certificates read
  *          -  CSSM_CERTGROUP will have NumCerts set and CertList
  *             will be the address of array of certificates
  *          returns CSSM_FALSE if error reading a file
  *********************************************************************/

 int buildCertGroup(CSSM_CERTGROUP * certGroupPtr,
                    char * certFile[], uint32 certCount)
 {
   FILE      * inFile;
   CSSM_DATA * certArray = (CSSM_DATA *) calloc(certCount,sizeof(CSSM_DATA));
   uint32    i, certSize;

   if (certArray == NULL) // If calloc failed, exit now
     return(CSSM_FAIL);                                        // @D3A

   certGroupPtr->NumCerts = certCount;
   certGroupPtr->CertList = certArray;

   for (i=0; i < certCount; i++) {
     inFile = fopen(certFile[i],"rb");
     if (!inFile) {
       printf("File %s could not be opened\n",certFile[i]);
       if (i > 1)          // if we've read any certs before this
       {
         certGroupPtr->NumCerts = i - 1;   // indicate how many read
         freeCertGroup(certGroupPtr);      // free alloc'd storage
       }
       return(CSSM_FAIL);
     }
     /* Find size of certificate file */
     fseek(inFile,0L,SEEK_END);
     certSize = ftell(inFile);
     rewind(inFile);
```

```
    /* Read in certificate data*/
    certArray[i].Length = certSize;
    certArray[i].Data   = (uint8 *)calloc(certSize, sizeof(char));
    if (certArray[i].Data == NULL) {  // If calloc failed         @D3A
      if (i > 1)          // if we've read any certs before this  @D3A
      {                                                          // @D3A
        certGroupPtr->NumCerts = i - 1;   // indicate how many read @D3A
        freeCertGroup(certGroupPtr);      // free alloc'd storage @D3A
      }                                                          // @D3A
      return(CSSM_FAIL);                                         // @D3A
    }                                                            // @D3A
    fread(certArray[i].Data, 1, certSize, inFile);
    fclose(inFile);
  }
  return(CSSM_OK);
}


 /*********************************************************************
  *  name: verifyCertGroup - call the Trust Policy (FINALLY)
  *
  *  purpose:  call CSSM_TP_PassThrough (PKITP) to verify certificate(s)
  *            call reportCertGroupVerify (internal routine to display
  *               results to stdout
  *            call CSSM_TP_PassThrough (PKITP) to free storage related to
  *               results
  *
  *  input:    CSSM_CERTGROUP containing number of and array of certificates
  *            CSSM_DL_DB_LIST containing CSSM handles for LDAP and OCEP
  *            CSSM_TP_HANDLE  CSSM handle for PKITP
  *
  *  output:   none
  *
  *********************************************************************/

  void verifyCertGroup(CSSM_CERTGROUP certgroup,
                       CSSM_DL_DB_LIST * datasources_ptr,
                       CSSM_TP_HANDLE tphandle)
  {
  ////////////////////////////////////////////////////////////////////////
  //
  // While there are only 3 parameters on CSSM_TP_PassThrough call to PKITP:
  //  - the CSSM_TP_HANDLE,
  //  - the function code "TP_VERIFY_PASSTHROUGH" and
  //  - a pointer to the TP_VERIFY_EXTRA structure.
  // TP_VERIFY_EXTRA structure contains many parameters, including the address of
  // TP_INITIALPOLICY structure that can be used to override the default
  // policy settings and the address of TP_VERIFY_EXTRA which PKITP can use
  // to pass back more detailed results.
  ////////////////////////////////////////////////////////////////////////
  TP_INITIALPOLICY initialPolicyPreferences;
  TP_EVIDENCE pkixEvidence;
  TP_VERIFY_EXTRA extraVerifyInfo;

  memset(&extraVerifyInfo, 0x00, sizeof(TP_VERIFY_EXTRA));     // @D3A
  ////////////////////////////////////////////////////////////////////////
  // The field initialPolicyMappingInhibit in TP_INITIALPOLICY is not used
  // by PKITP, therefore we do not set it.
  ////////////////////////////////////////////////////////////////////////
  initialPolicyPreferences.NumberofPolicyIdentifiers = NUM_POLICIES;
#if NUM_POLICIES != 0   //@D1A
  initialPolicyPreferences.PolicyIdentifiers = policydata;
#else                   //@D1A
  initialPolicyPreferences.PolicyIdentifiers = NULL;    //@D1A
#endif                  //@D1A
  initialPolicyPreferences.initialExplicitPolicy = INITIALExplicitPolicy;
  initialPolicyPreferences.initialPolicyMappingInhibit = CSSM_FALSE;
  initialPolicyPreferences.useCRLs = USECRLS;

  ////////////////////////////////////////////////////////////////////////
  // Initialize TP_EVIDENCE fields for printEvidence in case call
  // to PKITP, is not successful.
  pkixEvidence.CompleteCertGroup = NULL;  /* @D0A */
  pkixEvidence.CRL       = NULL;          /* @D0A */
  pkixEvidence.Cert      = NULL;          /* @D0A */
  ////////////////////////////////////////////////////////////////////////

  ////////////////////////////////////////////////////////////////////////
  // The following fields in TP_VERIFY_EXTRA are not used by PKITP.
  // CLHandle, PolicyIdentifiers and NumberofPolicyIdentifiers
  // (not to be confused with fields of same name in TP_INITIALPOLICY structure),
  // AnchorCerts and NumberofAnchorCerts.
  // Therefore we do not set these fields below.
  ////////////////////////////////////////////////////////////////////////
  extraVerifyInfo.DBList                    = datasources_ptr;
  extraVerifyInfo.VerificationAbortOn       = CSSM_TP_STOP_ON_POLICY;
  extraVerifyInfo.CertToBeVerified          = &certgroup
  extraVerifyInfo.InitialPolicy             = &initialPolicyPreferences
  extraVerifyInfo.Evidence                  = &pkixEvidence
  extraVerifyInfo.ValidationTime            = time(0);
```

```
   (void*)CSSM_TP_PassThrough(tphandle,
                              0,
                              0,
                              0,
                              0,
                              TP_VERIFY_PASSTHROUGH,
                              (void *)&extraVerifyInfo);

   reportCertGroupVerify(extraVerifyInfo);
   (void*)CSSM_TP_PassThrough(tphandle,
                              0,
                              0,
                              0,
                              0,
                              TP_FREE_EVIDENCE,
                              (void *)&extraVerifyInfo);
   }

 //============================================================================
// function: reportCertGroupVerify
//============================================================================

void
reportCertGroupVerify
   (TP_VERIFY_EXTRA extraVerifyInfo)
{
   //---------------------------------------
   // report success or failure
   //---------------------------------------
   unsigned int reported_err = CSSM_GetError()->error;

   printf("TP_VERIFY_PASSTHROUGH : ");
   if (CSSM_FALSE == extraVerifyInfo.result)
   {
    printf("FAILED.  Error code: %d\n",reported_err);
   }
   else
   {
    printf("PASSED\n");
   }

   //---------------------------------------
   // report evidence
   //---------------------------------------
   printEvidence(extraVerifyInfo.Evidence);


}

void printEvidence(TP_EVIDENCE_PTR evidence_ptr)
{
 if (evidence_ptr == NULL) return;
 if (evidence_ptr->CompleteCertGroup)
   {
   printf("CompleteCertGroup was returned containing %d certificates at address %x\n",
     evidence_ptr->CompleteCertGroup->NumCerts,
     evidence_ptr->CompleteCertGroup->CertList);
   }
 else printf("CompleteCertGroup was NULL.\n");

 if (evidence_ptr->CRL)
   {
   printf("CRL was returned of %d bytes (decimal) at address %x\n",
     evidence_ptr->CRL->Length,
     evidence_ptr->CRL->Data);
   }
 else printf("CRL was NULL.\n");

  if (evidence_ptr->Cert)
   {
   printf("Cert (failed certificate) was returned of %d bytes (decimal) at address %x\n",
     evidence_ptr->Cert->Length,
     evidence_ptr->Cert->Data);
   }
 else printf("Cert was NULL.\n");
}

  /*********************************************************************
   *  name: freeCertGroup - Free certificate data storage              *
   *********************************************************************/

   void freeCertGroup(CSSM_CERTGROUP * certGroupPtr)
   {
   CSSM_DATA      * certArray = certGroupPtr->CertList;
   uint32           i;
   uint32           certCount = certGroupPtr->NumCerts;

   for (i=0; i <= certCount-1; i++)
    {
    free(certArray[i].Data);
```

```
 }
free(certArray);
return;
}
```

End of Programming Interface information

Part 7. Troubleshooting

This part explains using logs for troubleshooting, including the following:

- Chapter 22, "Using information from SYS1.LOGREC," on page 449 discusses SYS1.LOGREC, which is used to record unusual runtime events, such as an exception.
- Chapter 23, "Using information from the PKI Services logs," on page 455 discusses using the PKI Services logs, which are ongoing, to debug problems and explains how to change logging options and display log options settings.

You can also use the iclview and vosview utilities for troubleshooting. For more information about these utilities, see "Using the iclview utility" on page 359 and "Using the vosview utility" on page 372.

© Copyright IBM Corp. 2001, 2011

Chapter 22. Using information from SYS1.LOGREC

SYS1.LOGREC keeps records of unusual runtime events, such as exceptions or unexpected return codes from calls to system services. It records hardware errors, selected software errors, and selected system conditions in the LOGREC data set. You can use the LOGREC data set as a starting point for diagnosing a problem. It supplies symptom data about the failure and shows the order in which errors occurred. After you have collected this information, you should report the problem to the IBM support center.

The following table describes the contents of the LOGREC data for PKI Services:

Table 96. LOGREC data for PKI Services

CSECT	Description	
IKYP0N IKYP81 IKYP8A IKYP8B	Issued when an ABEND occurs in the one of the CSECTs running on the Monitor Thread.	
	Primary symptom string:	
	Component ID (PIDS):	5752XXPKI
	Load module:	IKYPKID#L
	CSECT:	IKYP0N, IKYP81, IKYP8A, or IKYP8B
	Recovery routine:	ESTEXIT
	Error Information:	Consists of an abend code and reason code: **Abend code:** The character *S* followed by 4 hexadecimal digits or the character *U* followed by 4 decimal digits. **Reason code:** 8 hexadecimal digits.

© Copyright IBM Corp. 2001, 2011

Table 96. LOGREC data for PKI Services (continued)

CSECT	Description
IKYP8A	Issued when an exception is caught in the service thread routine IKYP8A01 or in the services thread request routine IKYP8A02.
	Primary symptom string: **Component ID (PIDS):** 5752XXPKI **Load module:** IKYPKID#L **CSECT:** IKYP8A **Failing routine:** IKYP8A01 or IKYP8A02 **Error information:** Consists of *either* an abend code and a reason code *or* a facility ID and a message number. **Abend code:** If present, either the character *U* followed by 4 decimal digits or the character *S* followed by 3 hexadecimal digits. **Reason code:** If present, 8 hexadecimal digits. **Facility ID:** If present, 3 characters. **Message number:** If present, 8 hexadecimal digits.
	Secondary symptom string: **USER** The user ID of the requestor. **FUNC** A function code of 8 hexadecimal digits.
IKYP8B	Issued when an ABEND occurs in the PC routine (or helper routines).
	Primary symptom string: **Component ID (PIDS):** 5752XXPKI **Load module:** IKYPKID#L **CSECT:** IKYP8B **Recovery routine:** ARREXIT **Error information:** Consists of an abend code and a reason code. **Abend code:** The character *S* followed by 4 hexadecimal digits or the character *U* followed by 4 decimal digits. **Reason code:** 8 hexadecimal digits.

Table 96. LOGREC data for PKI Services (continued)

CSECT	Description
IKYSCHDR	Issued from the **dispatcher()** function when an exception is caught while creating and posting a CRL to LDAP.
	Primary symptom string: **Component ID (PIDS):** 5752XXPKI **Load module:** IKYAPI#L **CSECT:** IKYSCHDR **Failing routine:** IKYDSPER **Error information:** Consists of *either* an abend code and a reason code *or* a facility ID and a message number. **Abend code:** If present, either the character *U* followed by 4 decimal digits or the character *S* followed by 3 hexadecimal digits. **Reason code:** If present, 8 hexadecimal digits. **Facility ID:** If present, 3 characters. **Message number:** If present, 8 hexadecimal digits.
	Secondary symptom string: **THREAD** The string `DISPATCHR`.
IKYSTART	Issued when an exception occurs during `daily_timer()` processing (general housekeeping for certificate requests and issued certificates).
	Primary symptom string: **Component ID (PIDS):** 5752XXPKI **Load module:** IKYAPI#L **CSECT:** IKYSTART **Failing routine:** IKYDAYTM **Error information:** Consists of *either* an abend code and a reason code *or* a facility ID and a message number. **Abend code:** If present, either the character *U* followed by 4 decimal digits or the character *S* followed by 3 hexadecimal digits. **Reason code:** If present, 8 hexadecimal digits. **Facility ID:** If present, 3 characters. **Message number:** If present, 8 hexadecimal digits.
	Secondary symptom string: **THREAD** The string `DAY_TIMR`.

Table 96. LOGREC data for PKI Services (continued)

CSECT	Description
IKYTIMER	Issued when an exception is caught while processing a timer event in **wakeup_rtn()**.
	Primary symptom string: **Component ID (PIDS):** 5752XXPKI **Load module:** IKYOSSRV#L **CSECT:** IKYTIMER **Failing routine:** IKYWAKUP **Error Information:** Consists of *either* an abend code and a reason code *or* a facility ID and a message number. **Abend code:** If present, either the character *U* followed by 4 decimal digits or the character *S* followed by 3 hexadecimal digits. **Reason code:** If present, 8 hexadecimal digits. **Facility ID:** If present, 3 characters. **Message number:** If present, 8 hexadecimal digits.
	Secondary symptom string: **EVENTFUNC** The name of the event routine being processed (postEvt, createEvt, or removeEvt).

Sample LOGREC data

The following is a sample of a LOGREC data for PKI Services:

```
TYPE:  SYMPTOM RECORD       REPORT:  SOFTWARE EDIT REPORT           DAY YEAR
                                                        REPORT DATE: 221  01
SCP:   VS 2 REL 3                                        ERROR DATE: 221  01
                            MODEL:  9672                             HH MM SS.TH
                            SERIAL: 048288                    TIME: 19:05:16.02
SEARCH ARGUMENT ABSTRACT:
    PIDS/5752XXPKI RIDS/IKYPKID#L RIDS/IKYP8A RIDS/IKYP8A01 AB/S0C4
    FLDS/RSNCODE VALU/H00000000

SYSTEM ENVIRONMENT:
    CPU MODEL:  9672               DATE:  221  01
    CPU SERIAL: 048288             TIME:  19:05:16.02
    SYSTEM:     DCEIMGUI           BCP:   MVS
    RELEASE LEVEL OF SERVICE ROUTINE:      HBB7703
    SYSTEM DATA AT ARCHITECTURE LEVEL:     10
    COMPONENT DATA AT ARCHITECTURE LEVEL:  10
    SYSTEM DATA:  00000000 00000000                          |........|
COMPONENT INFORMATION:
    COMPONENT ID:             5752XXPKI
    COMPONENT RELEASE LEVEL:  7706
    SERVICE RELEASE LEVEL:    HKY7706
    DESCRIPTION OF FUNCTION:  PKI SERVICES DAEMON

PRIMARY SYMPTOM STRING:
    PIDS/5752XXPKI RIDS/IKYPKID#L RIDS/IKYP8A RIDS/IKYP8A01 AB/S0C4
    FLDS/RSNCODE VALU/H00000000

    SYMPTOM            SYMPTOM DATA     EXPLANATION
    ---------------    -------------    -----------
    PIDS/5752XXPKI     5752XXPKI        COMPONENT IDENTIFIER
    RIDS/IKYPKID#L     IKYPKID#L        ROUTINE IDENTIFIER
    RIDS/IKYP8A        IKYP8A           ROUTINE IDENTIFIER
    RIDS/IKYP8A01      IKYP8A01         ROUTINE IDENTIFIER
    AB/S0C4            0C4              ABEND CODE - SYSTEM
    FLDS/RSNCODE       RSNCODE          DATA FIELD NAME
    VALU/H00000000     00000000         ERROR RELATED HEXADECIMAL VALUE

SECONDARY SYMPTOM STRING:
    FLDS/USER VALU/CG422253 FLDS/FUNC VALU/H00000000

    SYMPTOM            SYMPTOM DATA     EXPLANATION

    ---------------    -------------    -----------
    FLDS/USER          USER             DATA FIELD NAME
    VALU/CG422253      G422253          ERROR RELATED CHARACTER VALUE
    FLDS/FUNC          FUNC             DATA FIELD NAME
    VALU/H00000000     00000000         ERROR RELATED HEXADECIMAL VALUE

THE SYMPTOM RECORD DOES NOT CONTAIN FREE FORMAT COMPONENT INFORMATION.
HEX DUMP OF RECORD:
  HEADER
    +000    4C831800    00000000    0001221F    19051602    |<C.............|
    +010    FF048288    96720000                            |..BHO...       |
```

Figure 76. Sample LOGREC data (Part 1 of 2)

```
SYMPTOM RECORD
   +000    E2D9F9F6    F7F2F0F4    F8F2F8F8    FFFFCA5B    |SR9672048288...$|
   +010    B64312D1    0360F103    40404040    40404040    |...J.-1.        |
   +020    4040C4C3    C5C9D4C7    E4C9F5F7    F5F2C8C2    |  DCEIMGUI5752HB|
   +030    C2F7F7F0    F3400080    00000000    00000000    |B7703 ..........|
   +040    F1F00030    00640070    005C0138    003101A0    |10.......*......|
   +050 LENGTH(0032) ==> ALL BYTES CONTAIN X'00'.
   +070    E2D9F2F1    F1F0F5F7    F5F2E7E7    D7D2C900    |SR21105752XXPKI.|
   +080    F7F7F0F6    C8D2E8F7    F7F0F640    00000000    |7706HKY7706 ....|
   +090    00000000    00000000    00000000    D7D2C940    |............PKI |
   +0A0    E28599A5    898385A2    40848185    94969540    |SERVICES DAEMON |
   +0B0    40404040    40404040    40404040    00000000    |            ....|
   +0C0    00000000    00000000    00000000    00000000    |................|
   +0D0    00000000    0B41465C    0B414668    0B414699    |.......*.......R|
   +0E0    0B4146A8    0B4146A8    0B4146A8    01000000    |...Y...Y...Y....|
   +0F0    0B4144C8    00000000    00000000    F0F1F2F3    |...H........0123|
   +100    F4F5F6F7    F8F9C1C2    C3C4C5C6    00680040    |456789ABCDEF... |
   +110    0000000F    0B414530    00000000    0B414374    |................|
   +120    00000000    F0F00000    00000008    00000008    |....00..........|
   +130    00000000    40E70030    D7C9C4E2    61F5F7F5    |.... X..PIDS/575|
   +140    F2E7E7D7    D2C940D9    C9C4E261    C9D2E8D7    |2XXPKI RIDS/IKYP|
   +150    D2C9C47B    D340D9C9    C4E261C9    D2E8D7F8    |KID#L RIDS/IKYP8|
   +160    C140D9C9    C4E261C9    D2E8D7F8    C1F0F140    |A RIDS/IKYP8A01 |
   +170    C1C261E2    F0C3F440    C6D3C4E2    61D9E2D5    |AB/S0C4 FLDS/RSN|
   +180    C3D6C4C5    40E5C1D3    E461C8F0    F0F0F0F0    |CODE VALU/H00000|
   +190    F0F0F040    0B414780    00000001    00000000    |000 ............|
   +1A0    C6D3C4E2    61E4E2C5    D940E5C1    D3E461C3    |FLDS/USER VALU/C|
   +1B0    C7F4F2F2    F2F5F340    C6D3C4E2    61C6E4D5    |G422253 FLDS/FUN|
   +1C0    C340E5C1    D3E461C8    F0F0F0F0    F0F0F0F0    |C VALU/H00000000|
   +1D0    40                                              |                |
```

Figure 76. Sample LOGREC data (Part 2 of 2)

Chapter 23. Using information from the PKI Services logs

This topic explains viewing SYSOUT information. It describes the `_PKISERV_MSG_LEVEL` environment variable and lists subcomponents and message levels you can select. It explains how to display and change logging options.

Viewing SYSOUT information

To start PKI Services, you use the PKISERVD sample procedure (see "PKISERVD sample procedure to start PKI Services daemon" on page 562 for a code sample of the JCL). When you start PKI Services, error and informational messages for the PKISERVD job are written to the STDOUT and STDERR file streams. Unless you change the DD statements that specify STDOUT and STDERR in the PKISERVD sample procedure, PKI Services writes these messages to SYSOUT.

To view the SYSOUT information of a job, you use the Spool Display Search Facility (SDSF) or a comparable facility. If you are using SDSF, you can use the question mark line command (by entering a question mark in the prefix area in front of the file name) to separate the job files, including STDOUT and STDERR. Figure 77 on page 456 shows this.

© Copyright IBM Corp. 2001, 2011

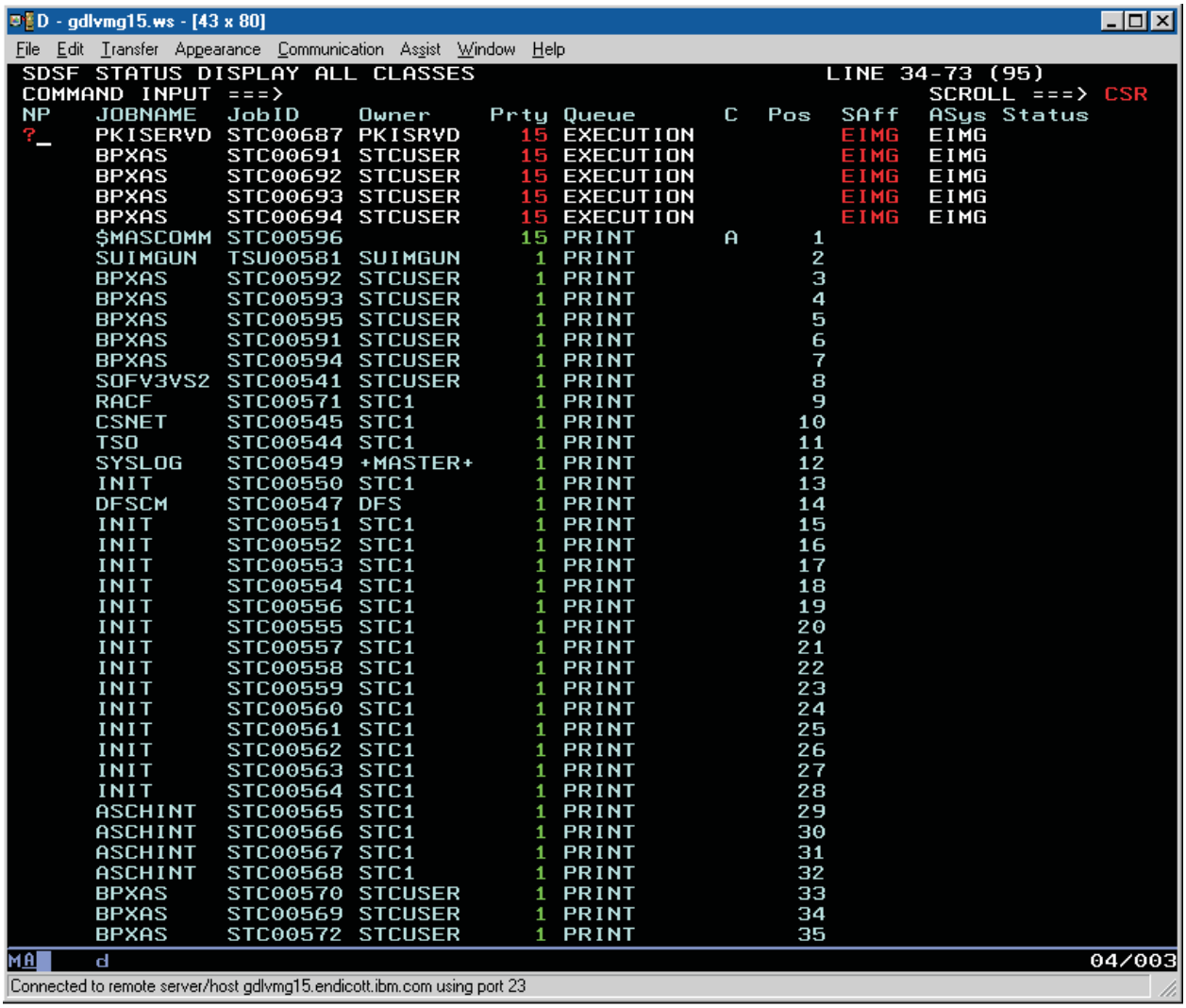


Figure 77. Separating the job files

After using the question mark line command, you can select the file you want to view by entering an *S* before this file name. Figure 78 on page 457 shows this:

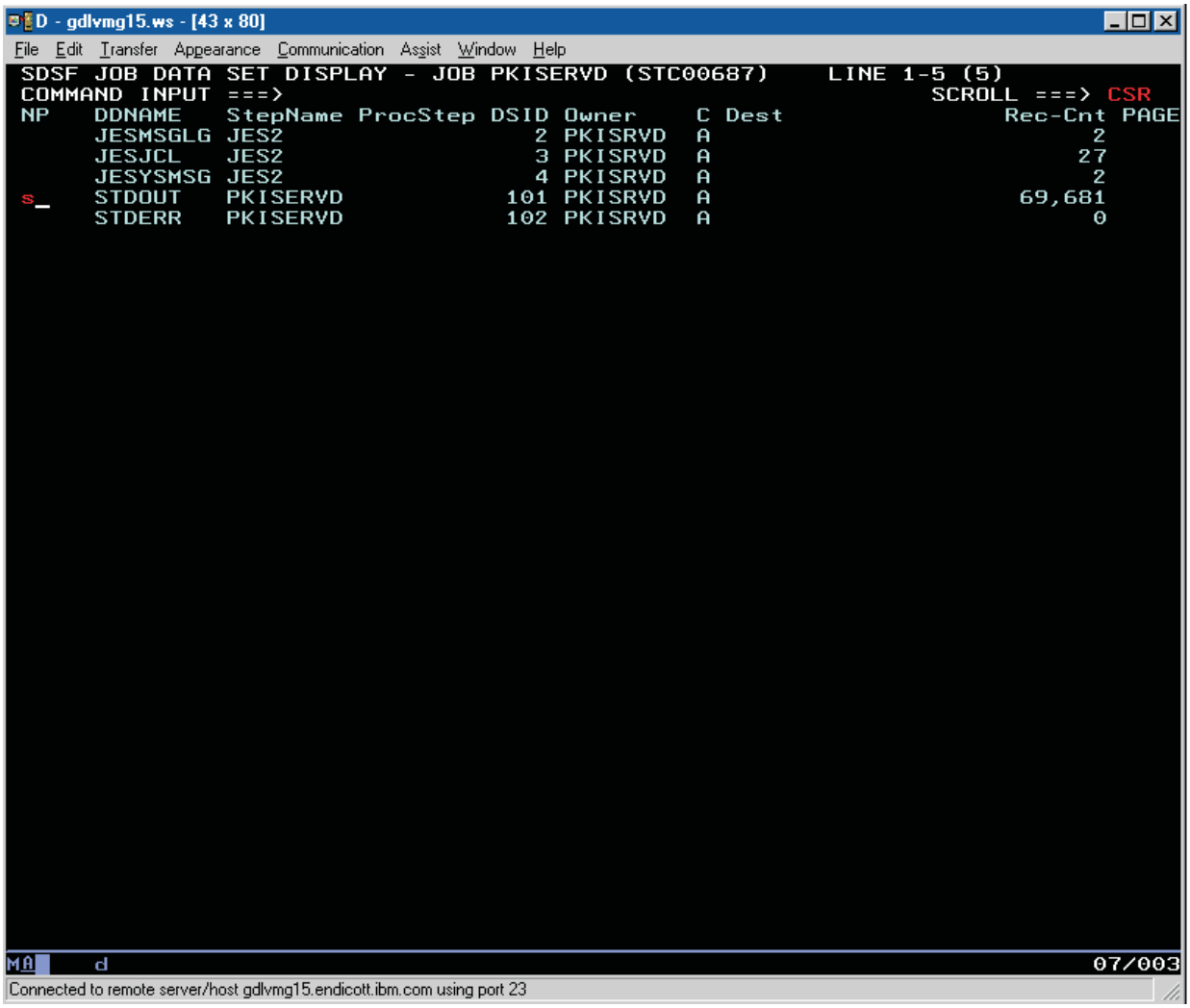


Figure 78. Selecting a file to view

Figure 79 on page 458 shows the messages contained in the file:

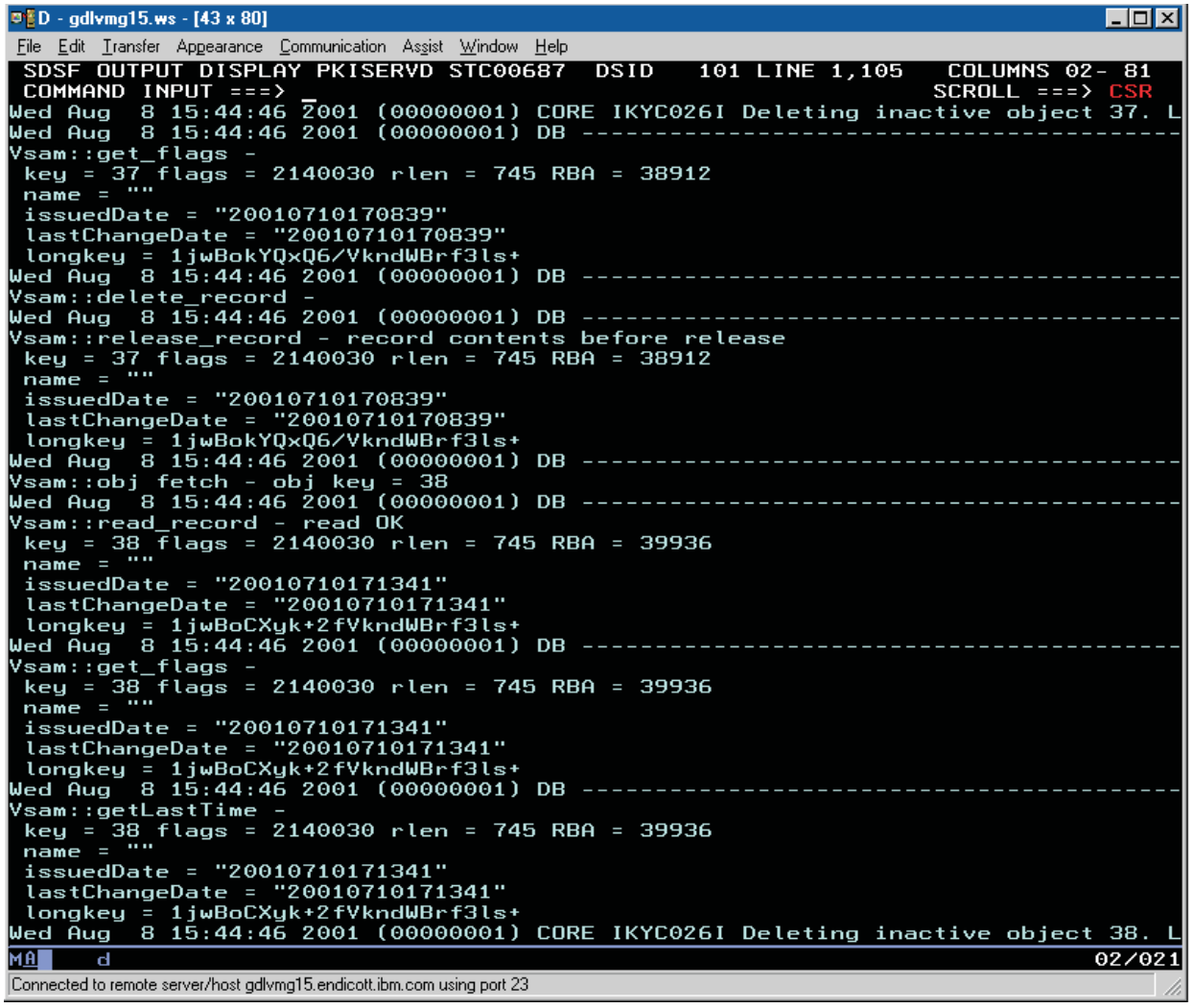


Figure 79. Messages contained in the file

Notes:

1. These messages were produced when Verbose tracing was active.
2. The SYSOUT records have a logical record length of 133, so you might have to scroll to the right to see the entire record.

From left to right, each record contains:

- A time stamp
- The thread identifier, in parenthesis
- The subcomponent name (in the example that follows, this is CORE)
- The message itself, which might span multiple lines

  Informational, warning, error, and severe level messages begin with a message number. (See Chapter 24, “Messages,” on page 463.) Verbose and diagnostic level messages do not have message numbers and are not documented.

The following is an example of an informational message:

```
Wed Aug 8 15:44:46 2001 (00000001) CORE IKYC026I Deleting inactive object 37.
     Last changed at 2001/07/10 17:08:39
```

_PKISERV_MSG_LEVEL subcomponents and message levels

The _PKISERV_MSG_LEVEL environment variable specifies the subcomponent and message level for logging messages.

The subcomponents are listed below:

Subcomponent	Meaning
*	The wildcard character (represents all subcomponents)
CORE	The core functions of PKI Services that are not specific to the other subcomponents
DB	Activity related to the object store or issued certificate list repositories
LDAP	LDAP posting operations
PKID	The PKI Services daemon address setup and infrastructure
POLICY	Certificate creation and revocation policy processing
SAF	SAF key ring, OCEP, and R_datalib calls
TPOLICY	Trust policy plug-in processing

The message levels are listed hierarchically below:

Debug level	Meaning
S	This indicates logging only severe messages.
E	This indicates logging severe and error messages.
W	This indicates logging severe, error, and warning messages. This is the *default* message level for all subcomponents if you do not set the environment variable.
I	This indicates logging severe, error, warning, and informational messages.
D	This indicates logging severe, error, warning, informational, and diagnostic messages.
V	This indicates logging *all* messages, including verbose diagnostic messages. This is very verbose. **Guideline:** Do not use **V** level unless IBM support personnel instruct you to do so.

(For information about updating environment variables during configuration, see "Optionally updating PKI Services environment variables" on page 57.)

After PKI Services is up and running, if a problem occurs, the MVS programmer can:

- Change the logging options dynamically - by using the MODIFY (or **F**) console command
- Display the current settings - by using another MODIFY console command.

Changing logging options

To change logging options dynamically, execute the following MODIFY (or **F**) console command:

```
F PKISERVD,LOG sub-component.level[,sub-component.level...]
```

subcomponent.level
Sets the message level setting(s) for the subcomponent(s). Use one of the subcomponents and message levels listed previously.

Displaying log options settings

To display the current logging options, execute the following MODIFY (or **F**) console command:

```
F PKISERVD,DISPLAY
```

The result of this command is information message IKYP025I. **Sample output:**

```
10.37.39 STC00146: IKYP025I PKI SERVICES SETTINGS:
  CA DOMAIN NAME: Customers
  SUBCOMPONENT              MESSAGE LEVEL
     LDAP                   ERROR MESSAGES AND HIGHER
     SAF                    WARNING MESSAGES AND HIGHER
     DB                     INFORMATIONAL MESSAGES AND HIGHER
     CORE                   WARNING MESSAGES AND HIGHER
     PKID                   VERBOSE DIAGNOSTIC MESSAGES AND HIGHER
     POLICY                 WARNING MESSAGES AND HIGHER
     TPOLICY                WARNING MESSAGES AND HIGHER
  MESSAGE LOGGING SETTING: STDOUT_LOGGING
  CONFIGURATION FILE IN USE:
/etc/pkiserv/pkiserv.conf
  TEMPLATE FILE IN USE:
/etc/pkiserv/pkiserv.tmpl
  CA CERTIFICATE FINGERPRINTS:
   SHA1:   BB:B5:AF:38:BA:3B:33:61:46:F5:FE:AD:20:33:10:98:C2:D7:9A:BC
   MD5:    C6:E6:B2:F3:39:F0:7C:B5:A6:B6:F0:36:5F:2F:7D:C8
   SHA256: FC:F6:DE:AF:CF:48:15:90:0E:91:9B:8F:5C:93:9B:FF:
           1D:2D:FC:B1:10:33:2C:CB:B5:02:F4:8E:5E:41:FA:F8
   SHA512: 14:DD:45:4C:78:66:47:0D:7B:BB:BE:56:33:F0:18:52:
           F4:AD:0C:96:B9:78:5B:40:FF:AE:D5:EB:62:87:A6:22:
           48:45:37:D6:4B:3A:DD:5C:F0:7D:6F:A5:D8:6F:6E:36:
           E5:8C:77:D2:B5:BC:3E:14:E2:34:F8:A1:11:31:2B:E3
```

The PKI administrator can use this command to display the fingerprints of the PKI Services CA certificate in support of Simple Certificate Enrollment Protocol (SCEP) certificate requests. For more information, see “Checking certificate fingerprints” on page 249.

Part 8. Reference information

This part provides reference information, including listings of code samples for certain important files.

Note: The listings in this part might not be identical to the code samples shipped with the product. For the most current sample, see the appropriate source directory.

- Chapter 24, "Messages," on page 463 explains PKI Services messages.
- Chapter 25, "File directory structure," on page 497 describes product and file system directories for PKI Services and files contained in them.
- Chapter 27, "Environment variables," on page 507 explains the `pkiserv.envars` environment variables file and provides a code sample.
- Chapter 28, "The IKYSETUP REXX exec," on page 511 explains the contents of the IKYSETUP REXX exec that performs RACF administration and provides a code sample.
- Chapter 29, "Other code samples," on page 545 provides additional code samples. Table 97 summarizes information about these code samples and those in the preceding chapters, summarizing their use, directory location, and the page where the code sample begins.
- Chapter 30, "SMF recording," on page 565 describes the content of the System Management Facility (SMF) record that is generated by PKI Services.

Table 97. Summary of information about important files

File	Description	Source location (default)	For code sample...
`httpd.conf` and `httpd2.conf`	Contain z/OS HTTP Server directives.	`/usr/lpp/pkiserv/samples/`	See page 545.
IKYCDB2	Sample to create DB2 objects for the object store and issued certificate list (ICL)	`SYS1.SAMPLIB`	See page 548
IKYCVSAM	Sample IDCAMS JCL to create VSAM data sets (regardless of whether you are using a sysplex or non-sysplex).	`SYS1.SAMPLIB`	See page 551.
IKYRVSAM	Sample IDCAMS JCL to add VSAM record-level sharing (RLS) support. IKYRVSAM reallocates your VSAM data sets in preparation for sharing in a sysplex.	`SYS1.SAMPLIB`	See page 555.
IKYSBIND	Sample job to create the DB2 package and plan for the object store and issued certificate list (ICL)	`SYS1.SAMPLIB`	See page 559
IKYSETUP	REXX exec to set up RACF profiles.	`SYS1.SAMPLIB`	See page 511.
IKYVBKUP	Sample JCL to back up the PKI Services VSAM data sets using the DFSMSdss DUMP utility		See page560.
IKYVREST	Sample JCL to restore the PKI Services VSAM data sets from a backup taken with the DFSMSdss DUMP utility		See page 560.

© Copyright IBM Corp. 2001, 2011

Table 97. Summary of information about important files (continued)

File	Description	Source location (default)	For code sample...
pkiserv.conf	PKI Services configuration file.	/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)	See page 501.
PKISERVD	Sample procedure to start PKI Services daemon.	SYS1.PROCLIB	See page 562.
pkiserv.envars	PKI Services environment variables file.	/usr/lpp/pkiserv/samples/ (You might need to copy this file to the runtime directory, /etc/pkiserv.)	See page 510.
pkiserv.tmpl	PKI Services certificate template file for REXX CGI execs.	/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)	Not provided.
pkitmpl.xml	PKI Services certificate template file for Java server pages.	/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)	Not provided.
pkixgen.tmpl	PKI Services certificate template file for the daemon's use when you implement the Web application using Java server pages.	You create this file from pkitmpl.xml using the TemplateTool utility.	Not provided.

Chapter 24. Messages

PKI Services message numbers begin with the three-character component prefix (**IKY**), followed by a fourth character that identifies the subcomponent. The following table lists the characters representing various subcomponents and describes where the messages appear.

Table 98. Meaning of fourth character in message number

Character	Meaning	Component producing messages	Where messages appear
C	CORE	Core subcomponent	PKI Services log
D	DB	Database accessing subcomponent	PKI Services log
I	INTERFACE	PKISERV CGIs or JSPs	In the user's Web browser window
L	LDAP	LDAP bind subcomponent	PKI Services log
O	POLICY	Certificate creation and revocation policy subcomponent	PKI Services log
P	PKID	PKI Services daemon address space controller	• PKI Services log • (For those with destination and routing codes) operators console
S	SAF	SAF interfacing subcomponent	PKI Services log
U	UTILITY	Utility programs	UNIX standard error (`stderr`)

Characters five through seven are numeric. The eighth character is the message type:

Table 99. Meaning of eighth character in message number

Character	Meaning	Action required
I	Informational (status message)	No action required
E	Eventual action	Possible problem that might require eventual action
A	Action required	Problem that requires immediate attention

For information about setting messages options using environment variables, see 508.

IKYC001I **Error** *nnnn action-being-performed*: *error-code-description*

Explanation: PKI Services is processing a request and has encountered an internal error. The action being performed and the error code encountered are displayed. A description of the error is also displayed, if known.

System action: The request is not processed.

User response: Report the error to the IBM support center.

IKYC002I **Error** *nnnn* **returned from CP_NewCertCreate:** *error-code-description*

Explanation: PKI Services is attempting to create a certificate and has encountered an internal error. The action being performed and the error code encountered

© Copyright IBM Corp. 2001, 2011

are displayed. A description of the error is also displayed, if known.

System action: The certificate is not created.

System programmer response: Report the error to the IBM support center.

IKYC003I **Error** *nnnn* **registering the next CRL cutting job:** *error-code-description*

Explanation: PKI Services has just finished creating the current CRL and is attempting to schedule the next CRL creation thread. An error was encountered. The error code encountered is displayed. A description of the error is also displayed, if known.

System action: Future CRLs are not created until the problem is corrected and PKI Services is restarted.

System programmer response: Look for other error messages that might be issued such as IKYC011I. If no other messages were issued, report the error to the IBM support center.

IKYC004I **Error** *nnnn* **creating and sending CRLs:** *error-code-description*

Explanation: PKI Services is attempting to create the current CRL and has encountered an error. The error code encountered is displayed. A description of the error is also displayed, if known. Note: If the error code is an LDAP return code, no error description is displayed. This would indicate a problem posting the CRL to the LDAP directory.

System action: If the CRL was created and the post to LDAP was unsuccessful, the post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database and deleted. For all other errors, PKI Services tries again to create the CRL during the next CRL interval.

System programmer response: If this is a problem with posting to LDAP, you should also see messages IKYC007I or IKYC008I or both. If so, follow the instructions for these messages. Otherwise, report the error to the IBM support center.

IKYC005I **Error** *nnnn* **posting {User | CA} Certificate to LDAP for distinguished-name:** *error-code-description*

Explanation: PKI Services is attempting to post a certificate to the LDAP directory and has encountered an error. The distinguished name for which the post was attempted and the error code encountered are displayed. A description of the error is also displayed, if known. If the error code is an LDAP return code, no error description is displayed.

System action: If the post is unsuccessful for a given certificate, PKI Services retries the post at the next post interval. If the post continues to be unsuccessful after 3 attempts, the post frequency for the certificate is reduced to no more than once per hour. After 26 unsuccessful attempts, it is further reduced to no more than once per day. After 33 unsuccessful attempts, the post request for the certificate is deleted from the request database.

System programmer response: Determine if the error occurred on the call to LDAP or within PKI Services, based on the presence of an error code description in the message. If no error code description is displayed in the message, the error occurred on the call to LDAP. If the error code is `LDAP_NO_SUCH_OBJECT`, the LDAP entry could not be created because the required suffix does not exist. Check the message to determine the entry that could not be created. If the entry should be posted to LDAP, you need to define the suffix in the LDAP server configuration file, and then stop and restart the LDAP server. For all other LDAP errors, follow the instructions in *IBM Tivoli Directory Server Client Programming for z/OS*. If an error code description is displayed in the message, the error occurred within PKI Services.

If the error code description is `Missing LDAP information`, then the `CreateOUValue` directive is missing from the **LDAP** section of the PKI Services configuration file. Add the directive, then stop and restart PKI Services. See Chapter 8, “Tailoring the PKI Services configuration file for LDAP,” on page 97 for more information.

Report any other PKI Services error to the IBM support center. If message IKYC009I is also displayed, report that information as well.

IKYC007I **Error** *nnnn* **posting {CRL | ARL} to LDAP:** *error-code-description*

Explanation: PKI Services is attempting to post a CRL or ARL to the LDAP directory and has encountered an error. The error code encountered is displayed. A description of the error is also displayed, if known. Note: If the error code is an LDAP return code, no error description is displayed.

System action: The post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database and deleted.

System programmer response: If the error is `LDAP_NO_SUCH_OBJECT`, the LDAP entry to contain the CRL or ARL does not yet exist. This is expected if you are starting PKI Services for the first time. For all other LDAP errors, follow the instructions in *IBM Tivoli Directory Server Client Programming for z/OS*. Report errors to the IBM support center. If message IKYC009I is also displayed, report that information as well.

IKYC008I **Error** *nnnn* **creating an entry for {CA Certificate | User Cert | CRL | ARL} to LDAP for** *distinguished-name*: *error-code-description*

Explanation: PKI Services is attempting to post a certificate, CRL or ARL to the LDAP directory and has encountered an error. The distinguished name for which the post was attempted and the error code encountered are displayed. A description of the error is also displayed, if known. Note: If the error code is an LDAP return code, no error description is displayed.

System action: The post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database and deleted.

System programmer response: You might also see message IKYC005I or IKYC007I displayed. If so, follow the instructions for the message displayed. Follow related instructions in *IBM Tivoli Directory Server Client Programming for z/OS*. Report errors to the IBM support center. If message IKYC009I is also displayed, report that information as well.

IKYC009I **LDAP post unsuccessful for object id =** *nnnn*, **state =** *nnnn*, **status =** *nnnn*: *status-code-description*

Explanation: This message appears as supplemental information for messages IKYC005I and IKYC008I.

System programmer response: If reporting message IKYC005I or IKYC008I to the IBM support center, report this information as well.

IKYC010I **Error** *nnnn* **returned from action-being-performed:** *error-code-description*

Explanation: PKI Services is processing a request and has encountered an error. The action being performed and the error code encountered are displayed. A description of the error is also displayed, if known.

System action: The request is not processed.

System programmer response: Check the PKI Services logs for additional information about this error. You can use the _PKISERV_MSG_LEVEL environment variable to increase the messages that PKI Services generates. For more information about the logs and _PKISERV_MSG_LEVEL, see Chapter 23, "Using information from the PKI Services logs," on page 455.

If the error code description is not self-explanatory, and you cannot find additional information in the logs, report the error to the IBM support center.

IKYC011I **Bad TimeBetweenCRLs value in pkiserv.conf file:** *incorrect-value*

Explanation: PKI Services is reading its configuration file to locate the value specified for `TimeBetweenCRLs` in the **CertPolicy** section. The value specified has an incorrect syntax.

System action: CRL processing is suspended until the problem is corrected and PKI Services is restarted.

System programmer response: Correct the value and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC012I **Bad CRLDuration value in pkiserv.conf file:** *incorrect-value*

Explanation: PKI Services is reading its configuration file to locate the value specified for `CRLDuration` in the **CertPolicy** section. The value specified has an incorrect syntax.

System action: CRL processing is suspended until the problem is corrected and PKI Services is restarted.

System programmer response: Correct the value and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC013I **Bad CreateInterval value in pkiserv.conf file**

Explanation: PKI Services is reading its configuration file to locate the value specified for `CreateInterval` in the **CertPolicy** section. The value specified has an incorrect syntax.

System action: PKI Services uses the default value of 3 minutes.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC014I **Bad RemoveCompletedReqs or RemoveInactiveReqs value in pkiserv.conf file**

Explanation: PKI Services is reading its configuration file to locate the value specified for either `RemoveCompletedReqs` or `RemoveInactiveReqs` in the **ObjectStore** section. The value specified has an incorrect syntax.

System action: Completed and inactive requests are not removed until the problem is corrected and PKI Services is restarted.

System programmer response: Correct the value and restart PKI Services. For more information, see

"(Optional) Steps for updating the configuration file" on page 60.

IKYC015I **Bad PostInterval value in pkiserv.conf file**

Explanation: PKI Services is reading its configuration file to locate the value specified for `PostInterval` in the **LDAP** section. The value specified has an incorrect syntax.

System action: PKI Services uses the default value of 5 minutes.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "Steps for tailoring the LDAP section of the configuration file" on page 98.

IKYC016I *action-being-performed* **returned** *nnnn* **in sub-function:** *error-code-description*

Explanation: PKI Services is processing a request and has encountered an internal error. The action being performed, the sub-function that returned the error, and the error code encountered are displayed. A description of the error is also displayed, if known.

System action: The request is not processed.

System programmer response: Report the error to the IBM support center.

IKYC017I **JNH_inquire_certreq_startdate (***object-id***) found neither certificate request nor response (***nnnn***):** *error-code-description*

Explanation: PKI Services is processing the start date in a request and has encountered an internal error. The request's ID and the error code encountered are displayed. A description of the error is also displayed, if known.

System action: The request is not processed.

System programmer response: Report the error to the IBM support center.

IKYC018I **{read | get_value} of** *certificate-or-CRL-extension-name* **returned** *nnnn***:** *error-code-description*

Explanation: PKI Services is processing a CRL or certificate extension field and has encountered an internal error. The field name and the error code encountered are displayed. A description of the error is also displayed, if known.

System action: The CRL or certificate is not processed.

System programmer response: Report the error to the IBM support center.

IKYC020I **Retrieving CA value failed** *nnnn***:** *error-code-description*

Explanation: PKI Services is processing a certificate extension field in preparation of posting the certificate to the LDAP directory. The processing has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

System action: The certificate is not posted to the LDAP directory.

System programmer response: Report the error to the IBM support center.

IKYC021I **CRL claims to have only User and only CA certs**

Explanation: PKI Services is processing a CRL extension field in preparation of posting the CRL to the LDAP directory. The processing has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

System action: The CRL is not posted to the LDAP directory.

System programmer response: Report the error to the IBM support center.

IKYC022I **Invalid type for object** *object-id* **in JNH_set_revreq_invaliditydate:** *error-code-description*

Explanation: PKI Services is processing a revocation request and has encountered an internal error. The revocation request's ID and the error code encountered are displayed. A description of the error is also displayed, if known.

System action: The revocation request is not processed.

System programmer response: Report the error to the IBM support center.

IKYC023I **Request index (***index-number***) greater than number of revocations (***nnnn***) in JNH_set_revreq_invaliditydate**

Explanation: PKI Services is processing a revocation request and has encountered an internal error.

System action: The revocation request is not processed.

System programmer response: Report the error to the IBM support center.

IKYC024I **Failed to schedule event in** *nnnn* **seconds, status =** *nnnn*: *error-code-description*

Explanation: PKI Services is attempting to schedule a timed event and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

System action: The event is not scheduled.

System programmer response: Report the error to the IBM support center.

IKYC025I **Failed to schedule event status =** *nnnn*: *error-code-description*

Explanation: PKI Services is attempting to schedule a timed event and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

System action: The event is not scheduled.

System programmer response: Report the error to the IBM support center.

IKYC026I **Deleting {inactive | completed} object** *object-id*. **Last changed at** *YYYY*/*MM*/*DD HH*:*MM*:*SS*

Explanation: PKI Services is attempting to purge the request database of inactive and completed requests. A request that has met the criteria for deletion has been found. The request's ID is displayed along with information on when it was last changed. This is an informational message only.

System action: The request is deleted. PKI Services continues normal processing.

IKYC027I **Removing certificate post request after** *nnnn* **unsuccessful attempts**

Explanation: PKI Services is attempting to purge the request database of unsuccessful LDAP post requests. A request that has met the criteria for deletion has been found. The number of unsuccessful attempts for this request is displayed. This is an informational message only.

System action: The request is deleted. PKI Services continues normal processing.

IKYC028I **Export for CertId** *certificate-id* **unsuccessful. Request is still pending approval or yet to be issued**

Explanation: A client has requested a certificate and is attempting to retrieve it. The retrieval was unsuccessful because the certificate is not yet available. The request either has yet to be approved by a PKI Services administrator or has been approved, but has not yet been issued by PKI Services. This is an informational message only.

System action: The state of the request is unchanged. PKI Services continues normal processing.

PKI Services administrator response: Use PKI Services administrative functions to query the request to check its state. If the request is still pending approval, determine whether the request should be approved or rejected and take action accordingly. For more information, see "Processing certificate requests" on page 338.

IKYC029I **Error: certificate request type is invalid for certificate creation**

Explanation: PKI Services is processing a certificate request and has encountered an internal error.

System action: The certificate request is not processed.

System programmer response: Report the error to the IBM support center.

IKYC030I **Error** *nnnn* **retrieving LDAP** *attribute-name* **attribute data from** *distinguished-name*: *error-code-description*

Explanation: PKI Services is trying to retrieve some attribute data from an entry in the LDAP directory and has encountered an error. The attribute name and distinguished name for which the retrieve was attempted and the error code encountered are displayed. If known, a description of the error is also displayed.

Note: If the error code is an LDAP return code, no error description is displayed.

System action: If the attribute being retrieved is `'MAIL'`, PKI Services is trying to retrieve the client's e-mail address to send the client a certificate expiration warning message or a renewed certificate. The warning message or the renewed certificate is not sent at this time but sending will be tried later.

System programmer response: Follow related instructions in *IBM Tivoli Directory Server Client Programming for z/OS*. Report errors to the IBM support center.

IKYC031I **Error** *nnnn* **invoking sendmail with email address** *email-address* **retrieved from LDAP entry** *distinguished-name*

Explanation: PKI Services tried to call the sendmail utility to notify a client that his or her certificate is expiring or has been renewed. The call was unsuccessful. This message displays the e-mail address and distinguished name from which it was retrieved and the error code encountered.

System action: The warning message or the renewed

certificate might or might not have been sent. If the e-mail address appears to be genuine, PKI Services retries sending later.

System programmer response: Diagnose the problem by consulting *z/OS Communications Server: IP Diagnosis Guide* and related documents. Report non-Communications Server errors to the IBM support center.

IKYC032I **Error** *nnnn* **invoking sendmail with email address** *email-address* **provided by** *distinguished-name*

Explanation: PKI Services is trying to notify a client that his or her certificate is ready, rejected, or recovered. Notification is accomplished by calling the sendmail utility. The call was unsuccessful. This message displays the e-mail address and the subject's distinguished name from the request. The error code encountered is also displayed.

System action: The message might or might not have been sent.

System programmer response: Diagnose the problem by consulting *z/OS Communications Server: IP Diagnosis Guide* and related documents. Report non-Communications Server errors to the IBM support center.

IKYC033I **Error** *nnnn* **accessing {ReadyMessageForm | RejectMessageForm | ExpiringMessageForm | AdminNotifyForm | RenewCertForm}** *form-value*

Explanation: PKI Services is attempting to notify a client that his or her certificate is ready, has been rejected, is expiring, is pending for approval, or has been renewed. The message to be sent is derived by reading the message form from a file or data set specified in the **General** section of the PKI Services configuration file. Either the file name was not specified correctly, or the file read was unsuccessful. The configuration file keyword in error is displayed. The name of the failing file or data set and the error code encountered are also displayed, if known. For ExpiringMessageForm an error code of zero with no file or data set name displayed indicates that the keyword is required but is missing from the PKI Services configuration file.

System action: The message is not sent. If this is the expiring warning message, notify pending message or the renewed certificate message, it will be attempted later.

System programmer response: Locate the failing `form-typeMessageForm` value in the `pkiserv.conf` file. Make sure that the value specifies the correct file or data set name and that the file or data set exists. If no errors are found, contact your RACF administrator to ensure that the user ID assigned to the PKI Services daemon has permission to open the file or data set for reading. After making a correction, restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC034I **Error issuing DEQ for resource** *resource-name***, return code was** *return-code*

Explanation: PKI Services background certificate processing has encountered an internal error trying to release control of a resource using the DEQ service. The resource name and return code from the DEQ macro are displayed.

System action: PKI Services processing continues. However, further processing of certificate requests might fail until PKI Services is stopped and restarted.

System programmer response: Stop and restart PKI Services. If the problem reoccurs, report the error to the IBM support center.

IKYC035I **Bad ExpireWarningTime value in pkiserv.conf file**

Explanation: PKI Services is reading its configuration file to locate the value specified for `ExpireWarningTime` in the **CertPolicy** section. The value specified has an incorrect syntax.

System action: PKI Services continues, but no expiration warning messages will be issued.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC036I **Bad MaxSuspendDuration value in pkiserv.conf file**

Explanation: PKI Services is reading its configuration file to locate the value specified for `MaxSuspendDuration` in the **CertPolicy** section. The value specified has an incorrect syntax.

System action: CRL processing continues. PKI Services will process as if the suspension grace period is unlimited.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC037I **Bad RemoveExpiredCerts or RemoveExpiredCertsAndKeys value in pkiserv.conf file**

Explanation: PKI Services is reading its configuration file to locate the value specified for `RemoveExpiredCerts`

or RemoveExpiredCertsAndKeys in the **ObjectStore** section. The value specified has an incorrect syntax.

System action: PKI Services continues, but expired certificates will not be removed from the issued certificate list (ICL).

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC038I Deleting expired certificate with serial number *certificate-serial-number*

Explanation: PKI Services is attempting to purge the issued certificate list (ICL) of expired certificates. A certificate that has met the criteria for deletion has been found. The certificate's serial number is displayed. This is an informational message only.

System action: The request is deleted. PKI Services continues normal processing.

IKYC039I Bad CRLDistName value in pkiserv.conf file

Explanation: PKI Services is initializing and is reading its configuration file to locate the value specified for CRLDistName in the **CertPolicy** section. The value specified does not contain all alphanumeric characters.

System action: PKI Services stops.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC040I Bad CRLDistURI*n* value in pkiserv.conf file: LdapServer*n*

Explanation: PKI Services is reading its configuration file to locate the LDAP server specified in the **LDAP** section of the PKI Services configuration file. The value is to be used to create the URI format for the CRLDistributionPoints extension for the LDAP protocol. The Server value specified by LdapServer*n* cannot be found or contains incorrect information.

System action: PKI Services continues, but the URI format for that protocol distribution point is not created.

System programmer response: Ensure that the CRLDistURI*n* value locates the correct LdapServer*n* or BindProfile*n* value in the **LDAP** section or the default FACILITY class profile, IRR.PROXY.DEFAULTS. For profile values, ensure the profile exists and contains the correct information. Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC041I Bad CRLDistURI*n* value in pkiserv.conf file, exceeds the number of LDAP servers: LdapServer*n*

Explanation: PKI Services is reading its configuration file to locate the LDAP server specified in the **LDAP** section of the PKI Services configuration file. The value is to be used to create the URI format for the CRLDistributionPoints extension for the LDAP protocol. The value *n* indicated in LdapServer*n* is greater than that specified by NumServers.

System action: PKI Services continues, but the URI format for that protocol distribution point is not created.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC042I Bad CRLDistURI*n* format in pkiserv.conf file: CRLDistURI*n*

Explanation: PKI Services is reading its configuration file to create the URI format for the CRLDistributionPoints extension. The value specified has incorrect syntax.

System action: PKI Services continues, but the URI format for that protocol distribution point is not created.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC043I Error *nnnn* **in creating HFS file** *file name* **to store distribution point CRL**

Explanation: PKI Services is trying to store the distribution point CRL in a file system file. An I/O error occurred during the processing.

System action: PKI Services continues, but the distribution point CRL is not created.

System programmer response: Fix the I/O error and wait for the next distribution point CRL to be created.

IKYC044I Bad OCSPType value in pkiserv.conf file: *value*

Explanation: PKI Services responder is reading its configuration file to check if the OCSP responder is enabled when receiving an OCSP request. The expected value is either 'none' or 'basic'. The value specified is not one of these.

System action: The responder is not enabled. The client will get a response back with status 'Try later'.

System programmer response: Correct the value and

restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC045I Unknown section or keyword in pkiserv.conf file: Section: [*section*], Keyword *keyword*

Explanation: PKI Services is reading its configuration file during initialization. One of the following conditions occurred:
- An unknown section name was found.
- An unknown keyword was found.
- A valid keyword was placed in the wrong section.
- A keyword was placed before any sections were defined.

System action: The keyword is ignored and PKI Services continues.

System programmer response: Correct the section name or the keyword and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC046I Incorrect encoding of SCEP {request | PKCS10}

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation contains incorrect ASN.1 encoding.

System action: PKI Services rejects the SCEP request.

System programmer response: Correct the SCEP client to produce a correctly encoded ASN.1 request. Report the error to the support center for the provider of your SCEP client.

IKYC047I Incorrect signature on SCEP {request | PKCS10}

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation or its enclosed PKCS #10 certificate request is incorrectly signed or was altered.

System action: PKI Services rejects the SCEP request.

System programmer response: Correct the SCEP client to produce a correctly signed request. Report the error to the support center for the provider of your SCEP client.

IKYC048I Unsupported algorithm in SCEP {request | PKCS10}

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation or its enclosed PKCS #10 certificate request contains an algorithm identifier that is unsupported by PKI Services.

System action: PKI Services rejects the SCEP request.

System programmer response: Correct the SCEP client to produce algorithms supported by PKI Services. Report the error to the support center for the provider of your SCEP client. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC049I SCEP signing certificate is {expired | revoked | not known to PKI Services}

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation contains a signing certificate that is expired, revoked, or unknown to PKI Services. The signing certificate cannot be used to authenticate the SCEP client.

System action: PKI Services rejects the SCEP request.

System programmer response: Determine if the SCEP client should request certificates from PKI Services. If so, correct the SCEP client to use a valid signing certificate previously issued by PKI Services. If none exists, reconfigure the SCEP client to remove any existing certificates and start the certificate request from the beginning using a new key-pair and a new self-signed certificate. Report the error to the support center for the provider of your SCEP client. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC050I Requested SCEP certificate is {expired | revoked}

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation requests a certificate that was previously issued by PKI Services but is currently revoked or expired, as indicated by the message displayed. The requested certificate cannot be used by the SCEP client.

System action: PKI Services rejects the SCEP request.

System programmer response: Determine if the SCEP client should request certificates from PKI Services. If so, reconfigure the SCEP client to remove any existing certificate request entries and start the certificate request from the beginning using a new key-pair and a new self-signed certificate. Report the error to the support center for the provider of your SCEP client. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC051I Error 0x*nnnn* decrypting SCEP request

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. PKI Services is attempting to recover the encrypted portion of the request using System SSL services but is unable to do so. The error code returned by System SSL is displayed in the message.

System action: PKI Services rejects the SCEP request.

System programmer response: Look up the **0x***nnnn* error code in *z/OS Cryptographic Services System SSL Programming*. Most likely, the SCEP client incorrectly encrypted the request, for example by encrypting it for a different host server. Report the error to the support center for the provider of your SCEP client. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC052I Unsupported SCEP message type: *nn*

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. The message type displayed in the message is either unrecognized or unsupported by PKI Services. PKI Services supports only message types `PKCSReq` (19) and `GetCertInitial` (20).

System action: PKI Services rejects the SCEP request.

System programmer response: Reconfigure the SCEP client to produce message types supported by PKI Services. Report the error to the support center for the provider of your SCEP client. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC053I SCEP key or name mismatch. SCEP transaction ID: *SCEP-transaction-ID*

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. While processing the displayed SCEP request transaction ID, PKI Services found one of the following inconsistencies:

- The signing certificate is self-signed but the public key within it does not match the key in the enclosed PKCS #10 request.
- The signing certificate's subject name does not match the subject name in the PKCS #10 request or the subject portion of the `IssuerAndSubject` field in the SCEP request.
- The signing certificate is not self-signed and not issued by PKI Services.
- The issuer portion of the `IssuerAndSubject` field in the SCEP request does not identify PKI Services.

System action: PKI Services rejects the SCEP request.

System programmer response: Correct the SCEP client to remove the inconsistency. Report the error to the support center for the provider of your SCEP client. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC054I Incorrect reuse of SCEP transaction ID by client name: *client-name*

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client named in the message. The request contains a SCEP transaction ID that was previously used in a different request. Transaction IDs must uniquely identify a transaction and cannot be reused.

System action: PKI Services rejects the SCEP request.

System programmer response: Determine if the SCEP client should request certificates from PKI Services. If so, reconfigure the SCEP client to remove any existing certificate request entries and start the certificate request from the beginning using a new key-pair and a new self-signed certificate. Report the error to the support center for the provider of your SCEP client. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC055I Bad PREREGISTER section in certificate template, nickname: *template-nickname*

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. The named certificate template used to preregister the SCEP client contains one of the following errors related to its PREREGISTER section:

- The PREREGISTER section does not exist.
- The PREREGISTER section is incorrectly terminated.
- A required directive is missing.
- A directive has an incorrect value.

System action: PKI Services rejects the SCEP request.

System programmer response: Edit the PKI Services certificate templates file (`pkiserv.tmpl`) and correct the error. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC056I Template missing from certificate templates file, nickname: *template-nickname*

Explanation: PKI Services cannot find the template nickname in the templates file in one of the following circumstances:

1. PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from an SCEP client, and the named certificate template used to preregister the SCEP client no longer exists.

2. During the automatic renewal process, the named certificate template used to generate the original certificate no longer exists.
3. PKI Services is trying to construct a quick link which contains the transaction ID and the certificate template name in the email message (Ready message) to notify a requestor that the certificate is ready to be picked up. The certificate template identified by the nickname indicated in the message cannot be found. If the indicated nickname is <NONICK>, the <NICKNAME=>directive was missing at the time the request was submitted; otherwise it had been there at the time of submission but was removed later.

System action: For case 1, PKI Services rejects the SCEP request. For case 2, the certificate cannot be automatically renewed. For case 3, the constructed quick link in the Ready message will not work and the certificate requestor will need to use the other link to pick up the certificate, which requires manual input of certificate type and transaction ID.

System programmer response: For case 1, edit the PKI Services certificate templates file (`pkiserv.tmpl`) and add the template. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246. For case 2, restore the nickname in the corresponding template and wait for the next automatic renewal processing. For case 3, make sure the template section has the <NICKNAME=> directive.

IKYC057I **SCEP request for client name** *client-name* **rejected by {SemiauthenticatedClient | UnauthenticatedClient | SubsequentRequest | RenewalRequest} directive.**

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. The certificate template used to preregister the named SCEP client contains a directive in the PREREGISTER section indicating that the request should be rejected. The directive is displayed in the message.

System action: PKI Services rejects the SCEP request. The rejected certificate request is recorded in the request database.

System programmer response: Determine if the SCEP client should request certificates from PKI Services. If so, reconfigure the SCEP client to remove any existing certificate request entries and start the certificate request from the beginning using a new key-pair and a new self-signed certificate. Create or recreate a PKI Services preregistration record as needed. Also ensure that the SCEP client is using a subject name that is consistent with the preregistration record. Make corrections to the SCEP client or delete and recreate the preregistration record as needed. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC058I **No preregistration record found for SCEP request, client name:** *client-name*

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from the named SCEP client. No preregistration record matching the SCEP client was found.

System action: PKI Services rejects the SCEP request.

System programmer response: Determine if the SCEP client should request certificates from PKI Services. If so, ensure that a preregistration record exists for the SCEP client and that the client is using a client name that matches the preregistration record. Make corrections to the SCEP client, or delete and recreate the preregistration record as needed. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC059I **No previous SCEP request found for SCEP GetCertInitial, client name:** *client-name*

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from the named SCEP client. The request is `GetCertInitial` (a polling operation to pick up a previously requested certificate). PKI Services is unable to locate the previous SCEP request.

System action: PKI Services rejects the SCEP request.

System programmer response: Determine if the SCEP client should request certificates from PKI Services. If so, reconfigure the SCEP client to remove any existing certificate request entries and start the certificate request from the beginning using a new a key-pair and new self-signed certificate. Make corrections to the SCEP client, or delete and recreate the preregistration record as needed. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246.

IKYC060I **CONSTANT section of certificate template, nickname:** *template-nickname* **contains an incorrect value:** *field-name*=*value*

Explanation: PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. The certificate template used to preregister the SCEP client is in error. A field found in the **<CONSTANT>** section of the template contains a missing or incorrect value. The nickname of the certificate template and the incorrect value are displayed in the message.

System action: PKI Services rejects the SCEP request.

System programmer response: Edit the certificate templates file and correct the error. For more information, see "Enabling Simple Certificate

Enrollment Protocol (SCEP)" on page 246.

IKYC061I Can not generate a certificate or a request with automatic renewal

Explanation: PKI Services is trying to process a GENCERT/REQCERT or GENRENEW/REQRENEW request. The <AUTORENEW=Y or N>tag is specified at the right position in its template, but NotifyEmail has no input.

System action: PKI Services does not process the request.

System programmer response: Enforce the input of NotifyEmail.

IKYC062I Can not enable or set up automatic renewal service for this request or certificate

Explanation: PKI Services is trying to enable or set up automatic certificate renewal for this request or certificate. But NotifyEmail is not in the request, or in the case of a certificate, one of the following has occurred:

- The certificate was created without NotifyEmail specified.
- The certificate has already been renewed.
- The expiration warning notification has already been sent for this certificate.

System action: PKI Services does not process the request.

System programmer response: None

IKYC063I The list of the pending approval requests may not be complete

Explanation: PKI Services is trying to construct a list of requests that are pending administrator approval. An error occurred when attempting to add an entry to the list.

System action: The request is not added to the pending request list.

System programmer response: None.

IKYC064I Error *nnnn* invoking sendmail to notify the administrator with email address *email-address* of requests awaiting approval

Explanation: PKI Services is trying to notify an administrator that one or more certificate requests are waiting for approval, and called the sendmail utility, which failed with the return code indicated in the error message. This error code can indicate an improperly defined e-mail address, a sendmail configuration error, or a network interruption.

System action: The message might or might not have been sent.

System programmer response: Diagnose the problem by consulting *z/OS Communications Server: IP Diagnosis Guide* and related documents. Report non-Communications Server errors to the IBM support center.

IKYC065I The value *value* specified for tag *tag* in template *template-nickname* is ignored

Explanation: PKI Services is processing a tag and its value in the templates file, and either the tag is not in the expected position or it does not contain an expected value. The message indicates problem tag, its value, and the template nickname.

System action: PKI Services ignores the tag.

System programmer response: None.

IKYC066I Can not find the template for GENRENEW or REQRENEW with nickname: *template-nickname*

Explanation: PKI Services is processing a GENRENEW or REQRENEW request and cannot find the template with the indicated nickname.

System action: PKI Services renews a certificate with no automatic renewal set up.

System programmer response: None

IKYC067I An attempt to check if the templates file has been updated failed

Explanation: PKI Services attempted to determine whether the templates file has been updated since it was last read, but encountered an error.

System action: PKI Services will open and read the templates file again to ensure that it uses the current values in the file.

System programmer response: None.

IKYC068I The templates file used may not be current

Explanation: If the Web application was implemented using REXX CGI execs, PKI Services determined that it needed to read the templates file, but was unable to either open or read the file.

If the Web application was implemented using Java server pages (JSPs), the XML template file `pkitmpl.xml` has been updated, but the file `pkixgen.tmpl` has not been regenerated.

System action: PKI Services uses the last updated version of the templates file.

System programmer response: If the Web application

was implemented using Java server pages (JSPs), use the TemplateTool utility to create an updated copy of `pkixgen.tmpl`.

IKYC069I **The renewed certificate with transaction id** *transaction-id* **can not be sent to email address** *email-address*

Explanation: PKI Services attempted to create or retrieve the note with the renewed certificate. An internal error occurred.

System action: None

System programmer response: Record the information from the message and manually send the transaction ID to the e-mail address to tell the user to pick up the renewed certificate.

IKYC070I **The exit program** {*exit-program*} **was cancelled for** {*pre* | *post*} **processing**

Explanation: PKI Services is executing the specified exit program. The program did not return within the time limit specified by the ExitTimeout keyword in the configuration file for the PKI Services daemon (`pkiserv.conf`).

System action: PKI Services postpones the automatic renewal processing until the next day.

System programmer response: Check the exit program to see if there are any problems. If you determine that it needs a longer time to run, adjust the ExitTimeout value in the `pkiserv.conf` file.

IKYC071I **Bad ExitTimeout value in pkiserv.conf file:** {*value*}

Explanation: PKI Services is reading its configuration file to locate the value specified for ExitTimeout in the General section. The value specified is incorrect.

System action: PKI Services continues. If the value specified is greater than 1 hour, then PKI Services uses 1 hour for the ExitTimeout value when the automatic renewal exit is invoked, otherwise it uses the default value of 30 seconds.

System programmer response: Correct the ExitTimeout value and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC072I **Unexpected return value from automatic renewal exit program** {*exit-program*}**:** {*unexpected-value*}

Explanation: PKI Services invoked the automatic certificate renewal preprocessing exit program specified, which returned an unexpected return value.

System action: PKI Services treats the unexpected return value as if a return value of 4 was returned. The automatic renewal of the certificate is deferred until the next time automatic renewal is performed.

System programmer response: Update the exit program to ensure that it returns only the documented expected values.

IKYC073I **A problem was encountered during automatic renewal** {*pre* | *post*} **exit processing in program** {*exit program*}

Explanation: PKI Services has encountered an error during automatic renewal exit processing. For example, it cannot invoke the specified program or it cannot retrieve the status from the exit program.

System action: PKI Services postpones the automatic renewal processing until the next day.

System programmer response: If diagnostic level messages are enabled for the CORE component, additional diagnostic level messages are written to the log to assist in identifying the exact cause of this failure. If diagnostic level messages were not enabled at the time of the error, enable diagnostic level messages for the CORE component and re-create the error. For information on enabling diagnostic level messages, see"_PKISERV_MSG_LEVEL subcomponents and message levels" on page 459.

IKYC074I **Error** *nnnn* **accessing the required** {**ReadyMessageForm** | **RecoverForm**} *form-value*

Explanation: PKI Services is attempting to process a request or a certificate that involves a key generated by PKI Services. The processing requires the set up of e-mail notification to notify a client of the transaction ID or the key ID so that the client can pick up the certificate through a provided link. The message to be sent is derived by reading the message form from a file or data set specified in the **General** section of the PKI Services configuration file. Either the file name was not specified correctly, or the file read was unsuccessful. The configuration file keyword in error is displayed. The name of the failing file or data set and the error code encountered are also displayed, if known.

System action: The request is not processed.

System programmer response: Locate the failing ReadyMessageForm or RecoverForm value in the `pkiserv.conf` file. Make sure that the value specifies the correct file or data set name and that the file or data set exists. If no errors are found, contact your RACF administrator to ensure that the user ID assigned to the PKI Services daemon has permission to open the file or data set for read. After making a correction, restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC075I Bad MaintRunTime value in pkiserv.conf file: {*value*}

Explanation: PKI Services is reading its configuration file, `pkiserv.conf`, to locate the value specified for `MaintRunTime` in the **General** section. The value specified is incorrect.

System action: PKI Services continues processing, using the default value of midnight, `00:00`.

System programmer response: Correct the value of `MaintRunTime` and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC076I Bad MaintRunDays value in pkiserv.conf file: {*value*}

Explanation: PKI Services is reading its configuration file, `pkiserv.conf`, to locate the value specified for `MaintRunDays` in the **General** section. The value specified is incorrect.

System action: PKI Services continues processing, using the default value of every day of the week, `0123456`.

System programmer response: Correct the value of `MaintRunDays` and restart PKI Services. For more information about the `MaintRunDays` parameter, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC077I Incorrect DBType value in pkiserv.conf file: *value*

Explanation: PKI Services is reading its configuration file, `pkiserv.conf`, to locate the value specified for `DBType` in the **ObjectStore** section. The value specified is incorrect. The value of `DBType` must be `VSAM` or `DB2`.

System action: PKI Services stops.

System programmer response: Correct the value of `DBType` and restart PKI Services. For more information about the `DBType` parameter, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC078I Incorrect DBSubsystem value in pkiserv.conf file: *value*

Explanation: PKI Services is reading its configuration file, `pkiserv.conf`, to locate the value specified for `DBSubsystem` in the **ObjectStore** section. The value specified is incorrect. The value of `DBSubsystem` must be a 4-character DB2 subsystem name.

System action: PKI Services stops.

System programmer response: Correct the value of `DBSubsystem` and restart PKI Services. For more information about the `DBSubsystem` parameter, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC079I Incorrect DBPackage value in pkiserv.conf file: *value*

Explanation: PKI Services is reading its configuration file, `pkiserv.conf`, to locate the value specified for `DBPackage` in the **ObjectStore** section. The value specified is incorrect. The value of `DBPackage` must be the name of a DB2 package.

System action: PKI Services stops.

System programmer response: Correct the value of `DBPackage` and restart PKI Services. For more information about the `DBPackage` parameter, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC080I Error *nnnn* in creating UNIX file *file_name* to post CRL

Explanation: PKI Services is trying to store a CRL in the z/OS UNIX file system file *file_name* for posting. An I/O error occurred during the processing.

System action: PKI Services continues, but the file is not created. The CRL is not posted.

Fix the I/O error and wait for the next CRL to be created.

IKYC081I Error *nnnn* in reading UNIX file *file_name* to post CRL

Explanation: PKI Services is trying to read a CRL from the file named *file_name* in the z/OS UNIX file system. An I/O error occurred during the processing.

System action: PKI Services continues, but does not post the CRL.

System programmer response: Fix the I/O error and wait for the next posting interval. For information about the posting interval, see the description of the `PostInterval` configuration parameter in Table 23 on page 98.

IKYC082I Bad URI CRL posting path and DP name value in pkiserv.conf file.

Explanation: PKI Services is reading its configuration file to locate the values specified for `CRLDistDirPath` and `CRLDistName` in order to save CRLs in the z/OS UNIX file system. The combination of the values specified is not valid because it results in a path length greater than the maximum length of 240. The path length includes a trailing "/" character on `CRLDistDirPath` if it does not include one.

System action: PKI Services continues, but CRLs are not saved in the file system.

System programmer response: Correct the values of CRLDistDirPath and CRLDistName in the pkiserv.conf file, and ensure that the resulting path length (including a trailing "/" character on CRLDistDirPath if it does not include one) is not greater than 240. Then restart PKI Services.

IKYC083I Bad Large CRL posting path and DP name value in pkiserv.conf file.

Explanation: PKI Services is reading its configuration file to locate the values specified for LargeCRLPostPath and CRLDistName to save CRLs in the z/OS UNIX file system for LDAP posting. The combination of values specified is not valid because it results in a path length greater than the maximum length of 240. The path length includes a trailing "/" character on LargeCRLPostPath if it does not include one.

System action: PKI Services continues, but large CRL posting cannot be enabled. CRLs whose size is 32KB or smaller will be stored in the object store for posting. CRLs larger than 32KB will not be posted. (32KB is an approximate value.)

System programmer response: Correct the values of LargeCRLPostPath and CRLDistName in the pkiserv.conf file. Ensure that the resulting path length (including a trailing "/" character on LargeCRLPostPath if it does not include one) is not greater than 240. Then restart PKI Services.

IKYC084I Large CRL posting object found while large CRL posting is not enabled.

Explanation: During LDAP posting, PKI Services has found a CRL posting object larger than 32KB, and EnableLargeCRLPosting has not been set to T in the pkiserv.conf file to enable posting of large CRLs. (32KB is an approximate value.)

System action: PKI Services deletes the large CRL posting object from the object store and continues.

System programmer response: None.

IKYC801I *nnnn* **bytes of unconsumed data transferring extensions to certificate template**

Explanation: PKI Services is processing a certificate renewal request and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

System action: The certificate renewal request is not processed.

System programmer response: Report the error to the IBM support center.

IKYC802I Error *nnnn* **{getting certificate-section from old certificate | setting certificate-section in certificate template | removing unnecessary extension from certificate template}:** *error-code-description*

Explanation: PKI Services is processing a certificate renewal request and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

System action: The certificate renewal request is not processed.

System programmer response: Report the error to the IBM support center.

IKYC901I Error *nnnn* **initializing** *sub-function-name***:** *error-code-description*

Explanation: PKI Services is initializing one of its sub-functions and has encountered an error. The sub-function name and error code encountered are displayed. A description of the error is also displayed, if known.

System action: PKI Services stops.

System programmer response: This message might accompany a message more specific to the sub-function that failed. Check the log for other error messages issued prior to this one, and diagnose accordingly. Restart PKI Services after making corrections. If you are unable to diagnose the error, report the error to the IBM support center.

Note: If the message indicates an LDAP error, the error code might be an OCSF error code. If you can't find the error code documented in *IBM Tivoli Directory Server Administration and Use for z/OS*, look in *z/OS Open Cryptographic Services Facility Application Programming*.

IKYC902I Error initializing the configuration file

Explanation: PKI Services is reading its configuration file to locate the object identifiers defined in the **OIDs** section. Either the section is missing, or a value has an incorrect syntax.

System action: PKI Services stops.

System programmer response: The OID values must be defined in dotted-decimal form, for example:

```
sha-1WithRSAEncryption=1.2.840.113549.1.1.5
```

Correct the configuration file, and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYC903I **Error** *nnnn* **adding CA certificate to ICL:** *error-code-description*

Explanation: PKI Services is initializing and is attempting to store its own certificate-authority certificate in the issued certificate list (ICL). The attempt was not successful. The error code encountered is displayed. A description of the error is also displayed, if known.

System action: PKI Services stops.

System programmer response: This message might accompany a more specific error message. Check the log for other error messages issued prior to this one and diagnose accordingly. Restart PKI Services after making corrections. If you are unable to diagnose the error, report the error to the IBM support center.

IKYD001I **Unable to open VSAM data set** *data-set-name*

Explanation: PKI Services is attempting to open one of the VSAM data sets specified in the **ObjectStore** section of the `pkiserv.conf` file or its default data set name. The open has failed. The data set name is displayed.

System action: PKI Services stops.

System programmer response: Locate the failing `DSN` value in the `pkiserv.conf` file. Make sure that the value specifies the correct VSAM data set name and that the data set has been created. If the data set name is not specified in the `pkiserv.conf` file, then PKI Services uses the default name for the data set. Make sure that this data set exists or add the appropriate `DSN` value to the `pkiserv.conf` file to specify the correct data set. If migrating from a previous release of PKI Services, make sure that the additional VSAM alternate index data sets have been created properly.

If no errors are found, contact your RACF administrator to ensure that the user ID assigned to the PKI Services daemon has permission to open the data set for update. Once corrected, restart PKI Services. For information about the values specified in the PKI Services configuration file (`pkiserv.conf`), including their defaults, see "(Optional) Steps for updating the configuration file" on page 60. See also "Steps for creating the VSAM object store and ICL data sets and indexes" on page 106.

IKYD002I **DB2 RRSAF failure: Command** *rrsaf_cmd* **failed with return code** *0xnnnn*, **reason code** *0xnnnn*

Explanation: A failure condition was detected when PKI Services issued Resource Recovery Services Attachment Facility (RRSAF) commands to the DB2 for z/OS database service. *rrsaf_cmd* is the RRSAF subcommand that detected the condition. This failure prevents PKI Services from using DB2 for z/OS as its storage mechanism.

System action: PKI Services stops.

System programmer response: Look up the return code and reason code given in this message in *DB2 for z/OS Codes* and perform any problem determination or problem resolution instructions.

IKYD003I **DB2 SQL failure: Instruction** *SQL_instruction* **failed with SQLCODE** *sqlcode*, **SQLSTATE** *sqlstate*

Explanation: A failure condition was detected when PKI Services issued Structured Query Language (SQL) instructions to the DB2 for z/OS database service. *SQL_instruction* is the SQL instruction that detected the condition.

System action: PKI Services stops.

System programmer response: Look up the SQLCODE and SQLSTATE values in this message in *DB2 for z/OS Codes* and perform any problem determination or problem resolution instructions.

IKYI001I **Request denied by installation exit. RC =** *nn*

Explanation: A user is requesting PKI Services. The `PKIServ` Web application called an installation-provided exit program. The exit program has determined that the request should be denied. The return code from the exit program is displayed in the message.

System action: The request in not performed.

User response: Contact your Web administrator.

Web administrator response: Determine why the exit program denied the request and correct the program if necessary.

IKYI002I **SAF Service IRRSPX00 Returned SAF RC =** *nn* **RACF RC =** *nn* **RACF RSN =** *nn* {*diagnostic-information*}

Explanation: A user is requesting PKI Services. The `PKIServ` Web application called the IRRSPX00 SAF callable service as requested. The service was unsuccessful. The diagnostic information that follows the message describes the problem in greater detail.

The text items listed here comprise all of the possible values for *diagnostic-information* in this message and in message IKYU002I.

1 Incorrect field name specified in CertPlist: <*field-name*>.
2 <*field-name*> has an incorrect value.
3 Required field <*field-name*> missing from the request.
4 Request denied, not authorized.

5 Certificate generation provider is not available [for CA domain <*CA-domain-name*>].
6 Certificate generation provider indicated the following error: <*provider-specific-error-msg*>.
7 Incorrect CertId PassPhrase specified.
8 Request has been rejected by the administrator.
9 Request is still pending approval or yet to be issued.
10 Incorrect certificate specified.
11 The certificate could not be {renewed | revoked} because of a state change.
12 Incorrect {CertId | Serial Number} specified.
13 The status of the {request | certificate} has been changed by another process.
14 {CertIds | SerialNums} has an incorrect length.
15 CertAnchor area missing.
16 CertAnchor area too small.
17 CertPlist has an incorrect length.
18 CertPlist DiagInfo field missing or has an incorrect length.
19 Conflicting field names specified in CertPlist : *field-name*.
20 Incorrect action specified.
21 Incorrect status criteria specified.
22 Incorrect transaction ID specified.
23 Incorrect reason specified.
24 Incorrect SerialNum specified.
25 SerialNums has an incorrect length.
26 Summary list or CertPlist area missing.
27 Summary list or CertPlist area too small.
28 A parameter list error has been detected.
29 An internal error has occurred during RACF processing.
30 Unable to establish recovery environment.
31 Function code specified is not defined.
32 Parameter list version specified is not supported.
33 RACF not installed.
34 Certificate generation provider internal error.
35 Unexpected error.
36 Incorrect value specified for CA domain.
37 Client already preregistered.
38 The ReadyMessageForm or the RecoverForm is not set up correctly. The ReadyMessageForm is required to request a certificate. The RecoverForm is required to recover a certificate whose keys were generated by PKI Services.
39 The e-mail containing the transaction ID link to pick up the certificate was not sent successfully. The requestor needs to contact the administrator.
40 The e-mail containing the key ID link to recover the certificate was not sent successfully. The recovery process stops.
41 The requestor's e-mail address for the certificate could not be modified because the key is not generated by PKI Services.
42 The certificate could not be renewed because the requestor's e-mail address has been changed.
43 The certificate could not be deleted from the token data set (TKDS) although it was deleted from the issued certificate list (ICL).

System action: The request is not performed.

User response: Correct the problem if applicable. If you cannot correct the problem, contact your Web administrator.

For problem 9, try to retrieve your certificate again at a later time. The amount of time you need to wait depends on your PKI Services operating procedures and settings. If you continue to get this message, contact your PKI Services administrator.

Web administrator response: Problems 1, 2, and 3 probably indicate an error with the certificate template. Change the certificate template definition in the `pkiserv.tmpl` file to correct the error.

Problem 4 indicates the user ID assigned to the unit of work calling the `IRRSPX00` callable service is not RACF-authorized to perform the request. Determine if the user should have access. If so, use RACF commands to permit the user ID to the required resources.

Problem 5 indicates the PKI Services daemon process has not been started. If PKI Services is configured for multiple-CA mode then the CA domain name is displayed as part of the diagnostic information. Start the correct instance of PKI Services; then retry the request.

For problems 6-13, 22, and 24, or for more information on any of the preceding problems, see earlier chapters in this document and *z/OS Security Server RACF Callable Services*.

For problems 14-21, 23, and 25-35, report the error to the IBM support center.

For problem 36, PKI Services is configured for multiple-CA mode, but the CA domain name as found in the URL contains characters that cannot be used as a CA domain name. Correct the value in the URL; then retry the request.

Problems 38, 39 and 40 probably indicate an error with the value specified in the ReadyMessageForm or the RecoverForm in the configuration file. Change the value in the `pkiserv.conf` file to correct the error.

For problem 43, you need to remove the orphaned TKDS objects yourself; for example, by using ICSF panels.

PKI Services administrator response: For problem 9, locate the pending certificate request using the PKI Services administration web pages, and approve or reject the request.

IKYI003I **PKI Services CGI error in cgi-program-name:** *diagnostic-error-information*

Explanation: A user is requesting PKI Services. The `PKIServ` Web application CGI program processing the request detected a problem. The name of the CGI program and additional diagnostic information is displayed in the message.

System action: The request in not performed.

User response: Contact your Web administrator.

Web administrator response: Locate the CGI program mentioned in the message. (Its default installation location is in a subdirectory under `/usr/lpp/pkiserv/PKIServ`.) Examine the CGI program's source code to determine the spot where it is failing and why. In most cases, the problem is caused by an error in the PKI Services template file (usually located in `/etc/pkiserv/pkiserv.tmpl`). Correct the problem and retry the request. For more information, see Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs," on page 123 and Chapter 12, "Customizing the administration Web pages if you use REXX CGI execs," on page 175.

IKYI004I Installation exit failed. RC = *nn*

Explanation: A user is requesting PKI Services. The `PKIServ` Web application called an installation-provided exit program. The exit program either terminated abnormally or returned an unsupported return code value. The return code from the invocation of the exit program is displayed in the message.

System action: The request in not performed.

User response: Contact your Web administrator.

Web administrator response: Determine why the exit program has failed and correct the program as necessary.

IKYI005I Invalid return type specified for certificate retrieval

Explanation: A user is retrieving a certificate from the PKIServ Web application. The result returned does not match the specified certificate type.

System action: The certificate is not returned through the Web page.

User response: Specify the correct return type for certificate retrieval and try again. Make sure that the transaction ID specified corresponds to PKI/SAF Browser Certificate, PKI/SAF Server Certificate or PKI Key Certificate,

---

IKYI006I PKI Services JSP error in *jsp-filename*:
diagnostic-information

Explanation: A user made a request through the PKI Services Web application. The PKIServ Web application detected a problem. The name of the JSP file and additional diagnostic information is displayed in the message.

System action: The request is not performed.

User response: Contact your Web administrator.

Web administrator response: If you can't determine the error and have modified the failing JSP file, try temporarily replacing the modified JSP with a copy of the original JSP file that shipped with PKI Services to determine whether your change is causing the problem.

IKYI007I PKI Services web pages have not been configured for the client authentication needed for this function.

Explanation: To renew or revoke a browser certificate, the Websphere application server must be set up for client authentication. This step wasn't completed or client authentication is not working properly.

System action: The request is not performed.

User response: Contact your Web administrator.

Web administrator response: Set up the Websphere application server for client authentication. For instructions, see "Allowing Websphere users to renew and revoke browser certificates" on page 189.

IKYK001I Unexpected PKCS#11 *function-name* **return code 0x***nnnn***. The request is not processed.**

Explanation: PKI Services is calling the PKCS #11 function *function-name* to perform an action on a token. *function-name* returns an unexpected hexadecimal return code, *nnnn*.

System action: The request is not processed

System programmer response: For more information, see *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*.

IKYK002I PKCS#11 token unavailable for *function-name* **with return code 0x***nnnn* **. The request is not processed.**

Explanation: PKI Services is calling the PKCS #11 function *function-name* to perform an action on a token. *function-name* returns a hexadecimal return code, *nnnn*. The possible return codes and reasons are:

X'00e0' ICSF is not active or is not configured for PKCS #11 services.

X'00e1' The PKI Services daemon has insufficient authority to access the token.

System action: PKI Services does not process the request.

System programmer response: Ensure that ICSF is properly configured and operational. If the return code is X'00e1', look for the ICH408I message that is issued for insufficient authority to the CRYPTOZ class resource and give the PKI Services daemon the required access. For more information, see *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*.

IKYK003I **{Certificate | key | PKCS12 package} with KEYID** *keyid* **is not deleted from TKDS object {Certificate object | Public key object | Private key object | Data object}.**

Explanation: During certificate deletion processing, PKI Services is calling a PKCS #11 function to delete a certificate and its related objects from the token data set (TKDS) . The deletion is completed in the issued certificate list (ICL) but not in the TKDS.

System action: The certificate is deleted from the ICL but the certificate and its related objects are not deleted from the TKDS.

System programmer response: Look at message IKYK001I to find out the return code. For information about deleting objects from the TKDS, see the PKCS #11 Cryptographic Token Interface Standard at ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf.

IKYL001I **Error** *nnnn* **{importing | converting} LDAP username** *user's-distinguished-name*: *error-code-description*

Explanation: PKI Services is reading its configuration file to locate one of the values specified for `AuthName` in the **LDAP** section. The value specified has a syntax error. The incorrect value is displayed. A description of the error is also displayed, if known.

System action: PKI Services binds to the LDAP directory anonymously and continues processing. When PKI Services attempts to post certificates and CRLs to this directory, it might fail due to insufficient access. Look for message IKYC007I to determine this is happening. (RC = `LDAP_INSUFFICIENT_ACCESS`)

System programmer response: Locate the incorrect `AuthName` value in the `pkiserv.conf` file and correct it. The value must be specified as an LDAP distinguished name, for example, `CN=root,O=IBM`. Note: The OID qualifiers must be specified in uppercase and there cannot be any spaces surrounding the equal signs or commas separating the attribute value assertions (AVAs). Make corrections as needed, then stop and restart PKI Services. For more information, see "Steps for tailoring the LDAP section of the configuration file" on page 98.

IKYL002I **LDAP bind to** *LDAP-server-domain-name:port* **failed, status = nnnn:** *status-code-description*

Explanation: PKI Services is attempting to bind to one of the LDAP servers specified in the **LDAP** section of the `pkiserv.conf` file. The bind has failed. The failing server name is displayed. A description of the error is also displayed, if known. Note: If the error code is an LDAP return code, no error description will be displayed.

System action: PKI Services attempts to bind to your other LDAP servers, if any. If PKI Services is unable to bind to any LDAP servers, the LDAP posting of certificates and CRLs is temporarily suspended. PKI Services attempts to bind again during the next posting interval. All post requests will remain in the request database to be attempted later, subject to being deleted after one week of unsuccessful attempts.

System programmer response: Diagnose the problem indicated by the return code. For `LDAP_SERVER_DOWN`, ensure that your LDAP server is running. If so, you might have specified the server name incorrectly in the PKI Services configuration file. Locate the failing `Server` value in the `pkiserv.conf` file. Correct the value if it does not specify the correct LDAP server domain name and port, then stop and restart PKI Services. For all other LDAP errors, follow the instructions in *IBM Tivoli Directory Server Client Programming for z/OS*. Report errors to the IBM support center. If message IKYC009I is also displayed, report that information as well. For more information, see "Steps for tailoring the LDAP section of the configuration file" on page 98.

IKYL003I **Incorrect value specified for LDAPBIND or FACILITY Class profile** *profile-name*

Explanation: PKI Services LDAP bind processing is trying to retrieve its LDAP bind information in preparation for communicating with the LDAP server. The bind information is contained in either an LDAPBIND class profile or the IRR.PROXY.DEFAULTS profile in the FACILITY class. Either the profile does not exist or some of the information is missing or incorrect. The name of the profile in question is displayed.

System action: PKI Services attempts to bind to your other LDAP servers, if any. If PKI Services is unable to bind to any LDAP servers, the LDAP posting of certificates and CRLs is temporarily suspended. PKI Services attempts to bind again during the next posting interval. All post requests remain in the request database to be attempted later, subject to being deleted after one week of unsuccessful attempts.

System programmer response: Locate the profile name in the PKI Services configuration file and correct it if needed. If you make corrections, stop and restart PKI Services. If the profile name is already correct, contact your RACF administrator. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

RACF administrator response: Display the PROXY segment of the profile using the RLIST TSO command. Check the LDAPHOST for accuracy, and correct it if needed. If non-anonymous access is required, do the same for the BINDDN and BINDPW.

Note: The BINDPW value is not displayed. Respecify it to ensure that it is accurate.

To alter the fields, use the RALTER TSO command. If

the profile does not exist, create it using the RDEFINE TSO command. For more information, see *z/OS Security Server RACF Command Language Reference*.

IKYL004I **Bad LDAP Server value** *server-value* **in pkiserv.conf file**

Explanation: PKI Services LDAP bind processing is trying to retrieve its LDAP bind information in preparation for communicating with the LDAP server. (The *Server1*, *Server2*, and so forth keywords in the **LDAP** section of the PKI Services configuration file specify the server host name information.) The host name has been specified incorrectly. Its value is displayed.

System action: PKI Services attempts to bind to your other LDAP servers, if any. If PKI Services is unable to bind to any LDAP servers, the LDAP posting of certificates and CRLs is temporarily suspended. PKI Services attempts to bind again during the next posting interval. All post requests remain in the request database to be attempted later, subject to being deleted after one week of unsuccessful attempts.

System programmer response: Locate the server name in the PKI Services configuration file, and correct it if needed. If you make corrections, stop and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYO001I **Error** *nnnn* **{setting | getting}** *certificate-field* **{in certificate | from template}:** *error-code-description*

Explanation: PKI Services is processing a certificate request field and has encountered an internal error. The field name and the error code encountered are displayed. A description of the error is also displayed, if known.

System action: The certificate request is not processed.

System programmer response: Report the error to the IBM support center.

IKYO002I *nnnn* **bytes of unconsumed data transferring** *certificate-field* **to certificate**

Explanation: PKI Services is processing a certificate request field and has found that the field is larger than it should be. This is an internal error. The field name and the number of extra bytes are displayed.

System action: The certificate request is not processed.

System programmer response: Report the error to the IBM support center.

IKYO003I **The certificate request failed validity checks. Status is** *nnnn***:** *status-code-description*

Explanation: PKI Services is processing a certificate request field and has encountered an internal error. The field name and the status (error) code encountered are displayed. A description of the error is also displayed, if known.

System action: The certificate request is not processed.

System programmer response: Report the error to the IBM support center.

IKYO004I *action-being-performed* **returned** *nnnn***:** *error-code-description*

Explanation: PKI Services is processing a request and has encountered an internal error. The action being performed and the error code encountered are displayed. A description of the error is also displayed, if known.

System action: The request is not processed.

System programmer response: Report the error to the IBM support center.

IKYO005I **PKI CA certificate last used serial number was regressed.**

Explanation: During certificate fulfillment, PKI Services has detected that the CA certificate's last serial number has regressed to a serial number below the next available PKI Services serial number.

System action: PKI Services calls the `R_DataLib` callable service `IncSerialNum` function to increment the regressed serial number back to the next available PKI Services serial number or to one that was never used by PKI Services. If this is an asynchronous certificate request, the request is retried with a new serial number.

IKYP001E **ICSF UNAVAILABLE. CERTIFICATE PROCESSING SUSPENDED**

Explanation: PKI Services background certificate processing is attempting to create a digital signature. ICSF manages the private key required for digital signing but it is not available for any of the following possible reasons:

- ICSF is inactive or incorrectly configured.
- The user ID of the PKI Services daemon has insufficient authority to use the ICSF private key.
- A system administrator inadvertently deleted the ICSF signing certificate and its private key.

After the ICSF problem has been corrected, PKI Services must be stopped and restarted.

Routing code: 2

Descriptor code: 6

System action: PKI Services background certificate processing is suspended. No certificates or CRLs are issued until the problem is corrected and PKI Services is stopped and restarted. However, certificate request management functions are still available through the `R_PKIServ` callable service and the PKI Services Web pages.

System programmer response: Ensure that ICSF and the PCI cryptographic coprocessor (if applicable) are properly configured and operational. Follow the documentation for any issued message with the **CSF** prefix.

If ICH408I messages are issued for insufficient authority to CSFKEYS or CSFSERV class resources, then the user ID of the PKI Services daemon has insufficient authority to use the key. Give the user ID the required access to the specified resource.

To determine if the key you are using requires the PCI cryptographic coprocessor, see Chapter 20, "RACF administration for PKI Services," on page 403.

For more information, see "Installing and configuring ICSF (optional)" on page 23, *z/OS Cryptographic Services ICSF System Programmer's Guide*, and *z/OS Cryptographic Services ICSF Administrator's Guide*.

If you make changes to ICSF to correct the problem, stop and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYP002I PKI SERVICES INITIALIZATION COMPLETE

Explanation: PKI Services has just been started and has finished initializing.

System action: PKI Services processing continues.

Routing code: 2

Descriptor code: 6

IKYP003I PKI SERVICES SHUTDOWN REQUESTED

Explanation: An operator command was issued to stop PKI Services.

Routing code: 2

Descriptor code: 5

System action: PKI Services stops.

IKYP004I LOG OPTION PROCESSED: *log-option*

Explanation: A MODIFY operator command was issued to alter the current log setting for PKI Services.

System action: The log setting for PKI Services is changed as requested.

Routing code: 2

Descriptor code: 5

IKYP005I INCORRECT LOG OPTION SPECIFIED

Explanation: A MODIFY operator command was issued to alter the current log setting for PKI Services. The log parameter syntax or value is incorrect.

System action: The MODIFY command is not processed. The log setting for PKI Services is unchanged.

System programmer response: Execute the MODIFY command specifying a correct log parameter. For more information, see "Changing logging options" on page 459.

Routing code: 2

Descriptor code: 5

IKYP006I UNRECOGNIZED PKI SERVICES COMMAND: SPECIFY LOG, DISPLAY, OR STOP

Explanation: A MODIFY operator command was issued for PKI Services. The command specified is not a supported PKI Services command.

System action: The MODIFY command is not processed. PKI Services continues processing unchanged.

System programmer response: Execute the MODIFY command specifying a supported PKI Services command. For more information, see "Stopping the PKI Services daemon" on page 119 and "Changing logging options" on page 459.

Routing code: 2

Descriptor code: 5

IKYP007E INSUFFICIENT STORAGE AVAILABLE

Explanation: PKI Services is attempting to allocate storage for processing a MODIFY operator command, but is unsuccessful because of a storage shortage.

System action: The console command is not processed. However, PKI Services might continue processing normally.

Operator response: Report the problem to your system programmer. After the problem is corrected, you can execute the command again.

System programmer response: Increase the region size for the PKI Services started procedure. Stop and restart PKI Services. For more information, see "Steps for starting the PKI Services daemon" on page 117 and "Stopping the PKI Services daemon" on page 119.

Routing code: 2

Descriptor code: 5

IKYP008E DIRECTORY POST UNSUCCESSFUL. LDAP DATA LIBRARY MODULE RC = *nnnn*

Explanation: PKI Services background certificate processing is attempting to post information (such as a certificate or CRL) to a directory. The post was unsuccessful. The OCSF Data Library Module (LDAPDL) return code is displayed in the message.

System action: The information is not posted at this time. The post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database.

System programmer response: Determine the cause of the failure from the return code displayed and take appropriate action. These return codes are documented in *z/OS Open Cryptographic Services Facility Application Programming*. If the error is `LDAPDL_NO_SUCH_OBJECT`, the LDAP entry could not be created because the required suffix does not exist. Check the PKI Services log to determine the entry that could not be created, as indicated on messages IKYC005I and IKYC008I. If the entry should be posted to LDAP, you need to define the suffix in the LDAP server configuration file and recycle the LDAP server. For more information, see "Steps for installing and configuring LDAP" on page 21 and *IBM Tivoli Directory Server Administration and Use for z/OS*.

If you want PKI Services to bypass LDAP posting for certificates with missing suffixes, set `RetryMissingSuffix=F` in the PKI Services `pkiserv.conf` configuration file. Then, stop and restart the PKI Services daemon. For more information, see "Steps for tailoring the LDAP section of the configuration file" on page 98.

Routing code: 2

Descriptor code: 6

IKYP009I PKI SERVICES IS STARTING, FMID *product-fmid*

Explanation: The START operator command was issued to start PKI Services. The START command could have been entered directly at the operator's console or indirectly through a COMMND*xx* PARMLIB member.

System action: PKI Services initialization proceeds.

Routing code: 2

Descriptor code: 6

IKYP010I THE CONFIGURATION FILE NAME EXCEEDS THE MAXIMUM LENGTH OF *nnnn* CHARACTERS

Explanation: The PKI Services daemon process is starting. Initialization processing is reading the `_PKISERV_CONFIG_PATH` environment variable. The value specified is too long.

System action: PKI Services stops.

System programmer response: Determine the location of your PKI Services environment variables file, and correct the value specified for `_PKISERV_CONFIG_PATH`. Then, restart PKI Services.

Routing code: 2

Descriptor code: 6

IKYP011I PKI SERVICES ADDRESS SPACE COULD NOT BE MADE NON-SWAPPABLE: ERROR *nnnn*

Explanation: The PKI Services daemon process is starting. Initialization processing is attempting to make the PKI Services address space non-swappable. The attempt was unsuccessful. The SYSEVENT TRANSWAP error code is displayed.

System action: PKI Services stops.

System programmer response: Look up the error code for SYSEVENT TRANSWAP in *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO* to determine what to do. Then, restart PKI Services.

Routing code: 2

Descriptor code: 6

IKYP012I SYSTEM FUNCTION *function-name* DETECTED ERROR — *error-string*

Explanation: PKI Services processing received an error when calling a system service. The service name and error message are displayed.

System action: PKI Services stops.

System programmer response: See documentation related to the service that failed. Make any necessary corrections. Then, restart PKI Services.

Routing code: 2

Descriptor code: 6

IKYP013I PKI SERVICES DETECTED AN ERROR DURING INITIALIZATION: ERROR *nnnn*, REASON 0x*nnnn*

Explanation: PKI Services is starting. Initialization processing is attempting to set up the Program Call (PC) interface. The attempt was unsuccessful. The error and reason codes are displayed.

System action: PKI Services stops.

System programmer response: Determine the failing service by examining the error code. The values are as follows:

Note: This message is also issued after various C function calls. In these cases, ERROR is the value of `errno`, and REASON is the value of `__errno2()`.

1 The PKI Services daemon (IKYPKID) is not APF-authorized.

3 Unable to establish recovery. The reason code displayed is the ESTAEX macro return code.

5 Unable to create a PC linkage table index. The reason code displayed is the LXRES macro return code.

6 Unable to create a PC entry table. The reason code displayed is the ETCRE macro return code.

7 Unable to connect the PC entry table to the linkage table. The reason code displayed is the ETCON macro return code.

8, 10, or 11
Unable to create a name token entry. The reason code displayed is the IEANTCR callable service return code.

For error code **1**, make the IKYPKID load module in SYS1.LINKLIB APF-authorized. For all other error codes, see the documentation associated with the MVS service that failed. Make corrections as necessary. Then, restart PKI Services.

Routing code: 2

Descriptor code: 6

IKYP014I PKI Services detected an error during termination: Error *nnnn*, Reason *nnnn*

Explanation: PKI Services is stopping. Termination processing is attempting to free resources allocated. The attempt was unsuccessful. The error and reason codes are displayed.

System action: PKI Services termination processing continues.

System programmer response: PKI Services should end normally. If so, no action is needed. However, you might want to diagnose the problem. Determine the failing service by examining the error code:

16 Unable to establish recovery. The reason code displayed is the ESTAEX macro return code.

See associated documentation for the MVS service that failed. Make corrections as necessary.

IKYP015I A PKI Services program call request failed: Error *nnnn*

Explanation: PKI Services is processing a PC request. The PC request was cancelled before PKI Services completed processing on it. The error code that was posted at the time of the cancel is displayed.

System action: PKI Services processing continues.

System programmer response: If the error code is 8, no action is required. This is an informational message only. For all other error codes, contact your IBM support center.

IKYP016I THE PKI SERVICES RUNTIME ENVIRONMENT COULD NOT BE INITIALIZED

Explanation: The PKI Services daemon process is starting. Initialization processing is trying to initialize the PKI Services runtime environment within the daemon address space. The attempt was unsuccessful.

System action: PKI Services stops.

System programmer response: Look for other PKI Services log messages related to this error. For more information, see Chapter 23, "Using information from the PKI Services logs," on page 455.

Routing code: 2

Descriptor code: 6

IKYP017I PKI SERVICES IS ALREADY RUNNING

Explanation: An attempt was made to start more than one instance of the PKI Services daemon.

System action: The first instance of PKI Services continues processing. The second instance stops.

Routing code: 2

Descriptor code: 6

IKYP018I PKI Services initialization failed because the program is not APF authorized

Explanation: PKI Services is starting. Initialization processing is attempting to initialize the PKI Services runtime environment within the daemon address space. The attempt was unsuccessful because the PKI Services daemon (IKYPKID) is not APF-authorized.

System action: PKI Services stops.

Routing code: 2

Descriptor code: 6

IKYP019I PKI Services dump created.

Explanation: PKI Services encountered a severe error during processing and has dumped the process (using the CEE3DMP callable service).

System action: PKI Services processing ends.

Operator response: Contact your system programmer.

System programmer response: Examine the dump to determine the error. Contact the IBM support center if needed. After the error has been corrected, restart PKI Services. For more information, see "Steps for starting the PKI Services daemon" on page 117 and "Stopping the PKI Services daemon" on page 119.

IKYP020I PKI SERVICES RESTART REGISTRATION COMPLETE ON *system-name*

Explanation: PKI Services is starting. Initialization processing has successfully registered PKI Services for automatic restart (ARM).

System action: PKI Services processing continues.

Routing code: 2

Descriptor code: 6

IKYP021I PKI SERVICES RESTARTING ON *system-name*

Explanation: The PKI Services daemon stopped and is being restarted by the Automatic Restart Manager (ARM). The restart was successful.

System action: PKI Services processing continues.

Routing code: 2

Descriptor code: 6

IKYP022I UNABLE TO REGISTER PKI SERVICES FOR RESTART: ERROR *nnnn*, **REASON 0x***nnnn*

Explanation: PKI Services is starting. Initialization processing is attempting to register PKI Services for automatic restart (ARM), using the IXCARM macro service. The attempt was unsuccessful. The IXCARM return and reason codes are displayed. Note: The reason code is displayed in hexadecimal.

System action: PKI Services initialization continues without automatic restart capability.

System programmer response: Determine and correct the problem with IXCARM as indicated by the error codes displayed. Then, stop and restart PKI Services if automatic restart capability is desired. For more information, see *z/OS MVS Programming: Sysplex Services Reference*.

Routing code: 2

Descriptor code: 6

IKYP023I PKI Services failed to format the display message

Explanation: A MODIFY operator command was issued to display the current settings for PKI Services. Formatting of the display information failed.

System action: The settings are not displayed. PKI Services processing continues.

System programmer response: Report the error to the IBM support center.

IKYP024I PKI SERVICES DUMPING FOR ABEND *abend-code* **RC** *nnnn*

Explanation: PKI Services has incurred an abend. The abend and reason codes are displayed.

System action: PKI Services stops.

System programmer response: Use IPCS to examine the dump and diagnose the problem. Contact the IBM support center if necessary. Restart PKI Services after the error has been corrected.

Routing code: 2

Descriptor code: 6

IKYP025I PKI SERVICES SETTINGS:

Explanation: A MODIFY operator command was issued to display the current settings for PKI Services.

System action: The settings are displayed.

```
IKYP025I PKI SERVICES SETTINGS:
   CA DOMAIN NAME: {CA-domain-name-for-this-PKI-daemon}
   SUBCOMPONENT             MESSAGE LEVEL
     LDAP                   {current-message-level}
     SAF                    {current-message-level}
     DB                     {current-message-level}
     CORE                   {current-message-level}
     PKID                   {current-message-level}
     POLICY                 {current-message-level}
     TPOLICY                {current-message-level}
   MESSAGE LOGGING SETTING: {STDERR_LOGGING | STDOUT_LOGGING}
   CONFIGURATION FILE IN USE:
 {full-UNIX-pathname-of-configuration-file-being-used}
   TEMPLATE FILE IN USE:
 {full-UNIX-pathname-of-template-file-being-used}
   CA CERTIFICATE FINGERPRINTS:
      SHA1:   {EBCDIC-representation-of-sha1-hash}
      MD5:    {EBCDIC-representation-of-md5-hash}
      SHA256: {EBCDIC-representation-of-sha256-hash}
      SHA512: {EBCDIC-representation-of-sha512-hash}
```

Restrictions:

- The `CA DOMAIN NAME:` line is suppressed if the daemon is running in single CA mode.
- For long CA domain names, the *CA-domain-name-for-this-PKI-daemon* value is truncated to 50 characters.

The possible *current-message-level* values for each subcomponent are:

- SEVERE MESSAGES ONLY
- ERROR MESSAGES AND HIGHER
- WARNING MESSAGES AND HIGHER
- INFORMATIONAL MESSAGES AND HIGHER
- DIAGNOSTIC MESSAGES AND HIGHER
- VERBOSE DIAGNOSTIC MESSAGES AND HIGHER

Operator response: You can change the subcomponent

message levels with the MODIFY operator command if desired. For more information, see "Changing logging options" on page 459.

Routing code: 2

Descriptor code: 5

IKYP026E PKI SERVICES {CA | RA} CERTIFICATE EXPIRES ON *yyyy/mm/dd*

Explanation: The certificate that contains the PKI Services CA or RA public key expires on the date shown.

System action: If the certificate has not yet expired, processing continues as normal. After the CA certificate expires, certificates issued by PKI Services might be unusable depending on their usage.

System programmer response: You should renew the certificate before it expires. If your security product is RACF, your certificate is contained in a RACF profile established when you first configured PKI Services. Follow RACF documentation on how to renew a certificate. This is done using either the RACDCERT TSO command or RACF ISPF panels. For more information, see "Renewing your PKI Services CA and RA certificates" on page 410 and *z/OS Security Server RACF Security Administrator's Guide*.

Routing code: 2

Descriptor code: 6

IKYP027E ERROR ACCESSING PKI SERVICES CA CERTIFICATE

Explanation: The PKI Services CA certificate is stored in the security product's database. PKI Services background certificate processing is attempting to access the certificate using the `R_datalib` SAF callable service. The attempt failed. Message IKYS015I should also appear in the PKI Services log.

System action: PKI Services background certificate processing is suspended. No certificates are issued until the problem is corrected. However, certificate request management functions are still available through the `R_PKIServ` callable service and the PKI Services Web pages.

System programmer response: You need to determine why the access failed. Look up the `R_datalib` return code displayed on message IKYS015I in *z/OS Security Server RACF Callable Services*. If your security product is RACF, your certificate is contained in a RACF profile established when you first configured PKI Services. That certificate must be connected as the default certificate to the key ring identified by the KeyRing keyword in the PKI Services configuration file. (The default location for this file is `/etc/pkiserv/pkiserv.conf`.) If you have only renewed your certificate and have not recycled PKI Services, stopping and restarting the PKI Services daemon might solve the problem. If not, use the RACF RACDCERT LIST and LISTRING commands to determine if the correct certificate is connected to the key ring. Also, use the RACF RLIST command to check that the PKI Services daemon user ID has proper authority to access the profile. Make any required changes. Then, stop and restart PKI Services. For more information, see Chapter 20, "RACF administration for PKI Services," on page 403 and *z/OS Security Server RACF Security Administrator's Guide*.

Routing code: 2

Descriptor code: 6

IKYP028E PKI SERVICES DISTINGUISHED NAME OR KEY CHANGE ERROR

Explanation: PKI Services is starting. Initialization processing has retrieved the PKI Services signing certificate from the key ring assigned to PKI Services. The certificate is incompatible with certificate processing that has previously transpired. The subject's distinguished name or the public key or both differ from the previous values used. The subject's distinguished name cannot be changed without reconfiguring PKI Services. The public key can be changed, but only if the key rollover process is performed.

System action: PKI Services stops.

System programmer response: Determine if PKI Services is processing the correct certificate. If your security product is RACF, your certificate is contained in a RACF profile established when you first configured PKI Services. That certificate must be connected as the default certificate to the key ring identified by the KeyRing keyword in the PKI Services configuration file. (The default location for this file is `/etc/pkiserv/pkiserv.conf`.) Use the RACF RACDCERT LIST and LISTRING commands to determine if the correct certificate is connected to the key ring. If you are attempting to rekey the PKI Services CA, you must follow the rollover process detailed in Chapter 20, "RACF administration for PKI Services," on page 403. Make any required changes. Then, restart PKI Services. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

Routing code: 2

Descriptor code: 6

IKYP029I PKI Services can only be started from a started procedure

Explanation: An attempt made at starting the PKI Services daemon was rejected because it was not made from a started procedure.

System action: The PKI Services daemon halts its initialization and stops after displaying this message to

the standard output (STDOUT) of the process.

System programmer response: Use the started procedure that PKI Services supplies in SYS1.PROCLIB(PKISERVD). For more information, see "Steps for starting the PKI Services daemon" on page 117.

IKYP030I CRL APPROACHING MAXIMUM SIZE

Explanation: PKI Services is creating CRLs as part of CRL processing and has encountered at least one CRL that is approaching the maximum size for CRL posting objects in the object store. This can occur when large CRL posting has not been configured.

System action: PKI Services CRL processing continues. If the CRLs are all less than the record size limit of approximately 32K bytes, CRL processing within PKI Services functions normally. However, CRL processing outside of PKI Services might be adversely affected due to the size of the CRL. If any CRL exceeds the record size limit, PKI Services CRL processing will be unsuccessful, and the large CRLs will not be published to the LDAP directory. When this happens you will also receive message IKYC010I with the error code description `Record too long`.

System programmer response: It is imperative that you correct the situation immediately. You can take either of these approaches:

- If you want to continue to use VSAM records or DB2 tables for LDAP posting, and if you are not yet using distribution point CRLs, start using them now. Edit the PKI Services configuration file and add the `CRLDistSize` directive to the **CertPolicy** section. If you are already using distribution point CRLs, decrease the value specified for the `CRLDistSize` directive. Make the appropriate changes and save the configuration file.

  Note: These changes will not result in an immediate reduction in the size of the CRL. You will continue to see this message until the revoked certificates on the CRL expire and are removed from the CRL.

- Alternatively, you can enable large CRL posting. If you do this, PKI Services stores CRLs in a z/OS UNIX file system instead of in a VSAM data set or DB2 table, and the record size limit of approximately 32K bytes does not apply. Edit the PKI Services configuration file and add the `EnableLargeCRLPosting` and `LargeCRLPath` directives to the **CertPolicy** section. In addition, you need to configure a z/OS UNIX file system to hold CRLs. For more information, see "Enabling support for large CRLs" on page 223.

Guideline: Enable large CRL posting.

Once the configuration file has been saved, stop and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYP031E [RSA | DSA | ECC] signing key algorithm error

Explanation: PKI Services is reading the **CertPolicy** section of its configuration file (`pkiserv.conf`) to find the signing algorithm. One of the following conditions occurred:

- The CA certificate key type does not match the signature algorithm you specified with the `SigAlg1` value in the **CertPolicy** section of the `pkiserv.conf` configuration file.
- The OID corresponding to the specified algorithm in the **OIDs** section is incorrect or is not specified at all.

System action: PKI Services stops.

System programmer response: Make sure the `SigAlg1` value in the **CertPolicy** section and its corresponding OID value in the **OIDs** section are correct and compatible with the CA certificate's key type.

If the CA certificate key type is RSA, specify the `SigAlg1` algorithm value as one of the following:

- `sha-1WithRSAEncryption` (OID value `1.2.840.113549.1.1.5`)
- `sha-256WithRSAEncryption` (OID value `1.2.840.113549.1.1.11`)
- `sha-384WithRSAEncryption` (OID value `1.2.840.113549.1.1.12`)
- `sha-512WithRSAEncryption` (OID value `1.2.840.113549.1.1.13`)
- `sha-224WithRSAEncryption` (OID value `1.2.840.113549.1.1.14`)
- `md-5WithRSAEncryption` (OID value `1.2.840.113549.1.1.4`)
- `md-2WithRSAEncryption` (OID value `1.2.840.113549.1.1.2`)

If the CA certificate key type is DSA, specify the `SigAlg1` algorithm value as follows:

- `id-dsa-with-sha1` (OID value `1.2.840.10040.4.3`)

If the CA certificate key type is ECC, specify the `SigAlg1` algorithm value as follows:

- `ecdsa-with-sha1` (OID value `1.2.840.10045.4.1`)
- `ecdsa-with-sha224` (OID value `1.2.840.10045.4.3.1`)
- `ecdsa-with-sha256` (OID value `1.2.840.10045.4.3.2`)
- `ecdsa-with-sha384` (OID value `1.2.840.10045.4.3.3`)
- `ecdsa-with-sha512` (OID value `1.2.840.10045.4.3.4`)

Correct the configuration values, and restart PKI Services. For more information, see "Updating the signature algorithm" on page 215.

IKYP032I PKI SERVICES DOES NOT HAVE RA CAPABILITY. SCEP PROCESSING SUSPENDED

Explanation: The PKI Services daemon process is starting. Initialization processing determines that the SCEP interface should be enabled and it reads the contents of the key ring to locate the certificate and key to be used for the SCEP registration authority (RA) function. No RA-capable certificate and key was found.

System action: Initialization continues but PKI Services SCEP processing is suspended.

System programmer response: The RA function of PKI Services requires a certificate and a private key capable of creating general purpose digital signatures and enciphering session keys, with `key usage`, `digitalSignature`, and `keyEncipherment` (`Handshaking`). Either the PKI Services CA certificate must have this capability (which is atypical) or an additional, dedicated RA certificate for PKI Services must be established. In either case, the certificate must have the proper key usage and must have an RSA private key. If a dedicated RA certificate is used, specify its label using the `RALabel` directive in the **SAF** section of the PKI Services configuration file (`pkiserv.conf`). The RA certificate must be assigned to the user ID of the PKI Services daemon and must be connected to the PKI Services key ring with USAGE PERSONAL and DEFAULT NO. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 246. If you make changes to the PKI Services key ring to correct the problem, stop and restart PKI Services.

Routing code: 2

Descriptor code: 6

IKYP033I Incorrect value specified for CA domain name

Explanation: The PKI Services daemon process is starting. Initialization processing has determined that the CA domain name value specified in the `_PKISERV_CA_DOMAIN` environment variable in the `pkiserv.envars` file contains illegal characters.

System action: PKI Services stops.

System programmer response: The first eight characters of the CA domain name are limited to the following character set: alphanumeric characters (`a-z`, `A-Z`, `0-9`) and the hyphen (`-`). In addition, the first character must not be a number or hyphen. Edit the PKI Services environment variables file `pkiserv.envars` for the CA instance in error and correct the value specified for `_PKISERV_CA_DOMAIN`. Restart PKI Services. For more information, see "Adding a new CA domain" on page 230.

IKYP034E ICSF UNAVAILABLE. SCEP PROCESSING SUSPENDED

Explanation: PKI Services is attempting to decrypt a Simple Certificate Enrollment Protocol (SCEP) request received from a SCEP client or to sign its response. ICSF manages the private key required for SCEP decryption and signing but ICSF is unavailable for any of the following possible reasons:

- ICSF is inactive or incorrectly configured.
- The user ID of the PKI Services daemon has insufficient authority to use the ICSF private key.
- A system administrator inadvertently deleted the certificate and its ICSF private key.

System action: PKI Services rejects the SCEP request.

System programmer response: Ensure that ICSF and the PCI cryptographic coprocessor (if applicable) are properly configured and operational. Follow the documentation pertaining to any issued messages having the **CSF** prefix. If you make changes to ICSF to correct the problem, stop and restart PKI Services.

If ICH408I messages are issued for insufficient authority to CSFKEYS or CSFSERV class resources, then the user ID of the PKI Services daemon has insufficient authority to use the private key. Give the user ID the required access to the specified resource.

To determine if your key still exists or requires the PCI cryptographic coprocessor, see Chapter 20, "RACF administration for PKI Services," on page 403. To determine if your SCEP configuration requires a CA certificate or a CA/RA combination, see "Installing and configuring ICSF (optional)" on page 23.

Routing code: 2

Descriptor code: 6

IKYP035I Unsupported character(s) in CA's name

Explanation: The PKI Services daemon process is starting. Initialization processing has determined that the CA's distinguished name contains one or more characters that have UTF-8 or BMP encoding that does not map to code page IBM-1047.

System action: PKI Services stops.

System programmer response: The characters in the distinguished name in the CA certificate must map to code page IBM-1047. Choose another CA certificate whose name has only characters that map to code page IBM-1047.

IKYP036I UNSUPPORTED CHARACTER(S) in RA'S NAME. SCEP PROCESSING SUSPENDED

Explanation: The PKI Services daemon process is starting. Initialization processing has determined that the RA's distinguished name contains one or more characters that have UTF-8 or BMP encoding that does

not map to code page IBM-1047.

System action: Initialization continues, but PKI Services SCEP processing is suspended.

System programmer response: If SCEP support is needed, choose another RA certificate in which the name has only characters that map to code page IBM-1047.

Routing code: 2

Descriptor code: 6

IKYP037I ONE OR MORE AUTOMATICALLY RENEWED CERTIFICATES CAN NOT BE SENT

Explanation: PKI Services attempted to create or retrieve the notes with the renewed certificates. An internal error occurred.

System action: PKI Services continues with the automatic renewal processing.

System programmer response: Check for more details in IKYC069I messages in the PKI Services log file and manually send the transaction IDs to the e-mail addresses to tell the users to pick up the renewed certificates.

Routing code: 2

Descriptor code: 6

IKYP038I IKYP038I THE DIRECTORY OR FILE SPECIFIED EXCEEDS THE MAXIMUM LENGTH OF *nnn* CHARACTERS

Explanation: The PKI Services daemon process is starting. Initialization processing has determined that the value specified by the _PKISERV_VARDIR variable or the _PKISERV_EXIT variable in the pkiserv.envars file exceeds the limit indicated in the message.

System action: PKI Services stops.

System programmer response: Correct the name specified by _PKISERV_VARDIR or _PKISERV_EXIT and restart PKI Services.

Routing code: 2

Descriptor code: 6

IKYP039E DIRECTORY POST UNSUCCESSFUL. ERROR CODE = *nnnn*

Explanation: PKI Services background certificate processing is attempting to post information (such as a certificate or CRL) to a directory. The post was unsuccessful. The return code from the associated LDAP client API is displayed in the message.

System action: The information is not posted at this time. The post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database.

System programmer response: Determine the cause of the failure from the return code displayed and take appropriate action. The return codes are documented in *IBM Tivoli Directory Server Client Programming for z/OS*. If the error is LDAP_NO_SUCH_OBJECT, the LDAP entry could not be created because the required suffix does not exist. Check the PKI Services log to determine the entry that could not be created, as indicated by messages IKYC005I and IKYC008I. If the entry should be posted to LDAP, you need to define the suffix in the LDAP server configuration file and recycle the LDAP server.

For more information, see "Installing and configuring LDAP" on page 21 and *IBM Tivoli Directory Server Administration and Use for z/OS*. If you want PKI Services to bypass LDAP posting for certificates with missing suffixes, set `RetryMissingSuffix=F` in the PKI Services `pkiserv.conf` configuration file. Then, stop and restart the PKI Services daemon. For more information, see "Steps for tailoring the LDAP section of the configuration file" on page 98.

Routing code: 2

Descriptor code: 6

IKYP040I PKI SERVICES DOES NOT HAVE KEY GENERATION CAPABILITY

Explanation: The PKI Services daemon process is starting. Initialization processing has determined that key generation capability is required, but an error has occurred in locating or creating the token specified in the `TokenName` field of the **SAF section** of the configuration file `pkiserv.conf`.

System action: Initialization continues but PKI Services does not have key generation capability.

System programmer response: Make sure that the token data set (TKDS) has been set up and the system has the required hardware. For information about setting up the TKDS, see *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*.

Routing code: 2

Descriptor code: 6

IKYP041E PKI SERVICES CA CERTIFICATE HAS *nnnn* SERIAL NUMBERS REMAINING.

Explanation: The PKI Services CA certificate can sign a finite number of certificates before it exceeds the maximum unique serial number usable by PKI Services (X'FFFFFFFE', or 4294967294). PKI Services has detected that the CA certificate will run out of unique serial numbers soon.

System action: PKI Services processing continues.

System programmer response: Add a new CA domain to continue certificate generation capability under a new CA certificate. For more information, see "Adding a new CA domain" on page 230.

Routing code: 2

Descriptor code: 6

IKYP042E PKI SERVICES HAS NO REMAINING SERIAL NUMBERS FOR CERTIFICATE GENERATION.

Explanation: The PKI Services CA certificate can sign a finite number of certificates before it exceeds the maximum unique serial number usable by PKI Services (X'FFFFFFFE', or 4294967294). PKI Services has detected that the CA certificate has exceeded this maximum serial number.

System action: PKI Services background certificate processing is suspended. No certificates are issued until the problem is corrected. However, certificate request management functions are still available through the `R_PKIServ` callable service and the PKI Services Web pages.

System programmer response: Add a new CA domain to continue certificate generation capability under a new CA certificate. For more information, see "Adding a new CA domain" on page 230.

Routing code: 2

Descriptor code: 6

IKYP043I PKI Services CA certificate cannot be a Diffie-Hellman certificate.

Explanation: An elliptic curve cryptography (ECC) certificate with only the keyAgreement keyUsage bit set, or with the keyAgreement bit set with either encipherOnly or decipherOnly set, is an ECC Diffie-Hellman certificate. Its intended usage is for key exchange, not for signing. Therefore, a CA certificate cannot be of this type.

System action: PKI Services stops.

System programmer response: Make sure that the key ring specified in the **SAF** section of the PKI Services configuration file (`pkiserv.conf`) contains a CA certificate that is not an ECC Diffie-Hellman certificate.

IKYS001I Error *nnnn* **{attaching | detaching}** *OCSF-service-provider-description*

Explanation: PKI Services is attaching or detaching an OCSF or OCEP service provider module. The attach or detach failed. The service provider in error and the error code encountered are displayed.

System action: PKI Services stops.

System programmer response: Look up the error code in either *z/OS Open Cryptographic Services Facility Application Programming* or *z/OS Integrated Security Services Open Cryptographic Enhanced Plug-ins Application Programming*. Diagnose the problem indicated by the return code. Restart PKI Services after corrections are made.

IKYS002I Error *nnnn* **in** *OCSF-API-name*

Explanation: PKI Services is calling an OCSF or OCEP API. The invocation has failed. The API name and error code encountered are displayed.

System action: If the error occurs during PKI Services initialization, PKI Services stops. Otherwise, PKI Services continues processing. However, needed cryptographic services might not be available.

System programmer response: If you are using ICSF for your CA's private key operations and the failing service is either `CSP_CreateSignatureContext` or `CSSM_SignData`, check that ICSF is functioning and configured properly for PKA operations. For this problem, you will also see console message IKYP001E. Follow the instructions for message IKYP001E. For all other errors, look up the error code in either *z/OS Open Cryptographic Services Facility Application Programming* or *z/OS Integrated Security Services Open Cryptographic Enhanced Plug-ins Application Programming*. Diagnose the problem indicated by the return code. Restart PKI Services after corrections are made, if needed.

IKYS003I Error *nnnn* **in getting {subject name | public key} from certificate:** *error-code-description*

Explanation: PKI Services is retrieving its CA certificate from the SAF key ring. An error occurred while PKI Services was extracting the subject name or public key from the certificate. The error code encountered is displayed. A description of the error is also displayed, if known. This might indicate a problem with the certificate stored in the SAF key ring or it might be an internal error.

System action: PKI Services stops.

System programmer response: Ensure that the certificate stored in the SAF key ring is correct. If no problems are found, report the error to the IBM support center. For more information, see Chapter 20, "RACF administration for PKI Services," on page 403 and *z/OS Security Server RACF Security Administrator's Guide*.

IKYS004I Error 0x*nnnn* **in opening key ring** *key-ring-name*

Explanation: PKI Services is initializing and is calling System SSL services to open the SAF key ring containing the CA certificate. The open failed. The key ring name and System SSL services error code encountered is displayed.

System action: PKI Services stops.

System programmer response: Look up the error code in *z/OS Cryptographic Services System SSL Programming*. Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

IKYS005I Error 0x*nnnn* in closing key ring

Explanation: PKI Services is terminating and is invoking System SSL services to close the SAF key ring containing the CA certificate. The close failed. The System SSL services error code encountered is displayed.

System action: PKI Services continues termination.

System programmer response: Look up the error code in *z/OS Cryptographic Services System SSL Programming*. Diagnose the problem indicated by the return code. Make corrections as indicated. Restart PKI Services if desired.

IKYS006I Cannot delete the signing context

Explanation: PKI Services is attempting to sign a certificate or CRL and is invoking the OCSF API `CSSM_DeleteContext`. The invocation failed.

System action: The certificate or CRL is not created.

System programmer response: Report the error to the IBM support center.

IKYS007I No KeyRing value specified under SAF section in pkiserv.conf file

Explanation: PKI Services is reading its configuration file to locate the value specified for `KeyRing` in the **SAF** section. The value is missing or has an incorrect syntax.

System action: PKI Services stops.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYS008I Signing key is from unknown crypto service provider

Explanation: PKI Services is retrieving its private key from the SAF key ring. The private key type is not known to PKI Services. This might indicate a problem with the certificate and private key stored in the SAF key ring or it might be an internal error.

System action: PKI Services stops.

System programmer response: Ensure that the certificate and private key stored in the SAF key ring are correct. If no problems are found, report the error to the IBM support center. For more information, see Chapter 20, "RACF administration for PKI Services," on page 403 and *z/OS Security Server RACF Security Administrator's Guide*.

IKYS009I Profile for key ring *key-ring-name* not found

Explanation: PKI Services is reading its configuration file to locate the value specified for `KeyRing` in the **SAF** section. The key ring specified is incorrect. No such key ring exists.

System action: PKI Services stops.

System programmer response: Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 60.

IKYS010I Profile for key ring or default certificate or private key not found

Explanation: PKI Services is attempting to retrieve data from the SAF key ring specified by the `KeyRing` value in the **SAF** section of the `pkiserv.conf` file. The key ring specified does not appear to be set up properly. Possible problems are:

- Key ring is empty.
- CA certificate in the key ring not connected as PERSONAL DEFAULT.
- CA certificate in key ring has no private key.
- User ID assigned to the PKI Services daemon has insufficient authority to read the key ring or private key.

System action: PKI Services stops.

System programmer response: Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 4, "Running IKYSETUP to perform RACF administration," on page 29 and *z/OS Security Server RACF Security Administrator's Guide*.

IKYS011I Error *error-description* in pthread_rwlock_rdlock/wrlock

Explanation: PKI Services is retrieving its CA certificate from the SAF key ring. An internal error occurred while PKI Services was calling the `pthread_rwlock_rdlock` or `pthread_rwlock_wrlock` UNIX function. A description of the error is displayed.

System action: PKI Services stops.

System programmer response: Report the error to the IBM support center.

IKYS012I Error *error-description* in pthread_rwlock_unlock

Explanation: PKI Services is retrieving its CA certificate from the SAF key ring. An internal error occurred while PKI Services was invoking the

pthread_rwlock_unlock UNIX function. A description of the error is displayed.

System action: PKI Services stops.

System programmer response: Report the error to the IBM support center.

IKYS013I Cannot find the private key associated with the {default | RA} certificate

Explanation: PKI Services is attempting to retrieve data from the SAF key ring specified by the KeyRing value in the **SAF** section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. The problem is related to either the CA (default) certificate or the RA certificate, as indicated in the message. Possible problems are:

- The certificate is not connected to the ring.
- The certificate is incorrectly connected to the key ring. Both must have USAGE PERSONAL and the CA certificate must be the DEFAULT.
- The certificate has no private key.
- The user ID assigned to the PKI Services daemon has insufficient authority to read the key ring or the private key.

System action: If the problem is with the default certificate, PKI Services stops. If the problem is with the RA certificate, PKI Services continues but the Simple Certificate Enrollment Protocol (SCEP) is disabled.

System programmer response: Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 4, "Running IKYSETUP to perform RACF administration," on page 29 and *z/OS Security Server RACF Security Administrator's Guide*.

IKYS014I Cannot find the {default | RA} certificate with private key associated in key ring

Explanation: PKI Services is attempting to retrieve data from the SAF key ring specified by the KeyRing value in the **SAF** section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. The problem is related to either the CA (default) certificate or the RA certificate, as indicated in the message. Possible problems are:

- The certificate is not connected to the ring.
- The certificate is incorrectly connected to the key ring. Both must have USAGE PERSONAL and the CA certificate must be the DEFAULT.
- The user ID assigned to the PKI Services daemon has insufficient authority to read the key ring or the private key.

System action: If the problem is with the default certificate, PKI Services stops. If the problem is with the RA certificate, PKI Services continues but the Simple Certificate Enrollment Protocol (SCEP) is disabled.

System programmer response: Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see "Locating your PKI Services certificates and key ring" on page 406 and *z/OS Security Server RACF Security Administrator's Guide*.

IKYS015I RACF callable service, R_datalib, with function code nnnn returns with SAF return code=*nnnn*, RACF return code=*nnnn*, RACF reason code=*nnnn*

Explanation: PKI Services is attempting to retrieve data from the SAF key ring specified by the KeyRing value in the **SAF** section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. Possible problems are:

- Key ring is empty.
- CA certificate in the key ring not connected as PERSONAL DEFAULT.
- CA certificate in key ring has no private key.
- User ID assigned to the PKI Services daemon has insufficient authority to read the key ring or private key.

System action: PKI Services stops.

System programmer response: Look up the return and reason code displayed in *z/OS Security Server RACF Callable Services*. Make corrections as needed. Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 20, "RACF administration for PKI Services," on page 403 and *z/OS Security Server RACF Security Administrator's Guide*.

IKYS016I Error 0x*nnnn* getting the {default | RA | CA} key

Explanation: PKI Services is initializing and calling System SSL services to get one of its private keys from the key ring. The obtain failed. The problem is related to one of the following, as indicated in the message:

- default - the CA certificate private key
- RA - the RA certificate private key
- CA - a former CA certificate that was re-keyed and rolled over

System action: If either the default or RA key cannot be obtained, PKI Services stops.

If issued for a former CA certificate, PKI Services continues initialization. However, SCEP processing might be disabled if the RA certificate cannot be signature verified.

System programmer response: Look up the error code in *z/OS Cryptographic Services System SSL Programming*. Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

IKYS017I Error 0x*nnnn* exporting the {default | RA} key

Explanation: PKI Services is initializing and calling System SSL services to export one of its private keys from the key ring. The export failed. The problem is related to either the CA (default) certificate private key or the RA certificate private key, as indicated in the message.

System action: PKI Services stops.

System programmer response: Look up the error code in *z/OS Cryptographic Services System SSL Programming*. Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

IKYS018I Error 0x*nnnn* signing Certificate/CRL

Explanation: PKI Services is attempting to sign a certificate or CRL. The signing failed.

System action: PKI Services continues processing. However, the needed cryptographic services might not be available.

System programmer response: If you are using ICSF for your CA's private key operations, check that ICSF is functioning and configured properly for PKA operations. For all other errors, look up the error code in *z/OS Cryptographic Services System SSL Programming*. Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

IKYU001I Unable to open file *file-pathname* for {READ | WRITE}

Explanation: A user is running a PKI Services utility program. The program is unable the open the input file specified. The name of the file is displayed in the message.

System action: The program ends.

User response: Check that the specified file pathname is correct and that the file exists. Also check the file's permissions to ensure that the user is allowed to process the file. Make changes as necessary and retry the program.

IKYU002I SAF Service IRRSPX00 Returned SAF RC = *nn* RACF RC = *nn* RACF RSN = *nn* {*diagnostic-information*}

Explanation: A user is running a PKI Services utility program. The program encountered an error while calling the `R_PKIServ` (IRRSPX00) callable service. The diagnostic information is displayed at the end of the message. For meanings of the diagnostic information, see message IKYI002I.

System action: The program ends.

RACF administrator response: Determine if the user should be permitted to perform the task. Make RACF authorization changes if necessary. For authorization information, see "Authorizing users for the PKI Services administration group" on page 403.

System programmer response: See message IKYI002I for information. For more information, see Chapter 18, "Using PKI Services utilities," on page 357.

User response: Correct the problem if applicable. If you cannot correct the problem, contact your Web administrator.

Web administrator response: See the Web administrator responses listed for message IKYI002I.

IKYU003I Unknown field name found in SCEP data file, *error-field-name*

Explanation: A user is running the `pkiprereg` PKI Services utility program. The program is reading the SCEP data file and has encountered a field name that it does not recognize. The erroneous field name is displayed.

System action: The program continues. If the user specified the **-l** option, the client is not preregistered.

User response: Correct the data file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 363.

IKYU004I Required field name {"ClientName" | "Template"} missing from input file at line *nnn*

Explanation: A user is running the `pkiprereg` PKI Services utility program. The program is reading the SCEP data file and encountered a preregistration entry that is missing a required field name. The missing field name is displayed. The line number where the error was found is also displayed.

System action: The program continues. If the user specified the **-l** or **-r** option of the `pkiprereg` utility, the current record is not processed.

User response: Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 363.

IKYU005I Duplicate ClientName=*error-value* entry found in SCEP data file at line *nnn*

Explanation: A user is running the `pkiprereg` PKI Services utility program. The program is reading the SCEP data file and encountered a preregistration entry with a `ClientName` value that is already in use. The value was either used earlier in the file or is already registered in PKI Services. The `ClientName` value and the line number where the error was found are displayed.

System action: The program continues. If the user specified the **-l** option of `pkiprereg` utility, the client is not preregistered.

User response: Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 363.

IKYU006I Preregistration record for ClientName=*error-value* not found in PKI Services

Explanation: A user is running the `pkiprereg` PKI Services utility program with the **-r** (remove) option. The program is reading the SCEP data file and encountered a preregistration entry with a `ClientName` value that does not exist in PKI Services. The erroneous field name is displayed.

System action: The program continues. The current record is not processed.

User response: Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 363.

IKYU007I Incorrect *field-name*=*error-value* entry found in SCEP data file

Explanation: A user is running the `pkiprereg` PKI Services utility program. The program is reading the SCEP data file and encountered a field name that has an incorrect value. If the error is length-related, the field name and error value are displayed; otherwise, only the field name is displayed.

System action: The program continues. If the user specified the **-l** option of `pkiprereg` utility, the current entry is not processed.

User response: Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 363.

IKYU008I Template nickname *template* not found in certificate templates file

Explanation: A user is running the `pkiprereg` PKI Services utility program. The program is reading the SCEP data file and encountered a encountered a value pair of *template=nickname* where the nickname is not found in the certificate templates file.

System action: The program continues. If the user specified the **-l** option of `pkiprereg` utility, the client is not preregistered.

User response: Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 363.

IKYU009I Out of memory

Explanation: A user is running a PKI Services utility program. The program is unable to allocate more memory.

System action: The program ends.

User response: Increase the amount of memory available to the program if possible. Otherwise report the error to your system programmer.

System programmer response: Increase the amount of REGION available to the user or adjust the Language Environment® runtime memory settings. (For instructions, see *z/OS Language Environment Programming Guide*.)

IKYU010I Internal error occurred in function *function-name*: *diagnostic-information*

Explanation: A user is running a PKI Services utility program. The program has encountered an internal error. The function in error is displayed. Additional diagnostic information might also be displayed.

System action: The program ends.

User response: Report the error to the IBM support center.

IKYU011I System function *function-name* detected error -- *error-string*

Explanation: A user is running a PKI Services utility program. The program received an error calling a system service. The service name and error message are displayed.

System action: The program ends.

User response: See documentation related to the service that failed. Make any necessary corrections. Then, rerun the program.

IKYU012I Unable to open message catalog *message-catalog-filename* -- *error-string*

Explanation: A user is running a PKI Services utility program. The program is attempting to open an external message catalog using the default location for the message catalog as set by the NLSPATH environment variable.

System action: The program continues using default messages.

User response: If you require an external message catalog, correct the indicated error. Then, rerun the program. For system errors, report the problem to your system programmer. For more information, see Chapter 18, "Using PKI Services utilities," on page 357.

System programmer response: If the user requires an external message catalog, correct the indicated error.

There are several reasons that could cause this error, such as file or directory permissions not allowing read access.

For information about updating the NLSPATH environment variable, see *z/OS UNIX System Services Programming Tools*. If default messages are acceptable, no action is necessary.

IKYU013I **Incorrect command syntax. The** *error-value* **parameter is {unknown | missing | incorrectly specified}**
Usage: *command-usage*

Explanation: A user is running a PKI Services utility program. A command parameter was entered incorrectly. The correct command usage is displayed.

System action: The program ends.

User response: Execute the command with the correct syntax. For more information, see Chapter 18, "Using PKI Services utilities," on page 357.

IKYU014I **pkiprereg complete.** *nnn* **preregistration records processed.** *mmm* **successful.** *000* **errors found**

Explanation: A user is running the `pkiprereg` PKI Services utility program. The program completed and is reporting the results. The value of *nnn* is the total number of preregistration records (entry groups) found in the SCEP data file. The value of *mmm* is the number of preregistration records that were successfully processed. The value of *000* is the number of errors found. Because multiple errors can occur for each preregistration record, *mmm* plus *000* can exceed *nnn*.

System action: The program ends.

User response: If no errors were found, no action is required. If errors were found, correct the errors in the SCEP data file. (You can remove the entries that were successful.) Rerun the program with the corrected SCEP data file. For more information, see Chapter 18, "Using PKI Services utilities," on page 357.

IKYU015I **Template with nickname** *template* **cannot be used for preregistration**

Explanation: A user is running the `pkiprereg` PKI Services utility program. The program is reading the certificates template file and found the certificate template with the specified nickname.

- If you are using CGI Web pages and pkiserv.tmpl, the template either has no **<PREREGISTER>** section or the **<PREREGISTER>** section is not properly terminated.
- If you are using JSP Web pages and PKIServTemplate.xml, the `certreq_template` tag for this nickname did not include a valid `preregrules` element.

System action: The program continues. If the **-l** option was specified, the client is not preregistered.

System programmer response: Correct the certificate template. Edit the SCEP data file to remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 363.

IKYU016I **pkiprereg complete.** *nnn* **passwords generated**

Explanation: A user is running the `pkiprereg` PKI Services utility program in **generate** mode (password generate mode). The program completes and reports the results. *nnn* is the total number of passwords generated.

System action: The program ends.

User response: No action is required.

IKYU017I *program-name* **terminated abnormally**

Explanation: A user is running a PKI Services utility program. The program can not complete due to a user or system error. The name of the utility program is displayed in the message.

System action: The program ends.

User response: Check for previously issued messages and their associated actions. Make changes as necessary and retry the program.

IKYU018I **Conversion from VSAM to DB2 failed at record** *record-key* **in VSAM file** *vsam-file-name*

Explanation: A user is running the PKI Services utility program `vsam2db2` to convert the existing object store and ICL VSAM files to DB2 tables. An error occurred when processing the record from the VSAM file indicated in the message. This message should be preceded by one indicating whether the source of the problem is VSAM or DB2.

System action: The `vsam2db2` utility program ends.

User response: Fix the error, drop the DB2 tables, and rerun the `vsam2db2` utility. For more information, see "Using the vsam2db2 utility" on page 377.

If you cannot fix the error based on the information in the message, report it to the IBM support center.

Chapter 25. File directory structure

This topic discusses the location of files in:

- z/OS product libraries
- File system directory `/usr/lpp/pkiserv/` and its subdirectories.

Product libraries

SMP/E installs PKI Services into the following product libraries:

- SAMPLIB/ASAMPLIB
  - IKYALLOC
  - IKYCDB2
  - IKYCVSAM
  - IKYDDDEF
  - IKYISMKD
  - IKYMKDIR
  - IKYRVSAM
  - IKYSBIND
  - IKYSETUP
  - IKYVBKUP
  - IKYVREST
- PROCLIB/APROCLIB
  - IKYSPROC with alias PKISERVD
- SIEALNKE/AIEALNKE
  - IKYPKID - The PKI Services daemon
  - IKYPRTM - The Resource Termination Manager for the daemon

File system directory and subdirectories

Unless you change the default, SMP/E installs PKI Services into the file system directory `/usr/lpp/pkiserv`. Table 100 describes the directory structure and contents:

Table 100. Files contained in subdirectories

Subdirectory	Contains...
`ActiveX`	The unsigned PKI Services ActiveX program installation files and the source files for use when creating signed ActiveX program installation files. For more information, see "Using the PKI Services Web application with Internet Explorer on Windows systems" on page 585. `ActiveX` contains the following subdirectories: • `signsrc` contains the unsigned ActiveX control files for Microsoft Windows Vista and later versions of Windows (files named PKICEnroll.*) and for Windows XP and earlier versions of Windows (files named PKIXEnroll.*). It also contains the file PKIActiveX.lic, which is used for all versions of Windows. Use these files to build signed ActiveX program installation files. • `PKICEnroll` contains the unsigned ActiveX control installation files for Windows Vista and later versions of windows. • `PKIXEnroll` contains the unsigned ActiveX control installation files for Windows XP and earlier versions of Windows.

© Copyright IBM Corp. 2001, 2011

Table 100. Files contained in subdirectories (continued)

Subdirectory	Contains...
bin	Utilities programs: • createcrls - Program to create certificate revocation lists (CRLs). (For more information, see "Using the createcrls utility" on page 358.) • iclview - Utility for viewing issued certificate list (certificate database). (For more information, see "Using the iclview utility" on page 359.) • pkiprereg - Utility for creating preregistration records in batch for Simple Certificate Enrollment Protocol (SCEP) clients, see "Using the pkiprereg utility" on page 363.) • pkitp_install - Program to register the PKI Services Trust Policy plug-in with OCSF. (For more information, see "Configuring and getting started with PKITP" on page 429.) • pkitp_ivp - Program to verify that the PKI Services Trust Policy plug-in installed successfully. (For more information, see "Configuring and getting started with PKITP" on page 429.) • postcerts - Program to post issued certificates to an LDAP directory. (For more information, see "Using the postcerts utility" on page 368.) • TemplateTool - Utility that works with certificate template files to perform functions you need if you implement the Web application using Java server pages (JSPs). (For more information, see "Using the TemplateTool utility" on page 370.) • vosview - Utility for viewing the object store (request database). (For more information, see "Using the vosview utility" on page 372.) • vsam2db2 - Program to copy data from the issued certificate list (ICL) and object store VSAM data sets into DB2 tables. (For more information, see "Using the vsam2db2 utility" on page 377.)
include	C header files: • pkitp.h - C language header file for writing application programs that use the PKI Trust Policy Plug-in. (For more information, see "Files for PKITP" on page 429.)
lib	Loadable files: • pkitp.so - OCSF Trust Policy plug-in for PKI Services. (For more information, see "Files for PKITP" on page 429.) • *.dll - Dynamic link libraries (DLLs) that the PKI Services daemon uses. • nls/msg/En_US.IBM-1047/*.cat - The PKI Services message catalogs. (These message catalogs are also symbolically linked in the /usr/lpp/pkiserv/lib/nls/msg/C directory as well as the /usr/lib/nls/msg/En_US.IBM-1047 and /usr/lib/nls/msg/C directories.) • *.jar - Java archives. • librpkisJNI.so - JNI shared object.
pkijsp	The Java server page (JSP) enterprise archive (EAR) file. (For information about the EAR file, see "(Optional) Modifying the JSP files and the EAR file" on page 198).
PKIServ	CGIs that make up the PKIServ Web application. (For information about CGIs, see "Relationship between CGIs and the pkiserv.tmpl file" on page 158 and Table 40 on page 175.) PKIServ contains the following subdirectories: • public-cgi - Public (non-SSL) directory • ssi-cgi-bin - SSL-protected – auth - SSL with user ID and password protection. Work runs under client's ID. – surrogateauth - SSL with user ID and password protection. Work runs under surrogate ID (PKISERV). • clientauth-cgi-bin - SSL with client certificate protection. Work runs under surrogate ID (PKISERV). – auth - SSL with client certificate protection. Work runs under administrator's ID.

Table 100. Files contained in subdirectories (continued)

Subdirectory	Contains...
samples	Various sample files, including: • expiringmsg.form - The e-mail message sent to a user as notification about an certificate that will expire • httpd.conf - Contains z/OS HTTP Server directives that you should *integrate* into your existing z/OS HTTP Server configuration file (httpd.conf). (For a code sample, see "z/OS HTTP Server configuration directives" on page 545.) • httpd2.conf - Contains z/OS HTTP Server directives for the second Web server that you should *integrate* into your existing z/OS HTTP Server configuration file for the second Web Server (httpd1443.conf). (For a code sample, see "z/OS HTTP Server configuration directives" on page 545.) • httpd.envvars - A sample of the environment variables needed for PKI Services that you should *integrate* into your existing z/OS HTTP Server environment variables file (httpd.envvars). • Makefile.pkiexit - The makefile for the PKI Services exit. (For more information, see Chapter 15, "Customizing with installation exit routines," on page 267.) • Makefile.pkitpsamp - The makefile for pkitpsamp.c, which is a sample application to call the PKI Trust Policy plug-in. (For more information, see "Files for PKITP" on page 429.) • pendingmsg.form - The e-mail message sent to an administrator listing pending requests. • pkiexit.c - The sample PKI Services exit, which PKI Services provides. (For more information, see Chapter 15, "Customizing with installation exit routines," on page 267.) • pkiserv.envars - The PKI Services environment variables file. (For more information, see "Optionally updating PKI Services environment variables" on page 57 and "The pkiserv.envars environment variables file" on page 510.) • pkiserv.tmpl - The PKI Services certificate templates file used with the REXX CGI execs. (For more information, see Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs," on page 123.) • pkiserv.conf - The PKI Services configuration file. (For more information, see "(Optional) Steps for updating the configuration file" on page 60 and Chapter 26, "The pkiserv.conf configuration file," on page 501.) • PKIServ.xsd - The XML schema for the PKI Services XML template, pkitmpl.xml. (For more information, see Chapter 13, "Implementing the Web application using Java server pages," on page 179.) • pkitmpl.xml - The PKI Services XML template file used with Java server pages (JSPs). (For more information, see Chapter 13, "Implementing the Web application using Java server pages," on page 179.) • pkitpsamp.c - Sample application to call the PKI Trust Policy plug-in. (For more information, see "Files for PKITP" on page 429 and "Building the sample application to invoke the certificate validation service" on page 434.) • readymsg.form - The e-mail message sent to a user as notification a certificate is ready for retrieval • recoverymsg.form - The e-mail message sent to a user who has requested that PKI Services recover a certificate for which PKI Services generated the keys • rejectmsg.form - The e-mail message sent to a user as notification a request for a certificate has been rejected • renewcertmsg.form - The e-mail message sent to a user as notification that PKI Services has automatically renewed an expiring certificate

Chapter 26. The pkiserv.conf configuration file

This topic includes a code sample of the pkiserv.conf configuration file.

The pkiserv.conf file is the configuration file for the PKI Services daemon. By default, you can find this file in the /usr/lpp/pkiserv/samples/ directory. For more information about the sections of the pkiserv.conf configuration file and the parameters, see "(Optional) Steps for updating the configuration file" on page 60 and Table 19 on page 60.

The following listing might not be identical to the code sample shipped with the product. For the most current sample, see the pkiserv.conf file in the source directory /usr/lpp/pkiserv/samples/.

```
# Licensed Materials - Property of IBM
# 5694-A01
# Copyright IBM Corp. 2001, 2011
# Status = HKY7780

[OIDs]
#
# Supported Distinguished Name OIDs
#
C=2.5.4.6
O=2.5.4.10
OU=2.5.4.11
CN=2.5.4.3
L=2.5.4.7
ST=2.5.4.8
TITLE=2.5.4.12
POSTALCODE=2.5.4.17
STREET=2.5.4.9
MAIL=0.9.2342.19200300.100.1.3
EMAIL=1.2.840.113549.1.9.1
SERIALNUMBER=2.5.4.5
UNSTRUCTUREDNAME=1.2.840.113549.1.9.2
UNSTRUCTUREDADDRESS=1.2.840.113549.1.9.8
DNQUALIFIER=2.5.4.46
DC=0.9.2342.19200300.100.1.25
UID=0.9.2342.19200300.100.1.1

#
# Signature Algorithm OIDs
#
sha-1WithRSAEncryption=1.2.840.113549.1.1.5
sha-256WithRSAEncryption=1.2.840.113549.1.1.11
sha-384WithRSAEncryption=1.2.840.113549.1.1.12
sha-512WithRSAEncryption=1.2.840.113549.1.1.13
sha-224WithRSAEncryption=1.2.840.113549.1.1.14
md-5WithRSAEncryption=1.2.840.113549.1.1.4
md-2WithRSAEncryption=1.2.840.113549.1.1.2
id-dsa-with-sha1=1.2.840.10040.4.3
ecdsa-with-sha1=1.2.840.10045.4.1
ecdsa-with-sha224=1.2.840.10045.4.3.1
ecdsa-with-sha256=1.2.840.10045.4.3.2
ecdsa-with-sha384=1.2.840.10045.4.3.3
ecdsa-with-sha512=1.2.840.10045.4.3.4

MyPolicy=1.2.3.4

[ObjectStore]
# Database implementation, either VSAM or DB2. Default is VSAM.
# Specify
# DBType=VSAM
# or
```

© Copyright IBM Corp. 2001, 2011

```
# DBType=DB2

# If DBType is DB2, configure the following additional keywords:
#     DBPackage                DBSubsystem
#    These keywords will be ignored if DBType is set to VSAM.
#
# If DBType is VSAM, configure the following additional keywords:
#     ObjectDSN                ObjectTidDSN      ObjectStatusDSN
#     ObjectRequestorDSN       ICLDSN            ICLStatusDSN
#     ICLRequestorDSN
# These keywords will be ignored if DBType is set to DB2.
#
# Regardless of the setting for DBType, verify the setting of
# the following keywords in this ObjectStore section:
#     SharedPLEX               RemoveCompletedReqs
#     RemoveInactiveReqs       RemoveExpiredCerts
#     RemoveExpiredCertsAndKeys

# Is the database implementation, whether it be VSAM datasets
# or DB2, shared in a sysplex with other instances of PKI
# Services?  True (T) or False (F).
# Note: The SharedPLEX keyword below replaces the SharedVSAM
#       keyword in PKI Services V1R13.  This keyword has the
#       same meaning as the SharedVSAM keyword has in prior
#       version of PKI Services.
SharedPLEX=F

# Name of the DB2 Package this instance of PKI will be using
# in the DB2 sub system specified by the DBSubsystem keyword. If
# DBSubsystem is missing or not specified, this keyword will be
# ignored.
# DBPackage=MasterCA

# Name of the DB2 Sub system. If DBPackage is missing or not
# specified, this keyword will be ignored.
# DBSubsystem=DSN9

# Data set name of the VSAM request (object store) base CLUSTER
ObjectDSN='pkisrvd.vsam.ost'

# Data set name of the VSAM object store PATH for the transaction ID (TID)
# alternate index.
ObjectTidDSN='pkisrvd.vsam.ost.path'

# Data set name of the VSAM object store PATH for the status alternate index
ObjectStatusDSN='pkisrvd.vsam.ost.status'

# Data set name of the VSAM object store PATH for the requestor alternate index
ObjectRequestorDSN='pkisrvd.vsam.ost.requestr'

# Data set name of the VSAM issued certificate list (ICL) base CLUSTER
ICLDSN='pkisrvd.vsam.icl'

# Data set name of the VSAM ICL PATH for the status alternate index
ICLStatusDSN='pkisrvd.vsam.icl.status'

# Data set name of the VSAM ICL PATH for the requestor alternate index
ICLRequestorDSN='pkisrvd.vsam.icl.requestr'

# How many days (d) or weeks (w) should completed requests remain in the
# object store?
# Specify 0d to indicate completed requests should not be removed
RemoveCompletedReqs=1w

# How many days (d) or weeks (w) should inactive requests remain in the object
# store?
# Specify 0d to indicate inactive requests should not be removed
RemoveInactiveReqs=4w

# How many days (d) or weeks (w) should expired certificates remain in the ICL?
# Specify 0d to indicate expired certificates should not be removed
#RemoveExpiredCerts=26w
```

```
# How many days (d) or weeks (w) should expired certificates and Keys remain
# in the ICL and TKDS? Specify 0d to indicate expired certificates and keys
# should not be removed
#RemoveExpiredCertsAndKeys=520w

[CertPolicy]
SigAlg1=sha-256WithRSAEncryption
CreateInterval=3m

# When the warning message should be issued. (i.e. the number of days
# or weeks before the certificate expiration date/time). Defaults to never
ExpireWarningTime=4w

TimeBetweenCRLs=1d
CRLDuration=2d

# Maximum number of certificates that may appear on one distribution point CRL.
# The default is 0 which indicates distribution point CRLs should not be created.
CRLDistSize=500

# Constant portion of the CRL distribution point leaf-node relative
# distinguished name. The distribution point number is appended to this value
# to form the common name. The default value is "CRL".
CRLDistName=CRL

# Authority Revocation List(ARL) Distribution Point. 'F' (default) indicates
# no ARL distribution point will be created. 'T' indicates ARL distribution
# point will be created IF CRLDistSize is greater than zero.
ARLDist=F

# Full path of the HFS Directory where PKI Services is to save each
# distribution point CRL specified by the CRL distribution point extension
# "Uniform Resource Identifier" fields(URI) for the http protocols. Defaults
# to /var/pkiserv/. It will be ignored if you do not create the extension
# for the http protocol.
CRLDistDirPath=/var/pkiserv/

# Values for the CRL distribution point extension URI fields for the
# protocols(ldap, http) you choose. This is repeatable. The first one
# always starts with CRLDistURI1,  followed by CRLDistURI2, 3, ...n,
# if necessary.  Uncomment and update the desired directive to enable
# URI CRL distribution point that you need. If more than one URI field
# is needed, remember to increase the field number sequentially by the
# order of one, e.g. CRLDistURI2, CRLDistURI3...

# For ldap protocol, you may specify the LDAP server indicated in the LDAP
# section below, e.g.,
#CRLDistURI1=LdapServer1

# or specify a skeleton URL which contains the protocol type, the domain
# name and the port, if needed, e.g.,
#CRLDistURI1=ldap://myotherldapserver.mycompany.com:389/

# For http protocol, specify the complete URL minus the file name of the
# distribution point CRL file, e.g.,
#CRLDistURI1=http://www.mycompany.com/PKIServ/cacerts/

| # Enable posting to LDAP of CRLs larger than 32k by storing created CRLs
| # in the HFS instead of in the VSAM Object Store dataset.
| # T - True, CRLs are stored in the HFS before being posted to LDAP
| # F - False, CRLs are stored in the VSAM Object Store dataset before
| #            being posted to LDAP. CRLs larger than 32k will not be posted
| #            to LDAP. (This is the Default)
| #EnableLargeCRLPosting=F
|
| # Full path of the HFS Directory where PKI Services is to save each
| # CRL before posting CRLs to LDAP. Defaults to /var/pkiserv/. It will be
| # ignored if you do not enable large CRL posting.
| #LargeCRLPostPath=/var/pkiserv/

| # Should this CA create CRLs that contain a critical Issuing
```

```
| # Distribution Point extension?
| # T = True, CRLs will be created with a critical Issuing Distribution
| #           Point extension. (This is the default value if not
| #           specified.)
| # F = False, CRLs will be created with no Issuing Distribution Point
| #            extension.
| #
| #CRLIDPExt=T

# What type of OCSP request is desired?
#  'none'  - No OCSP responder support (This is the default)
#  or
#  'basic' - the signature in the request(if there is one) will be ignored
OCSPType=none

#
# Enable the Simple Certificate Enrollment Protocol (SCEP)
# T = True, SCEP is enabled
# F = False, SCEP is disabled (default if not specified)
#
EnableSCEP=F

#
# Enable the Certificate Management Protocol (CMP)
# T = True, CMP is enabled
# F = False, CMP is disabled (default if not specified)
#
EnableCMP=F

#
# Should the CA restrict certificate requests to a validity period
# that does not exceed the CA certificate life time?
# T = True, requests with a validity period that exceeds the CA's
#           will fail.
# F = False, requests are not constrained to the CA's validity
#            period(this is the default value if not specified)
#
#CertValidityConstraint=F

PolicyRequired=F
PolicyCritical=F

PolicyName1=MyPolicy
Policy1Org=MyOrganization
Policy1Notice1=3
Policy1Notice2=17
UserNoticeText1=This is some very lawyerly statement for the relying party to read
and make decisions based on.
CPS1=http://www.mycompany.com/cps.html

# Length of certificate suspension grace period in day or weeks (d,w).
# Certificates which remained suspended for longer than this period are
# automatically revoked.
# The default value is 0d which indicates the grace period is unlimited.
MaxSuspendDuration=120d

# Specify the email address(es) of the administrator(s) who will receive
# an email notification on pending requests immediately. This is repeatable.
# The first one always starts AdminNotifyNew1, followed by
# AdminNotifyNew2, 3, ...n, if necessary. The field number increases
# sequentially by the order of one. Uncomment and update the desired
# email address(es) to enable the notification.
#AdminNotifyNew1=adminA@abc.com

# Specify the email address(es) of the administrator(s) who will receive
# an email notification on pending requests every day. This is repeatable.
# The first one always starts AdminNotifyReminder1, followed by
# AdminNotifyReminder2, 3, ...n, if necessary. The field number increases
# sequentially by the order of one. Uncomment and update the desired email
# address(es) to enable the notification.
#AdminNotifyReminder1=adminA@abc.com
```

```
[General]
InitialThreadCount=10

# Timeout value for the exit program. Default is 30 seconds (30s).
ExitTimeout=30s

# full pathname or data set name containing the 'your certificate is ready'
# message form. Defaults to no message issued
ReadyMessageForm=/etc/pkiserv/readymsg.form

# full pathname or data set name containing the 'your certificate request
# has been rejected' message form. Defaults to no message issued
RejectMessageForm=/etc/pkiserv/rejectmsg.form

# full pathname or data set name containing the 'your certificate is about
# to expire' message form. Defaults to no message issued
ExpiringMessageForm=/etc/pkiserv/expiringmsg.form

# full pathname or data set name containing the request(s) pending for approval
# message form. Defaults to no notification sent.
AdminNotifyForm=/etc/pkiserv/pendingmsg.form

# full pathname or data set name containing the renewed certificate. Defaults to
# no certificate sent.
RenewCertForm=/etc/pkiserv/renewcertmsg.form

# full pathname or data set name containing information on certificate(s)
# needed to be recovered.
RecoverForm=/etc/pkiserv/recoverymsg.form

# Time of day to run the PKI maintenance task in 24 hour time format (HH:MM).
# The valid range is 00:00-23:59. The default value is 00:00 midnight.
#MaintRunTime=01:00

# Days of the week to run the PKI maintenance task in 0-6 format. The value
# specified is a list of numbers between 0 and 6. 0 represents Sunday and 6
# represents Saturday.  No spaces or any other characters are permitted.
# Order of the digits is not important. Repeat digits are not allowed.
# The default value is everyday of the week: 0123456
#MaintRunDays=0123456

# Should the PKI maintenance task run when the PKI daemon is started?
# True (T) or False (F). Default value is True.
#RunMaintAtStart=T


[SAF]
KeyRing=PKISRVD/CAring
#TokenName=PKISRVD.PKIToken

# The Label name for the PKI RA certificate connected to the Key ring
#   specified in the KeyRing value above
#
RALabel=Local PKI RA

[LDAP]
NumServers=1
PostInterval=5m
Server1=myldapserver.mycompany.com:389
AuthName1=CN=root
AuthPwd1=root
CreateOUValue= Created by PKI Services
RetryMissingSuffix=T
# Name of the LDAPBIND Class profile containing the bind information for LDAP
# server 1. This key is optional. Used in place of keys Server1, AuthName1.
# and AuthPwd1
#BindProfile1=LOCALPKI.BINDINFO.LDAP1
```

Chapter 27. Environment variables

This topic describes the environment variables that PKI Services uses and their possible values. It also includes a code sample of the environment variables file, `pkiserv.envars`. (See "The pkiserv.envars environment variables file" on page 510.) For information about the PKISERVD procedure, which specifies the pathname of the environment variables file, see "PKISERVD sample procedure to start PKI Services daemon" on page 562.

Environment variables in the environment variables file

The environment variables contained in `pkiserv.envars` and their values are:

_PKISERV_CA_DOMAIN

Specifies the CA domain. The first eight characters must be unique. The first eight characters of the CA domain name are limited to the following character set: alphanumeric characters (`a-z`, `A-Z`, `0-9`) and the hyphen (`-`). In addition, the first character must not be a number or hyphen.

Example:

```
_PKISERV_CA_DOMAIN=WebAppCA
```

_PKISERV_CONFIG_PATH

Specifies the pathname for the directory containing the configuration file, `pkiserv.conf`, and the certificate template file, `pkiserv.tmpl` for this CA domain. The default value (if you do not set the environment variable) is `/etc/pkiserv`.

Guideline: Copy both of these files from the install directory, `/usr/lpp/pkiserv/samples`, before making any changes.

Note: Because the PKISERV CGIs run in a z/OS HTTP Server address space, if the `pkiserv.tmpl` file is not in its default location of `/etc/pkiserv/pkiserv.tmpl`, you need to add the `_PKISERV_CONFIG_PATH` variable to the z/OS HTTP Server environment variable file. The HTTP servers environment variables file is usually in `/etc/httpd.envvars`. PKI Services uses two instances of the z/OS HTTP Server. Therefore, if the two servers are using different environment variables files, you need to update both files.

_PKISERV_EXIT

Specifies the full pathname for the installation-provided PKI exit program that the PKI Services daemon will invoke to perform autorenew preprocessing or postprocessing. (This exit is a UNIX-executable program or shell script.) If you do not define this variable or if it contains a null value, the PKI autorenew exit processing is disabled.

Note: The _PKISERV_EXIT environment variable is also used by the PKI Services CGI scripts to specify an exit program to be used by the Web application. The PKI Services CGI scripts run in a z/OS HTTP Server address space, so you must specify the `_PKISERV_EXIT` environment variable in the z/OS

© Copyright IBM Corp. 2001, 2011

HTTP Server environment variables file. The z/OS HTTP Server environment variables file is usually `/etc/httpd.envvars`. PKI Services uses two instances of the z/OS HTTP Server. Therefore, if the two servers are using different environment variables files, you need to update both files.

_PKISERV_MSG_LOGGING

Values include:

STDOUT_LOGGING Indicates writing *all* messages (verbose, diagnostic, informational, warning, error, and severe) to STDOUT and *additionally* writing the error and severe messages to STDERR. This is the default if the environment variable is not set.

STDERR_LOGGING Indicates writing verbose, diagnostic, informational, and warning messages to STDOUT and writing error and severe messages to STDERR.

_PKISERV_MSG_LEVEL

Specifies the subcomponent and message level to log. Messages for a particular subcomponent are logged only if the message level is greater than or equal to the specified level for that subcomponent. You can use an asterisk (`*`) to indicate all subcomponents. The subcomponent list consists of a subcomponent name and a message level separated by a period (`.`).

For example, the following sets the message level for all subcomponents to log warning messages or higher. (This is the default setting.)

Example:

```
_PKISERV_MSG_LEVEL=*.W
```

You can specify multiple subcomponents by separating entries with a comma (`,`). For example, the following indicates that all subcomponents are set to message level **W** (warning) and that the PKID subcomponent is set to message level **D** (diagnostic).

Example:

```
 _PKISERV_MSG_LEVEL=*.W,PKID.D
```

The subcomponents are:

Subcomponent	Meaning
*	The wildcard character (represents all subcomponents)
CORE	The core functions of PKI Services that are not specific to the other subcomponents
DB	Activity related to the object store or issued certificate list repositories
LDAP	LDAP posting operations
PKID	The PKI Services daemon address setup and infrastructure
POLICY	Certificate creation and revocation policy processing
SAF	SAF key ring, OCEP, and `R_datalib` calls
TPOLICY	Trust policy plug-in processing

The message levels are listed hierarchically:

Debug level	Meaning
S	This indicates logging only severe messages.
E	This indicates logging severe and error messages.
W	This indicates logging severe, error, and warning messages. This is the *default* message level for all subcomponents if you do not set the environment variable.
I	This indicates logging severe, error, warning, and informational messages.
D	This indicates logging severe, error, warning, informational, and diagnostic messages.
V	This indicates logging *all* messages, including verbose diagnostic messages. This is very verbose. **Guideline:** Do not use **V** level unless IBM support personnel instruct you to do so.

_PKISERV_VARDIR

Specifies the pathname for a directory in which PKI Services will write persistent data. The maximum length of the pathname is 256 characters, including the trailing /. The default value (if you do not set the environment variable) is `/var/pkiserv`.

_PKISERV_ENABLE_JSP

Specifies whether you use the JSP interface and XML templates for PKI Services Web pages, or the REXX CGI execs and text templates. Set to TRUE to use the JSP interface. The default (if you do not set the environment variable) is to use the REXX CGI exec interface.

The pkiserv.envars environment variables file

The following code sample is for the pkiserv.envars environment variables file. (For information about updating the environment variables file, see "Optionally updating PKI Services environment variables" on page 57.) The following listing might not be identical to the code sample shipped with the product. For the most current sample, see the pkiserv.envars file in the source directory /usr/lpp/pkiserv/samples/.

```
#-------------------------------------------------------------------#
#                                                                   #
# PKI Services sample environment variable file                     #
#                                                                   #
# Licensed Materials - Property of IBM                              #
# 5694-A01                                                          #
# Copyright IBM Corp.  2001, 2010                                   #
# Status = HKY7770                                                  #
#                                                                   #
#-------------------------------------------------------------------#
#
# Language and Path configurations
#
#  Note: The PATH statement should be set to ensure the PKI Services
#        daemon uses the correct version of sendmail command based
#        on your configuration of sendmail/smtp. When running the
#        smtp server, the PATH should be set to find sendmail in the
#        /bin directory. When sendmail is configured to not use an
#        smtp server, the PATH should be set to find sendmail in the
#        /usr/sbin directory.#
LANG=En_US.IBM-1047
PATH=/bin
LIBPATH=/usr/lpp/pkiserv/lib:/usr/lib
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/pkiserv/lib/nls/msg/%L/%N

#
# When running as a CA Domain, set the CA Domain name by assigning
# desired value to the _PKISERV_CA_DOMAIN variable.
# Note: The first eight characters must be unique.
#
# example: _PKISERV_CA_DOMAIN=WebAppCA

#
# Configuration File location and Message configuration Options
#
_PKISERV_CONFIG_PATH=/etc/pkiserv
_PKISERV_MSG_LOGGING=stdout_logging
_PKISERV_MSG_LEVEL=*.w

#
# Set up a directory for PKI Services to write persistent data. The
# maximum length is 256 characters including the trailing /.
# The default is /var/pkiserv/.
#
# example: _PKISERV_VARDIR=/var/pkiserv/

#
# Set up an exit program for autorenew. The maximum length is 256
# characters including the program name.
#
# example: _PKISERV_EXIT=/mydir/renewexit

#
# Enable the JSP Webpages and XML Template.
#
#_PKISERV_ENABLE_JSP=TRUE
```

Chapter 28. The IKYSETUP REXX exec

IKYSETUP is a REXX exec that issues RACF commands to perform RACF administration. This topic describes the actions IKYSETUP performs and provides a code sample of IKYSETUP.

Actions IKYSETUP performs by issuing RACF commands

In broad terms, the actions that IKYSETUP performs are as follows:

- Sets up the PKI Services daemon user ID
- Sets up the access control to protect PKI Services
  - Protects end-user functions
  - Protects administrative functions
- Defines one or more CA domains with associated administrative domains
- Creates the CA and RA certificates, their private keys, and key ring
- Creates the z/OS HTTP Server certificate, private key, and key ring
- Enables surrogate operation for the z/OS HTTP Server
- Allows PKI Services to generate key pairs for certificate requests

Setting up the PKI Services daemon user ID

Create the daemon user ID (by default, PKISRVD) using the RACF ADDUSER TSO command. Give it an OMVS segment because it needs access to z/OS UNIX. If you implement the object store and issued certificate list (ICL) using VSAM data sets, this user ID also needs update access to the VSAM data sets identified in the **ObjectStore** section of the `pkiserv.conf` file. If necessary, use the RACF ADDSD and PERMIT TSO commands to give this user ID UPDATE access to the VSAM data sets. If you implement the object store and ICL using DB2 tables, this user ID also needs access to the Resource Recovery Services Access Facility (RRSAF). If necessary, use the RACF RDEFINE and PERMIT commands to define the profile for RRSAF in the DSNR class and give this user ID READ access.

Guideline: Define the daemon user ID with the NOPASSWORD attribute.

To associate this user ID to the PKI Services started procedure, use the following RACF TSO commands:

```
RDEFINE  STARTED PKISERVD.* STDATA(USER(PKISRVD))
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
SETROPTS RACLIST(STARTED) REFRESH
```

Setting up access control to protect PKI Services

This task can be divided into two steps:

1. Protecting end-user functions
2. Protecting administrative functions.

Protecting end-user functions

You must first determine who your end-users are and how they will be using their certificates. In general there are two categories of end-users:

- Internal clients, such as employees who have SAF user IDs on the host system and who might be using their certificates to access resources on the host

© Copyright IBM Corp. 2001, 2011

- External clients, who have no access to the host system.

When PKI Services is called, the unit of work has some identity (user ID) associated with it. For external customers, a surrogate user ID is necessary.

Guideline: Although under certain circumstances it might be beneficial for internal clients to access PKI Services under their own identities, your implementation will be simpler if you use surrogate user IDs for internal clients as well.

Use the RACF ADDUSER command to create the surrogate user ID (PKISERV). Give it an OMVS segment because it needs access to z/OS UNIX. **Guideline:** Define the surrogate user ID with the PROTECTED and RESTRICTED attributes.

The `R_PKIServ` SAF callable service is protected by FACILITY class resources of the form IRR.RPKISERV.*function*[.*ca_domain*], where function is one of the following and *ca_domain* specifies an optional CA domain name. (Specify *ca_domain* when your installation has established multiple PKI Services CAs.)

The `R_PKIServ` functions are:

EXPORT Retrieves (exports) a previously requested certificate, or retrieves (exports) the PKI Services registration authority (RA) certificate or the certificate authority (CA) certificate.

GENCERT Generates an auto-approved certificate.

GENRENEW Generates an auto-approved renewal certificate. (The request submitted is automatically approved.)

QRECOVER Lists certificates whose key pairs were generated by PKI Services under a requestor's e-mail address and passphrase.

REQCERT Requests a certificate that an administrator must approve before it is created.

REQRENEW Requests certificate renewal. The administrator needs to approve the request before the certificate is renewed.

RESPOND Invokes the PKI OCSP responder.

REVOKE Revokes a certificate that was previously issued.

SCEPREQ Generates a certificate request using Simple Certificate Enrollment Protocol (SCEP).

VERIFY Confirms that a given user certificate was issued by this certificate authority and, if so, returns the certificate fields.

Create these resources and give the PKISERV user ID either READ or CONTROL access to them. CONTROL bypasses subsequent resource checks.

Additional FACILITY class resources of the form IRR.DIGTCERT.*function* protect the actual certificate generation and retrieval functions. If subsequent resource checks are not being bypassed, define these resources and their access.

There are two ways to handle certificate approval:

- An administrator can review certificate requests
- Requests can be auto-approved without administrator action (this should probably be reserved for internal clients only).

If you plan to have an administrator approve certificate requests before issuing certificates, PKISERV needs the following access:

Table 101. Access required if you plan to have an administrator approve certificate requests

Resource	Access
IRR.DIGTCERT.REQCERT	READ
IRR.DIGTCERT.REQRENEW	READ

If your clients can request certificates that are auto-approved without action by an administrator, PKISERV needs the following access:

Table 102. Access required if you plan to use auto-approval

Resource	Access
IRR.DIGTCERT.ADD	UPDATE
IRR.DIGTCERT.GENCERT	CONTROL
IRR.DIGTCERT.GENRENEW	READ

Finally, because the Web server will be switching identities to PKISERV, you must give it surrogate permission. This is done by creating another resource in the SURROGAT class (BPX.SRV.PKISERV) and giving the Web server daemon user ID READ access to it.

Protecting administrative functions

This is much easier to set up than protecting the end-user functions. Your PKI Services administrators must have SAF user IDs on the host system. When PKI Services is called for administrative functions, the unit of work is always tagged with the identity of the authenticated administrator. Each administrator needs the following FACILITY class resource access to:

Table 103. FACILITY class access needed for protecting administrative functions

Resource	Access	Purpose
IRR.RPKISERV.PKIADMIN[.*ca_domain*]	READ	For list and query operations
	UPDATE	To act on certificate requests, preregistration requests, and issued certificates

To grant user ID ADMID authority to administer PKI Services, use the following RACF TSO commands:

Example:

```
RDEFINE  FACILITY (IRR.RPKISERV.PKIADMIN.CUSTOMER) UACC(NONE)
PERMIT   IRR.RPKISERV.PKIADMIN.CUSTOMER CLASS(FACILITY) ACCESS(UPDATE) ID(ADMID)
SETROPTS RACLIST (FACILITY) REFRESH
```

Establishing your CA and RA certificates

To create and sign digital certificates for others, you need to establish a CA certificate, and optional RA certificate, and their associated private keys using the RACDCERT command.

Steps for establishing your CA and RA certificates

Perform the following steps to create your CA certificate, RA certificate, and their associated keys, back up the keys, connect them to a key ring and authorize PKI Services to use them.

Before you begin: Determine the CA or RA's distinguished name and where it will be located (under CERTAUTH for the CA and under the PKI Services daemon user ID for the RA). Typically, CAs and RAs have distinguished names in the following form:

OU=*your-CA-or-RA's-friendly-name*.O=*your-organization*.C=*your-two-letter-country-abbreviation*

1. Create your CA certificate, and optional RA certificate, and their associated private keys using the RACDCERT GENCERT command. **Rule:** If you create an optional RA certificate, it must be signed by the CA certificate.

   Example:

   a. This example creates a 20-year CERTAUTH certificate with a distinguished name of OU=Human Resources Certificate Authority.O=Your Company, Inc.C=US.

   ```
   RACDCERT CERTAUTH GENCERT SUBJECTSDN(
      OU('Human Resources Certificate Authority')
      O('Your Company, Inc')  C('US')) WITHLABEL('Local PKI CA') HIGHTRUST
      NOTAFTER(DATE(2026/05/06))
      SIZE(1024) KEYUSAGE(HANDSHAKE)
      SIGNWITH(CERTAUTH LABEL('Local Root CA'))
   ```

   b. This example creates a 20-year RA certificate signed by the CA certificate created in Example 1a.

   ```
   RACDCERT GENCERT ID(PKISERVD) SUBJECTSDN(
      CN('Registration Authority')
      OU('Human Resources Certificate Authority')
      O('Your Company, Inc')  C('US')) WITHLABEL('Local PKI RA')
      NOTAFTER(DATE(2026/05/06))
      SIZE(1024) KEYUSAGE(HANDSHAKE)
      SIGNWITH(CERTAUTH LABEL('Local PKI CA'))
   ```

---

2. Back up your CA certificate, RA certificate (if created), and their associated private keys to password-protected data sets using the RACDCERT EXPORT command.

   Example:

   ```
   RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA'))
      DSN('PKISRVD.PRIVATE.KEY.P12BIN')
      FORMAT(PKCS12DER) PASSWORD('your-passphrase')

   RACDCERT ID(PKISRVD) EXPORT(LABEL('Local PKI RA'))
      DSN('PKISRVD.PRIVATE.RAKEY.P12BIN')
      FORMAT(PKCS12DER) PASSWORD('your-passphrase')
   ```

---

3. (Optional) If you want to use ICSF for private key protection and signing, migrate the private keys to ICSF using the RACDCERT ADD command. For this step to be successful, ICSF must be operational and configured for RSA operations. (For additional information about ICSF, see *z/OS Cryptographic Services ICSF Administrator's Guide*.)

   Example:

```
RACDCERT CERTAUTH ADD('PKISRVD.PRIVATE.KEY.P12BIN') PASSWORD('your-passphrase') ICSF

RACDCERT CERTAUTH ADD('PKISRVD.PRIVATE.RAKEY.P12BIN') PASSWORD('your-passphrase') ICSF
```

4. Create a key ring for the PKI Services daemon and add the CA certificate and RA certificate (if created) to it so that PKI Services can use the certificates. The example creates a key ring called `CAring` for user ID PKISRVD and connects the CA and RA certificates to it.

   Important: Make sure your CA certificate is marked with the TRUST or HIGHTRUST attribute in RACF. (Otherwise, PKI Services will not be able to use the certificate.) Check this by issuing the RACDCERT LIST command and execute the RACDCERT ALTER command to change it if needed.

   Example:

   ```
   RACDCERT ADDRING(CAring) ID(PKISRVD)

   RACDCERT ID(PKISRVD) CONNECT(CERTAUTH LABEL('Local PKI CA') RING(CAring
      USAGE(PERSONAL) DEFAULT)

   RACDCERT ID(PKISRVD) CONNECT(ID(PKISRVD) LABEL('Local PKI RA') RING(CAring)
      USAGE(PERSONAL))
   ```

5. Authorize the PKI Services daemon to use RACF certificates and act as the CA. The daemon user ID (PKISRVD) needs access to the FACILITY class resources listed in Table 104. RACLIST the FACILITY class if it is not already RACLISTed. Define the FACILITY class resources. When the definitions are complete, refresh the FACILITY class.

   Examples:

   ```
   SETROPTS RACLIST(FACILITY)

   RDEFINE FACILITY IRR.DIGTCERT.GENCERT
   RDEFINE FACILITY IRR.DIGTCERT.LISTRING
   RDEFINE FACILITY IRR.DIGTCERT.LIST

   PERMIT IRR.DIGTCERT.GENCERT  CLASS(FACILITY) ID(PKISRVD) ACCESS(CONTROL)
   PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(PKISRVD) ACCESS(READ)
   PERMIT IRR.DIGTCERT.LIST     CLASS(FACILITY) ID(PKISRVD) ACCESS(READ)

   SETROPTS RACLIST(FACILITY) REFRESH
   ```

   Table 104. Access PKISRVD needs to use RACF certificates

Resource	Access
IRR.DIGTCERT.GENCERT	CONTROL (if the CA certificate was created under CERTAUTH)
	READ (if the certificate was created under the PKI Services daemon user ID)
IRR.DIGTCERT.LISTRING	READ
IRR.DIGTCERT.LIST	READ

Configuring the z/OS HTTP Server for SSL mode

The PKISERV application requires the z/OS HTTP Server to operate in three modes. That is why PKI Services requires two z/OS HTTP Servers. The modes are:

- Normal

- SSL without client authentication
- SSL with client authentication.

For SSL, your server needs to obtain a digital certificate. You can:

- Purchase one from an external source
- Create one using RACF

Note: If your server is already operating in SSL mode, you can skip the following section, "Using RACF to obtain a certificate for the Web server."

Using RACF to obtain a certificate for the Web server

The z/OS HTTP Server supports using either gskkyman key databases (`.kdb` files) or RACF (SAF) key rings for the server's certificate store. You are expected to use SAF key rings if setting up their Web server for the first time.

Note: If you have already set up your Web server using gskkyman, you can continue to use it.

Use RACDCERT to generate the server certificate signed by the new Certificate Authority.

Example:

```
RACDCERT GENCERT ID(WEBSRV) SIGNWITH(CERTAUTH LABEL('Local PKI CA'))
   WITHLABEL('SSL Cert') SUBJECTSDN(CN('www.YourCompany.com') O('Your Company Inc')
   L('Millbrook') SP('New York') C('US'))
```

The Web server needs a key ring containing its new certificate and any trusted CA certificate. The RACDCERT command with operands ADDRING and CONNECT also sets this up. For example, the RACDCERT commands to create a key ring called `SSLring` for user ID WEBSRV and to connect the Web server and CA certificates to it are:

Example:

```
RACDCERT ADDRING(SSLring) ID(websrv)
RACDCERT ID(websrv) CONNECT(CERTAUTH LABEL('Local PKI CA')) RING(SSLring)
   USAGE(PERSONAL) DEFAULT)
RACDCERT ID(websrv) CONNECT(ID(websrv) LABEL('SSL Cert') RING(SSLring)
   USAGE(PERSONAL) DEFAULT)
```

Export the CA certificate to an MVS data set. Then OPUT it to an file system file so that it can be made available to your clients.

Example:

```
RACDCERT EXPORT(LABEL(''Local PKI CA'))
   CERTAUTH DSN('pkisrvd.webroot.derbin') FORMAT(CERTDER)
```

Enabling the z/OS HTTP Server for surrogate operation

Your server must be able to act as a surrogate for clients requesting certificates. To enable this, create:

- Profile BPX.SERVER in the FACILITY class
- Profile BPX.SRV.PKISERV in the SURROGAT class.

Give the z/OS HTTP Server daemon user ID READ access to both of these profiles.

Allowing PKI Services to generate key pairs for certificate requests

You can choose to allow PKI Services to generate key pairs (public and private key) for certificate requests. The PKI Services daemon does this using the PKCS #11 API provided by ICSF. Set up profiles in the CRYPTOZ class to allow the PKI Services daemon to use the PKCS #11 API:

- Activate the CRYPTOZ class.
- Define the profile SO.*daemon_id*.* in the CRYPTOZ class.
- Give the daemon user ID UPDATE access to the profile
- Define the profile USER.*daemon_id*.* in the CRYPTOZ class.
- Give the daemon user ID CONTROL access to the profile.

IKYSETUP sample

IKYSETUP contains the commands to perform the RACF administrator tasks of adding groups and user IDs, setting up access control, creating CA, RA, and SSL certificates, and setting up daemon security. The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYSETUP.

```
/* REXX */
/*********************************************************************/
/*                                                                   */
/* DESCRIPTIVE NAME:  PKI Services RACF setup CLIST                  */
/*                                                                   */
/* Licensed Materials - Property of IBM                              */
/* 5694-A01                                                          */
| /* Copyright IBM Corp. 2001, 2011                                  */
| /* Status = HKY7780                                                */
/*                                                                   */
/*01* EXTERNAL CLASSIFICATION: OTHER                                 */
/*01* END OF EXTERNAL CLASSIFICATION:                               */
/*                                                                   */
/* FUNCTION:                                                         */
/*                                                                   */
/*   This CLIST will issue the RACF TSO commands necessary to set up*/
/*   security for PKI Services. It must be run from TSO by a user ID*/
/*   that is RACF SPECIAL.                                           */
/*                                                                   */
/* USAGE:                                                            */
/*                                                                   */
/*   1) Read accompanying PKI Services post installation             */
/*      instructions.                                                */
/*   2) Perform necessary prerequisite product installation for      */
/*      the webserver (websphere), LDAP, etc.                        */
/*   3) Make note of any predetermined values such as the LDAP       */
/*      suffix, webserver fully qualified domain name, and the       */
/*      settings contained in the pkiserv.conf file.                 */
/*   4) Copy the CLIST to a data set where you can edit it.          */
/*   5) Examine the entire CLIST, in particular, the configurable    */
/*      section.                                                     */
/*   6) Modify the values in the configurable section as needed for */
/*      your installation.                                           */
/*   7) Run the CLIST. Syntax:                                       */
/*                                                                   */
/*  EX 'data-set-name(IKYSETUP)' 'RUN(YES | NO | PROMPT)'            */
/*                                                                   */
```

```
/*  where:  YES - indicates to run CLIST as is                      */
/*          NO - indicates to display the commands only             */
/*          PROMPT - indicates to prompt the user prior             */
/*           to invoking each command                               */
/*                                                                  */
/* DISCLAIMER:                                                      */
/*                                                                  */
/*   This CLIST is not intended to cover every possible customer    */
/*   scenario. Modification of the actual commands to be issued     */
/*   may be required                                                */
/*                                                                  */
/********************************************************************/
/* Change-Activity:                                                 */
/*  $L1=PKIS4  , HKY7708, 020514, JWS: PKI Services                 */
/*  $D1=MG01103, HKY7708, 020813, BRW:                              */
/*  $D1=MG01405, HKY7708, 021114, BRW:                              */
/*  $L2=PKIS6  , HKY7720, 040204, TCG: PKI Services VI              */
/*  $L3=PKIS6  , HKY7720, 040130, MSF: PKI Services VI              */
/*  $L4=PKIS7  , HKY7730, 050301, TCG: PKI Services VII             */
/*  $D2=MG06477, HKY7730, 051018, TCG: Incorrectly formed command   */
/*  $O1=MG07496, HKY7730, 060607, SHN: Misc updates                 */
/*  $L5=PKIS9  , HKY7740, 060607, SHN: Removing private from dsn    */
/*  $L6=PKIS10J, HKY7750, 070529, RDC: Remove OCSF setup            */
/*  $L7=AADS,    HKY7750, 070726, RDC: 4096 Bit Updates             */
/*  $L8=AADS,    HKY7750, 071114, RDC: 4096 Bit Updates DCR         */
/*  $L9=PKIS11K, HKY7760, 080619, WFC: PKI Services XI              */
/*  $D3=MG11251, HKY7760, 080804, TCG: Misleading comment update    */
/*  $D4=MG11676, HKY7760, 081029, TCG: Bug fix and enhancements     */
/*  $LA=PKIS12E, HKY7770, 090710, RRG: Elliptic Curve Crypto Supt   */
/*  $D5=MG13537, HKY7770, 090903, RRG: Incorrectly formed command   */
/*  $D6=MG13623, HKY7770, 090921, RRG: Incorrectly formed command   */
|/*  $LB=ECC2,    HKY7780, 100524, SGW: ECC hardware support         */
|/*  $D7=MG15079, HKY7780, 100714, SGW: Update key type logsay       */
|/*  $LC=PKIS13D, HKY7780, 100831, RRG: DB2 Support                  */
|/*  $D8=MG15384, HKY7780, 101028, SGW: Incorrect ICSF profiles      */
|/*  $D9=MG15688, HKY7780, 101206, RRG: Expiration date improvement  */
/*                                                                  */
/* Change Descriptions:                                             */
/*  A - PCICC and CSSM support                                @L1A*/
/*  C - Use restrict_surrog when checking for whether         @D1A*/
/*      surrogate user should be restricted                   @D1A*/
/*  C - change SEZALINK to SEZALOAD                           @D1A*/
/*  C - LDAP and SSL loadlibs now in SYS1.SIEALNKE, updated   @L2A*/
/*      program control setup.                                @L2A*/
/*  A - DSA key support                                       @L3A*/
/*  A - Added RA certificate and CA Domain support            @L4A*/
/*  C - Updated RACDCERT command for connecting the RA cert       */
/*      to the PKI daemons' keyring.                          @D2A*/
/*  C - Added CA domain qualifier to RA cert backup dataset       */
/*      when run with CA Domain support.                          */
/*    - Added logic and doc to prevent usage of 2048 bit DSA      */
/*      keys for the CA certificate.                              */
/*    - Added CA domain qualifier to RA label when run with CA    */
/*      Domain support.                                       @01A*/
/*  C - Removed 'private' dataset qualifier from all dataset      */
/*      names.                                                @L5A*/
/*  C - Remove RACF Profile setup for daemon access to OCSF   @L6A*/
/*  C - Prevent usage of keysize greater than 4096 for        @L7A*/
/*      key type 2 (PCICC).                                   @L7A*/
/*  C - Allow keysize up to 4096 for key type 0 (software)    @L8A*/
```

```
/*  A - Key generation support                                      */
/*    - Distinguish datasets needed to be saved and downloaded  @L9A*/
/*  C - Misleading comment about keysize supported for a            */
/*      keytype of 0.                                             @D3A*/
/*  C - Fixed error in cryptoz permit commands                      */
/*    - Added commands to define generic dataset profile            */
/*      for daemon user ID.                                         */
/*    - Added conditional logic to not issue permit commands        */
/*      for DIGTCERT profiles when Web Server certificate and       */
/*      key ring is not being defined/created.                    @D4A*/
/*  A - Allow Elliptic Curve Cryptography based keys to be used     */
/*      as the CA certificate's private key.                      @LAA*/
/*  C - Append key type to RACDCERT GENCERT command for Elliptic    */
/*      Curve Cryptography keys.                                  @D5A*/
/*  C - Incomplete repair for @D5 modification.                   @D6A*/
|/*  A - Allow ECC keys in PKDS for CA certificate                 @LBA*/
|/*  C - Add 6 & 7 to key type logsay                              @D7A*/
|/*  A - Allow use of DB2 for ICL and object store repository.     @LCA*/
|/*  C - Provide greater flexibility in specifying CA certifiate     */
|/*  C - Specify correct ICSF profiles for key generation & SSL    @D8A*/
|/*      and web server certificate expiration dates.              @D9A*/
/*                                                                  */
/********************************************************************/

trace value('O')

/*-----------------------------------------------------------------*/
/* configurable section                                            */
/*-----------------------------------------------------------------*/

/*---------------------------------*/
/* Part 1 - Things you must change */
/*---------------------------------*/


/*****************************************************************/
/* This exec will create the certificate, private key,  and      */
/* keyring needed for your certificate authority.                */
/*                                                               */
/* You must update the distinguished name of your certificate    */
/* authority defined below. The suffix of this DN must match     */
/* the suffix set up for your LDAP directory (suffix value from  */
/* your slapd.conf file).                                        */
/*                                                               */
/* Typically, Certificate Authorities have distinguished names   */
/* in the following form:                                        */
/*                                                               */
/*  OU=<your-CA's-friendly-name>,O=<your-organization>,          */
/*        C=<your-2-letter-country-abbreviation>                 */
/*                                                               */
/* e.g., OU=Human Resources Certificate Authority.O=IBM,C=US     */
/*                                                               */
/* If you already have your CA certificate and private key set   */
/* up in RACF, set ca_dn="" and update the ca_label variable to  */
/* equal your CA certificate's label. Note, it must reside       */
/* under CERTAUTH                                                */
/*                                                               */
/* If you are running with Multiple-CAs:                         */
/*   You could run IKYSETUP once for each separate CA you        */
/*   want to operate, changing ca_domain everytime. The          */
```

```
/*   ca_domain value will help qualify the other variables thus */
/*   reducing the amount of work the RACF administrator needs   */
/*   to perform. Otherwise, set to NULL.                        */
/*                                                              */
/****************************************************************/
ca_domain = ""                                          /* @L4A*/

if LENGTH(ca_domain) > 8 then                           /* @L4A*/
   ca_domain_trunc = LEFT(ca_domain,8)                  /* @L4A*/
else                                                    /* @L4A*/
   ca_domain_trunc = ca_domain                          /* @L4A*/


OrgUnit = STRIP(ca_domain "Human Resources Certificate Authority")
                                                        /* @L4A*/
ca_dn= "OU('"||OrgUnit||"')",
       "O('Your Company')",
       "C('Your Country 2 Letter Abbreviation')"        /* @L4C*/
ca_label = STRIP(ca_domain "Local PKI CA")  /* Label for CA
                                            certificate with the
                                            CA Domain name
                                            prepended      @L4A*/

/****************************************************************/
/* ra_label:                                                    */
/*   A "must change" variable - default: "Local PKI RA"         */
/* ra_dn:                                                       */
/*   A "must change" variable. If you don't wish to have PKI    */
/*   Services operate with a separate RA certificate, set       */
/*   ra_dn=""                                                   */
/****************************************************************/

ra_label = STRIP(ca_domain "Local PKI RA") /*Label for
                                            RA Certificate @01C*/

if (ra_label = "") then      /* If no RA Label ...          @L4A*/
   ra_dn=""                                            /*@L4A*/
else                                                   /*@L4A*/
   ra_dn=,                                             /*@L4A*/
     "CN('Registration Authority')",
      ca_dn

/****************************************************************/
/* This exec will create the certificate, private key,  and     */
/* keyring needed for your webserver. (Required for SSL.)       */
/*                                                              */
/* You must update the distinguished name of your               */
/* webserver. The Common Name (CN) must match your webserver's  */
/* fully qualified domain name.                                 */
/*                                                              */
/* e.g., CN=www.ibm.com,O=IBM,C=US                              */
/*                                                              */
/* If you already have your webserver configured for SSL, set   */
/* web_dn="".                                                   */
/****************************************************************/
web_dn=,
 "CN('www.YourCompany.com')",
 "O('Your Company')",
 "L('Your City')",
 "SP('Your Full State or Province Name')",
```

```
 "C('Your Country 2 Letter Abbreviation')"

/*************************************************************/
/* The sample web server protection directives supplied by PKI  */
/* use SSLring for the web server's SAF key ring. If you change */
/* the value below, you will need to modify   the "KeyFile"     */
/* directive in the samples/httpd.conf and samples/httpd2.conf  */
/* files when configuring the web server.                       */
/*                                                              */
/* If you already have your webserver configured for SSL and    */
/* are using a SAF key ring (vs a gskkyman keyfile), then set   */
/* web_ring equal to your webserver's SAF key ring name. If you */
/* are using a gskkyman keyfile, then set web_ring="". Note,    */
/* you will have to manually add the PKI CA's certificate to    */
/* the webserver's keyfile/keyring and make the webserver's     */
/* root CA available in an HFS file for download.          @L9C */
/*************************************************************/
web_ring = "SSLring"             /* SAF keyring for web server */

/*************************************************************/
/* You must provide UID and GID values for the user IDs and     */
/* groups being created below                                   */
/*************************************************************/
daemon="PKISRVD"                 /* user ID for PKI daemon     */
daemon_uid="554"                 /* uid for PKI daemon         */
surrog="PKISERV"                 /* user ID for the surrogate  */
surrog_uid="555"                 /* uid for the surrogate id   */

/*************************************************************/
/* pkigroup members are authorized to administer PKI Services   */
/* certificates and certificate requests.  If you know the user */
/* IDs that should be connected to this group, update the       */
/* pkigroup_mem stem variable. If not, you can always connect   */
/* users later.                                                 */
/*                                                              */
/* If you do not wish to have this exec create this group,      */
/* set the group name to ""                                     */
/*                                                              */
/*************************************************************/
pkigroup="PKIGRP"             /* PKI Services Admin group name  */
pki_gid="655"                 /* PKI Services Admin group id    */
pkigroup_mem.0=0       /* Number of pkigroup members to connect */
pkigroup_mem.1=""


/*-------------------------------------*/
/* Part 2 - Questions you must answer */
/*-------------------------------------*/

/*************************************************************/
/* Question 1 -  Restrict the surrogate user ID?                */
/*                                                              */
/* The surrogate user ID is the identity assigned to client     */
/* processes when requesting certificate services. The          */
/* RESTRICTED attribute can be assigned to this ID to limit the */
/* resources available to this user should the user ID be       */
/* hijacked by an unfriendly client (hacker). We recommend      */
/* that you run the surrogate this way. However, this probably  */
/* will cause additional setup work. If you want the RESTRICTED */
/* attribute assigned now, set restrict_surrog=1. Note, you     */
```

```
/* can always do this at some later time.                        */
/*****************************************************************/
restrict_surrog=0

/*****************************************************************/
/* Question 2 -  Need PKI Services to generate the key pair      */
/*               for its certificates?                           */
/*                                                               */
/* You have the option of choosing PKI Services to generate the */
/* key pair for the certificates using PKCS#11 APIs. In order    */
/* for the daemon to generate the key pair, you need to specify */
/* the TokenName keyword in pkiserv.conf and the CRYPTOZ class   */
/* must be activated, the profiles must be defined and the       */
/* daemon needs to have appropriate access.                      */
/*                                                               */
/* If you don't need the key generation capability, leave        */
/* key_gen = 0, otherwise set key_gen = 1.                 @L9A */
/*                                                               */
/*****************************************************************/
key_gen = 0                                                /*@L9A*/

/*****************************************************************/
/* If you set key_gen=1 above, this exec will activate the       */
/* CRYPTOZ class, create profiles in this class and permit       */
/* the PKI Services daemon to access them. You may also have     */
/* a RACF group for authorized PKCS11 token users. The daemon    */
/* ID would need to be added to this group.                      */
/*                                                               */
/* Set the following variables as needed:                        */
/*                                                               */
/* cryptoz_profile_so - Profile to be created in the CRYPTOZ     */
/*     class to control the security officer (SO) role           */
/* cryptoz_profile_user - Profile to be created in the CRYPTOZ   */
/*     class to control the user role                            */
/* cryptoz_grp - Group name for authorized PKCS11 token users    */
/*                                                         @L9A */
/*****************************************************************/
cryptoz_profile_so = 'SO.'||daemon||'.*'                  /*@L9A*/
cryptoz_profile_user = 'USER.'||daemon||'.*'              /*@L9A*/
cryptoz_grp = ''                                          /*@L9A*/

/*****************************************************************/
/* Question 3 - Use ICSF?                                        */
/*                                                               */
| /* There are several possible choices for generation and    @LBC*/
/* protection of your CA's private key:                          */
/*                                                               */
/* - Generate the RSA key using software and retain it as a      */
/*   software key. This is the default. (Option 0)               */
/* - Generate the RSA key using software then store the key in   */
/*   ICSF. (Option 1)                                      @LAC*/
/*                                                               */
/* - Generate the RSA key through ICSF using the PCI             */
/*   cryptographic coprocessor (PCICC) then store the key in     */
/*   ICSF. (Option 2)                                      @LAC*/
/*                                                               */
/* - Generate the DSA key using software and retain it as a      */
/*   software key. This key cannot be saved in ICSF. (Option 3) */
/*                                                               */
/* - Generate the NIST Elliptic Curve Cryptography (ECC) key     */
```

```
                    /*   using software and retain it as a software key.           */
                    /*   (Option 4)                                          @LAA*/
                    /*                                                             */
                    /* - Generate the Brainpool ECC key using software and retain  */
                    /*   it as a software key. (Option 5)                    @LAA*/
                    /*                                                             */
|                   /* - Generate the NIST Elliptic Curve Cryptography (ECC)       */
|                   /*   key using the Crypto Express 3 cryptographic coprocessor  */
|                   /*   and store the key in the ICSF PKA data set (PKDS).        */
|                   /*   (Option 6)                                          @LBA*/
|                   /*                                                             */
|                   /* - Generate the Brainpool Elliptic Curve Cryptography (ECC)  */
|                   /*   key using the Crypto Express 3 cryptographic coprocessor  */
|                   /*   and store the key in the ICSF PKA data set (PKDS).        */
|                   /*   (Option 7)                                          @LBA*/
                    /*                                                             */
                    /* Notes:                                                      */
                    /*                                                             */
|                   /* - For options 1, 2, 4, 5, 6, 7, ICSF must be configured for */
|                   /*   support and running. Additionally, for option 2, a        */
|                   /*   PCICC must be present and operational. For options 6      */
|                   /*   and 7, a Crypto Express 3 cryptographic coprocessor       */
|                   /*   must be operational and configured with the PKDS.   @LBC*/
                    /*                                                             */
                    /* - Options 0 and 2 are the only way to generate an RSA       */
                    /*   private key larger than 1024 bits.                  @LAC*/
                    /*                                                             */
|                   /* - If option 2, 6, or 7 is selected, the certificate and     */
|                   /*   private key will not be backed up by this exec.     @LBC*/
                    /*                                                             */
                    /* - If you select option 0, you can always migrate the key    */
                    /* to ICSF later (recommended).  However, if you wish to use   */
                    /* the PCICC, you must select that option now.                 */
                    /*                                                             */
                    /* Select the option desired by setting key_type=0, 1, 2, 3, 4, */
|                   /* 5, 6, or 7.                                         @LBC*/
                    /*************************************************************/
                    key_type=0

                    /*************************************************************/
|                   /* If you set key_type=1, 2, 6, or 7 above, you will need to   */
                    /* restrict access to the CA's private key. Unless you indicate */
                    /* otherwise, this exec will activate the CSFKEYS class,       */
                    /* create a profile in the CSFKEYS class to protect the CA's   */
                    /* private key, and permit the PKI Services daemon to use it.  */
                    /*                                                     @LBC*/
|                   /* If you are already using ICSF, then you may have profiles in */
|                   /* the CSFSERV class protecting ICSF services. If you set      */
|                   /* key_type=1, 2, 4, 5, 6, or 7 above, the  PKI Services       */
|                   /* daemon would need access to the following profiles:         */
|                   /*  CSFDSV, CSFDSG, CSFPKI, CSFENC, CSFDEC                     */
|                   /*                                                     @D8C*/
|                   /* Also, the PKI Services surrogate ID would need access       */
|                   /* to the following profiles that cover the services used      */
|                   /* during the SSL/TLS handshake processing depending on        */
|                   /* the cipher and certificates being utilized:                 */
|                   /*  CSFENC, CSFDEC, CSFPKD, CSFPKE, CSFPKI, CSFDSG, CSFDSV,    */
|                   /*  CSFCKI, CSFCKM, CSF1DVK, CSF1GAV, CSF1GKP, CSFIQA, CSF1PKS, */
|                   /*  CSF1PKV, CSF1TRD, CSF1TRC                                  */
|                   /*                                                     @D8C*/
```

```
                  /* You may already have a RACF group for authorized ICSF users. */
                  /* If you specify the group name for csfusers_grp, the daemon   */
                  /* and surrogate IDs will be connected to the group, and the    */
                  /* group permitted to the profiles. Otherwise the two IDs will  */
                  /* be directly permitted to the profiles.                       */
                  /*                                                      @D8C*/
                  /*                                                              */
                  /* Set the following variables as needed:                       */
                  /*                                                              */
                  /* csfkeys_profile - Profile to be created in the CSFKEYS class */
                  /*     Set the value to '' if you don't want the profile        */
                  /* csfserv_profile - Profile to be created in the CSFSERV class */
                  /*     e.g., 'CSF*'                                             */
                  /* csfusers_grp - Group name for authorized ICSF users          */
                  /*     e.g., 'ICSFUGRP'                                         */
                  /****************************************************************/
                  csfkeys_profile='IRR.DIGTCERT.CERTIFAUTH.*'

                  if (key_type=1 | key_type=2 | key_type=4 |,
                      key_type=5 | key_type=6 | key_type=7 |,
                      key_gen=1)
                    then csfserv_profile='CSF*'
                    else csfserv_profile=''                              /* @D8C*/

                  csfusers_grp=''

                  /****************************************************************/
                  /* Question 4 - Back up your private key?                       */
                  /*                                                              */
                  /* The exec will prompt you to enter a pass phrase to encrypt a */
                  /* backup copy of your CA's certificate and private key.        */
                  /* Caution, the text you enter at the prompt WILL be displayed  */
                  /* at the terminal. Backup is highly recommended.  If you do not*/
                  /* wish to back up your CA's certificate and private key to a   */
                  /* pass phrase encrypted data set, set key_backup=0. The back up*/
                  /* may be done later if the key is not stored in ICSF.          */
                  /*                                                              */
                  /* Note, back up is not performed if the CA certificate was not */
                  /* created by this exec or if you specified key_type 2, 6, or 7 */
                  /* above.                                                 @LBC*/
                  /* *************************************************************/
                  key_backup=1

                  /****************************************************************/
                  /* Question 5 - Set up z/OS UNIX level security?                */
                  /*                                                              */
                  /* z/OS UNIX may be set up to operate with a higher level of    */
                  /* security than traditional UNIX. While we recommend this, it  */
                  /* difficult to set up. You may want to defer this until later. */
                  /*                                                              */
                  /* If you don't want to set up UNIX security now, leave         */
                  /* unix_sec=0.                                                  */
                  /*                                                              */
                  /* If you already have UNIX level security established and wish */
                  /* to continue it, set unix_sec=1.                              */
                  /*                                                              */
                  /* If you don't have UNIX level security established and wish   */
                  /* to establish it now, set unix_sec=2. Note additional manual  */
                  /* configuration probably will be required. This can be done    */
                  /* by adding, removing, updating members of the two stem        */
```

```
/* variables below. The pgmcntl_dsn stem contains the data set  */
/* names of load libraries that need program control. The       */
/* bpx_userid stem contains the user IDs of your server daemons.*/
/* (These need access to BPX.SERVER and BPX.DAEMON in the       */
/* FACILITY class.) Again, you can defer this until later by    */
/* leaving unix_sec=0                                           */
/*--------------------------------------------------------------*/
unix_sec=0
pgmcntl_dsn.0=8  /* Number of program controlled data sets below @L2C*/
pgmcntl_dsn.1="'CEE.SCEERUN'"
pgmcntl_dsn.2="'CBC.SCLBDLL'"
pgmcntl_dsn.3="'SYS1.SIEALNKE'"  /* Common LINKLIST PDSE dataset @L2A*/
pgmcntl_dsn.4="'SYS1.CSSLIB'"                                 /* @L2C*/
pgmcntl_dsn.5="'TCPIP.SEZALOAD'"                              /* @L2C*/
pgmcntl_dsn.6="'SYS1.LINKLIB'"                                /* @L2C*/
pgmcntl_dsn.7="'CSF.SCSFMOD0'"                                /* @L2C*/
pgmcntl_dsn.8="'CSF.SCSFMOD1'"                                /* @L2C*/
bpx_userid.0=1 /* Number of additional bpx server ids below */
bpx_userid.1="OMVSKERN"

/*--------------------------------------------------------------*/
/* Question 6 -  Use DB2 as the repository for the Issued       */
/*               Certificate List (ICL) and Object Store?       */
/*                                                              */
/* In the default configuration, PKI Services uses VSAM data    */
/* sets as the repository for the issued certificate list (ICL) */
/* and the object store.  VSAM is an included feature of z/OS.  */
/*                                                              */
/* Optionally, DB2 for z/OS may be used as the repository for   */
/* the ICL and the object store.  DB2 for z/OS is a separately  */
/* purchased and installed product which provides additional    */
/* data stability features and query capabilities beyond those  */
/* provided by VSAM.  PKI Services may use DB2 for z/OS as the  */
/* repository for the ICL and object store if DB2 is running on */
/* the same system as the PKI Services daemon.                  */
/*                                                              */
/* If you wish to use DB2 for z/OS as the repository for the    */
/* ICL and object store, set db2_repos to 1; otherwise, leave   */
/* db2_repos set to 0.                                          */
/*                                                        @LCA */
/*--------------------------------------------------------------*/
db2_repos = 0                                          /* @LCA */

/*--------------------------------------------------------------*/
/* If you set db2_repos = 1 above, this exec will permit the    */
/* PKI Services daemon access to the DB2 Resource Recovery      */
/* Services access facility (RRSAF).                            */
/*                                                              */
/* To grant this access, the name of the local DB2 subsystem    */
/* that will be used to provide the repository must be          */
/* provided.  This is a name of up to 4 characters.  You can    */
/* obtain this name from your DB2 for z/OS systems programmer.  */
/* Change the value of db2_subsys below to contain the name of  */
/* the local DB2 subsystem.                                     */
/*                                                        @LCA */
/*--------------------------------------------------------------*/
db2_subsys = 'DSN9'                                    /* @LCA */
/*---------------------------------*/
/* Part 3 - Things you can change  */
/*---------------------------------*/
```

```
/*****************************************************************/
/* Label of the CA certificate that is the superior (signer) of */
/* the PKI Services CA, if self sign leave blank                 */
/*****************************************************************/
signing_ca_label = ""                                    /*@L4A*/

/*****************************************************************/
/* This exec will record results to a log data set if desired.  */
/* the name of the data set is specified below. If you do not   */
/* want log data set recording, set log_dsn="" (Not recommended)*/
/*****************************************************************/
if (ca_domain = "") then          /* If no CA Domain...     @L4A*/
   log_dsn="IKYSETUP.LOG"         /* Under your ID              */
else                              /* Else use CA Domain     @L4A*/
   log_dsn=ca_domain_trunc||".IKYSETUP.LOG" /* CA Domain qulified @L4A*/
/*****************************************************************/
/* Note IKYCVSAM, the sample JCL to create VSAM datasets and    */
/* pkiserv.conf expect the object store and ICL datasets to     */
/* have PKISRVD as their high level qualifier.                  */
/* Changing either "daemon" or "vsamhlq" will                   */
/* require making the same change to IKYCVSAM and pkiserv.conf  */
/*****************************************************************/
vsamhlq=daemon     /* HLQ for VSAM data sets. Same as daemon ID */

/*****************************************************************/
/* The following variables are used to create the certificates  */
/* for thw Certificate Authority (CA) and the web server. If    */
/* the default settings are used, these certificates will       */
/* expire as follows:                                           */
/*    - The CA certificate will expire 20 years from the date   */
/*      when this exec is run.                                  */
/*    - The web server SSL certificate will expire 5 years from */
/*      the date when this exec is run.                         */
/*                                                              */
/* You may tailor the expiration dates to suit your own need in */
/* one of two ways, which are described below.  If you decide   */
/* to tailor the expiration dates for these certificates, make  */
/* sure that the web server SSL certificate's expiration date   */
/* does not exceed the expiration date of the CA certificate;   */
/* failure to ensure this will result in the web server SSL     */
/* certificate being added as NOTRUST in the RACF database.     */
/*    - Method 1: You can change the number of years used to    */
/*      set the CA certificate and web server SSL certificate.  */
/*      If you choose this method, leave the code segment       */
/*      labelled "Method 2" below commented out of the exec,    */
/*      and alter the values for the "ca_exyears" and           */
/*      "web_exyears" to set the number of years for the CA     */
/*      certificate and the web server SSL certificate,         */
/*      respectively.  Ensure that the value for "ca_exyears"   */
/*      equals or exceeds the  value for "web_exyears".         */
/*    - Method 2: You can hardcode the expiration date to be    */
/*      used for the CA certificate and the web server SSL      */
/*      certificate.   If you choose this method, comment out   */
/*      the code segment labelled "Method 1" below, uncomment   */
/*      the code segment labelled "Method 2" below, and alter   */
/*      the values for the "ca_expires" and "web_expires" to    */
/*      set the expiration dates for the CA certificate and the */
/*      web server SSL certificate, respectively.  Ensure that  */
/*      the expiration date given in "ca_expires" is later      */
```

```
/*      than the date specified in "web_expires".               */
/********************************************************* @D9A */

/*
   Method 1: Set the CA certificate and web server
   certificate expiration dates to a number of years in
   the future.  If using Method 2 to set the expiration
   dates, comment out the code in this section.              @D9A
 */
ca_exyears  = 20                                        /* @D9C */
web_exyears = 5                                         /* @D9C */

if (datatype(ca_exyears,'N')^=1) then do
  say "The CA certificate expiration years value",
      ca_exyears "is not valid."                        /* @D9A */
  return 8                                              /* @D9A */
end                                                     /* @D9A */
if (datatype(web_exyears,'N')^=1) then do
  say "The webserver SSL certificate expiration years value",
      web_exyears "is not valid."                       /* @D9A */
  return 8                                              /* @D9A */
end

today = date('S')                                       /* @D9A */

exday = ca_exyears * 10000 + today                      /* @D9A */
ca_expires = substr(exday,1,4)"/"||,
             substr(exday,5,2)"/"||,
             substr(exday,7,2)                          /* @D9A */

exday = web_exyears * 10000 + today                     /* @D9A */
web_expires = substr(exday,1,4)"/"||,
              substr(exday,5,2)"/"||,
              substr(exday,7,2)                         /* @D9A */
/* End of Method 1.                                        @D9A */

/*
   Method 2: Set the CA certificate and web server
   certificate expiration dates to a specific future
   date.  If using Method 1 to set the expiration
   dates, leave this code segment commented out of
   the exec.  If using this method, remove the
   comment characters from the following two
   instructions, change the dates to the desires
   values if necessary, and comment out all
   instructions in Method 1.                                @D9A
 */
/* ca_expires  = "2030/01/01" */                        /* @D9A */
/* web_expires = "2015/01/01" */                        /* @D9A */
/* End of Method 2.                                        @D9A */
web_label = "SSL Cert"           /* Label for web server cert  */

if (ca_domain = "") then         /* If no CA Domain...     @L4A*/
   ca_ring="CAring"              /* keyring name for PKI Srvs  */
else                             /* Else use CA Domain     @L4A*/
   ca_ring="CAring."||ca_domain  /* CA Domain qualified    @L4A*/

/*************************************************************/
/* You can select the size (in bits) of your CA's private key.  */
/* Acceptable values depend upon the value that is used for     */
```

```
/* "key_type" in Question 2.                                      */
/*    - When key_type=0 or key_type=2, the acceptable range is  */
/*      512 to 4096.                                             */
/*    - When key_type=1 or key_type=3, the acceptable range is  */
/*      512 to 1024.                                             */
|  /*    - When key_type=4 or Key_type=6,                          */
/*      the acceptable set of values are 192,                    */
/*      224, 256, 384, and 521 (yes, that is really 521).        */
|  /*    - When key_type=5 or key_type=7,                          */
/*      the acceptable set of values are 160,                    */
/*      192, 224, 256, 320, 384, and 512.                        */
/* The default value to use for the key size is given in the     */
/* "ca_keysize" value below.  This default value has been        */
/* chosen with the assumption that an RSA or a DSA key will be   */
/* used in the CA certificate (when "key_type" is set to 0, 1,   */
/* 2, or 3 in Question 3 above).  Please notice that the value   */
/* of "ca_keysize" must be changed if an ECC key (key_type of 4 */
/* or 5) is selected in Question 3.  Also, please notice that    */
/* to use an RSA key size greater than 1024 bits, you must       */
/* select key_type=0 or key_type=2 in Question 3 above.     @LBC*/
/*****************************************************************/
ca_keysize="1024"

/*****************************************************************/
/* Data set to contain the backup copy of the CA certificate     */
/* and private key. (pass phrase encrypted PKCS#12 format)       */
/*****************************************************************/
if (ca_domain = "") then    /* If no CA Domain...            @L4A*/
   backup_dsn = "'"||daemon||".KEY.BACKUP.P12BIN'"       /* @L5C*/
else                        /* Else use CA Domain            @L4A*/
   backup_dsn = "'"||daemon||"."||ca_domain_trunc||,
                ".KEY.BACKUP.P12BIN'"                    /* @L5C*/
                            /* CA Domain qualify backup dsn @L4A*/

/*****************************************************************/
/* Data set to contain the exported copy of the CA certificate   */
/* (DER encoded). This is to assist the backup process.     @L9C*/
/*                                                                */
/*****************************************************************/
if (ca_domain = "") then    /* If no CA Domain...            @L4A*/
   cacert_dsn = "'"||daemon||".CACERT.DERBIN'"
else                        /* Else use CA Domain            @L9C*/
   cacert_dsn = "'"||daemon||"."||ca_domain_trunc||".CACERT.DERBIN'"
                            /* CA Domain qualify export dsn @L9C*/

/*****************************************************************/
/* Data set to contain the exported copy of the webserver's      */
/* root certificate (DER encoded). This is to be OPUT to an      */
/* HFS file later to enable easy downloading by clients.    @L9A*/
/*****************************************************************/
if (ca_domain = "") then    /* If no CA Domain...            @L9A*/
   export_dsn = "'"||daemon||".WEBROOT.DERBIN'"
else                        /* Else use CA Domain            @L9A*/
   export_dsn = "'"||daemon||"."||ca_domain_trunc||".WEBROOT.DERBIN'"
                            /* CA Domain qualify export dsn @L9A*/

/*****************************************************************/
/* Data set to contain the backup copy of the RA certificate     */
/* and private key. (pass phrase encrypted PKCS#12 format)       */
/*****************************************************************/
```

```
if (ca_domain = "") then    /* If no CA Domain...           @01A*/
   ra_backup_dsn = "'"||daemon||".RAKEY.BACKUP.P12BIN'"  /* @L5C*/
else                        /* Else use CA Domain           @01A*/
   ra_backup_dsn = "'"||daemon||"."||ca_domain_trunc||,
                   ".RAKEY.BACKUP.P12BIN'"               /* @L5C*/
                        /* CA Domain qualify RA backup dsn @01A*/

/*****************************************************************/
/* This EXEC expects the web server to be set up.  If this is    */
/* not the case, please refer to:                                */
/* z/OS HTTP Server Planning, Installing and Using.              */
/* If the user ID assigned to the IBM HTTP Server Daemon is not */
/* WEBSRV, please update the assignment below.                   */
/*****************************************************************/
webserver="WEBSRV"

/*---------------------------------------------------------------*/
/* End of configurable section                                   */
/*---------------------------------------------------------------*/

parse upper arg "RUN(" runopt ")"

if runopt = '' then
  runopt="NO"
if runopt ^= "YES" & runopt ^= "PROMPT" & runopt ^= "NO" then do
  say "syntax ex 'data-set-name(IKYSETUP)' 'run(yes | no | prompt)'"
  return 8
end
if runopt ^= "YES" & runopt ^= "PROMPT" then
  runopt="NO"

say 'IKYSETUP EXEC invoked ...'
return_code= '0'
max_return_code= '0'
logdata.0=0

if log_dsn ^= "" then do
  say "Allocating log data set" log_dsn "..."
  x = OUTTRAP(MSGS.)
  "FREE FI(IKYLOGDD)"
  "FREE DA("||log_dsn||")"
  "DELETE" log_dsn
  x = OUTTRAP('OFF')
  "ALLOCATE DA("||log_dsn||") FILE(IKYLOGDD) RECFM(V B)" ,
  " LRECL(256) DSORG(PS) BLKSIZE(2560) SP(1,1) TRACKS "
  al_rc= rc
  IF al_rc ^= 0 THEN
    do
      say 'Allocation of log data set failed.'
      return 8
    end
end

call logsay "RUN("runopt") requested on" DATE() 'at' TIME() '...'
if runopt="NO" then
  call logsay "Running in test mode. Commands are not being invoked"


/*****************************************************************/
/* Verify the requested key size, based upon the selected key    */
```

```
/* key type.                                                          */
/*    - When key_type=0 (RSA software generated key) or               */
/*      key_type=2 (RSA PCICC generated key), the acceptable          */
/*      range is 512 to 4096.                                         */
/*    - When key_type=1 (RSA software generated key stored in         */
/*      ICSF) or key_type=3 (DSA software generated key), the         */
/*      acceptable range is 512 to 1024.                              */
/*    - When key_type=4 (NIST ECC software generated key),    @LBC*/
|/*        or key_type=6 (NIST ECC key in PKDS), the           @LBA*/
/*      acceptable set of values are 192, 224, 256, 384, and 521. */
/*    - When key_type=5 (Brainpool ECC software generated key),   */
|/*        or key_type=7 (Brainpool ECC in PKDS),              @LBA*/
/*      the acceptable set of values are 160, 192, 224, 256, 320, */
/*      384, and 512.                                               */
/* If invalid key size is chosen, we will issue message and end   */
/* execution.                                                     @LAC*/
/*********************************************************************/
select                                                          /*@LAA*/
  when key_type = 0 then do                                     /*@LAA*/
    if ca_keysize < 512                                         /*@LAA*/
    then do                                                     /*@LAA*/
      call logsay "Key size cannot be less than 512",
                  "for key type" key_type                       /*@LAA*/
  return 8
  end                                                           /*@LAA*/
  else do                                                       /*@LAA*/
      if ca_keysize > 4096                                      /*@LAA*/
      then do                                                   /*@LAA*/
        call logsay "Key size cannot be greater than",
                    "4096 for key type" key_type                /*@LAA*/
        return 8                                                /*@LAA*/
      end                                                       /*@LAA*/
    end                                                         /*@LAA*/
  end                                                           /*@LAA*/
  when key_type = 1 then do                                     /*@LAA*/
    if ca_keysize < 512                                         /*@LAA*/
    then do                                                     /*@LAA*/
      call logsay "Key size cannot be less than 512",           /*@LAA*/
                  "for key type" key_type                       /*@LAA*/
      return 8                                                  /*@LAA*/
    end                                                         /*@LAA*/
    else do                                                     /*@LAA*/
      if ca_keysize > 1024                                      /*@LAA*/
      then do                                                   /*@LAA*/
        call logsay "Key size cannot be greater than",
                    "1024 for key type" key_type                /*@LAA*/
        return 8                                                /*@LAA*/
      end                                                       /*@LAA*/
    end                                                         /*@LAA*/
  end                                                           /*@LAA*/
  when key_type = 2 then do                                     /*@LAA*/
    if ca_keysize < 512                                         /*@LAA*/
    then do                                                     /*@LAA*/
      call logsay "Key size cannot be less than 512",
                  "for key type" key_type                       /*@LAA*/
      return 8                                                  /*@LAA*/
    end                                                         /*@LAA*/
    else do                                                     /*@LAA*/
      if ca_keysize > 4096                                      /*@LAA*/
      then do                                                   /*@LAA*/
```

```
            call logsay "Key size cannot be greater than",
                        "4096 for key type" key_type            /*@LAA*/
            return 8                                            /*@LAA*/
          end                                                   /*@LAA*/
        end                                                     /*@LAA*/
      end                                                       /*@LAA*/
      when key_type = 3 then do                                 /*@LAA*/
        if ca_keysize < 512                                     /*@LAA*/
        then do                                                 /*@LAA*/
          call logsay "Key size cannot be less than 512",
                      "for key type" key_type                   /*@LAA*/
          return 8                                              /*@LAA*/
        end                                                     /*@LAA*/
        else do                                                 /*@LAA*/
          if ca_keysize > 1024                                  /*@LAA*/
          then do                                               /*@LAA*/
            call logsay "Key size cannot be greater than",
                        "1024 for key type" key_type            /*@LAA*/
            return 8                                            /*@LAA*/
          end                                                   /*@LAA*/
        end                                                     /*@LAA*/
      end                                                      /*@LAA*/
|     when key_type = 4 |,                                      /*@LBC*/
|          key_type = 6 then do                                 /*@LBA*/
        if ca_keysize ^= 192 & ca_keysize ^= 224 &,
           ca_keysize ^= 256 & ca_keysize ^= 384 &,
           ca_keysize ^= 521                                    /*@LAA*/
        then do                                                 /*@LAA*/
          call logsay "Key size must be: 192, 224, 256, 384,",
                      "or 521 for key type" key_type            /*@LAA*/
          return 8                                              /*@LAA*/
        end                                                     /*@LAA*/
      end                                                       /*@LAA*/
|     when key_type = 5 |,                                      /*@LBC*/
|          key_type = 7 then do                                 /*@LBA*/
        if ca_keysize ^= 160 & ca_keysize ^= 192 &,
           ca_keysize ^= 224 & ca_keysize ^= 256 &,
           ca_keysize ^= 320 & ca_keysize ^= 384 &,
           ca_keysize ^= 512                                    /*@LAA*/
        then do                                                 /*@LAA*/
          call logsay "Key size must be: 160, 192, 224, 256,",
                      "320, 384, or 512 for key type",
                      key_type                                  /*@LAA*/
          return 8                                              /*@LAA*/
        end                                                     /*@LAA*/
      end                                                       /*@LAA*/
      otherwise do                                              /*@LAA*/
|       call logsay "Key type must be: 0, 1, 2, 3, 4, 5, 6 or 7"/*@D7C*/
        return 8                                                /*@LAA*/
      end                                                       /*@LAA*/
    end                                                         /*@LAA*/
                                                               /*10@LAD*/

  /*****************************************************************/
  /* Create the daemon and surrogate user IDs using RACF ADDUSER TSO*/
  /* command. Give them an OMVS segment since they will need access */
  /* to UNIX System Services.                                      */
  /*****************************************************************/
```

```
call logsay2 "Creating users and groups ..."
call tsoserv "ADDUSER " daemon "name('PKI Srvs Daemon')",
  " nopassword",
  " omvs(uid("daemon_uid")",
  " assize(256000000)",
  " threads(512))"

if restrict_surrog=1 then   /*@D1C*/
  resattr="restricted"
else
  resattr=""
call tsoserv "ADDUSER " surrog "nopassword",
  resattr,
  " omvs(uid("surrog_uid"))",
  " name('PKI Srvs Surrogate')"

if pkigroup ^= "" then do                                        /* @LCM*/
  call tsoserv "ADDGROUP " pkigroup "OMVS(GID("pki_gid"))"       /* @LCM*/
  do i = 1 to pkigroup_mem.0                                     /* @LCM*/
    call tsoserv "CONNECT" pkigroup_mem.i "GROUP("pkigroup")"    /* @LCM*/
  end                                                            /* @LCM*/
end                                                              /* @LCM*/

/*****************************************************************/
/* Set up the permission needed for the DB2 tables or the VSAM  */
/* data sets.                                                @LCA*/
/*****************************************************************/

/*****************************************************************
 * The DB2 for z/OS Resource Recovery Services access facility
 * (RRSAF) is protected by the RRSAF DSNR class resource that
 * is specific to the DB2 subsystem.  The following command
 * gives READ access to the PKI Services daemon, which allows
 * the daemon the capability of interacting with the DB2
 * subsystem through this access facility.                  @LCA
 *****************************************************************/
if db2_repos=1 then do                                           /* @LCA*/
   call logsay2 "Granting DB2 RRSAF access to the PKI",
                "Services daemon."                               /* @LCA*/
   call tsoserv "RDEFINE DSNR "db2_subsys".RRSAF",
                "UACC(NONE)"                                     /* @LCA*/
   call tsoserv "PERMIT "db2_subsys".RRSAF",
                "CLASS(DSNR) ACCESS(READ)",
                "ID("daemon")"                                   /* @LCA*/
   call tsoserv "SETROPTS CLASSACT(DSNR)"                        /* @LCA*/
end                                                              /* @LCA*/
/*****************************************************************
 * The VSAM data sets created for PKI Services are protected.
 * Grant access to these data sets for the PKI Services daemon
 * and the administrator.                                   @LCA
 *****************************************************************/
else do                                                          /* @LCA*/
   call tsoserv "SETROPTS EGN GENERIC(DATASET)"

   call tsoserv "ADDSD '"vsamhlq".**' UACC(NONE)"
   call tsoserv "PERMIT '"vsamhlq".**' ID("daemon")",
                "ACCESS(ALTER)"                                  /* @D4A*/

   if (vsamhlq ^= daemon) then do                                /* @D4A*/
     call tsoserv "ADDSD '"daemon".**' UACC(NONE)"               /* @D4A*/
```

```
|                      call tsoserv "PERMIT '"daemon".**' ID("daemon")",
|                                   "ACCESS(ALTER)"                          /* @D4A*/
|                    end                                                     /* @D4A*/
                     /*************************************************************
                      * Give the administrators access to the VSAM data sets
                      * identified in the [ObjectStore] section of
                      * the pkiserv.conf file.
                      *************************************************************/
                      call logsay2 "Allowing administrators to access",
|                                  "PKI VSAM databases ..."                  /* @LCC*/
                      call tsoserv "PERMIT '"vsamhlq".**' ID("pkigroup")",
                                   "ACCESS(CONTROL)"
                      call tsoserv "SETROPTS GENERIC(DATASET) REFRESH"
|                  end                                                       /* @LCA*/

                                                                             /*6@LCM*/

                  /*****************************************************************/
                  /* In order to create and sign digital certificates for others  */
                  /* you need to define or import in RACF a Certificate Authority */
                  /* certificate and associated private key.                       */
                  /* This is done using the RACF RACDCERT GENCERT command.         */
                  /*****************************************************************/
                  if ca_dn ^= "" then do
                    call logsay2 "Creating the CA certificate ..."
                    if signing_ca_label = "" then                             /*@L4A*/
                      certcmd = "RACDCERT GENCERT CERTAUTH SUBJECTSDN("ca_dn")",
                       " WITHLABEL('"ca_label"') NOTAFTER(DATE("ca_expires"))",
                       " SIZE("ca_keysize")"
                     else                                                     /*@L4A*/
                      certcmd = "RACDCERT GENCERT CERTAUTH SUBJECTSDN("ca_dn")",
                       " WITHLABEL('"ca_label"')",
                       " SIGNWITH(CERTAUTH LABEL('"signing_ca_label"'))",
                       " NOTAFTER(DATE("ca_expires"))",
                       " SIZE("ca_keysize")"                                 /*@L4A*/
                    if key_type=1 & key_backup=0 then
                      certcmd= certcmd || " ICSF"
                    else if key_type=2 then
                      certcmd= certcmd || " PCICC"
                    else if key_type=3 then
                      certcmd =certcmd || " DSA"
                    else if key_type=4 then
                      certcmd =certcmd || " NISTECC"                         /*@D5A*/
                    else if key_type=5 then
                      certcmd =certcmd || " BPECC"                           /*@D6A*/
|                   else if key_type=6 then
|                     certcmd =certcmd || " NISTECC(PKDS)"                 /*@LBA*/
|                   else if key_type=7 then
|                     certcmd =certcmd || " BPECC(PKDS)"                   /*@LBA*/

                    call tsoserv certcmd

|                     if key_type^=2 & key_type^=6 & key_type^=7 & key_backup=1
|                     then do                                              /* @D8C*/
                  /*****************************************************************/
                  /* Export certificate and key to PKCS#12 dataset                 */
                  /*****************************************************************/
                      say ""
                      say "Enter a passphrase to protect the key. You will need"
                      say " this value later if you need to restore the key."
```

```
    say ""
    say "Attention, the value will be displayed in the screen:"
    parse pull pp
    call logsay2 "Backing up the CA certificate ..."
    certcmd = "RACDCERT CERTAUTH EXPORT(LABEL('"ca_label"'))",
     " DSN("backup_dsn") FORMAT(PKCS12DER)",
     " PASSWORD('"pp"')"

    call tsoserv certcmd
  end

  if key_type=1 & key_backup=1 then do
/*****************************************************************/
/* If ICSF was requested and key backup, reload the certificate */
/* to get the key migrated to ICSF                               */
/*****************************************************************/
    call logsay2 "Migrating the CA's private key to ICSF ..."
    certcmd = "RACDCERT CERTAUTH ADD("backup_dsn")",
     " PASSWORD('"pp"') ICSF"

    call tsoserv certcmd
  end

end /* ca_dn ^= "" */
/*****************************************************************/
/* Mark the CA certificate as HIGHTRUST so HostIdMappings        */
/* are honored                                                   */
/*****************************************************************/
call logsay2 "Marking CA certificate as HIGHTRUST ..."
certcmd = "RACDCERT CERTAUTH ALTER(LABEL('"ca_label"')) HIGHTRUST"
call tsoserv certcmd

/*****************************************************************/
/* The CA certificate must be saved to a data set to assist the  */
/* backup process.                                        @L9C   */
/*****************************************************************/
call logsay2 "Saving the CA certificate to a data set ..."/* @L9C*/
certcmd = "RACDCERT CERTAUTH EXPORT(LABEL('"ca_label"'))",
  " DSN("cacert_dsn") FORMAT(CERTDER)"                    /* @L9C*/
call tsoserv certcmd

if (ra_label ^= "") then do                               /* @L4A*/
   /*************************************************************/
   /* Creating RA Certifcate                                    */
   /*************************************************************/
   call logsay2 "Creating the RA certificate ..."           /* @L4A*/
   certcmd = "RACDCERT ID("daemon") GENCERT SUBJECTSDN("ra_dn")",
     " KEYUSAGE(HANDSHAKE) SIGNWITH(CERTAUTH LABEL('"ca_label"'))",
     " NOTAFTER(DATE("ca_expires")) WITHLABEL('"ra_label"')" /* @L4A*/
   call tsoserv certcmd                                   /* @L4A*/

   /*************************************************************/
   /* Backing up RA Certifcate                                  */
   /*************************************************************/
   call logsay2 "Backing up RA certificate ..."             /* @L4A*/
   certcmd = "RACDCERT ID("daemon") EXPORT(LABEL('"ra_label"'))",
     " DSN("ra_backup_dsn") FORMAT(PKCS12DER)",
     " PASSWORD('"pp"')"                                    /* @L4A*/
   call tsoserv certcmd                                   /* @L4A*/
end
```

```
                                                          /* @L4A*/
/*****************************************************************/
/* The CA/RA certificate must be placed in a key ring so that    */
/* PKI Services can access it.                                   */
/*****************************************************************/
call logsay2 "Creating the PKI Services keyring ..."
call tsoserv "RACDCERT ADDRING("ca_ring") ID("daemon")"
call tsoserv "RACDCERT ID("daemon") CONNECT(CERTAUTH",
 " LABEL('"ca_label"')",
 " RING("ca_ring") USAGE(PERSONAL) DEFAULT) "
if (ra_label ^= "") then                                  /* @L4A*/
   call tsoserv "RACDCERT ID("daemon") CONNECT(LABEL('"ra_label"')",
   " RING("ca_ring") USAGE(PERSONAL))"                    /* @D2C*/

/*****************************************************************/
/* Create the certificate for the webserver signed by your new CA  */
/*****************************************************************/

if web_dn ^= "" then do
  call logsay2 "Creating the Webserver SSL certificate and keyring ..."
  call tsoserv "RACDCERT GENCERT ID("webserver") SIGNWITH(CERTAUTH",
   " LABEL('"ca_label"'))",
   " WITHLABEL('"web_label"') SUBJECTSDN("web_dn")",
   " NOTAFTER(DATE("web_expires"))"

/*****************************************************************/
/* Add the webserver's certificate to the webserver's RACF (SAF)*/
/* key ring                                                 @L9C */
/*****************************************************************/
  call tsoserv "RACDCERT ADDRING("web_ring") ID("webserver")"
  call tsoserv "RACDCERT  ID("webserver") CONNECT(ID("webserver")",
    " LABEL('"web_label"') RING("web_ring") USAGE(PERSONAL) DEFAULT)"
end /* web_dn ^= "" */

/*****************************************************************/
/* Add the PKI CA certificate to the webserver's RACF (SAF)     */
/* key ring                                                 @L9C */
/*****************************************************************/
if web_ring ^= "" then
  call tsoserv "RACDCERT  ID("webserver") CONNECT(CERTAUTH",
     " LABEL('"ca_label"') RING("web_ring"))"

/*****************************************************************/
/* The webserver's root CA certificate must be saved to a data  */
/* set so that it may be OPUT to an HFS file for download   @L9A*/
/*****************************************************************/
/* If webserver certificate is generated in this exec and       */
/* it is issued by the self-signed PKI CA, ie. the self-signed  */
/* PKI CA is the webserver's root CA, export it for OPUT    @L9A*/

if web_dn ^= "" & signing_ca_label = "" then do           /*@L9A*/
  call logsay2 "Saving the webserver's root CA certificate to a ",
               "data set for OPUT ..."                    /*@L9A*/

  certcmd = "RACDCERT CERTAUTH EXPORT(LABEL('"ca_label"'))",
            " DSN("export_dsn") FORMAT(CERTDER)"          /*@L9A*/
  call tsoserv certcmd
end
else                                                      /*@L9A*/
```

```
  /* Need to manaully export the webserver's root
     CA certificate                                          @L9A*/
  call logsay2 "You need to manually export the webserver's ",
               "root CA certificate."                     /*@L9A*/

if unix_sec = 0 then do
/*****************************************************************/
/* Not setting up z/OS UNIX higher security. However, the        */
/* daemon does need access to one server service. So, if the     */
/* daemon user ID is not uid 0, then it must be given read       */
/* access to FACILITY class profile BPX.SERVER                   */
/*****************************************************************/
  if strip(daemon_uid,L,'0') ^= "" then do /* if daemon not uid 0 */
    call logsay2 "Giving" daemon "access to BPX.SERVER ..."
    call tsoserv "RDEFINE FACILITY BPX.SERVER"
    call tsoserv "PERMIT BPX.SERVER CLASS(FACILITY)",
        " ID("daemon") ACCESS(READ)"
  end
end
else do
  call logsay2 "Setting up or modifying z/OS UNIX security ..."
  if unix_sec = 2 then do
/*****************************************************************/
/* Set up z/OS UNIX to operate with a higher level of            */
/* security than traditional UNIX, by defining BPX.SERVER and    */
/* BPX.DAEMON classes.                                           */
/*****************************************************************/
    call tsoserv "RDEFINE FACILITY BPX.SERVER"
    call tsoserv "RDEFINE FACILITY BPX.DAEMON"
    do i = 1 to bpx_userid.0
      call tsoserv "PERMIT BPX.SERVER CLASS(FACILITY)",
        " ID("bpx_userid.i") ACCESS(READ)"
      call tsoserv "PERMIT BPX.DAEMON CLASS(FACILITY)",
        " ID("bpx_userid.i") ACCESS(READ)"
    end
  end


/*****************************************************************/
/* To use the higher level of security, you need to establish    */
/* RACF program control and enable the PKI Services daemon       */
/* user ID and webserver daemon user ID to access protected      */
/* UNIX daemon services.                                         */
/*****************************************************************/
  call tsoserv "PERMIT BPX.SERVER CLASS(FACILITY) ID("daemon")",
   " ACCESS(READ)"
  call tsoserv "PERMIT BPX.DAEMON CLASS(FACILITY) ID("daemon")",
   " ACCESS(READ)"
  call tsoserv "PERMIT BPX.SERVER CLASS(FACILITY) ID("webserver")",
   " ACCESS(UPDATE)"
  call tsoserv "PERMIT BPX.DAEMON CLASS(FACILITY) ID("webserver")",
   " ACCESS(READ)"

  if unix_sec = 2 then do
/*****************************************************************/
/* Set the PKI Services daemon and DLLs up for program control   */
/*****************************************************************/
    call tsoserv "RDEFINE PROGRAM * UACC(NONE)"
    do i = 1 to pgmcntl_dsn.0
     call tsoserv "RALTER PROGRAM * ADDMEM("pgmcntl_dsn.i"//NOPADCHK)",
```

```
      " UACC(READ)"
    end
    call tsoserv "SETROPTS WHEN(PROGRAM)"
  end
  call tsoserv "PERMIT * CLASS(PROGRAM)",
   " ID("surrog") ACCESS(READ)"
  call tsoserv "SETROPTS WHEN(PROGRAM) REFRESH"
end /* unix_sec ^= 0 */

/*************************************************/
/* Allow the daemon to be a certificate authority */
/*************************************************/
call logsay2 "Allowing the PKI Services daemon to act as a CA ..."
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.GENCERT"
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.LISTRING"
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.LIST"
call tsoserv "PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY)",
 " ID("daemon") ACCESS(CONTROL)"
call tsoserv "PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)",
 " ID("daemon") ACCESS(READ)"
call tsoserv "PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)",
 " ID("daemon") ACCESS(READ)"

/*****************************************************************/
/* If the webserver cert and keyring were created in this exec,  */
/* Allow the webserver to access its keyring                     */
/*****************************************************************/
if web_dn ^= "" then do                                    /* @D4A*/
  call logsay2 "Allowing the Webserver to access its keyring ..."
  call tsoserv "PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)",
   " ID("webserver") ACCESS(READ)"
  call tsoserv "PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)",
   " ID("webserver") ACCESS(READ)"
end /* web_dn ^= "" */                                     /* @D4A*/

/*****************************************************************/
/* Permit the webserver daemon User ID to switch identity to the */
/* surrogate Id                                                  */
/*****************************************************************/

call logsay2 "Allowing the Webserver to switch identity to "surrog" ..."
call tsoserv "SETROPTS CLASSACT(SURROGAT)"
call tsoserv "RDEFINE SURROGAT BPX.SRV."surrog
call tsoserv "PERMIT BPX.SRV."surrog" CLASS(SURROGAT)",
 " ID("webserver") ACCESS(READ)"
call tsoserv "SETROPTS RACLIST(SURROGAT) REFRESH"

                                                           /* @D8D*/
/*************************************************/
/* Allow the daemon to use ICSF                  */
/*************************************************/
if csfkeys_profile ^= '' | csfserv_profile ^= '' then do   /* @D8M*/
  call logsay2 "Allowing the PKI Services daemon to use ICSF ..."
  call tsoserv "SETROPTS  GENERIC(CSFKEYS CSFSERV)"
  call tsoserv "SETROPTS  GENERIC(CSFKEYS CSFSERV) REFRESH"
                                                           /* @D8D*/
  if csfusers_grp ^= '' then do                            /* @D8M*/
    call tsoserv "CONNECT" daemon "GROUP(" csfusers_grp ")"  /* @D8M*/
    call tsoserv "CONNECT" surrog "GROUP(" csfusers_grp ")"  /* @D8M*/
  end                                                      /* @D8M*/
```

```
|
|                     if csfkeys_profile ^= '' then do
|                       call tsoserv "RDEFINE CSFKEYS" csfkeys_profile "UACC(NONE)"
|                       if csfusers_grp ^= '' then                              /* @D8A*/
|                         call tsoserv "PERMIT" csfkeys_profile "CLASS(CSFKEYS)", /* @D8A*/
|                          " ID("csfusers_grp") ACCESS(READ)"                   /* @D8A*/
|                       else                                                    /* @D8A*/
|                         call tsoserv "PERMIT" csfkeys_profile "CLASS(CSFKEYS)",
|                          " ID("daemon") ACCESS(READ)"
|                       call tsoserv "SETROPTS CLASSACT(CSFKEYS) RACLIST(CSFKEYS)"
|                       call tsoserv "SETROPTS RACLIST(CSFKEYS) REFRESH"
|                     end
|                     if csfserv_profile ^= '' then do
|                       call tsoserv "RDEFINE CSFSERV" csfserv_profile "UACC(NONE)"
|                       if csfusers_grp ^= '' then do                           /* @D8A*/
|                         call tsoserv "PERMIT" csfserv_profile "CLASS(CSFSERV)", /* @D8A*/
|                          " ID("csfusers_grp") ACCESS(READ)"                   /* @D8A*/
|                       end                                                     /* @D8A*/
|                       else do                                                 /* @D8A*/
|                         call tsoserv "PERMIT" csfserv_profile "CLASS(CSFSERV)",
|                          " ID("daemon") ACCESS(READ)"
|                         call tsoserv "PERMIT" csfserv_profile "CLASS(CSFSERV)",
|                          " ID("surrog") ACCESS(READ)"
|                       end                                                     /* @D8A*/
|                       call tsoserv "SETROPTS CLASSACT(CSFSERV) RACLIST(CSFSERV)"
|                       call tsoserv "SETROPTS RACLIST(CSFSERV) REFRESH"
|                     end
|                   end                                                        /* @D8D*/
                    if (key_gen=1) then do                                     /*@L9A*/
                    /************************************************/
                    /* Allow the daemon to generate key pairs     @L9A */
                    /************************************************/
                      call logsay2 "Allowing the PKI Services daemon to ",
                      "generate key pairs ..."                                 /*@L9A*/
                      if cryptoz_profile_so ^= '' & cryptoz_profile_user ^= ''
                      then do                                                  /*@L9A*/
                        call tsoserv "SETROPTS CLASSACT(CRYPTOZ)"              /*@L9A*/
                        call tsoserv "SETROPTS GENERIC(CRYPTOZ)"               /*@L9A*/
                        call tsoserv "SETROPTS RACLIST(CRYPTOZ)"               /*@L9A*/
                        call tsoserv "RDEFINE CRYPTOZ",
                             cryptoz_profile_so "UACC(NONE)"                   /*@L9A*/
                        call tsoserv "RDEFINE CRYPTOZ",
                             cryptoz_profile_user "UACC(NONE)"                 /*@L9A*/
                        call tsoserv "PERMIT" cryptoz_profile_so "CLASS(CRYPTOZ)",
                             " ID("daemon") ACCESS(UPDATE)"                    /*@D4C*/
                        call tsoserv "PERMIT" cryptoz_profile_user "CLASS(CRYPTOZ)",
                             " ID("daemon") ACCESS(CONTROL)"                   /*@D4C*/
                        call tsoserv "SETROPTS RACLIST(CRYPTOZ) REFRESH"       /*@L9A*/
                        call tsoserv "SETROPTS GENERIC(CRYPTOZ) REFRESH"       /*@L9A*/
                      end                                                      /*@L9A*/
                      if cryptoz_grp ^= '' then do                             /*@L9A*/
                        call tsoserv "CONNECT" daemon "GROUP(" cryptoz_grp ")" /*@L9A*/
                      end                                                      /*@L9A*/
                    end

                    /************************************************/
                    /* daemon no longer needs access to OCSF        */
                    /************************************************/

                    /*********************************************************
```

```
 * Tie the daemon user ID to PKI Services started procedure
 ************************************************************/
 call logsay2 "Creating the STARTED class profile for the daemon ..."
 call tsoserv "RDEFINE STARTED PKISERVD.* STDATA(USER("daemon"))"
 call tsoserv "SETROPTS CLASSACT(STARTED) RACLIST(STARTED)"
 call tsoserv "SETROPTS RACLIST(STARTED) REFRESH"

/*****************************************************************/
/* Give the surrogate user ID authority to request certificate  */
/* generation functions.                                         */
/*****************************************************************/
call logsay2 "Allowing "surrog" to request certificate functions ..."
call tsoserv "SETR GENERIC(FACILITY)"
/*
 * When a CA Domain is not specified, use the default value for
 * the end user functions generic profile name. When a CA Domain
 * is specified, append a dot/period(".") followed by the CA
 * Domain value to the default value.
 */
if (ca_domain = "") then                                  /* @L4A*/
   profname = "IRR.RPKISERV.**"                           /* @L4A*/
else                                                      /* @L4A*/
   profname = "IRR.RPKISERV.*."||ca_domain_trunc          /* @L4A*/

call tsoserv "RDEFINE FACILITY "||profname                /* @L4C*/
call tsoserv "PERMIT "||profname||" CLASS(FACILITY)",
             " ID("surrog") ACCESS(CONTROL)"              /* @L4C*/

/*****************************************************************/
/* The administrative functions of PKI Services are protected   */
/* by the IRR.RPKISERV.PKIADMIN FACILITY class resource.        */
/* The following commands give UPDATE access to the PKI         */
/* services group to allow them to act on certificate           */
/* requests and issued certificates.                            */
/*****************************************************************/
call logsay2 "Creating the profile to protect PKI Admin functions ..."

/*
 * When a CA Domain is not specified, use the default value for
 * the administrative function profile name. When a CA Domain
 * is specified, append a dot/period(".") followed by the CA
 * Domain value to the default value.
 */
if (ca_domain = "") then                                  /* @L4A*/
   profname = "IRR.RPKISERV.PKIADMIN"                     /* @L4A*/
else                                                      /* @L4A*/
   profname = "IRR.RPKISERV.PKIADMIN."||ca_domain_trunc   /* @L4A*/

call tsoserv "RDEFINE FACILITY "||profname                /* @L4C*/
call tsoserv "PERMIT "||profname||" CLASS(FACILITY)",
             " ID("pkigroup") ACCESS(UPDATE)"             /* @L4C*/
call tsoserv "PERMIT "||profname||" CLASS(FACILITY)",
             " ID("surrog") ACCESS(NONE)"                 /* @L4C*/
call tsoserv "SETROPTS RACLIST(FACILITY) REFRESH"

/*****************************/
/* Done. Now write to the log */
/*****************************/
upper daemon vsamhql export_dsn
call logsay " "
```

```
call logsay "------------------------------------------------"
call logsay "Information needed for PKI Services UNIX set up:"
call logsay "------------------------------------------------"
call logsay " "
call logsay "The daemon user ID is:"
call logsay " " daemon
call logsay " "
call logsay "The VSAM high level qualifier is:"
call logsay " " vsamhlq
call logsay,
"This is needed for the [ObjectStore] section in pkiserv.conf"
call logsay " "
call logsay "The PKI Services' DER encoded certificate is in data set:"
call logsay " " cacert_dsn                                  /*@L9C*/
call logsay " "
if export_dsn ^= "" then do                                 /*@L9A*/
  call logsay "The webserver's DER encoded root "
  call logsay "CA certificate is in data set:"
  call logsay " " export_dsn                                /*@L9A*/
  call logsay,
  "This must be OPUT to /var/pkiserv/cacert.der with ",
  "the BINARY option"
  call logsay " "
end                                                         /*@L9A*/
else do                                                     /*@L9A*/
  call logsay "You need to find the webserver's root CA ",
              "certificate and export it manually."         /*@L9A*/
  call logsay " "                                           /*@L9A*/
end
call logsay "The fully qualified PKI Services' SAF keyring is:"
call logsay " " daemon"/"ca_ring
call logsay,
"This is needed for the [SAF] section in pkiserv.conf"
if ra_label ^= "" then do                                   /*@L4A*/
  call logsay " "                                          /*2@L4A*/
  call logsay "The label of the PKI Services' RA certificate is:"
  call logsay " " ra_label                                  /*@L4A*/
  call logsay,                                              /*@L4A*/
  "This is needed for the [SAF] section in pkiserv.conf"    /*@L4A*/
end                                                         /*@L4A*/
call logsay " "
if ca_dn ^= "" then do
  call logsay "The PKI Services CA DN is:"
  call norm_dn ca_dn
  call logsay " " dn
  call logsay "The suffix must match the LDAP suffix in slapd.conf"
end
else
  call logsay "CA certificate not created by this exec"
call logsay " "

if ra_dn ^= "" then do                                       /*@L4A*/
  call logsay "The PKI Services RA DN is:"                   /*@L4A*/
  call norm_dn ra_dn                                         /*@L4A*/
  call logsay " " dn                                        /*2@L4A*/
  call logsay "The suffix must match the LDAP suffix in slapd.conf"
end                                                          /*@L4A*/
else                                                         /*@L4A*/
  call logsay "RA certificate not created by this exec"      /*@L4A*/
call logsay " "                                              /*@L4A*/
```

```
call logsay,                                                /*2@L4A*/
"The recommended location for the pkiserv.conf and pkiserv.tmpl is:"
if ca_domain = "" then                                      /*@L4A*/
  call logsay "  /etc/pkiserv"                             /*@L4A*/
else                                                        /*@L4A*/
  call logsay "  /etc/pkiserv/"ca_domain                   /*@L4A*/
call logsay " "                                             /*@L4A*/

call logsay,                                                /*2@L4A*/
"Set the following environment variables in pkiserv.envars:"
if ca_domain = "" then                                      /*@L4A*/
  call logsay "  _PKISERV_CONFIG_PATH=/etc/pkiserv"         /*@L4A*/
else do                                                     /*2@L4A*/
  call logsay "  _PKISERV_CA_DOMAIN="ca_domain              /*@L4A*/
  call logsay "  _PKISERV_CONFIG_PATH=/etc/pkiserv/"ca_domain
end                                                         /*@L4A*/
call logsay " "                                             /*@L4A*/
call logsay,                                                /*2@L4A*/
"Set the following environment variable in your httpd envvars files:"
if ca_domain = "" then                                      /*@L4A*/
  call logsay "  _PKISERV_CONFIG_PATH=/etc/pkiserv"         /*@L4A*/
else                                                        /*2@L4A*/
  call logsay "  _PKISERV_CONFIG_PATH_"TRANSLATE(ca_domain),
              "=/etc/pkiserv/"ca_domain                     /*@L4A*/
call logsay " "                                             /*@L4A*/
if web_dn ^= "" then do
  call logsay "The webserver's SAF keyring is:"
  call logsay " " web_ring
  call logsay,
  "This is needed for the KeyFile directive in httpd*.conf files"
  call logsay " "
  call logsay "The Webserver's DN is:"
  call norm_dn web_dn
  call logsay " " dn
  call logsay "The left most RDN must be the webserver's fully",
              "qualified domain name"
end
else
  call logsay,
"Webserver certificate and keyring not created. You must add the CA",
"certificate as a 'trusted root' manually"
call logsay " "
if log_dsn ^= "" then do
  x = OUTTRAP(MSGS.)
  'EXECIO' logdata.0 'DISKW IKYLOGDD (FINIS STEM LOGDATA.'
  'FREE FI(IKYLOGDD)'
  x=OUTTRAP('OFF')
  say "Commands complete. Results written to log data set" log_dsn
end
/***********************/
/* Exit                */
/***********************/
say 'The IKYSETUP EXEC has completed.'
Exit max_return_code

/*************************************************************/
/* tsoserv - echo rc and commands and track highest rc       */
/*************************************************************/
tsoserv:
Parse arg cmd
```

```
return_code = 0
skipit= 0
if runopt = "NO" | runopt = "PROMPT" then
  call logsay cmd
if runopt = "PROMPT" then do
  say "Run command (y/n) ?"
  parse pull ans
  if substr(ans,1,1) ^= 'Y' & substr(ans,1,1) ^= 'y' then
    skipit= 1
end
if skipit = 0 then
  if runopt = "YES" | runopt = "PROMPT" then do
    msg_status= MSG('ON')
    x=OUTTRAP('rac_ret.')
    Address TSO cmd
    return_code=rc
    y=OUTTRAP('OFF')
    call logsay 'Return code' return_code 'from->' cmd
    If return_code\=0 then do
      Do j=1 to rac_ret.0
      call logsay rac_ret.j
      end
    end
  end
max_return_code= max(max_return_code,return_code)
return return_code
return 0

/*************************************************************/
/* logsay - echo messages to the terminal and logdata stem   */
/*************************************************************/
logsay:
Parse arg cmd
parse var cmd leftpart " PASSWORD('" pw "') " rightpart
if pw ^= "" then
  cmd= leftpart "PASSWORD('******')" rightpart
say cmd


cmdlen = length(cmd)                                  /* @L5A */
Do Ln = 1 to length(cmd) by 252                       /* @L5A */
  k= logdata.0 + 1
  logdata.k=substr(cmd, Ln, min(252, cmdlen))         /* @L5C */
  logdata.0= k
  cmdlen = cmdlen - min(252, cmdlen)                  /* @L5A */
end                                                   /* @L5A */

return 0

/*************************************************************/
/* logsay2 - echo a blank line before echoing the command    */
/*************************************************************/
logsay2:
Parse arg cmd2
call logsay " "
call logsay cmd2

return 0

/*************************************************************/
```

```
/* norm_dn - transform the RACF dn keywords to an LDAP dn    */
/*************************************************************/
norm_dn:
parse arg in_dn
parse var in_dn q.1 "('" v.1 "')",
                q.2 "('" v.2 "')",
                q.3 "('" v.3 "')",
                q.4 "('" v.4 "')",
                q.5 "('" v.5 "')",
                q.6 "('" v.6 "')",
                q.7 "('" v.7 "')" rest
dns.= ""
do i = 1 to 7
  q= strip(q.i)
  upper q
  if q = "" then
    leave
  if q = "CN" then
    dns.1= "CN=" ||  v.i
  else
  if q = "T" then
    dns.2= "T=" ||  v.i
  else
  if q = "OU" then
    dns.3= "OU=" ||  v.i
  else
  if q = "O" then
    dns.4= "O=" ||  v.i
  else
  if q = "L" then
    dns.5= "L=" ||  v.i
  else
  if q = "SP" then
    dns.6= "ST=" ||  v.i
  else
    dns.7= "C=" ||  v.i
end
dn= ""
do i = 1 to 7
  if dns.i ^= "" then
    if dn = "" then
      dn= dns.i
    else
      dn= dn || "," || dns.i
end
return 0

Parse arg cmd2
call logsay " "
call logsay cmd2

return 0

/*************************************************************/
/* norm_dn - transform the RACF dn keywords to an LDAP dn    */
/*************************************************************/
norm_dn:
parse arg in_dn
parse var in_dn q.1 "('" v.1 "')",
                q.2 "('" v.2 "')",
```

```
                q.3 "('" v.3 "')",
                q.4 "('" v.4 "')",
                q.5 "('" v.5 "')",
                q.6 "('" v.6 "')",
                q.7 "('" v.7 "')" rest
dns.= ""
do i = 1 to 7
  q= strip(q.i)
  upper q
  if q = "" then
    leave
  if q = "CN" then
    dns.1= "CN=" ||  v.i
  else
  if q = "T" then
    dns.2= "T=" ||  v.i
  else
  if q = "OU" then
    dns.3= "OU=" ||  v.i
  else
  if q = "O" then
    dns.4= "O=" ||  v.i
  else
  if q = "L" then
    dns.5= "L=" ||  v.i
  else
  if q = "SP" then
    dns.6= "ST=" ||  v.i
  else
    dns.7= "C=" ||  v.i
end
dn= ""
do i = 1 to 7
  if dns.i ^= "" then
    if dn = "" then
      dn= dns.i
    else
      dn= dn || "," || dns.i
end
return 0

Parse arg cmd2
call logsay " "
call logsay cmd2

return 0
```

Chapter 29. Other code samples

This topic provides code samples for the following files:

- `httpd.conf` and `httpd2.conf`, which contain z/OS HTTP Server directives. (See "z/OS HTTP Server configuration directives.")
- IKYCDB2, which is a sample to build DB2 objects for the PKI Services object store and issued certificate list (ICL). (See "IKYCDB2" on page 548.)
- IKYCVSAM, which is sample IDCAMS JCL to create VSAM data sets (regardless of whether you are using a sysplex or non-sysplex). (See "IKYCVSAM" on page 551.)
- IKYRVSAM, which is sample IDCAMS JCL to add VSAM record-level sharing (RLS) support. IKYRVSAM reallocates your VSAM data sets in preparation for sharing in a sysplex. (See "IKYRVSAM" on page 555.)
- IKYSBIND, which is a sample job to build the DB2 package and plan for the PKI Services object store and issued certificate list (ICL). (See "IKYSBIND" on page 559.)
- IKYVBKUP, which is sample JCL to back up the PKI Services VSAM data sets using the DFSMSdss DUMP utility. (See "IKYVBKUP" on page 560.)
- IKYVREST, which is sample JCL to restore the PKI Services VSAM data sets from a backup taken with the DFSMSdss DUMP utility. (See "IKYVREST" on page 561.)
- PKISERVD, which is a sample procedure to start PKI Services daemon. (See "PKISERVD sample procedure to start PKI Services daemon" on page 562.)

Note: Other important programs are contained in other chapters:

- IKYSETUP (a REXX exec to set up RACF profiles). See Chapter 28, "The IKYSETUP REXX exec," on page 511.
- `pkiserv.envars` (the PKI Services environment variables file). See "The pkiserv.envars environment variables file" on page 510.
- `pkiserv.conf` (the PKI Services configuration file). See Chapter 26, "The pkiserv.conf configuration file," on page 501.

z/OS HTTP Server configuration directives

The following listing might not be identical to the code sample shipped with the product. For the most current sample, see the `httpd.conf` sample z/OS HTTP Server configuration directives in the source directory `/usr/lpp/pkiserv/samples/`.

```
#--------------------------------------------------------------------#
# Licensed Materials - Property of IBM                               #
# 5694-A01                                                           #
# (C) Copyright IBM Corp. 2001,2011                                  #
# Status = HKY7780                                                   #
#                                                                    #
# Change-Activity:                                                   #
#  $L1=PKIS4  , HKY7708, 020429, JWS: PKI Services                   #
#  $L2=PKIS13B, HKY7780, 100628, SSD: PKI Release XIII Updates       #
#                                                                    #
# Change Descriptions:                                               #
#   A - Multiple application support                          @L1A #
#   A - Added support for PKI ActiveX                         @L2A #
#--------------------------------------------------------------------#
# For a secure system, set the default User ID to %%CLIENT%%
```

© Copyright IBM Corp. 2001, 2011

```
UserId     %%CLIENT%%

# SSL support using a SAF keyring
keyfile SSLring SAF
#         OR
# May use a gskkyman key database instead of SAF keyring
#keyfile  /etc/key.kdb

sslmode on
sslport 443
Normalmode on
Protection PublicUser {
        ServerId        PublicUser
        UserID          PKISERV
        Mask            Anyone
}
Protect /PKIServ/public-cgi/*  PublicUser
Protect /PKIServ/ssl-cgi-bin/*  PublicUser
Protect /PKIServ/*  PublicUser
Protect /Customers/public-cgi/* PublicUser
Protect /Customers/ssl-cgi-bin/* PublicUser
Protect /Customers/* PublicUser

Protection AuthenticatedUser {
        ServerId        AuthenticatedUser
        AuthType        Basic
        PasswdFile      %%SAF%%
        UserID          %%CLIENT%%
        Mask            All
}
Protect /PKIServ/ssl-cgi-bin/auth/* AuthenticatedUser
Protect /Customers/ssl-cgi-bin/auth/* AuthenticatedUser

Protection SurrogateUser {
        ServerId        SurrogateUser
        AuthType        Basic
        PasswdFile      %%SAF%%
        UserID          PKISERV
        Mask            All
}
Protect /PKIServ/ssl-cgi-bin/surrogateauth/* SurrogateUser
Protect /Customers/ssl-cgi-bin/surrogateauth/* SurrogateUser

Redirect  /PKIServ/ssl-cgi/*
https://<server-domain-name>/PKIServ/ssl-cgi-bin/*
Redirect  /PKIServ/ssl-cgi/auth/*
https://<server-domain-name>/PKIServ/ssl-cgi-bin/auth/*
Redirect  /PKIServ/ssl-cgi/surrogateauth/*
https://<server-domain-name>/PKIServ/ssl-cgi-bin/surrogateauth/*
Redirect  /Customers/ssl-cgi/*
https://<server-domain-name>/Customers/ssl-cgi-bin/*
Redirect  /Customers/ssl-cgi/auth/*
https://<server-domain-name>/Customers/ssl-cgi-bin/auth/*
Redirect  /Customers/ssl-cgi/surrogateauth/*
https://<server-domain-name>/Customers/ssl-cgi-bin/surrogateauth/*


Redirect  /PKIServ/clientauth-cgi/*
https://<server-domain-name>:1443/PKIServ/clientauth-cgi/*
Redirect  /Customers/clientauth-cgi/*
https://<server-domain-name>:1443/Customers/clientauth-cgi/*

Exec      /PKIServ/public-cgi/*    <application-root>/PKIServ/public-cgi/*
Exec      /PKIServ/ssl-cgi-bin/*   <application-root>/PKIServ/ssl-cgi-bin/*
Exec      /Customers/public-cgi/*  <application-root>/PKIServ/public-cgi/*
Exec      /Customers/ssl-cgi-bin/* <application-root>/PKIServ/ssl-cgi-bin/*
Pass      /PKIServ/cacerts/*       /var/pkiserv/*
Pass      /PKIServ/PKIXEnroll/*    /usr/lpp/pkiserv/ActiveX/PKIXEnroll/*
```

```
Pass      /PKIServ/PKICEnroll/*    /usr/lpp/pkiserv/ActiveX/PKICEnroll/*
AddType  .cer  application/x-x509-user-cert      ebcdic  0.5 # Browser Certificate
AddType  .der  application/x-x509-ca-cert        binary  1.0 # CA Certificate
```

The source of the following sample z/OS HTTP Server configuration directives for your /etc/httpd1443.conf file is /usr/lpp/pkiserv/samples/httpd2.conf.

```
#--------------------------------------------------------------------#
# Licensed Materials - Property of IBM                               #
# 5694-A01                                                           #
# (C) Copyright IBM Corp. 2001,2011                                  #
# Status = HKY7780                                                   #
#                                                                    #
# Change-Activity:                                                   #
#  $L1=PKIS4  , HKY7708, 020429, JWS: PKI Services                   #
#  $L2=PKIS13B, HKY7780, 100628, SSD: PKI Release XIII Updates       #
#                                                                    #
# Change Descriptions:                                               #
#   A - Multiple application support                          @L1A #
#   A - Added support for PKI ActiveX                         @L2A #
#--------------------------------------------------------------------#

# For a secure system, set the default User ID to %%CLIENT%%
UserId     %%CLIENT%%


# SSL support using a SAF keyring
keyfile SSLring SAF
#         OR
# May use a gskkyman key database instead of SAF keyring
#keyfile  /etc/key.kdb

sslmode on
sslport 1443
Normalmode off
SSLClientAuth   strong
SSLX500CARoots local_and_x500
SSLX500Host <ldap-server-name>
SSLX500Port <ldap-port-number>
SSLX500UserID <ldap-distinguished-name>
SSLX500Password <ldap-password>

Protection RenewRevokeUser {
        ServerId        RenewRevokeUser
        AuthType        Basic
        UserID          PKISERV
        SSL_CLIENTAUTH  Client
        Mask            Anyone
}

Protect /PKIServ/clientauth-cgi/*   RenewRevokeUser
Protect /Customers/clientauth-cgi/* RenewRevokeUser
Protect /PKIServ/PKIXEnroll/*       RenewRevokeUser
Protect /PKIServ/PKICEnroll/*       RenewRevokeUser
Protection AuthenticatedAdmin {
        ServerId        AuthenticatedAdmin
        AuthType        Basic

        UserID          %%CERTIF%%
        SSL_CLIENTAUTH  Client
        Mask            Anyone
}
Protect /PKIServ/clientauth-cgi/auth/* AuthenticatedAdmin
Protect /Customers/clientauth-cgi/auth/* AuthenticatedAdmin

Redirect  /PKIServ/public-cgi/*
http://<server-domain-name>/PKIServ/public-cgi/*
Redirect  /PKIServ/ssl-cgi/*
```

```
https://<server-domain-name>/PKIServ/ssl-cgi-bin/*
Redirect  /Customers/public-cgi/*
http://<server-domain-name>/Customers/public-cgi/*
Redirect  /Customers/ssl-cgi/*
https://<server-domain-name>/Customers/ssl-cgi-bin/*

Exec      /PKIServ/clientauth-cgi/* <application-root>/PKIServ/clientauth-cgi-bin/*
Exec      /Customers/clientauth-cgi/* <application-root>/PKIServ/clientauth-cgi-bin/*

Pass      /PKIServ/PKIXEnroll/*       /usr/lpp/pkiserv/ActiveX/PKIXEnroll/*
Pass      /PKIServ/PKICEnroll/*       /usr/lpp/pkiserv/ActiveX/PKICEnroll/*
```

IKYCDB2

IKYCDB2 is a sample to create DB2 objects for the object store and issued certificate list (ICL). IKYCDB2 is a member of SYS1.SAMPLIB.

Note: The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYCDB2.

```
--********************************************************************
--* SAMPLE: IKYCDB2
--*
--* Licensed Materials - Property of IBM
--* 5694-A01
--* Copyright IBM Corp.  2011
--* Status = HKY7780
--*
--********************************************************************
--
-- This sample may be used to create the DB2 database using SPUFI that
-- PKI Services utilizes to store certificate requests and issued
-- certificates.
--
--********************************************************************
--
-- Before using this sample, you may need to make the following
-- modifications:
--
-- 1) Change all the occurences of 'SYSDEFLT' to the storage group you
--    want to contain the PKI Services DB2 tablespaces if SYSDEFLT
--    if SYSDEFLT is not suitable for your installation.
--
-- 2) Change all the occurences of 'MASTERCA' to the package name.
--    The package name should match the first eight characters of
--    the CA domain name.
--
-- 3) Change all the occurences of 'IKYPKIDB' to the database name of
--    your choosing. If you are running multiple PKI CA domains, each
--    domain must have a unique database name.
--
-- 4) Change all the occurences of 'PKISRVD' to the PKI Services
--    daemon user ID. If you used the IKYSETUP sample, the user ID
--    should match the value you specified for the "daemon variable.
--
--********************************************************************
-- Change Activity:
--
--   $L0=PKIS13       HKY7780   100706  PDWFC1: DB2 support
--
-- Change Description:
--
--   A: Initial code                                              @L0A
--
--
```

```
--*********************************************************************
-- If you have already run this sample before, uncomment these
-- statements to drop existing indexes, tables, tablespaces and
-- database.
--
--*********************************************************************
--  ;
--DROP INDEX MASTERCA.TIDAIX;
--DROP INDEX MASTERCA.OREQAIX;
--DROP INDEX MASTERCA.OSTATAIX;
--DROP TABLE MASTERCA.OST;
--DROP INDEX MASTERCA.IREQAIX;
--DROP INDEX MASTERCA.ISTATAIX;
--DROP TABLE MASTERCA.ICL;

--DROP TABLESPACE IKYPKIDB.OSTSPACE;
--DROP TABLESPACE IKYPKIDB.ICLSPACE;

--DROP DATABASE IKYPKIDB;

--COMMIT;
--
--*********************************************************************
--          Create the database
--*********************************************************************
CREATE DATABASE IKYPKIDB STOGROUP SYSDEFLT;
--
--*********************************************************************
--          Create the TABLESPACE for the ObjectStore table
--
--*********************************************************************
CREATE TABLESPACE OSTSPACE IN IKYPKIDB
     LOCKSIZE       ROW
     SEGSIZE        4
     PCTFREE        0
     BUFFERPOOL     BP32K
     USING STOGROUP SYSDEFLT
     PRIQTY         144400
;
--*********************************************************************
--          Create the TABLESPACE for the ICL table
--
--*********************************************************************
CREATE TABLESPACE ICLSPACE IN IKYPKIDB
     LOCKSIZE       ROW
     SEGSIZE        4
     PCTFREE        0
     BUFFERPOOL     BP32K
     USING STOGROUP SYSDEFLT
     PRIQTY         144400
;
--*********************************************************************
--         Create OBJECTSTORE table and index
--         Table name must be <package name>.OST
--
--*********************************************************************
CREATE TABLE MASTERCA.OST(
      RECORD_KEY                    BINARY(4)         NOT NULL,
      RECORD_STATE                  BINARY(4)         NOT NULL,
      REQDATA_LEN                   INTEGER           NOT NULL,
      REQUESTOR_NAME                VARCHAR(32)               ,
      TRANS_ID                      CHAR(24)          NOT NULL,
      COMMENT                       VARCHAR(64)               ,
      ISSUED_TIME                   TIMESTAMP         NOT NULL,
      LAST_CHANGE_TIME              TIMESTAMP         NOT NULL,
      TEMPLATE_NICKNAME             VARCHAR(8)                ,
      SERIAL_NUM                    BINARY(4)                 ,
```

```
      REQDATA                      VARBINARY(32512)  NOT NULL,
      PRIMARY KEY (RECORD_KEY)
      )
 IN IKYPKIDB.OSTSPACE
 ;
 CREATE UNIQUE INDEX MASTERCA.RKEYIX   ON MASTERCA.OST (RECORD_KEY);
 CREATE        INDEX MASTERCA.TIDAIX   ON MASTERCA.OST (TRANS_ID);
 CREATE        INDEX MASTERCA.OREQAIX  ON MASTERCA.OST (REQUESTOR_NAME);
 CREATE        INDEX MASTERCA.OSTATAIX ON MASTERCA.OST (RECORD_STATE,
                                                       REQDATA_LEN,
                                                       REQUESTOR_NAME);

--*********************************************************************
--         Create ICL table and index
--         Table name must be <package name>.ICL
--
--*********************************************************************
CREATE TABLE MASTERCA.ICL(
      SERIAL_NUM                   BINARY(4)         NOT NULL,
      CERT_STATE                   BINARY(4)         NOT NULL,
      CERT_LEN                     INTEGER           NOT NULL,
      REQUESTOR_NAME               VARCHAR(32)       NOT NULL,
      REVOKE_DATE                  TIMESTAMP                 ,
      INVALID_DATE                 TIMESTAMP                 ,
      REVOKE_REASON                INTEGER                   ,
      COMMENT                      VARCHAR(64)               ,
      ISSUED_TIME                  TIMESTAMP         NOT NULL,
      LAST_CHANGE_TIME             TIMESTAMP         NOT NULL,
      TEMPLATE_NICKNAME            VARCHAR(8)                ,
      OBFUS_PW                     VARCHAR(32)               ,
      PROCESS_FLAGS                BINARY(4)                 ,
      KEYID                        BINARY(20)                ,
      CRLDP_NUM                    INTEGER                   ,
      EXPIRE_EPOCH_DAYS            INTEGER           NOT NULL,
      EXPIRE_DATE                  TIMESTAMP         NOT NULL,
      KU_DIGTSIG                   BINARY(1)                 ,
      KU_NONRPU                    BINARY(1)                 ,
      KU_KEYENC                    BINARY(1)                 ,
      KU_DATAENC                   BINARY(1)                 ,
      KU_KEYAGR                    BINARY(1)                 ,
      KU_CRTSGN                    BINARY(1)                 ,
      KU_CRLSGN                    BINARY(1)                 ,
      KU_ENCONLY                   BINARY(1)                 ,
      KU_DECONLY                   BINARY(1)                 ,
      EKU_SEVAUTH                  BINARY(1)                 ,
      EKU_CLIAUTH                  BINARY(1)                 ,
      EKU_CODESGN                  BINARY(1)                 ,
      EKU_EMLPROT                  BINARY(1)                 ,
      EKU_TMESTMP                  BINARY(1)                 ,
      EKU_OCSPSGN                  BINARY(1)                 ,
      EKU_MSSCLNON                 BINARY(1)                 ,
      PREV_SERIAL_NUM              BINARY(4)                 ,
      SUBJ_DN                      VARCHAR(1024)     NOT NULL,
      X509CERT                     VARBINARY(10240)  NOT NULL,
      PRIMARY KEY (SERIAL_NUM)
      )
 IN IKYPKIDB.ICLSPACE
 ;
CREATE UNIQUE INDEX MASTERCA.SERIX    ON MASTERCA.ICL (SERIAL_NUM);
CREATE        INDEX MASTERCA.IREQAIX  ON MASTERCA.ICL (REQUESTOR_NAME);
CREATE        INDEX MASTERCA.ISTATAIX ON MASTERCA.ICL (CERT_STATE,
                                                       CERT_LEN,
                                                       REQUESTOR_NAME);

COMMIT;

--*********************************************************************
```

```
--         Grant privileges
--
--*********************************************************************
GRANT EXECUTE ON PACKAGE MASTERCA.* TO PKISRVD;

GRANT DBADM ON DATABASE IKYPKIDB TO PKISRVD;

COMMIT;
```

IKYCVSAM

IKYCVSAM contains sample IDCAMS JCL to create VSAM data sets. IKYCVSAM is installed as a member of SYS1.SAMPLIB.

Use IKYCVSAM if you are creating VSAM data sets for the first time, regardless of whether you intend to use Parallel Sysplex support. However, if you intend to use Parallel Sysplex support, execute the IKYRVSAM job *after* this job to add VSAM record-level sharing (RLS) support. (See "IKYRVSAM" on page 555.)

Note: The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYCVSAM.

```
//IKYCVSAM JOB <job card parameters>
//*********************************************************************
//*  SAMP:      IKYCVSAM                                              *
//*                                                                   *
//*    Licensed Materials - Property of IBM                           *
//*    5694-A01                                                       *
//*    Copyright IBM Corp. 2001, 2011                                 *
//*    Status = HKY7780                                               *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*  This sample JCL may be used to create the VSAM data sets         *
//*  PKI Services utilizes to store certificate requests and          *
//*  issued certificates.                                             *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*  Caution: This is neither a JCL procedure nor a complete job.     *
//*  Before using this job step, you will have to make the following  *
//*  modifications:                                                   *
//*                                                                   *
//*  1) Change the job card to meet your system requirements.         *
//*                                                                   *
//*  2) If you wish to change the data set qualifiers from the        *
//*     default value change all occurrences of "PKISRVD.VSAM"        *
//*     to a preferred value. If you choose to modify this value, be  *
//*     be sure to also modify the sample configuration file          *
//*     appropriately(/etc/pkiserv/pkiserv.conf). If you are using    *
//*     multiple CA Domains, IBM recommends using the first eight     *
//*     characters of the CA Domain as one of the data set            *
//*     qualifiers.                                                   *
//*                                                                   *
//*  3) If you are using VSAM record level sharing (RLS), perform     *
//*     the following steps:                                          *
//*                                                                   *
//*     a) Replace the VOL(vvvvvv) statements in the DEFKSDS step     *
//*        with STORCLAS(class-name) where class-name is the name of  *
//*        the storage class defined for VSAM RLS.                    *
//*                                                                   *
//*     b) Remove the VOL(vvvvvv) statements from the DEFALTDX step.  *
//*                                                                   *
//*     c) Remove all the SPANNED and CISIZE statements.              *
//*                                                                   *
//*     If not using VSAM RLS, change all occurrences of vvvvvv to    *
//*     the VOLSER value appropriate for the system this job is to be *
```

```
//*     run on. Do not remove the SPANNED and CISIZE statements.      *
//*                                                                   *
//*                                                                   *
//*  4) If you wish to change the default userid to own the VSAM      *
//*     data set, change the OWNER(PKISRVD) operand to the userid you *
//*     want to own the data sets. If you choose to modify this value *
//*     ensure you have modified the sample setup REXX exec (IKYSETUP)*
//*     to account for this change.                                   *
//*                                                                   *
//*  5) If you wish to change either the primary or secondary record  *
//*     allocation sizes for either the OST or ICL datasets from the  *
//*     default value, update the RECORDS(50 50) operands on the      *
//*     DEFINE CLUSTER or DEFINE ALTERNATE INDEX commands.            *
//*                                                                   *
//*  **Note, do not change any of the numeric values other than       *
//*       CYL or TRK                                                  *
//*-------------------------------------------------------------------*
//* Change Activity:                                                  *
//*                                                                   *
//*    $L1=PKIS3   HKY7707 020314 PDJWS1: VSAM RLS               @L1A*
//*    $P1=MG00719 HKY7707 020416 PDJWS1: VSAM RLS 2             @P1A*
//*    $L2=MG01176 HKY7708 020826 PDJWS1: VSAM scaling           @L2A*
//*    $P2=MG01346 HKY7708 021022 PDJWS1: JCL errors             @P2A*
//*    $L3=PKIS7   HKY7730 050228 PDTCG1: Multi-CA Support       @L3A*
|   //*    $P3=MG06654 HKY7730 051130 PDTCG1: Missing Qualifier      @P3A*
|   //*    $P4=MG15386 HKY7780 100927 PDTCG1: Update CISIZE values   @P4A*
//*                                                                     *
//* Change Description:                                               *
//*                                                                   *
//*    C: Added STORCLAS instructions, LOG. Removed VOLUME DDs    @L1C*
//*    D: Removed FILE(VOLUME) statements                         @P1A*
//*    C: Added more alt indexes and changed allocation parms     @L2A*
//*    C: Removed VOL keywords from ALTERNATEINDEX statements.    @P2A*
//*       Removed DD statements from BLDINDEX step. Added         @P2A*
//*       IEBGENER step to remove hardcoded binary zeros          @P2A*
//*    C: Updated prolog with CA Domain information               @L3A*
//*    C: Updated DEFALTDX step to use PKISRVD.VSAM.OST.AIX.IX       *
//*       dataset name instead of the former dataset name of         *
//*       PKISRVD.VSAM.AIX.IX.                                    @P3A*
|   //*    C: Updated CISIZE (CISZ) values on DEFINE CLUSTER             *
|   //*       statements to minimize contention.                   @P4A*
//*                                                                     *
//*-------------------------------------------------------------------*
//*
//*-------------------------------------------------------------------*
//* Delete existing clusters, paths, alt indexes
//*-------------------------------------------------------------------*
//DELCLUST EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
    DELETE -
        PKISRVD.VSAM.OST -
        CLUSTER -
        PURGE -
        ERASE
    DELETE -
        PKISRVD.VSAM.ICL -
        CLUSTER -
        PURGE -
        ERASE
    IF MAXCC LT 9 THEN SET MAXCC = 0
/*
//*-------------------------------------------------------------------*
//* Define KSDS                                                       *
//*-------------------------------------------------------------------*
//DEFKSDS  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
    DEFINE CLUSTER -
(NAME(PKISRVD.VSAM.OST) -
        VOL(vvvvvv) -
        RECSZ(1024 32756) -
```

```
          INDEXED -
          NOREUSE -
          KEYS(4 0) -
          SHR(2) -
          CYL(3,1) -
          LOG(NONE) -
          OWNER(PKISRVD) ) -
        DATA -
          (NAME(PKISRVD.VSAM.OST.DA) -
          CISZ(4096) -
          SPANNED) -
        INDEX -
          (NAME(PKISRVD.VSAM.OST.IX))

    DEFINE CLUSTER -
        (NAME(PKISRVD.VSAM.ICL) -
        VOL(vvvvvv) -
        RECSZ(1024 32756) -
        INDEXED -
        NOREUSE -
        KEYS(4 0) -
        SHR(2) -
        CYL(3,1) -
        LOG(NONE) -
        OWNER(PKISRVD) ) -
      DATA -
        (NAME(PKISRVD.VSAM.ICL.DA) -
        CISZ(4096) -
        SPANNED) -
      INDEX -
        (NAME(PKISRVD.VSAM.ICL.IX))
/*
//*-------------------------------------------------------------------*
//* Repro record of all binary zeros into KSDS                        *
//*-------------------------------------------------------------------*
//MKZEROS  EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   *

//SYSUT2   DD   DSN=&&GENTMP,UNIT=SYSDA,DISP=(,PASS),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=640),SPACE=(TRK,(1,1))
//SYSIN    DD   *
 GENERATE MAXFLDS=4,MAXLITS=80
 RECORD FIELD=(20,X'00000000000000000000000000000000000000000000',,1),
        FIELD=(20,X'00000000000000000000000000000000000000000000',,21),
        FIELD=(20,X'00000000000000000000000000000000000000000000',,41),
        FIELD=(20,X'00000000000000000000000000000000000000000000',,61)
/*
//REPROKSD EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSDATA  DD   DSN=*.MKZEROS.SYSUT2,DISP=(OLD,DELETE)
//SYSIN    DD   *
   REPRO INFILE(SYSDATA) -
      OUTDATASET(PKISRVD.VSAM.OST)
   REPRO INFILE(SYSDATA) -
      OUTDATASET(PKISRVD.VSAM.ICL)
/*
//*-------------------------------------------------------------------*
//* Define ALTERNATE INDEX and PATH                                   *
//*-------------------------------------------------------------------*
//DEFALTDX EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.OST.AIX) -
        RELATE(PKISRVD.VSAM.OST)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(24 44) ) -
      DATA -
        (NAME(PKISRVD.VSAM.OST.AIX.DA)) -
      INDEX -
```

```
        (NAME(PKISRVD.VSAM.OST.AIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.PATH) -
         PATHENTRY(PKISRVD.VSAM.OST.AIX))
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.OST.STATAIX) -
        RELATE(PKISRVD.VSAM.OST)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(40 4) ) -
      DATA -
        (NAME(PKISRVD.VSAM.OST.STATAIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.OST.STATAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.STATUS) -
         PATHENTRY(PKISRVD.VSAM.OST.STATAIX))
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.ICL.STATAIX) -
        RELATE(PKISRVD.VSAM.ICL)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(40 4) ) -
      DATA -
        (NAME(PKISRVD.VSAM.ICL.STATAIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.ICL.STATAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.ICL.STATUS) -
         PATHENTRY(PKISRVD.VSAM.ICL.STATAIX))
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.OST.REQAIX) -
        RELATE(PKISRVD.VSAM.OST)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(32 12) ) -
      DATA -
        (NAME(PKISRVD.VSAM.OST.REQAIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.OST.REQAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.REQUESTR) -
         PATHENTRY(PKISRVD.VSAM.OST.REQAIX))
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.ICL.REQAIX) -
        RELATE(PKISRVD.VSAM.ICL)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(32 12) ) -
      DATA -
        (NAME(PKISRVD.VSAM.ICL.REQAIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.ICL.REQAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.ICL.REQUESTR) -
         PATHENTRY(PKISRVD.VSAM.ICL.REQAIX))
/*
//*-------------------------------------------------------------------*
//* BUILD ALTERNATE INDEX                                             *
//*-------------------------------------------------------------------*
//BLDINDEX EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.AIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.STATAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
        OUTDATASET(PKISRVD.VSAM.ICL.STATAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.REQAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
```

```
       OUTDATASET(PKISRVD.VSAM.ICL.REQAIX)
/*
//*-------------------------------------------------------------------*
//* Print out the cluster                                             *
//*-------------------------------------------------------------------*
//PRTCLUST EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
   PRINT -
       INDATASET(PKISRVD.VSAM.OST) CHAR
   PRINT -
       INDATASET(PKISRVD.VSAM.ICL) CHAR
/*
```

IKYRVSAM

IKYRVSAM contains sample IDCAMS JCL to migrate the PKI Services VSAM data sets to support VSAM record-level sharing (RLS) when you intend to use Parallel Sysplex support. IKYRVSAM is installed as a member of SYS1.SAMPLIB.

Execute this job *after* executing the IKYCVSAM job. This job renames the VSAM data sets created by IKYCVSAM and copies their contents to newly allocated RLS data sets.

Note: The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYRVSAM.

```
//IKYRVSAM JOB <job card parameters>
//*********************************************************************
//*  SAMP:      IKYRVSAM                                              *
//*                                                                   *
//*    Licensed Materials - Property of IBM                           *
//*    5694-A01                                                       *
//*    (C) Copyright IBM Corp. 2002, 2006                             *
//*    Status = HKY7730                                               *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*  This sample JCL may be used to reallocate the VSAM data sets     *
//*  in a storage class acceptable to VSAM record level sharing (RLS).*
//*  This is a prerequisite to using PKI Services SYSPLEX support.    *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*  Caution: This is neither a JCL procedure nor a complete job.     *
//*  Before using this job step, you will have to make the following  *
//*  modifications:                                                   *
//*                                                                   *
//*  1) Change the job card to meet your system requirements.         *
//*                                                                   *
//*  2) Change the STORCLAS statements to provide the name of the     *
//*     storage class defined for use with VSAM RLS.                  *
//*                                                                   *
//*  3) This job assumes you are using the default VSAM data set      *
//*     names (all have high level qualifiers "PKISRVD.VSAM"). If     *
//*     you have changed these data set names, you will need to       *
//*     modify the source data set names in the ALTER                 *
//*     statements of the RENAMEDS step. If you are using             *
//*     multiple CA Domains, IBM recommends using the first eight     *
//*     characters of the CA Domain as one of the data set            *
//*     qualifiers.                                                   *
//*                                                                   *
//*  4) This job creates destination data sets with the same default  *
//*     names as the source data sets. (The source data sets are      *
//*     renamed.) If you wish to use different destination data set   *
//*     names, you will need to modify the data set names in all      *
//*     steps except the RENAMEDS step. If you modify these names,    *
```

```
//*     be sure to also modify your configuration file               *
//*     appropriately(/etc/pkiserv/pkiserv.conf). If you are using   *
//*     multiple CA Domains, IBM recommends using the first eight    *
//*     characters of the CA Domain as one of the data set           *
//*     qualifiers.                                                  *
//*                                                                  *
//*  5) This job renames the source data sets to begin with high     *
//*     level qualifiers "PKISRVD.OLDVSAM". If you wish to change    *
//*     these names, you will need to do so in the RENAMEDS and      *
//*     and REPROCL steps. If you are using multiple CA Domains,     *
//*     IBM recommends using the first eight characters of the CA    *
//*     Domain as one of the data set qualifiers.                    *
//*                                                                  *
//*  6) If you wish to change either the primary or secondary space  *
//*     allocation sizes for either the OST or ICL datasets from the *
//*     default value, update the CYL or TRK operands on the         *
//*     DEFINE CLUSTER or DEFINE ALTERNATE INDEX commands.           *
//*                                                                  *
//*  **Note, do not change any of the numeric values other than      *
//*        CYL or TRK                                                *
//*------------------------------------------------------------------*
//* Change Activity:                                                 *
//*                                                                  *
//*    $L0=PKIS3   HKY7707 020314 PDJWS1: VSAM RLS                   *
//*    $P1=MG00719 HKY7707 020416 PDJWS1: VSAM RLS 2              @P1A*
//*    $L1=MG01176 HKY7708 020826 PDJWS1: VSAM scaling            @L1A*
//*    $P2=MG01346 HKY7708 021022 PDJWS1: JES3 JCL error          @P2A*
//*    $P3=MG01521 HKY7708 030106 PDBRW1: Incorrect source names  @P3A*
//*    $L2=PKIS7   HKY7730 050228 PDTCG1: Multi-CA Support        @L2A*
//*                                                                  *
//* Change Description:                                              *
//*                                                                  *
//*    C: Removed SPANNED, CISIZE, and FILE(VOLUME) statements    @P1A*
//*    C: Added more alt indexes and changed allocation parms     @L1A*
//*    C: Removed DD statements from BLDINDEX step                @P2A*
//*    C: Correct souce dataset names for ICL alternate indexes   @P3A*
//*    C: Updated prolog with CA Domain information               @L2A*
//*    C: - Updated RENAMEDS step to use PKISRVD.VSAM.OST.AIX.IX      *
//*         dataset name and conditionally handle the former dataset  *
//*         name (PKISRVD.VSAM.AIX.IX).                              *
//*       - Updated DEFALTDX step to use PKISRVD.VSAM.OST.AIX.IX     *
//*         dataset name instead of the former dataset name of       *
//*         PKISRVD.VSAM.AIX.IX.                                     *
//*       - Updated DEFKSDS to have a line continuation character    *
//*         after the CYL parameters.                              @P4A*
//*------------------------------------------------------------------*
//*
//*------------------------------------------------------------------*
//* Rename source clusters, alternate indexes and PATH               *
//*------------------------------------------------------------------*
//RENAMEDS  EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
    ALTER -
        PKISRVD.VSAM.OST -
        NEWNAME(PKISRVD.OLDVSAM.OST)
    ALTER -
        PKISRVD.VSAM.OST.* -
        NEWNAME(PKISRVD.OLDVSAM.OST.*)
    ALTER -
        PKISRVD.VSAM.OST.AIX.* -
        NEWNAME(PKISRVD.OLDVSAM.OST.AIX.*)
    ALTER -
        PKISRVD.VSAM.ICL -
        NEWNAME(PKISRVD.OLDVSAM.ICL)
    ALTER -
        PKISRVD.VSAM.ICL.* -
        NEWNAME(PKISRVD.OLDVSAM.ICL.*)
    ALTER -
        PKISRVD.VSAM.OST.AIX.IX -
        NEWNAME(PKISRVD.OLDVSAM.OST.AIX.IX)
    IF LASTCC EQ 8 THEN
```

```
      DO
        SET MAXCC EQ 0
        ALTER -
            PKISRVD.VSAM.AIX.IX -
            NEWNAME(PKISRVD.OLDVSAM.OST.AIX.IX)
      END
    ALTER -
        PKISRVD.VSAM.OST.STATAIX.* -
        NEWNAME(PKISRVD.OLDVSAM.OST.STATAIX.*)
    ALTER -
        PKISRVD.VSAM.OST.REQAIX.* -
        NEWNAME(PKISRVD.OLDVSAM.OST.REQAIX.*)
    ALTER -
        PKISRVD.VSAM.ICL.STATAIX.* -
        NEWNAME(PKISRVD.OLDVSAM.ICL.STATAIX.*)
    ALTER -
        PKISRVD.VSAM.ICL.REQAIX.* -
        NEWNAME(PKISRVD.OLDVSAM.ICL.REQAIX.*)
/*
//*-------------------------------------------------------------------*
//* Define destination Clusters                                       *
//*-------------------------------------------------------------------*
//DEFKSDS  EXEC PGM=IDCAMS,COND=(8,LE)
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
    DEFINE CLUSTER -
        (NAME(PKISRVD.VSAM.OST) -
        STORCLAS(class-name) -
        RECSZ(1024 32756) -
        INDEXED -
        NOREUSE -
        KEYS(4 0) -
        SHR(2) -
        CYL(3,1) -
        LOG(NONE) -
        OWNER(PKISRVD) ) -
      DATA -
        (NAME(PKISRVD.VSAM.OST.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.OST.IX))

    DEFINE CLUSTER -
        (NAME(PKISRVD.VSAM.ICL) -
        STORCLAS(class-name) -
        RECSZ(1024 32756) -
        INDEXED -
        NOREUSE -
        KEYS(4 0) -
        SHR(2) -
        LOG(NONE) -
        CYL(3,1) -
        OWNER(PKISRVD) ) -
      DATA -
        (NAME(PKISRVD.VSAM.ICL.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.ICL.IX))
/*
//*-------------------------------------------------------------------*
//* Repro source cluster to destination cluster                       *
//*-------------------------------------------------------------------*
//REPROCL EXEC PGM=IDCAMS,COND=(8,LE)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
   REPRO INDATASET(PKISRVD.OLDVSAM.OST) -
      OUTDATASET(PKISRVD.VSAM.OST)
   REPRO INDATASET(PKISRVD.OLDVSAM.ICL) -
      OUTDATASET(PKISRVD.VSAM.ICL)
/*
//*-------------------------------------------------------------------*
//* Define ALTERNATE INDEX AND PATH                                   *
//*-------------------------------------------------------------------*
//DEFALTDX EXEC PGM=IDCAMS,COND=(8,LE)
```

```
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.OST.AIX) -
        RELATE(PKISRVD.VSAM.OST)-
        TRK(5,1) -
        KEYS(24 44) ) -
      DATA -
        (NAME(PKISRVD.VSAM.OST.AIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.OST.AIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.PATH) -
         PATHENTRY(PKISRVD.VSAM.OST.AIX))
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.OST.STATAIX) -
        RELATE(PKISRVD.VSAM.OST)-
        TRK(5,1) -
        KEYS(40 4) ) -
      DATA -
        (NAME(PKISRVD.VSAM.OST.STATAIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.OST.STATAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.STATUS) -
         PATHENTRY(PKISRVD.VSAM.OST.STATAIX))
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.ICL.STATAIX) -
        RELATE(PKISRVD.VSAM.ICL)-
        TRK(5,1) -
        KEYS(40 4) ) -
      DATA -
        (NAME(PKISRVD.VSAM.ICL.STATAIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.ICL.STATAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.ICL.STATUS) -
         PATHENTRY(PKISRVD.VSAM.ICL.STATAIX))
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.OST.REQAIX) -
        RELATE(PKISRVD.VSAM.OST)-
        TRK(5,1) -
        KEYS(32 12) ) -
      DATA -
        (NAME(PKISRVD.VSAM.OST.REQAIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.OST.REQAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.REQUESTR) -
         PATHENTRY(PKISRVD.VSAM.OST.REQAIX))
    DEFINE ALTERNATEINDEX -
       (NAME(PKISRVD.VSAM.ICL.REQAIX) -
        RELATE(PKISRVD.VSAM.ICL)-
        TRK(5,1) -
        KEYS(32 12) ) -
      DATA -
        (NAME(PKISRVD.VSAM.ICL.REQAIX.DA)) -
      INDEX -
        (NAME(PKISRVD.VSAM.ICL.REQAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.ICL.REQUESTR) -
         PATHENTRY(PKISRVD.VSAM.ICL.REQAIX))
/*
//*-------------------------------------------------------------------*
//* BUILD ALTERNATE INDEX                                             *
//*-------------------------------------------------------------------*
//BLDINDEX EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.AIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
```

```
        OUTDATASET(PKISRVD.VSAM.OST.STATAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
        OUTDATASET(PKISRVD.VSAM.ICL.STATAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.REQAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
        OUTDATASET(PKISRVD.VSAM.ICL.REQAIX)
/*
```

IKYSBIND

IKYSBIND is a sample job to create the DB2 package and plan for the object store and issued certificate list (ICL). IKYSBIND is a member of SYS1.SAMPLIB.

Note: The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYSBIND.

```
//IKYSBIND JOB <job parameters>
//*********************************************************************
//* SAMPLE: IKYSBIND
//*
//* Licensed Materials - Property of IBM
//* 5694-A01
//* Copyright IBM Corp.  2011
//* Status = HKY7780
//*
//*********************************************************************
//*
//* This sample job may be used to build the DB2 package and plan for
//* PKI Services.
//*
//*********************************************************************
//*
//* Before running this job, you may need to make the following
//* modifications:
//*
//* 1) Change all the occurences of 'MASTERCA' to the package name.
//*    The package name should match the first eight characters of
//*    the associated CA domain name and match the value used when
//*    creating the DB2 database, tablespaces, tables, and indexes.
//*
//* 2) Change DSN9 to the DB2 subsystem name which PKI Services will
//*    be running under.
//*
//* 3) Change MASTERPL to the plan name of your choosing.
//*
//* 4) Change 'PKISRVD' to the ID you would like to be the owner of
//*    the package and plan.
//*
//*********************************************************************
//* Change Activity:
//*
//*   $L0=PKIS13D      HKY7780   100706  PDWFC1: DB2 support
//*
//* Change Description:
//*
//*   A: Initial code                                          @L0A
//*
//*
//*********************************************************************
//EXECTSO  EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//DBRMLIB  DD DSN=SYS1.CBRDBRM,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

```
//SYSTSIN  DD *
DSN SYSTEM(DSN9)
BIND PACKAGE(MASTERCA)   -
     MEMBER(IKYPDBRM)    -
     QUALIFIER(MASTERCA) -
     VALIDATE(BIND)      -
     ENABLE(RRSAF)       -
     ENCODING(1047)      -
     ISOLATION(RS)       -
     RELEASE(COMMIT)     -
     OWNER(PKISRVD)

BIND PLAN(MASTERPL)            -
     PKLIST(MASTERCA.IKYPDBRM) -
     QUALIFIER(MASTERCA)       -
     ENABLE(RRSAF)             -
     ENCODING(1047)            -
     ACTION(REPLACE)           -
     ACQUIRE(USE)              -
     CURRENTDATA(NO)           -
     DBPROTOCOL(DRDA)          -
     NODEFER                   -
     RELEASE(COMMIT)           -
     VALIDATE(BIND)            -
     OWNER(PKISRVD)
END
```

IKYVBKUP

IKYVBKUP contains sample JCL to back up the PKI Services VSAM data sets using the DFSMSdss DUMP utility. IKYVBKUP is installed as a member of SYS1.SAMPLIB.

Note: The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYVBKUP.

```
//IKYVBKUP JOB <job card parameters>
//*********************************************************************
//*  SAMP:      IKYVBKUP                                              *
//*                                                                   *
//*    Licensed Materials - Property of IBM                           *
//*    5694-A01                                                       *
//*    Copyright IBM Corp. 2009                                       *
//*    Status = HKY7760                                               *
//*                                                                   *
//*********************************************************************
//*                                                                   *
//*  This sample JCL may be used to backup the VSAM data sets         *
//*  PKI Services utilizes to store certificate requests and          *
//*  issued certificates. To ensure data integrity, the PKI Services  *
//*  address space must be stopped before a backup is attempted.      *
//*                                                                   *
//*********************************************************************
/*                                                                    *
//*  Caution: This is neither a JCL procedure nor a complete job.     *
//*  Before using this job step, you will have to make the following  *
//*  modifications:                                                   *
//*                                                                   *
//*  1) Change the job card to meet your system requirements.         *
//*                                                                   *
//*  2) If you are not using the default data set qualifiers,         *
//*     change all occurrences of "PKISRVD.VSAM" to the qualifiers    *
//*     you are using.                                                *
//*                                                                   *
//*  3) Change all occurrences of vvvvvv to a VOLSER value            *
```

```
//*     that contains sufficient free space to contain a complete     *
//*     backup of both of the PKI Services VSAM data set clusters.    *
//*                                                                    *
//*  4) Change the primary and secondary allocation values for the    *
//*     backup dataset to values that will ensure a complete backup   *
//*     of both VSAM data set clusters. Change the xxx value for the  *
//*     primary allocation, and the yyy value for the secondary       *
//*     allocation.                                                    *
//*                                                                    *
//*--------------------------------------------------------------------*
//* Change Activity:                                                   *
//*                                                                    *
//*    $L0=PKIS11M HKY7760 080828 PDTCG1: VSAM Backup job              *
//*                                                                    *
//* Change Description:                                                *
//*    A000000-999999 New sample written for z/OS release 11           *
//*                                                                    *
//*--------------------------------------------------------------------*
//*
//*--------------------------------------------------------------------*
//* Delete existing backup dataset
//*--------------------------------------------------------------------*
//CATALOG  EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=*
//DD1      DD UNIT=3390,VOLUME=SER=vvvvvv,DISP=OLD,SPACE=(TRK,0)
//SYSIN    DD *
    SCRATCH VOL=3390=vvvvvv,DSNAME=PKISRVD.VSAM.BACKUP
/*
//*--------------------------------------------------------------------*
//* Perform a DFSMS/dss Dump of the two PKI Services VSAM clusters
//*--------------------------------------------------------------------*
//BACKUP1  EXEC PGM=ADRDSSU,COND=(8,LT)
//BACKUPDS DD DSN=PKISRVD.VSAM.BACKUP,DISP=(NEW,CATLG,DELETE),
//       SPACE=(CYL,(xxx,yyy)),VOL=SER=(vvvvvv)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DUMP DATASET(INCLUDE(PKISRVD.VSAM.OST -
                         PKISRVD.VSAM.ICL)) -
         OUTDDNAME(BACKUPDS) -
         CANCELERROR         -
         COMPRESS            -
         OPTIMIZE(4)         -
         SPHERE              -
         WAIT(0,0)           -
         SHR
/*
```

IKYVREST

IKYVREST contains sample JCL to restore the PKI Services VSAM data sets from a backup taken with the DFSMSdss DUMP utility. IKYVREST is installed as a member of SYS1.SAMPLIB.

Note: The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYVREST.

```
//IKYVREST JOB <job card parameters>
//********************************************************************
//*  SAMP:      IKYVREST                                              *
//*                                                                   *
//*    Licensed Materials - Property of IBM                           *
//*    5694-A01                                                       *
//*    Copyright IBM Corp. 2009                                       *
//*    Status = HKY7760                                               *
//*                                                                   *
```

```
//*******************************************************************
//*                                                                 *
//*  This sample JCL may be used to restore the PKI Services VSAM   *
//*  data sets from a backup taken with the DFSMS/dss DUMP utility. *
//*  The PKI Services address space must be stopped before running  *
//*  this restore job.                                              *
//*                                                                 *
//*******************************************************************
//*                                                                 *
//*  Caution: This is neither a JCL procedure nor a complete job.   *
//*  Before using this job step, you will have to make the following *
//*  modifications:                                                 *
//*                                                                 *
//*  1) Change the job card to meet your system requirements.       *
//*                                                                 *
//*  2) If you are not using the default data set qualifiers,       *
//*     change all occurrences of "PKISRVD.VSAM" to the qualifiers  *
//*     you are using.                                              *
//*                                                                 *
//*  3) If you changed the default data set name for the BACKUPDS   *
//*     DD in the IKYVBKUP jcl, change the dataset name (DSN) value *
//*     in the BACKUPDS DD statement is this jcl to match the value *
//*     used in the backup jcl (IKYVBKUP).                          *
//*                                                                 *
//*-----------------------------------------------------------------*
//* Change Activity:                                                *
//*                                                                 *
//*    $L0=PKIS11M HKY7760 080828 PDTCG1: VSAM Restore from backup  *
//*                                                                 *
//* Change Description:                                             *
//*    A000000-999999 New sample written for z/OS release 11        *
//*                                                                 *
//*-----------------------------------------------------------------*
//*
//*-----------------------------------------------------------------*
//* Perform a DFSMS/dss RESTORE of the two PKI Services VSAM clusters
//*-----------------------------------------------------------------*
//RESTORE1 EXEC PGM=ADRDSSU
//BACKUPDS DD DSN=PKISRVD.VSAM.BACKUP,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    RESTORE DATASET(INCLUDE(PKISRVD.VSAM.OST,  -
                            PKISRVD.VSAM.ICL)) -
         INDDNAME(BACKUPDS)   -
         CATALOG              -
         CANCELERROR          -
         REPLACEUNCONDITIONAL -
         SPHERE               -
         WAIT(0,0)
/*
```

PKISERVD sample procedure to start PKI Services daemon

PKISERVD is the sample procedure to start PKI Services daemon. The PKI Services daemon runs as a started task. The procedure for this can be found in 'SYS1.PROCLIB' member PKISERVD. (PKISERVD is an alias for IKYSPROC.)

PKISERVD contains the **TZ** (time zone) environment variable, which is the environment variable most likely to change. You need to specify any other environment variables that PKI Services needs in an environment variables file, by default pkiserv.envars. PKISERVD contains FN (file name) and DIR (directory) parameters, to specify the pathname of the environment variables file. You can make any needed changes in PKISERVD, such as updating this pathname.

Guideline: By default, the pathname for the `pkiserv.envars` environment variables file is `/usr/lpp/pkiserv/samples/pkiserv.envars`. If you need to make changes in the environment variables file, you need to copy it from the samples directory to another directory. Specify your environment variables using an environment variables file under the `/etc` directory, for example `/etc/pkiserv/pkiserv.envars`.

The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.PROCLIB member PKISERVD.

```
//*******************************************************************
//*                                                                 *
//*             Licensed Materials - Property of IBM                *
//*             5694-A01                                            *
//*             (C) Copyright IBM Corp.  2001                       *
//*             Status=HKY7706                                      *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//* Procedure for starting the PKI Services Daemon                  *
//*                                                                 *
//*******************************************************************
//PKISERVD PROC REGSIZE=256M,                                        X
//             OUTCLASS='A',                                         X
//             TZ='EST5EDT',                                         X
//             FN='pkiserv.envars',                                  X
//             DIR='/usr/lpp/pkiserv/samples',                       X
//             STDO='1>DD:STDOUT',                                   X
//             STDE='2>DD:STDERR'
//*------------------------------------------------------------------
//GO       EXEC  PGM=IKYPKID,REGION=&REGSIZE,TIME=1440,
//  PARM=('ENVAR("_CEE_ENVFILE=&DIR/&FN","TZ=&TZ") / &STDO &STDE')
//STDOUT   DD  SYSOUT=&OUTCLASS
//STDERR   DD  SYSOUT=&OUTCLASS
//SYSOUT   DD  SYSOUT=&OUTCLASS
//CEEDUMP  DD  SYSOUT=&OUTCLASS
```

Other code samples

Chapter 30. SMF recording

PKI Services produces one SMF record type - type 80. The first 18 bytes of type 80 records represent the standard SMF header without subtypes.

For more information:

1. See *z/OS MVS System Management Facilities (SMF)* for information about how to use SMF.
2. See *z/OS Security Server RACF Macros and Interfaces* and *z/OS Security Server RACF Auditor's Guide* for information about using the RACF SMF data unload utility (IRRADU00) to prepare reports with the RACF report writer.

PKI Services event code

Table 105 describes the SMF80EVT (event code) and SMF80EVQ (event code qualifier) fields for the PKI Services event code. It also lists the SMF80DTP and SMF80DA2 values for the relocate type sections.

Table 105. SMF event code and event code qualifier for PKI Services

Event Dec(Hex)	Command	Code qualifier Dec(Hex)	Description	Relocate type sections
79(4F)	CRL publication	0(0)	Successful publication of revocation information	318, 319, 366, 379, 380, 381, 382, 383, 384, 385, 387
85(55)	SUCCRNEW	0(0)	Successful AutoRenew	318, 319, 341, 342, 346, 358, 363, 373, 391, 408

Relocate section variable data

Table 106 describes the variable data elements of the extended-length relocate section.

Table 106. SMF data elements of the extended-length relocate section for PKI Services

Data type (SMF80TP2) Dec(Hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
318(13E)	1-255	EBCDIC	66, 67, 69, 72, 74, 79	Certificate or CRL serial number
319(13F)	1-255	EBCDIC	66, 67, 69, 72, 74, 79	Certificate or CRL issuer's distinguished name
366(16E)	4	Binary	74	Certificate revocation reason
379(17B)	1-255	EBCDIC	79	CRL issuing distribution point DN
380(17C)	10	EBCDIC	79	CRL's date of issue
381(17D)	8	EBCDIC	79	CRL's time of issue
382(17E)	10	EBCDIC	79	CRL's expiration date
383(17F)	8	EBCDIC	79	CRL's expiration time
384(180)	10	EBCDIC	79	CRL's date of publish
385(181)	8	EBCDIC	79	CRL's time of publish
387(183)	1-1024	EBCDIC	79	CRL's issuing distribution point URI
408(198)	256	EBCDIC	85	AutoRenew Exit path name

© Copyright IBM Corp. 2001, 2011

LDAP directory server requirements

PKI Services typically requires access to an LDAP directory server to store issued certificates and certificate revocation lists. The z/OS LDAP server provided by IBM Tivoli Directory Server for z/OS is preferred but not required. You can use a non-z/OS LDAP server if it can support the objectclasses and attributes PKI Services uses. These are listed in the following table:

Table 107. LDAP objectclasses and attributes that PKI Services sets

End-entity or branch node?	Visible RDN attribute	Objectclasses used	Additional attributes set (other than visible RDN attribute)
Creating a branch node	C=	country	—
Creating a branch node	L=	locality	—
Creating a branch node	O=	organization	—
Creating a branch node	OU=	organizationalUnit	—
Creating a branch node	DC=	domain	none
Creating a branch node	Any supported value other than the preceding	organizationalUnit, and extensibleObject	`ou` (the `ou` value from CreateOUValue in the **LDAP** section of `pkiserv.conf` file)
Creating a user end-entity	unstructuredName or unstructAddress	account, pkiUser, cEPDevice, and extensibleObject	userCertificate, and uid (hardcoded to `NoUid`)
Creating a user end-entity	serialNumber	account, pkiUser, pKCS10Device, and extensibleObject	userCertificate, and uid (hardcoded to `NoUid`)
Creating a user end-entity	DC	domain pkiUser, , and extensibleObject	userCertificate
Creating a user end-entity	dnQualifier	account, pkiUser, uniquelyQualifiedObject, and extensibleObject	userCertificate, and uid (hardcoded to `NoUid`)
Creating a user end-entity	UID	account, pkiUser, and extensibleObject	userCertificate
Creating a user end-entity	Any supported value other than unstructuredName, unstructAddress, serialNumber, DC, dnQualifier and UID	account, pkiUser, and extensibleObject	userCertificate, and uid (hardcoded to `NoUid`)
Creating a CA end-entity	O=	organization, and pkiCA	cACertificate, certificaterevocationlist, and authorityrevocationlist
Creating a CA end-entity	OU=	organizationalUnit, and pkiCA	cACertificate, certificaterevocationlist, and authorityrevocationlist
Creating a CA end-entity	DC	domain, pkiCA and extensibleObject	cACertificate, certificaterevocationlist, and authorityrevocationlist

© Copyright IBM Corp. 2001, 2011

Table 107. LDAP objectclasses and attributes that PKI Services sets (continued)

End-entity or branch node?	Visible RDN attribute	Objectclasses used	Additional attributes set (other than visible RDN attribute)
Creating a CA end-entity	dnQualifier	account, uniquelyQualifiedObject, pkiCA, and extensibleObject	cACertificate, certificaterevocationlist, authorityrevocationlist, and uid (hardcoded to `NoUid`)
Creating a CA end-entity	UID	account, pkiCA, and extensibleObject	cACertificate, certificaterevocationlist, and authorityrevocationlist
Creating a CA end-entity	Any supported value other than O, OU, DC, dnQualifier and UID	account, pkiCA, and extensibleObject	cACertificate, certificaterevocationlist, authorityrevocationlist and uid (hardcoded to `NoUid`)
User end-entity that already exists	unstructuredName or unstructAddress	pkiUser, cEPDevice	userCertificate
User end-entity that already exists	serialNumber	pkiUser, pKCS10Device	userCertificate
User end-entity that already exists	Any supported value other than unstructuredName, unstructAddress, and serialNumber	pkiUser	userCertificate
CA end-entity that already exists	Any supported value	pkiCA	cACertificate, certificaterevocationlist, and authorityrevocationlist
Creating a distribution point CRL end-entity	CN=	commonName and cRLDistributionPoint	certificateRevocationList
Distribution point CRL end-entity that already exists	Any supported value	cRLDistributionPoint	certificateRevocationList

The `R_PKIServ` SAF callable service supports specifying the subject's DN through named fields in the CertPlist. The CGIs invoke the `R_PKIServ` SAF callable service. For more information, see *z/OS Security Server RACF Callable Services*. PKI Services supports the subject's DN fields, plus some additional ones: postal code, street, and mail. They are mapped to LDAP attributes as Table 108 indicates.

Table 108. Relationship of named fields to LDAP attributes and object identifiers

Named field	Visible RDN attribute	OID
CommonName	CN	2.5.4.3
Title	TITLE	2.5.4.12
OrgUnit	OU	2.5.4.11
Org	O	2.5.4.10
Locality	L	2.5.4.7
StateProv	ST	2.5.4.8
Country	C	2.5.4.6
PostalCode	POSTALCODE	2.5.4.17
Street	STREET	2.5.4.9
Email[1]	MAIL	0.9.2342.19200300.100.1.3

Table 108. Relationship of named fields to LDAP attributes and object identifiers (continued)

Named field	Visible RDN attribute	OID
Mail	MAIL[2]	0.9.2342.19200300.100.1.3
EmailAddr	EMAIL	1.2.840.113549.1.9.1
UnstructName	UNSTRUCTUREDNAME	1.2.840.113549.1.9.2
UnstructAddr	UNSTRUCTUREDADDRESS	1.2.840.113549.1.9.8
SerialNumber	SERIALNUMBER	2.5.4.5
DNQualifier	DNQUALIFIER	2.5.4.46
DomainName	DC	0.9.2342.19200300.100.1.25
Uid	UID	0.9.2342.19200300.100.1.1

[1] The use of the field name Email is deprecated; use Mail instead.

[2] When a certificate is created and posted to LDAP, the NotifyEmail value, if specified, is posted as the MAIL attribute. (This replaces any MAIL attribute for the directory entry and for certificate renewals replaces the original NotifyEmail value).

Using a gskkyman key database for your certificate store

This appendix lists the steps the RACF programmer performs to use a gskkyman key database.

Steps for using a gskkyman key database for your certificate store

Perform the following steps to use a gskkyman key database for your server's certificate store:

Note: If the z/OS HTTP Server is installed and configured for SSL using gskkyman, you need to perform only steps 9, 10, 11, and 15.

1. From the UNIX shell, **cd** to `/etc` and enter `/usr/lpp/gskssl/bin/gskkyman`.

2. Choose option **1** to create a key database. Type in a name or let it default to key `.kdb` and enter a password you want to use. When asked "Work with the database now?", enter **1** for yes.

3. Choose option **3** to create new key pair and certificate request. Answer the prompts for file name, label, key size (1024 is suggested), and subject name fields.

   Note: Common Name should be your server's symbolic IP address (for example, www.*YourCompany*.com).

4. Exit gskkyman when you are done.

5. From TSO, use the OGET command to put the certificate request in an MVS data set.

   Example:

   ```
   OGET '/etc/certreq.arm' certreq.arm
   ```

6. Use the RACDCERT command to read the request and generate the server certificate.

   Example:

   ```
   RACDCERT GENCERT(certreq.arm) ID(WEBSRV) SIGNWITH(CERTAUTH
   LABEL('Local PKI CA')) WITHLABEL('SSL Cert')
   ```

7. Export both the new server certificate and the CA certificate to MVS data sets, and OPUT these to file system files.

   Example:

   ```
   RACDCERT EXPORT(LABEL('SSL Cert')) ID(WEBSRV) DSN(cert.arm)
   FORMAT(CERTB64)
   OPUT cacert.der '/var/pkiserv/cacert.der' BINARY
   ```

© Copyright IBM Corp. 2001, 2011

8. You can optionally delete both certificate TSO data sets (but not the file system files).

9. In the UNIX shell, **cd** to `/etc` and invoke `/usr/lpp/gskssl/bin/gskkyman`.

10. Choose option **2** to open the key database (created earlier). Reply to the name and password prompts.

11. Choose option **6** to store a CA certificate and specify the `'/var/pkiserv/cacert.der'` file.

12. When asked to "Exit gskkyman?", enter **0** for No.

13. Choose option **4** to receive a certificate issued for your request and specify the `'/etc/cert.arm'` file. Again enter `0` when asked to "Exit gskkyman?".

14. Choose option **11** to store encrypted database password.

15. Exit gskkyman.

16. You can optionally remove the `/etc/cert.arm` file.

Configuring PKI Services as an IdenTrust™ certificate authority

This appendix describes the configuration required to allow your z/OS installations to participate in the IdenTrust™ infrastructure. By performing the task described here, you can configure z/OS Cryptographic Services PKI Services to operate as an IdenTrust compliant certificate authority (CA). This allows you to use z/OS to manage the life cycle of digital certificates on behalf of your organization and in accordance with your security policy. You can then issue and revoke digital certificates as needed.

This appendix contains an overview, a task roadmap, a set of procedures you need to perform, and a set of code samples that you can customize as you complete the procedures.

z/OS Cryptographic Services PKI Services is the component of the IBM z/OS operating system that provides support for the Public Key Infrastructure (PKI) infrastructure. Beginning with z/OS Version 1 Release 5, PKI Services was certified at the IdenTrust 3.1 specification level as an IdenTrust compliant CA software program.

For more overview information about z/OS support for IdenTrust, see http://www.ibm.com/servers/eserver/zseries/security/identrus/.

Who should use this appendix

This appendix should be used by personnel who support member institutions of the IdenTrust network. It instructs the UNIX programmer or Web server programmer to configure PKI Services to operate as an IdenTrust compliant CA.

This appendix assumes that you have experience installing and configuring software in a network environment. You should be knowledgeable about PKI technology. Previous experience with PKI Services is helpful.

Related information from IdenTrust

You should be familiar with the certificate profiles defined by IdenTrust in the following IdenTrust document. As an IdenTrust compliant CA using z/OS, you will implement these policies using PKI Services.

- *Identrus Public Key Infrastructure and Certificate Profiles [IT-PKI]*, Operating Rules and System Documentation.

Overview of configuring z/OS PKI Services as a CA

This topic describes the configuration required operate z/OS Cryptographic Services PKI Services as an IdenTrust compliant certificate authority (CA). It instructs member institutions of the IdenTrust network to configure their z/OS installations to participate in the IdenTrust infrastructure.

By following the task outlined in this topic, you can use your z/OS systems to establish CAs within the IdenTrust infrastructure. This allows you to use z/OS to manage the life cycle of digital certificates on behalf of your organization in

© Copyright IBM Corp. 2001, 2011

accordance with your security policy, while issuing and revoking digital certificates as needed. See http://www.ibm.com/servers/eserver/zseries/security/identrus/ for additional overview information.

System prerequisites

To use your z/OS system to establish a CA within the IdenTrust infrastructure, the following requirements apply:

1. You must operate PKI Services with z/OS Version 1 Release 5, or higher.
2. Your PKI Services installation must be completed, with PKI Services established as a certificate authority (CA) and configured as a self-signed CA. (See "Configuring z/OS PKI Services as a CA" on page 575.)
3. You must supply your own (or OEM) online certificate status protocol (OCSP) responder.

Task overview

The topic "Configuring z/OS PKI Services as a CA" on page 575 leads you through subtasks and associated procedures needed to configure your z/OS PKI Services system as a CA. The procedures configure PKI Services to accomplish the following objectives:

- Establish PKI Services as an intermediate CA under the IdenTrust root
- Adjust your PKI Services general settings.
- Define PKI Services certificate templates for IdenTrust certificate types

Establish PKI Services as an intermediate CA under the IdenTrust root

In the IdenTrust infrastructure, IdenTrust operates the *root* or top certificate authority (CA). All financial institutions are subordinate CAs that are certified by the root. Therefore, you must acquire a certificate for PKI Services that is signed by IdenTrust. You might need additional certificates issued by IdenTrust, such as for your OCSP responder. Request these certificates directly from IdenTrust following the operational requirements of IdenTrust.

Adjust your PKI Services general settings

Some of your current PKI Services configuration settings might require adjusting to operate PKI Services within the parameters identified by IdenTrust. In particular, your settings related to the following certificate policy items:

- CRL processing time
- Distribution point CRLs

CRL processing time: Processing time for an authenticated revocation request must not exceed 60 minutes. (Your OCSP responder usually provides revocation information by reading the CRLs issued by PKI Services to determine revocation status.) Therefore, the CRLs in LDAP must be refreshed with sufficient time to meet the 60-minute window.

Applicable PKI Services settings: `TimeBetweenCRLs` and `CRLDuration`.

Distribution point CRLs: Depending on how your OCSP responder operates, you might have to disable creation of distribution point CRLs.

Applicable PKI Services setting: `CRLDistSize`.

Define PKI Services certificate templates for IdenTrust certificate types

The IdenTrust document *Identrus Public Key Infrastructure and Certificate Profiles [IT-PKI]* identifies and details the various certificate types used within the IdenTrust infrastructure. Using PKI Services, you create your IdenTrust compliant certificates. You decide which IdenTrust certificate types your CA will create and then define certificate templates for them. You can use the sample certificate templates provided to create your IdenTrust compliant certificates.

IdenTrust compliant certificates must include the following:

- IdenTrust certificate policy information (needed to build the `CertificatePolicies` extension)
- The location of your OCSP responder (needed to build the `AuthorityInformationAccess` extension)
- Other IdenTrust required fields and extensions.

The following sample PKI Services certificate templates are provided to help you create IdenTrust compliant certificates:

- Sample browser certificate template for IdenTrust compliance
- Sample server certificate template for IdenTrust compliance.

Configuring z/OS PKI Services as a CA

The following table contains a task roadmap to lead you through the subtasks and associated procedures to configure your z/OS PKI Services system as a CA. Where needed, some notes are included to provide reminders about additional activities that are not described in this document. (For background information on the subtasks and reasons they are required, see "Task overview" on page 574.)

The following procedures are provided in this topic:

- Steps to modify pkiserv.conf for different certificate types
- Steps to modify pkiserv.conf general settings
- Steps to create IdenTrust specific certificate templates.

Subtask	Associated instructions (see ...)	Notes
If you have not already done so, install and configure PKI Services as a self-signed certificate authority (CA).	Follow the instructions in Part 1, "Planning," on page 1, Part 2, "Configuring your system for PKI Services," on page 27, and Part 3, "Customizing PKI Services," on page 121.	Remember to store your PKI Services CA signing key in Integrated Cryptographic Service Facility (ICSF).
Establish PKI Services as an intermediate certificate authority under the IdenTrust root.	Follow the instructions in "Establishing PKI Services as an intermediate CA" on page 408 and inPart 5, "Administering security for PKI Services," on page 401.	Send your certificate request to IdenTrust for signing. To do this, follow the IdenTrust instructions in *IT-PKI* to request a certificate for a "Participant CA Key Signing and CRL Signing Certificate Profile".
Modify the PKI Services configuration file (`pkiserv.conf`).	"Steps to modify pkiserv.conf for different certificate types"	Make sure your certificate policies are in accordance with *IT-PKI*.
	"Steps to modify pkiserv.conf general settings" on page 577	Make sure your changes are in accordance with *IT-PKI*.
Create IdenTrust specific certificate templates in the PKI Services certificate templates file (`pkiserv.tmpl`).	"Steps to create IdenTrust specific certificate templates" on page 577	Make sure your certificate templates are in accordance with *IT-PKI*.
Stop and restart PKI Services to activate your changes.	Follow the instructions in Chapter 10, "Starting and stopping PKI Services," on page 117.	

Steps to modify pkiserv.conf for different certificate types

Before you begin: Refer to the sample configuration file directives in "Sample PKI Services configuration file directives for IdenTrust compliance" on page 579.

Perform the following steps to modify the PKI Services configuration file (`pkiserv.conf`) to add a certificate policy for each type of IdenTrust certificate you intend to issue:

1. Copy the sample **OIDs** directives to the **OIDs** section.

   _____

2. Copy the sample **CertPolicy** directives to the **CertPolicy** section.

   _____

3. For each IdenTrust certificate policy you add, replicate one of the **OIDs** directives copied in Step 1.

   _____

4. Change the name and value of the directive as needed for the particular certificate profile. The name you choose is arbitrary, but must be unique.

   _____

5. Replicate one pair of `PolicyName`*nn* and `UserNoticeText`*nn* (the **CertPolicy** directives) copied in Step 2 on page 576.

6. Change the value of the `PolicyName`*nn* directive to match the name defined in Step 4 on page 576.

7. Change the value of the `UserNoticeText`*nn* directive as needed for this policy.

8. Change the policy number in the directives' name (the *nn* in `PolicyName`*nn* and `UserNoticeText`*nn*) as needed for the particular policy being defined. The number you choose is arbitrary, but must be unique. Use the same number for both directives.

9. For each IdenTrust certificate policy you add, repeat Step 3 on page 576 through Step 8.

When you are done: You have defined a certificate policy for each type of IdenTrust certificate you intend to issue.

Steps to modify pkiserv.conf general settings

Before you begin:

- Refer to the sample configuration file directives in "Sample PKI Services configuration file directives for IdenTrust compliance" on page 579.
- For more information about creating directives for certificate policies, see "Using certificate policies" on page 212 in Part 3, "Customizing PKI Services."

Perform the following steps to modify the PKI Services configuration file (`pkiserv.conf`) to configure certain general settings for IdenTrust compliance:

1. Change your current setting to `TimeBetweenCRLs=30m`.

2. Change your current setting to `CRLDuration=60m`.

3. Optionally, change your current setting to `CRLDistSize=0`. This change is required only if your OCSP responder does not support distribution point CRLs.

When you are done: You have configured your general PKI Services settings for IdenTrust compliance.

Steps to create IdenTrust specific certificate templates

Before you begin:

- Refer to the sample browser certificate template in "Sample browser certificate template for IdenTrust compliance" on page 579 and the sample server certificate template in "Sample server certificate template for IdenTrust compliance" on page 581.
- For details about creating certificate templates, see Chapter 11, "Customizing the end-user Web application if you use REXX CGI execs," on page 123.

For each IdenTrust certificate profile you need, perform the following steps to create an IdenTrust specific certificate template in the PKI Services certificate templates file (`pkiserv.tmpl`):

1. Determine if you need to define a certificate profile for a browser certificate or a server certificate.

   ---

2. Copy the appropriate sample browser or server certificate template to the PKI Services certificate templates file.

   ---

3. Change the name of the template as desired.

   ---

4. Change the nickname of the template as desired.

   ---

5. Change the `<CONTENT>` section to add or remove name fields and matching JavaScript as required for the desired IdenTrust profile. For example, if the subject's alternate name e-mail address is not required, remove it or make it optional.

   ---

6. Change the `<CONSTANT>` section as follows:
   a. Change the `AuthInfoAcc` values to provide the URLs required by your OCSP responder.
   b. Change the `CertPolicies` value to provide the policy numbers needed for the desired IdenTrust profile. (See Step 3.)

When you are done: You have created an IdenTrust specific certificate template for each IdenTrust certificate profile you need. Stop and restart PKI Services to activate all of your changes, according to the task roadmap 576.

Code samples

This topic contains code samples to help you complete the task of configuring PKI Services as a CA.

- "Sample PKI Services configuration file directives for IdenTrust compliance" on page 579

  This sample from the PKI Services configuration file (`pkiserv.conf`) contains sample file directives that are IdenTrust compliant. The sample defines two IdenTrust policies (`IdentrusPolicy4` and `IdentrusPolicy16`). It also shows sample general settings that meet the requirements for IdenTrust compliance (`TimeBetweenCRLs` and `CRLDuration`), and an optional setting (`CRLDistSize`).
- "Sample browser certificate template for IdenTrust compliance" on page 579

  This sample from the PKI Services template file (`pkiserv.tmpl`) defines a certificate template for an IdenTrust compliant browser certificate.
- "Sample server certificate template for IdenTrust compliance" on page 581

  This sample from the PKI Services template file (`pkiserv.tmpl`) defines a certificate template for an IdenTrust compliant server certificate.

Sample PKI Services configuration file directives for IdenTrust compliance

```
[OIDs]
IdentrusPolicy4=1.2.840.114021.1.4.1
IdentrusPolicy16=1.2.840.114021.1.16.2

[CertPolicy]
PolicyName4=IdentrusPolicy4
UserNoticeText4=This certificate may be relied upon only by either: (1) a
Relying Customer of an Identrus Participant, or (2) a party bound to the
alternative policy regime specified elsewhere in this Certificate

PolicyName16=IdentrusPolicy16
UserNoticeText16=This certificate may be relied upon only by either: (1) a
Relying Customer of an Identrus Participant, or (2) a party bound to the
alternative policy regime specified elsewhere in this Certificate

TimeBetweenCRLs=30m
CRLDuration=60m
CRLDistSize=0
```

Sample browser certificate template for IdenTrust compliance

```
#
# ====================================================================
#
# Template Name - 4-Year Identrus EE Identity Software Consumer Type 2 Certificate
#
# Function - Creates a 4-year browser certificate for use within
#            the Identrus infrastructure. This certificate is used
#            to sign communications between the Subscribing Customer (SC)
#            and Relying Customer (RC) and for S/MIME Digital Signature.
#
# ====================================================================
#
<TEMPLATE NAME=4-Year Identrus EE Identity Software Consumer Type 2
Certificate>
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=4YIEEIC2>
<CONTENT>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
%%-AdditionalHead[browsertype]%%
</HEAD>

<BODY>
<H1>4-Year Identrus EE Identity Software Consumer Type 2 Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#               "/[application]/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#             "/[application]/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
 <FORM NAME="CertReq" METHOD=POST ACTION=
              "/[application]/ssl-cgi-bin/careq.rexx" onSubmit=
    "return ValidateEntry(this)">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
#-- User input fields and validation Javascript ---------------
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
```

```
  if (ValidRequestor(frm) &&
   ValidCommonName(frm) &&
   ValidOrgUnit(frm) &&
   ValidOrg(frm) &&
   ValidCountry(frm) &&
   ValidAltEmail(frm) &&
   ValidNotifyEmail(frm) &&
   ValidPassPhrase(frm) &&
   ValidPublicKey(frm)){
# Add your validation Javascript here if needed ---
 return true;
}
else
 return false;
}
//-->
</SCRIPT>
 %%Requestor (optional)%%
 %%CommonName%%
 %%OrgUnit (optional)%%
 %%Org (optional)%%
 %%Country (optional)%%
 %%AltEmail%%
 %%NotifyEmail (optional)%%
 %%PassPhrase%%
 %%PublicKey[browsertype]%%
#-- End user input fields and validation Javascript -----------
<p>
<INPUT TYPE="Submit" VALUE="Submit certificate request">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="/[application]/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<CONSTANT>
 %%NotBefore=0%%
 %%NotAfter=1461%%
 %%KeyUsage=digitalsig%%
 %%KeyUsage=docsign%%
 %%CertPolicies=16%%
 %%AuthInfoAcc=OCSP,URL=https://ocsp.bank1.com
 %%AuthInfoAcc=IdentrusOCSP,URL=https://tc.bank1.com
 %%SignWith=PKI:%%
</CONSTANT>
<SUCCESSCONTENT>
 %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
 %%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
%%-copyright%%
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/[application]/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#      "/[application]/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
```

```
# This ACTION is for non z/OS clients. The task runs under surrogate ID
 <FORM NAME=retrieveform METHOD=POST ACTION=
       "/[application]/ssl-cgi-bin/cagetcert.rexx" onSubmit=
      "return ValidateEntry(this)">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
#-- User input fields and validation Javascript ---------------
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
if (ValidTransactionId(frm) &&
    ValidChallengePassPhrase(frm)) {
# Add your own Javascript here if needed
 return true;
}
else
 return false;
}
//-->
</SCRIPT>
 %%-TransactionId%%
 %%ChallengePassPhrase (optional)%%
#-- End user input fields and validation Javascript -----------
<p>
<INPUT TYPE="submit" VALUE="Retrieve and Install Certificate">
</FORM>
<p>
<H2>To check that your certificate installed properly, follow the
procedure below:</h2>
<p><B>Netscape V6</B> - Click Edit->Preferences, then Privacy and Security->
Certificates. Click the Manage Certificates button to start the Certificate
Manager.
Your new certificate should appear in the Your Certificates list.
Select it then click View to see more information.
<p><B>Netscape V4</B> - Click the Security button, then Certificates->
Yours. Your certificate should appear in the list.  Select it then
click Verify.
<p><B>Internet Explorer V5</B> - Click Tools->Internet Options, then
Content, Certificates.
Your certificate should appear in the Personal list. Click Advanced to
see additional information.
<p>
<FORM METHOD=GET ACTION="/[application]/public-cgi/camain.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Home page">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>
```

Sample server certificate template for IdenTrust compliance

```
#
# ====================================================================
#
# Template Name - 4-Year Identrus EE Server Signing Certificate
#
# Function - Creates a 4-year server certificate for use within the Identrus
#            infrastructure. This certificate is used to sign communications
#            1) between the Subscribing Customer (SC) and Relying Customer (RC),
#            2) between the RC and the Relying Participant (RP), and
#            3) for S/MIME Digital Signature.
#
# ====================================================================
#
<TEMPLATE NAME=4-Year Identrus EE Server Signing Certificate>
<TEMPLATE NAME=PKI Server Certificate>
<NICKNAME=4YIEESS>
<CONTENT>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
</HEAD>
<BODY>
<H1>4-Year Identrus EE Server Signing Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
```

```
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq"  METHOD=POST ACTION=
#                "/[application]/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq"  METHOD=POST ACTION=
#              "/[application]/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
 <FORM NAME="CertReq"  METHOD=POST ACTION=
                "/[application]/ssl-cgi-bin/careq.rexx" onSubmit=
     "return ValidateEntry(this)">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
#-- User input fields and validation Javascript ---------------
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
  if (ValidRequestor(frm) &&
   ValidCommonName(frm) &&
   ValidOrgUnit(frm) &&
   ValidOrg(frm) &&
   ValidCountry(frm) &&
   ValidAltEmail(frm) &&
   ValidNotifyEmail(frm) &&
   ValidPassPhrase(frm) &&
   ValidPublicKey(frm)){
# Add your validation Javascript here if needed ---
 return true;
}
else
 return false;
}
//-->
</SCRIPT>
 %%Requestor (Optional)%%
 %%CommonName%%
 %%OrgUnit%%
 %%Org%%
 %%Country%%
 %%AltEmail (Optional)%%
 %%NotifyEmail (Optional)%%
 %%PassPhrase%%
 %%PublicKey%%
#-- End user input fields and validation Javascript -----------
<p>
<INPUT TYPE="submit" VALUE="Submit certificate request">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>

<FORM METHOD=GET ACTION="/[application]/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<CONSTANT>
 %%NotBefore=0%%
 %%NotAfter=1461%%
 %%KeyUsage=digitalsig%%
 %%KeyUsage=docsign%%
 %%CertPolicies=4%%
 %%AuthInfoAcc=OCSP,URL=https://ocsp.bank1.com
 %%AuthInfoAcc=IdentrusOCSP,URL=https://tc.bank1.com
 %%SignWith=PKI:%%
</CONSTANT>
<SUCCESSCONTENT>
 %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
 %%-requestbad%%
</FAILURECONTENT>
```

```
<RETRIEVECONTENT>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
%%-copyright%%
</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#      "/[application]/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#      "/[application]/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
 <FORM NAME=retrieveform METHOD=POST ACTION=
       "/[application]/ssl-cgi-bin/cagetcert.rexx" onSubmit=
      "return ValidateEntry(this)">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
#-- User input fields and validation Javascript ---------------
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
if (ValidTransactionId(frm) &&
    ValidChallengePassPhrase(frm)) {
# Add your own Javascript here if needed
 return true;
}
else
 return false;
}
//-->
</SCRIPT>
 %%-TransactionId%%
 %%ChallengePassPhrase (optional)%%
#-- End user input fields and validation Javascript -----------
<p>
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%-returnpkcs10cert%%
</RETURNCERT>
</TEMPLATE>
```

Using the PKI Services Web application with Internet Explorer on Windows systems

In order to use the PKI Services end-user Web application through the Internet Explorer browser on a Microsoft Windows system, a user must first set up the Windows system and Internet Explorer to work with PKI Services. This topic describes the tasks that a user needs to perform.

Note: Users do not need to perform these tasks for browsers other than Internet Explorer.

In order to install a renewed certificate, PKI Services requires function that CAPICOM has provided. However, Microsoft has announced that it is not supporting CAPICOM on Windows 7. Although you can still install and use CAPICOM on Windows 7, there is no guarantee that it will continue to function properly. To avoid loss of function, PKI Services provides two ActiveX programs that can be used instead of CAPICOM to provide the function that PKI Services requires to renew a certificate. One ActiveX program is for Windows XP and earlier versions of Windows, and one is for Windows Vista and later versions of Windows. You can choose to use either CAPICOM or the PKI Services ActiveX program for your version of Windows. The PKI Services administrator must decide whether to provide signed or unsigned versions of the ActiveX programs, and in which directory the ActiveX programs are stored on the PKI Services server. These administrator tasks are described in "Administrator tasks for setting up a Windows system and Internet Explorer to work with the PKI Services Web application" on page 593.

The script used in the template and Java Server page (JSP) to renew and install a certificate from Internet Explorer first calls the PKI Services ActiveX program for the required functions. If the ActiveX program is not found, the script calls the CAPICOM functions. If the CAPICOM functions are not found, PKI Services prompts the user to install the ActiveX program.

User tasks for setting up a Windows system and Internet Explorer to work with the PKI Services Web application

Table 109. Tasks to perform to set up a Windows system and the Internet Explorer browser to work with PKI Services

	Subtask	Associated instructions
Install code that PKI Services requires on the Microsoft Windows system. Perform *one* of the following steps:		
	Install CAPICOM on the Microsoft Windows system. *or*	"Installing CAPICOM on a Microsoft Windows system" on page 586
	Install the PKI Services ActiveX program for your version of Microsoft Windows.	"Installing the PKI Services ActiveX program" on page 587

© Copyright IBM Corp. 2001, 2011

Table 109. Tasks to perform to set up a Windows system and the Internet Explorer browser to work with PKI Services (continued)

Subtask	Associated instructions
Configure Internet Explorer to trust PKI Services. (Windows Vista and later versions of Windows.)	"Configuring Internet Explorer to trust PKI Services on a Windows system" on page 591
Install the z/OS PKI Services certificate authority (CA) certificate on the Microsoft Windows system.	"Installing the PKI Services CA certificate on a Microsoft Windows system" on page 592

Installing CAPICOM on a Microsoft Windows system

Note: You do not need to install CAPICOM if you install the PKI Services ActiveX program. For information on how to do this, see "Installing the PKI Services ActiveX program" on page 587.

CAPICOM is an ActiveX control created by Microsoft that provides access to cryptographic functions. It can be used to digitally sign data, verify digital signature and digital certificates, add certificates to the certificate stores, and encrypt or decrypt data. CAPICOM provides functions required for z/OS PKI Services certificate renewal through Internet Explorer browsers.

Microsoft provides two methods for installing CAPICOM:

- CAPICOM is available as part of the Microsoft Platform Software Development Kit (SDK). If you choose this method to install CAPICOM, you have two options:
  - Install the full Microsoft Platform SDK (not available for Windows 7). Doing this causes more software than required by z/OS PKI Services to be installed on the Microsoft Windows system.
  - Install the CAPICOM subset of the Microsoft Platform SDK. This approach installs CAPICOM as well as software development samples on the Microsoft Windows system.
- CAPICOM is available as a security patch from Microsoft. Installing the patch causes the least amount of software installation for z/OS PKI Services.

Either method for installing CAPICOM requires verification of the Microsoft Windows operating system through Windows Genuine Advantage. When attempting to download the desired version of CAPICOM, the Microsoft Web site might present you with a page requesting you to verify the operating system. Click **Continue**, and allow the browser to activate any necessary ActiveX controls to perform the verification.

Steps for installing CAPICOM on a Microsoft Windows system

Before you begin: You must run as the system administrator on the Microsoft Windows system on which you are installing CAPICOM.

Perform the following steps to install CAPICOM on a Microsoft Windows system.

1. Determine which method of installation to use:
   - Install the full Microsoft Platform SDK for Vista (not available for Windows 7).

- Install the CAPICOM subset of the Microsoft Platform SDK.
- Install the CAPICOM security patch.

---

2. If you choose to install the full Microsoft Platform SDK, go to http://www.microsoft.com/downloads/ to obtain the download package and instructions for installing the package.
3. If you choose to install the CAPICOM subset of the Microsoft Platform SDK, go to http://www.microsoft.com/downloads/ to obtain the download package and instructions for installing and registering the package.
4. If you choose to install the CAPICOM security patch, go to http://www.microsoft.com/downloads/. Download the install package and run it on the Windows system.

When you are done, you have installed CAPICOM on a Microsoft Windows system. Continue to the next task, "Configuring Internet Explorer to trust PKI Services on a Windows system" on page 591.

Installing the PKI Services ActiveX program

There are two ways to install the PKI Services ActiveX program for your version of Windows:

- From the PKI Services home page (see "Steps for installing the PKI Services ActiveX program from the PKI Services home page").
- From the certificate renewal page when you renew a certificate (see "Steps for installing the PKI Services ActiveX program when you renew a certificate" on page 589). If you do not have either CAPICOM or the PKI Services ActiveX program for your version of Microsoft Windows installed, PKI Services prompts you to install the ActiveX program.

Steps for installing the PKI Services ActiveX program from the PKI Services home page

Before you begin: You must run as the system administrator on the Microsoft Windows system on which you are installing the PKI Services ActiveX program, and you must be using the Internet Explorer browser.

Perform the following steps to install the PKI Services ActiveX program for your version of Windows from the PKI Services home page.

1. On the PKI Services home page (Figure 36 on page 309), click **Install the PKI ActiveX Control to renew certificates**

---

2. A file download window opens.

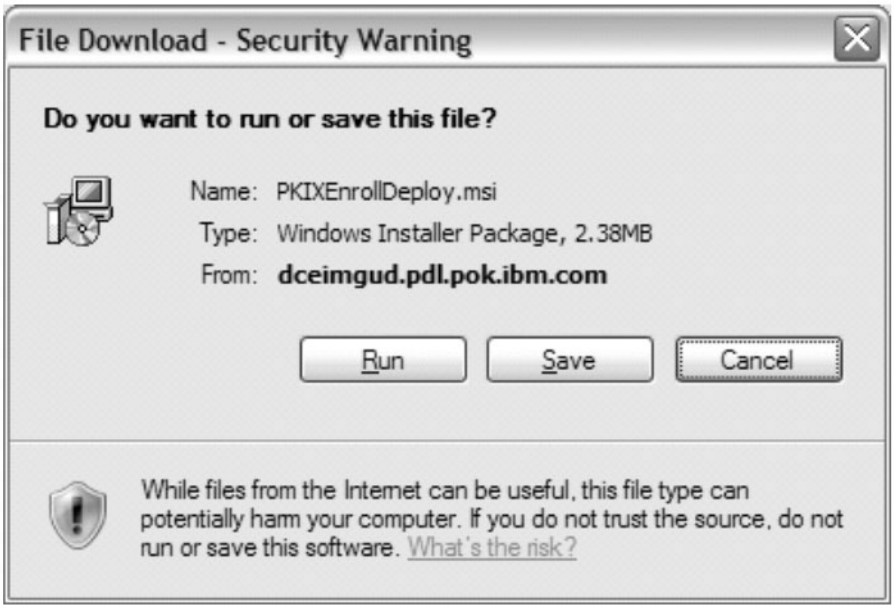


Figure 80. File download window

Click **Run**.

---

3. A security warning window opens. If the ActiveX program is signed, the name of the signer appears in the **Publisher** field. Figure 81 shows the security warning window for an unsigned ActiveX program. (The PKI Services administrator decides whether or not to sign the ActiveX program.)



Figure 81. Security warning window

Click **Run**.

---

4. The setup wizard for the PKI ActiveX control opens.

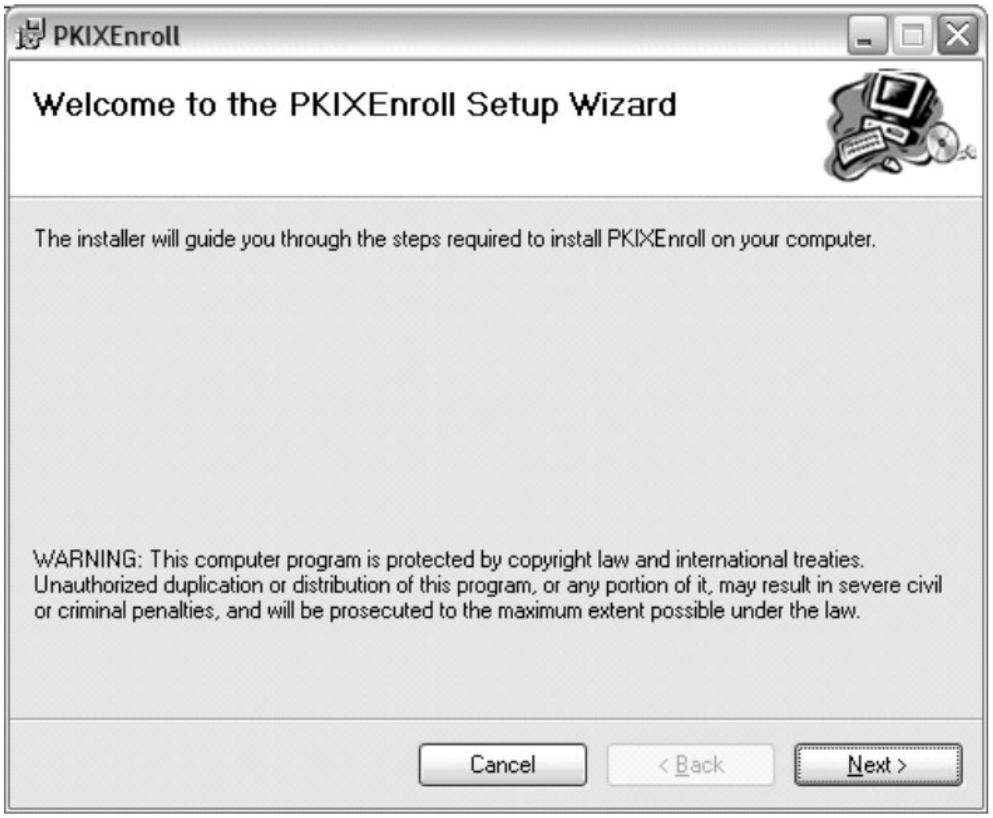


Figure 82. ActiveX control setup wizard

Click **Next**, and continue clicking **Next** on the windows that follow until you reach **Finish**. Click **Finish** to complete the installation.

When you are done, you have installed the PKI Services ActiveX program on a Microsoft Windows system. Continue to the next task, "Configuring Internet Explorer to trust PKI Services on a Windows system" on page 591.

Steps for installing the PKI Services ActiveX program when you renew a certificate

Before you begin: You must run as the system administrator on the Microsoft Windows system on which you are installing the PKI Services ActiveX program.

Perform the following steps to install the PKI Services ActiveX program for your version of Windows from the certificate renewal page.

1. On the certificate renewal page (Figure 49 on page 326), click **Renew**.

---

2. If PKI Services determines that neither CAPICOM nor the PKI Services ActiveX program is installed:
   a. A message window opens.

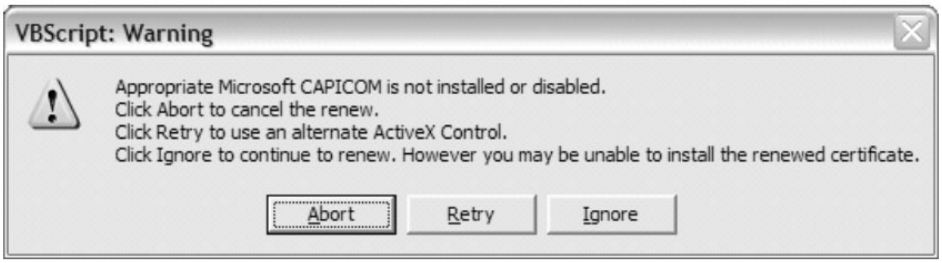


Figure 83. Message window

Click **Retry** to install the ActiveX program.

b. A confirmation window opens.

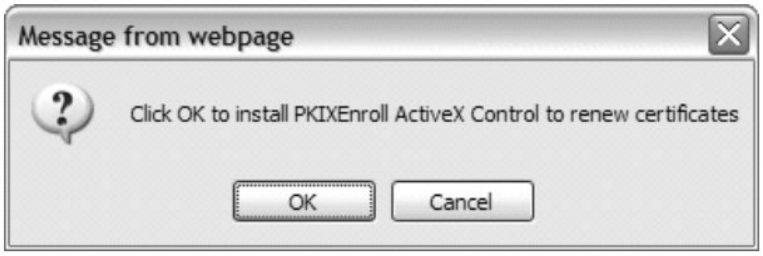


Figure 84. Confirmation window

Click **OK**.

c. Continue with the steps for installing the ActiveX control from the PKI Services home page, beginning with step 2 on page 587.

If PKI Services determines that CAPICOM is installed and the PKI Services ActiveX program is not installed:

a. PKI Services invokes CAPICOM, which opens a security alert window.

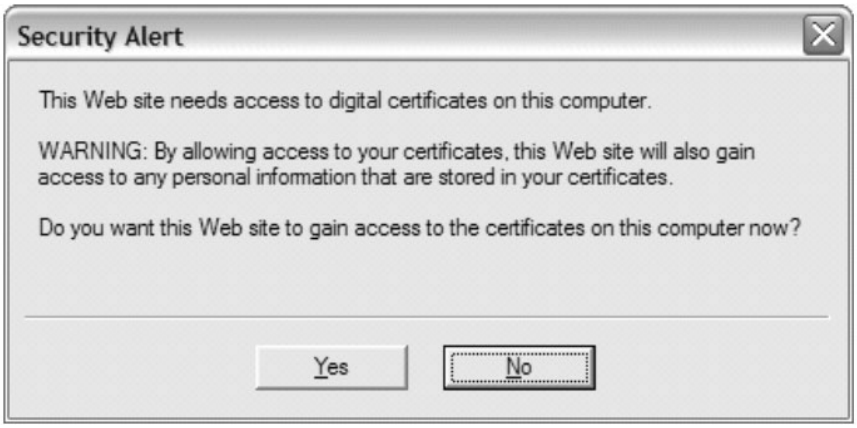


Figure 85. CAPICOM security alert window

Click **No** to install the ActiveX control. (Clicking **Yes** causes CAPICOM to run.)

b. The message window shown in Figure 83 on page 590 opens. Continue with the steps for installing the ActiveX control when neither CAPICOM nor the ActiveX control is installed, beginning with step 2a on page 589.

---

3. Refresh the certificate renewal Web page to load the ActiveX control on the browser.

When you are done, you have installed the PKI Services ActiveX program on a Microsoft Windows system from the certificate renewal web page, and loaded it on the Internet Explorer browser. PKI Services continues with the renewal of your certificate, using the ActiveX program that you just installed.

Configuring Internet Explorer to trust PKI Services on a Windows system

In order to request or administer certificates using the PKI Services end-user Web application through Internet Explorer on a system running Windows Vista or a later version of Windows, you must add the PKI Services system to the list of trusted sites recognized by Internet Explorer. You do not need to perform this task on systems running earlier versions of Windows.

Steps for configuring Internet Explorer to trust PKI Services

Perform the following steps to configure Internet Explorer to trust PKI Services on a system running Windows Vista or a later version of Windows.

1. From the Microsoft Windows system, launch the Internet Explorer browser.

---

2. Click `Tools->Internet Options` to see the "Internet Options" panel, and select the "Security" tab.

---

3. Select the "Trusted sites" icon in the window labeled "Select a zone to view or change security settings". The "Trusted sites" icon is usually represented as a green check mark.

---

4. If necessary, click "Custom Level..." to change the settings for the zone. Set "Security level" to `Medium` and ensure that "Enable protected mode (requires restarting Internet Explorer)" is not selected.

---

5. Click **Sites** to see the "Trusted Sites" panel.
6. In the area labeled "Add this website to the zone:" enter the URL for the PKI Services system. Use `https` as the protocol, not `http`. For example, if the URL for the PKI Services system is `alps4049.pok.ibm.com`, enter `https://alps4049.pok.ibm.com`. Click **Add** to add this site to the list of trusted sites.

---

7. Leave the box labeled "Require server verification (https:) for all sites in this zone" checked. Click **Close** to close the "Trusted Sites" panel and return to the "Security" tab of the "Internet Options" panel.

8. Click **Apply** on the "Internet Options" panel to confirm the configuration changes. Then click **OK** to close the" Internet Options" panel.

9. Shut down the Internet Explorer browser to allow the modifications to take effect.

When you are done, you have configured Internet Explorer to trust PKI Services on a system running Windows Vista or a later version of Windows.. Continue to the next task, "Installing the PKI Services CA certificate on a Microsoft Windows system."

Installing the PKI Services CA certificate on a Microsoft Windows system

Any system using Microsoft Windows to request or administer certificates using the PKI Services end-user Web application through Internet Explorer must install the z/OS PKI Services certificate authority (CA) certificate to enable SSL-protected sessions. Although the Internet Explorer browser can import the z/OS PKI Services CA certificate, the browser will not correctly install the certificate in the proper location by default.

Steps for installing the PKI Services CA certificate on a Microsoft Windows system

Before you begin: You need to know what version of Windows is running on your system.

Perform the following steps to install the z/OS PKI Services CA certificate in the correct location using Internet Explorer.

1. On the Microsoft Windows system, start the Internet Explorer browser. In the address field, enter the URL for the PKI Services home page.

2. Click the link labeled "Install the CA Certificate to enable SSL sessions for PKI Services". Internet Explorer presents a "File Download - Security Warning" pop-up panel, asking whether to open or save the file, and indicating that the file is from the PKI Services system. For example, if the PKI Services system is `alps4049.pok.ibm.com`, the following information should be displayed in this pop-up panel:

   ```
   Type: Security Certificate, number bytes
   From: alps4049.pok.ibm.com
   ```

   Click **Open**.

3. Internet Explorer might display a pop-up panel to warn that a Web site wants to open Web content using the browser. If it does, click **Allow**.

4. Internet Explorer presents a "Certificate" pop-up panel indicating that this CA Root certificate is not trusted, and to enable the trust, the certificate must be installed in the Trusted Root Certification Authorities store. Click **Install Certificate**.

5. Internet Explorer presents a " Certificate Import Wizard " welcome panel. Click **Next >**.

6. Internet Explorer displays the certificate store selections.
   a. If you are using Windows Vista or Windows 7, you *must* select `Place all certificates in the following store` and continue to step 7.
   b. If you are using Windows XP you can follow the instructions in step 6a. Alternatively, you can select `Automatically select the certificate store based on the type of certificate` and click **Next** to have the Wizard automatically select the certificate store. Then skip to step 8.

7. Internet Explorer presents a "Select Certificate Store" pop-up panel, allowing you to select the proper certificate store. Select the "Trusted Root Certification Authorities" item in the scrolled selection window, and click **OK**. Then click **Next >**.

8. Internet Explorer displays the "Completing the Certificate Import Wizard" panel. In the area labelled "You have specified the following settings:" the following information should be displayed:

   ```
   Certificate Store Selected By User       Trusted Root Certification Authorities
   Content                                  Certificate
   ```

   Click **Finish**.

9. Internet Explorer presents a "Security Warning" pop-up panel, indicating that a certificate is about to be installed, and asking you to verify that this is the intended action to take. Click **Yes**. Internet Explorer presents a confirmation pop-up, indicating that the certificate was successfully imported.

When you are done, you have successfully installed the PKI Services CA certificate, and your Windows system and Internet Explorer browser are ready to use the PKI Services Web application.

Administrator tasks for setting up a Windows system and Internet Explorer to work with the PKI Services Web application

PKI Services provides two ActiveX programs as alternatives to CAPICOM to provide function that PKI Services requires to install a renewed certificate. These programs are:

- PKIXEnroll, for Windows XP and earlier versions of Windows
- PKICEnroll, for Windows Vista and later versions of Windows

Both ActiveX programs require the Microsoft C Runtime Library and the Microsoft Active Template Library. PKI Services provides a Microsoft installer program for each of the ActiveX programs that is packaged with the ActiveX program and the required Microsoft libraries. You can use these programs in one of two ways:

- If you choose not to sign the ActiveX programs, you can use the installer programs as they are shipped.

- For greater security, you can sign the ActiveX programs, but if you do this you must repackage the installer programs. PKI Services provides the related registry files, type library files, and the license file that you need to sign and repackage the programs. After signing and repackaging the programs, you must put the .exe and .msi files in a directory on the PKI Services server that is accessible to PKI Services users so that they can install the ActiveX programs.

Guideline: For maximum security, sign the ActiveX programs with a certificate issued by your PKI Services CA. The ability of ActiveX programs to modify your system makes them a security risk. Signing your ActiveX programs helps to ensure that users are running unchanged versions free of viruses.

Signing the PKI Services ActiveX programs

To sign a PKI Services ActiveX program, you must sign the unsigned version that PKI Services provides, rebuild the install program for it, and sign the rebuilt install program. To make the signed programs available to PKI Services users, you must put the signed files in directories that are accessible to PKI Services users, and update the HTTP Server configuration files to specify those directories.

Steps for signing the PKI Services ActiveX programs

Before you begin:

- You need to have a tool such as Microsoft Visual Studio that builds an installer program.
- You need a code signing certificate with an Extended Key Usage of Code Signing and its private key. If you don't have one, you can request one from PKI Services using the 2-year Authenticode template. Follow the instructions for requesting a server certificate in “Steps for requesting a new certificate” on page 314. Then export the certificate and its private key to a PKCS #12 file and download it to the Windows platform. In step 3a on page 595, Microsoft Sign Tool will use the certificate to sign `PKIXEnroll.dll` and `PKICEnroll.dll`.

Perform the following steps to sign the ActiveX programs that PKI Services provides and make them available to PKI Services users.

1. Create directories on the PKI Services server for the .exe and .msi programs that you will build in step 3a on page 595. Create one directory for Windows XP and earlier versions of Windows, and one for Windows Vista and later versions of Windows.

   _____

2. For Windows XP and earlier versions of windows, download the following files from the directory `/usr/lpp/pkiserv/ActiveX/signsrc` on the z/OS system to a working directory on your workstation:
   - PKIXEnroll.dll
   - PKIXEnroll.tlb
   - PKIXEnroll.reg
   - PKIActiveX.lic

   For Windows Vista and later versions of windows, download the following files from the directory `/usr/lpp/pkiserv/ActiveX/signsrc` on the z/OS system to a working directory on your workstation:
   - PKICEnroll.dll
   - PKICEnroll.tlb
   - PKICEnroll.reg
   - PKIActiveX.lic

Ensure that the files are transferred in binary format so that they are not modified in transit.

---

3. Sign each of the ActiveX programs. Perform the following steps twice, once for `PKIXEnroll.dll` and once for `PKICEnroll.dll`.
   a. Use Microsoft Sign Tool (`signtool.exe`) to sign the ActiveX program (`PKIXEnroll.dll` or `PKICEnroll.dll`) with the code signing certificate. You can download Microsoft Sign Tool at http://msdn.microsoft.com/en-us/library/aa387764(VS.85).aspx.
   b. Build the installer programs. You'll need to use a tool such as Microsoft Visual Studio. For Windows XP and below, use PKIXEnrollDeploy for the project name, and the outputs for the installer program are:
      - `setup.exe`
      - `PKIXEnrollDeploy.msi`

      For Windows Vista and above, use PKICEnrollDeploy as the project name, and the outputs for the installer program are:
      - `setup.exe`
      - `PKICEnrollDeploy.msi`

      For detailed instructions, see "Steps for building the installer programs using Microsoft Visual Studio."
   c. Use Microsoft Sign Tool and the code signing certificate that you used in step 3a to sign the installer programs.

---

4. Upload the signed installer programs to the directories you created on the PKI Services server in step 1 on page 594.

   Note: Be sure you upload the programs in binary, so that the files are not altered during the transfer.

---

5. Update the HTTP configuration files.
   a. Update `httpd.conf`. Change the following statements to specify the directories you created in step 1 on page 594.

      ```
      Pass /PKIServ/PKIXEnroll/*   /usr/lpp/pkiserv/ActiveX/PKIXEnroll/*
      Pass /PKIServ/PKICEnroll/*   /usr/lpp/pkiserv/ActiveX/PKICEnroll/*
      ```

   b. Update `httpd2.conf`. Change the following statements to specify the directories you created in step 1 on page 594.

      ```
      Pass /PKIServ/PKIXEnroll/*   /usr/lpp/pkiserv/ActiveX/PKIXEnroll/*
      Pass /PKIServ/PKICEnroll/*   /usr/lpp/pkiserv/ActiveX/PKICEnroll/*
      ```

When you are done, you have signed the ActiveX programs provided by PKI Services, and made them available to PKI Services users

Steps for building the installer programs using Microsoft Visual Studio

Before you begin: You need to have a tool such as Microsoft Visual Studio that builds an installer program.

These instructions describe how to use Microsoft Visual Studio Software Application Version 2010 to create a setup and deployment Visual Studio project. The project creates a directory to save the project setup files along with the output

files. If you select `C:` as the location where the setup project is to be created, and PKIXEnrollDeploy as the project name, the directory structure created would be as shown in Table 110.

Table 110. Directory structure for a sample project named PKIXEnrollDeploy created in the C: directory

Directory or file	Description
C:\PKIXEnrollDeploy	solution directory
C:\PKIXEnrollDeploy\ PKIXEnrollDeploy.suo	solution property file
C:\PKIXEnrollDeploy\PKIXEnrollDeploy.sln	solution file
C:\PKIXEnrollDeploy\PKIXEnrollDeploy	root project directory
C:\PKIXEnrollDeploy\PKIXEnrollDeploy\ PKIXEnrollDeploy.vdproj	project property file
C:\PKIXEnrollDeploy\PKIXEnrollDeploy\ Debug	Build on debug version
C:\PKIXEnrollDeploy\PKIXEnrollDeploy\ Debug\PKIXEnrollDeploy.msi	
C:\PKIXEnrollDeploy\PKIXEnrollDeploy\ Debug\setup.exe	
C:\PKIXEnrollDeploy\PKIXEnrollDeploy\ Release	Build on release version
C:\PKIXEnrollDeploy\PKIXEnrollDeploy\ Release\PKIXEnrollDeploy.msi	
C:\PKIXEnrollDeploy\PKIXEnrollDeploy\ Release\setup.exe	

Perform the following steps to build the installer programs for a PKI Services ActiveX program.

1. Open Microsoft Visual Studio Software Application Version 2010 from the Windows Start menu, and create a new setup and deployment project.
   a. Click **File > New Project**.
   b. In the New Project panel, under **Other Project Types** select **Setup and Deployment**.
   c. In the "Visual Studio installed templates" pane (on the right) click **Setup Project**.
   d. In the **Name** field, enter the name of the project, for example PKIXEnrollDeploy.
   e. In the **Location** field, enter the directory where you want the project created, or click **Browse** to select a directory.
   f. Select **Create directory for solution**.
   g. Click **OK**.

   A new project directory is created in a separate file directory in the path given in the **Location** field.

   ---

2. Add the required ActiveX Dynamic link library, the ActiveX Type library file, and the ActiveX license file (PKIActiveX.lic) to the current project to create the installer program.

a. Click **View > Solution Explorer**.
b. Right click the name of the project that you just created, (for example, PKIXEnrollDeploy).
c. Click **Add > File**.
d. On the AddFiles window, navigate to the directory where the files were stored on the workstation in step 2 on page 594. For PKIXEnroll select these files:
   - PKIXEnroll.dll
   - PKIXEnroll.tlb
   - PKIActiveX.lic

   For PKICEnroll select these files:
   - PKICEnroll.dll
   - PKICEnroll.tlb
   - PKIActiveX.lic

   Click **Open**. The files you selected are listed under the project on Solution Explorer. The DLL file has a dependency on Microsoft .NET Framework. Microsoft Visual Studio automatically lists the dependency under the project.

---

3. Add the Microsoft C Runtime Library and Microsoft Active Template Library merge modules to be packaged in the installer program.
   a. On the Solution Explorer pane, right click the project name (for example, PKIXEnrollDeploy).
   b. Click **Add > Merge Module**. A window opens listing all the merge modules that were installed when Microsoft Visual Studio was installed.
   c. Click `Microsoft_VC100_CRT_x86.msm` and `Microsoft_VC100_ATL_x86.msm` and click **Open** to add these files to the project.

---

4. Modify the project properties.
   a. From the Solution Explorer pane, click the project name (for example, PKIXEnrollDeploy).
   b. On the toolbar click **View > Other Windows > Property Window**. A list of properties with default values is displayed.
   c. For the Author property, enter `IBM`.
   d. For the InstallAllUsers property, enter `True`.
   e. For the Manufacturer property, enter `IBM`.
   f. For the ProductName property, enter `PKIXEnroll` or `PKICEnroll`.
   g. For the RemovePreviousVersions property, enter `True`.

---

5. Determine the default location where the ActiveX program will be installed.
   a. Click **View > Solution Explorer**.
   b. Right click the project (for example, PKIXEnrollDeploy).
   c. Click **View > File System**. A File System pane opens on the right side.
   d. In the File System pane, right click **Application Folder**.
   e. Click **Properties Window**.
   f. Note the value listed in the **Default Location** field. This is the location where the ActiveX program will be installed. The default value is *[ProgramFilesFolder][Manufacturer]\[ProductName]*. The value of

ProgramFilesFolder has been set by Microsoft Visual Studio to the Program Files folder for the operating system: `C:\Program Files` for a 32-bit Windows system and `C:\Program Files(x86)` for a 64-bit operating system. You set the value of *Manufacturer* to IBM and the value of *ProductName* to PKIXEnroll or PKICEnroll when you set the project properties in step 4 on page 597. Do not modify any of these values. The ActiveX DLL will look for the license file (PKIActive.lic) in this directory. If it can't find it there, the ActiveX program will not be instantiated on the browser and certificate renewal processing might not work properly.

---

6. The User Interface command provides the interface for the user during installation. It allows the user installing the ActiveX program to select a directory for installation. Disable the folder selection step, so that the location listed in step 5f on page 597 is used.
   a. Click **View > Solution Explorer**.
   b. Right click the project (for example, PKIXEnrollDeploy).
   c. Click **View > User Interface**. A User Interface pane opens on the right side.
   d. Under **Install**, click **Installation Folder** and click **Delete**.
   e. Under **Install**, under **End** right click **Finished** and click **Properties**. Modify the UpdateText property to include instructions to be displayed to the user after the installation of the ActiveX program: `Please refresh the PKI Certificate Renewal web page to use the newly installed PKI ActiveX Control.`
   f. Under **Administrative Install**, click **Installation Folder** and click **Delete**.
   g. Under **Administrative Install**, under **End** right click **Finished** and click **Properties**. Modify the UpdateText property to include instructions to be displayed to the user after the installation of the ActiveX program: `Please refresh the PKI Certificate Renewal web page to use the newly installed PKI ActiveX Control.`

   ---

7. The setup and deployment project (for example, PKIXEnrollDeploy) can create registry entries for the ActiveX program. Once the ActiveX program is installed on the target machine the ActiveX program is registered and the browsers accessing this ActiveX program will instantiate looking at the windows registry. The registry entries are created using the Registry setup interface.
   a. Click **View > Solution Explorer**.
   b. Right click the project (for example, PKIXEnrollDeploy).
   c. Click **View > Registry**. A Registry pane opens on the right side.
   d. Right click **Registry on Target machine** .
   e. Click **Import**. The Import Registry File window opens.
   f. Click PKIXEnroll.reg or PKICEnroll.reg and click **Open**.

   ---

8. Set the ActiveX dynamic link library (PKIXEnroll.dll or PKICEnroll.dll) to be registered during installation.
   a. Click **View > Solution Explorer**.
   b. Right click PKIXEnroll.dll or PKICEnroll.dll.
   c. Click **Properties**.

d. Set the **Register** field to `vsdraCOM`.

---

9. Set the ActiveX type library (PKIXEnroll.tlb or PKICEnroll.tlb) to be registered during install.
   a. Click **View > Solution Explorer**.
   b. Right click PKIXEnroll.tlb or PKICEnroll.tlb.
   c. Click **Properties**.
   d. Set the **Register** field to `vsdrfCOM`.

---

10. Build the setup and deployment project.
    a. Click **Build > Release for the Build**.
    b. Right-click the project (for example, PKIXEnrollDeploy).
    c. Select Properties. The **Property Pages** window opens.
    d. Select Release in the Configuration list.
    e. On the right pane of the window that opens, click **Prerequisites**.
    f. Select the **Create setup program to install prerequisite components** check box.
    g. In the list of prerequisites, select the **.NET Framework 4.0 Client Profile (x86 and x64)** check box if it is not already selected.
    h. Select the **Visual C++ 2010 Runtime Libraries (x86)** check box.
    i. Click **OK**. The **Prerequisites** window closes.
    j. Click **OK**. The **Property Pages** window closes.
    k. Click **Build PKIXEnrollDeploy** (project name: PKIXEnrollDeploy).

       If the build is successful, the application creates an output directory under the root project directory which will be under the main solution directory. If PKIXEnrollDeploy is the project name:
       - C:\PKIXEnrollDeploy is the solution name.
       - C:\PKIXEnrollDeploy\PKIXEnrollDeploy is the root project directory.
       - C:\PKIXEnrollDeploy\PKIXEnrollDeploy\Release is the output directory.
       - There are two output files: PKIXEnrollDeploy.msi and setup.exe.

When you are done, you have built the installer program for a PKI Services ActiveX program. Continue with step 3c on page 595 to sign the installer program.

Accessibility

Publications for this product are offered in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when using PDF files, you may view the information through the z/OS Internet Library website or the z/OS Information Center. If you continue to experience problems, send an email to mhvrcfs@us.ibm.com or write to:

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A.

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer or Library Server versions of z/OS books in the Internet library at:

`http://www.ibm.com/systems/z/os/zos/bkserv/`

© Copyright IBM Corp. 2001, 2011

Notices

This information was developed for products and services offered in the U.S.A. or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

© Copyright IBM Corp. 2001, 2011

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted

for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Programming interface information

This document primarily documents information that is NOT intended to be used as Programming Interfaces of PKI Services.

This document also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of PKI Services. This information is identified where it occurs, either by an introductory statement to a topic or section or by the following marking:

Programming Interface information

End of Programming Interface information

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe is either a registered trademark or trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel is a trademark of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Bibliography

The following lists the titles and numbers of documents referenced in this publication.

- *IBM Health Checker for z/OS: User's Guide*, SA22-7994
- *IBM Tivoli Directory Server Administration and Use for z/OS*, SC23-5191
- *IBM Tivoli Directory Server Client Programming for z/OS*, SA23-2214
- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521
- *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SA22-7522
- *z/OS Cryptographic Services ICSF System Programmer's Guide*, SA22-7520
- *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*, SA23-2231
- *z/OS Cryptographic Services System SSL Programming*, SC24-5901
- *z/OS DFSMS Access Method Services for Catalogs*, SC26-7394
- *z/OS DFSMS Introduction*, SC26-7397
- *z/OS DFSMSdfp Storage Administration*, SC26-7402
- *z/OS HTTP Server Planning, Installing, and Using*, SC34-4826
- *z/OS Information Roadmap*, SA22-7500
- *z/OS Integrated Security Services Open Cryptographic Enhanced Plug-ins Application Programming*, SC24-5925
- *z/OS Introduction and Release Guide*, GA22-7502
- *z/OS Migration*, GA22-7499
- *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*, SA22-7612
- *z/OS MVS Programming: Sysplex Services Guide*, SA22-7617
- *z/OS Open Cryptographic Services Facility Application Programming*, SC24-5899
- *z/OS SDSF Operation and Customization*, SA22-7670
- *z/OS Security Server RACF Callable Services*, SA22-7691
- *z/OS Security Server RACF Command Language Reference*, SA22-7687
- *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683
- *z/OS Summary of Message and Interface Changes*, SA22-7505
- *z/OS TSO/E REXX Reference*, SA22-7790
- *z/OS UNIX System Services Command Reference*, SA22-7802
- *z/OS UNIX System Services Planning*, GA22-7800

© Copyright IBM Corp. 2001, 2011

Index

Special characters

_CEE_RUNOPTS, updating 430
_PKISERV_CA_DOMAIN environment variable 507
_PKISERV_CONFIG_PATH environment variable 507
_PKISERV_ENABLE_JSP
  description 509
_PKISERV_EXIT
  adding to environment variables file 268
  adding to Web server's environment variables file 273
  description 507
_PKISERV_MSG_LEVEL
  description 508
  message levels 459
  subcomponents 459
_PKISERV_MSG_LOGGING
  STDERR_LOGGING 508
  STDOUT_LOGGING 508
-AdditionalHeadIE 126
-ChallengePassphrase 126
-ChallengePassphrase2 126
-preregok 127
-RecoverEmail 126
-RecoverEmail2 126
-renewkeysetIE 127
-renewkeysetNS 127
-renewrevokebad 127
-renewrevokeok 127
-requestbad 126
-requestok 126
-requestor 126
-requestor2 127
-returnp12cert 127
/etc/pkiserv 11
/usr/lpp/pkiserv 11
  description 11
  subdirectories 497
/var/pkiserv 11
  description 11
[ ] for substitution variables 124
% in named fields 125
%%-renewrevokebad%% 134
%%-renewrevokeok%% 134
%%-requestok%% 142
%%AltOther_1_2_3_4_5%% 170
%%AltOther_1_2_3_4_6%% 170
%%cadomain%% variable 255
%%dn%% variable 255
%%notafter%% variable 255
%%pendreqlist%% variable 255
%%printcert%% variable 255
%%quicklink%% variable 256
%%recoverylink%% variable 256
%%recoverylist%% variable 256
%%rejectreason%% variable 256
%%requestor%% variable 256
%%returnp12cert%% 134
%%SelectCADomain%% 134
%%transactionid%% 256
%%transactionid%% variable 256
n-year PKI extensions demonstration certificate
  fields 144

Numerics

1YBSM 137
1YBSSL 137
2YBZOS 137
2YIACS 137
5YSCA 137
5YSCEPP 137
5YSIPS 137
5YSSSL 137

A

abends, recording 449
access
  READ, authorizing 404
  required for administrator 513
  to administration pages 335
    changing 178
  to end-user Web pages 308
  to RACF group 29
  to VSAM data sets 29
access control list (ACL), LDAP 88
access control, setting up 29, 511
accessibility 601
ACL (access control list), LDAP 88
actions
  on certificate requests 339
  on certificates 349
active (status of certificate) 349
active, AutoRenewDisabled (status of certificate) 349
active, NotRenewable (status of certificate) 349
active,AutoRenew (status of certificate) 349
ActiveX controls for PKI Services
  administrator tasks to set up 593
  building installer programs for using Microsoft Visual Studio 595
  description 585
  installing from PKI Services home page 587
  installing when renewing a certificate 589
  signing 594
ActiveX subdirectory 497
addtype directive 93
admactcert.rexx 176
admacttid.rexx 176
admacttid2.rexx 176
admicl.rexx 175
admiclall.rexx 177
admiclcert.rexx 176
ADMINAPPROVE subsection (in TEMPLATE section of pkiserv.tmpl) 140
adminDN keyword 87, 92
ADMINFOOTER subsection (in APPLICATION section of pkiserv.tmpl) 134, 177
ADMINHEADER subsection (in APPLICATION section of pkiserv.tmpl) 134, 177
administration
  changing log options 459
  changing requestor e-mail 351
  deleting certificate requests 344
  deleting certificates 351
  deleting preregistered requests 344
  disable automatic renewal of certificates 350
  displaying log options settings 460
  enable automatic renewal of certificates 351
  log options
    changing 459
    displaying 460
  modifying certificate requests 342
  processing
    certificate requests using searches 344
    certificates using searches 351
    multiple certificate requests 346
    multiple certificates 352
    selected certificate requests 346
    selected certificates 353
    single certificate 350
    single certificate request 340
    single preregistered request 340
  RACF
    ongoing administration 403
    running IKYSETUP 29
  rejecting certificate requests 344
  resuming certificates 351
  revoking certificates 351
  searching
    certificate requests 344
    certificates 351
  selected certificate requests 346
  selected certificates 353
  starting PKI Services 117
  stopping PKI Services daemon 119
  suspending certificates 351
administration group
  GID, IKYSETUP variable 32
  IKYSETUP variable 45
  setting up 29
  user IDs, IKYSETUP variable for 33
administration group PKIGRP 45
administration home page
  accessing 335
  using 339
administration tasks
  PKI Services 335

© Copyright IBM Corp. 2001, 2011

administration tasks *(continued)*
processing certificate requests 338
processing certificates 348
RACF
ongoing administration 403
running IKYSETUP 29
administration Web application
t description 4
administration Web pages
changing access to 178
customizing 175
fields 338
steps for customizing 177
using 335
administration, RACF
classroom courses xvi
administrative functions
protecting 29, 513
administrator
access required 513
administrator notification
copying files for 56
AdminNotifyForm (parameter in pkiserv.conf) 77
AdminNotifyNew (parameter in pkiserv.conf) 65
AdminNotifyReminder (parameter in pkiserv.conf) 66
adminPW
LDAP server configuration file 87, 92
ADMINSCOPE subsection (in APPLICATION section of pkiserv.tmpl) 134, 177
admmain.rexx 175
admmodtid.rexx 175
admpend.rexx 175
admpendall.rexx 176
admpendtid.rexx 175
advanced customization 211
AIEALNKE 497
alias
file (for sendmail) 24
for certificate template
changing 161
description 137
list 137
for database entry 286
for IKYSPROC 562
PKISERVD 562
AltDomain (named field in pkiserv.tmpl) 127
AltEmail (named field in pkiserv.tmpl) 127
alternate name
domain name 312
e-mail address 312
IP address 312
other name 312
uniform resource identifier (URI) 312
AltIPAddr (named field in pkiserv.tmpl) 128
AltOther (named field in pkiserv.tmpl) 128
AltURI (named field in pkiserv.tmpl) 128
APPL subsection (in TEMPLATE section of pkiserv.tmpl) 138
application domain
adding 226
steps for adding, when you use JSPs 229
steps for adding, when you use REXX CGIs 228
APPLICATION section of pkiserv.tmpl
ADMINFOOTER subsection 177
ADMINHEADER subsection 177
ADMINSCOPE subsection 134, 177
APPLICATION sections of pkiserv.tmpl
ADMINFOOTER subsection 134
ADMINHEADER subsection 134
CONTENT subsection 133
examining 149
FAILURECONTENT subsection 133
FINDRECOVERCONTENT subsection 133
RECONTENT subsection 134
RECONTENT2 subsection 134
REFAILURECONTENT subsection 134
RENEWEDCERT subsection 134
RESUCCESSCONTENT subsection 134
RETRIEVECONTENT2 subsection 134
RETURNCERT subsection 134
subsections 133
approve (action on certificate request) 339
approve with modifications (action on certificate request) 339
approved (status of certificate request) 338
APROCLIB 497
ARL (authority revocation list)
distribution point 225
ARLDist (parameter in pkiserv.conf) 66, 222
ASAMPLIB 497
associating
user ID with PKI Services started procedure 511
attributes
HIGHTRUST 405
LDAP, that PKI Services requires 567
OU 100, 102
PROTECTED 512
RDN 567
RESTRICTED 37, 512
Authenticode code signing PKI server certificate
fields 145
Authenticode–code signing PKI server certificate
description 136
AuthName1 (parameter in pkiserv.conf) 99
authority revocation list (ARL) 225
AuthorityInformationAccess
certificate extension 428
AuthorityKeyIdentifier
certificate extension 428
CRL extension 429
authorization checking, using exit routine 272
AuthPwd1 (parameter in pkiserv.conf) 100
auto-approval
access required 513
indicator in pkitmpl.xml 182
of certificates 135, 136
automatic certificate renewal
exit routine processing for 267
indicator in pkitmpl.xml 182
setting up 258
automatic deletion
from ICL 64, 65
from object store 64
automatic renewal of certificates 137
AUTORENEW tag on certificate templates 137

B

backing up
VSAM data sets
sample JCL 560
backup_dsn (variable in IKYSETUP) 43
base64-encoded
#10 certificate request 131
certificate 124, 157, 159
response 7
base64-encoded PKCS (field in end-user Web pages) 313
base64cert substitution variable 124, 127, 157
basic constraints extension, including in certificate requiest 130
BasicConstraints (certificate extension) 428
bibliography 607
bin subdirectory 498
bind passwords for LDAP
encrypted 420
in the clear 420
binding
distinguished name for LDAP 99
passwords for LDAP servers 100
BindProfile1 (parameter in pkiserv.conf) 101
bpx_userid. (variable in IKYSETUP) 39
brackets (in substitution variables) 124
browser certificates
n-year PKI certificate for extensions demonstration 136
aliases 137
IdenTrust compliant template 579
installing 321, 322
n-year PKI certificate for extensions demonstration 136
one-year PKI S/MIME browser certificate 135
one-year PKI SSL browser certificate 135
one-year SAF browser certificate 135
requesting 314
retrieving 321, 322
supported types 7
two-year PKI browser certificate for authenticating to z/OS 135
browsers supported 7
browsertype substitution variable 124

C

CA (certificate authority)
  distinguished name, IKYSETUP variable 32
  overview 3
CA certificate
  backing up 29
  creating 513
    using IKYSETUP 29
  expiration date, IKYSETUP variable 44
  exporting 29
  installing 309, 336
  installing on Microsoft Windows system 592
  key rollover 414
  label, IKYSETUP variable 32
  life span, IKYSETUP variable 44
  locating 406
  rekeying 414
  renewing 411
  specifying expiration 42
CA certificate and private key
  backing up 514
CA certificate profile, recovering 413
CA domain to which a CMP request is routed, determining 385
CA domains
  adding 230
CA functions
  authorizing PKI Services daemon user ID for 29
CA revocation list (ARL) 223
ca_dn (variable in IKYSETUP) 32
ca_domain (variable in IKYSETUP) 43
ca_expires (variable in IKYSETUP) 44
  changing value of 42
ca_exyears (variable in IKYSETUP) 44
  changing value of 42
ca_keysize (variable in IKYSETUP) 39
ca_label (variable in IKYSETUP) 32
ca_ring (variable in IKYSETUP) 44
CA, intermediate
  establishing PKI Services as 409
cacert_dsn (variable in IKYSETUP) 44
cadisplay.rexx 159
cadomain substitution variable 124
cagetcert.rexx 159
cagorcvr.rexx 159
camain.rexx 158
camodify.rexx 159
CAPICOM, installing 586
careq.rexx 159
caretrieve.rexx 159
CAring (default name of SAF key ring) 44, 78
catmpl.rexx 158
CBC.SCLBDLL 41
CDSA 3, 4
CEE.SCEERUN 41
CertGroupVerify 425
certificate authority (CA)
  certificate
    backing up 29
    creating 29
    exporting 29
    installing 309, 336
certificate authority (CA) *(continued)*
  certificate *(continued)*
    renewing 411
  overview 3
certificate data set, editing 410
certificate extensions 261
  customizing 9
  host identity mapping 9
  in PKI Services 9
  standard 8
  supported by PKITP 428
certificate generation application Web page 133
certificate management protocol (CMP), support for 379
  determining the CA domain to route request to 385
  enabling 68
  error codes and messages returned 397
  HTTP Server environment variables 392
  PKIBody structure, fields supported 381
  PKIHeader structure, fields supported 381
  PKIMessage structure, fields supported 380
  pkiserv.conf parameter to enable 68
  setting up client to make requests 388
  trace file, specifying 397
  trace options, specifying 396
  tracing 397
  type cp message, fields supported 383
  type cr message, fields supported 381
  type error message, fields supported 385
  type p10cr message, fields supported 383
  type rp message, fields supported 384
  type rr message, fields supported 384
certificate policies
  CertificatePolicies extension 212
  PKITP supports 427
certificate preregistration
  enabling in pkiserv.conf 69
certificate profile, recovering 413
certificate renewal, automatic
  exit routine processing for 267
certificate request message for CMP, fields supported 381
certificate requests
  actions on 339
  changing 342
  deleting 344
  modifying 342
  processing 335
    multiple 346
    selected 346
    single 340
    using searches 344
  processing selected 346
certificate requests *(continued)*
  rejecting 344
  relationship with certificates 355
  searching 344
  states 338
  statuses 338
  updating 342
certificate response message for CMP, fields supported 383
certificate revocation list (CRL)
  constant portion, distribution point URI 68
  constant portion, file system full path for the distribution point 67
  constant portion, relative distinguished name 67
  creating immediately 358
  maximum number of certificates 67
  revoked certificates on 349
  suspended certificates on 349
  time interval between issuances 73
  utility program for creation of 358
  validity period 68
certificate revocation list (CRL), allowing access to 88
certificate revocation request message for CMP, fields supported 384
certificate revocation response message for CMP, fields supported 384
certificate store, using gskkyman for 571
certificate suspension grace period 71
certificate templates
  adding 167
  alias 137
  AUTORENEW tag 137
  file
    customization, additional first-time 161
    customization, minimal 160, 161
    customizing for IdenTrust compliance 573
    customizing the OtherName field for REXX CGI execs 171
    for JSPs 179
    retrofitting release changes 165
  IdenTrust compliant
    browser certificate 579
    server certificate 581
  name 137
  nickname 137
  one-year PKI generated key 135
  pkiserv.tmpl 134
  subsections, summary 143
  true name 137
certificate validation service 425
CertificateIssuer (CRL entry extension) 429
CertificatePolicies extension
  defining 212
  in certificate 72
  organization name for 72
  PolicyCritical (parameter in pkiserv.conf) 71
  supported by PKITP 428
  using 212
certificates
  actions on 349

certificates *(continued)*
auto-approval 135, 136
automatic renewal 258
capturing 272
changing requestor e-mail 351
defining custom extensions for 261
deleting 351
disabling automatic renewal
by administrator 350
enabling automatic renewal
by administrator 351
expiration messages, when processed 82
expired, removing 81
extensions 9
generating keys for 76, 259
locating 406
posting 368
preregistering a SCEP client 331
preregistration 136
processing 335, 348
processing selected 353
recovering if PKI Services generated the keys 328
relationship with certificate requests 355
renewal messages, when processed 82
renewing
via Web page, steps for 324
renewing automatically 258
requesting 314
resuming 351
retrieving
from bookmarked page 321
from home page 322
revoking
by administrator 351
by user 327
searching 351
single 350
standard extensions 8
states 348
statuses 348
supported types 7
suspending
by administrator 351
by user 327
uses 7
X.509v3 support 8
certificates, authorization for creation 88
Certification Practice Statement
Uniform Resource Identifier 66
CertPlist class 300
CertPolicy section (of pkiserv.conf)
default values 65
description 59
excerpt 59
information needed 65
CertValidityConstraint (parameter in pkiserv.conf) 66
CF lock structure
defining in SMS base configuration 106
defining to MVS 105
CGI debugging, flag in pkiserv.tmpl 123
CGIs
admactcert.rexx 176
admacttid.rexx 176
admacttid2.rexx 176
admicl.rexx 175
admiclall.rexx 177
admiclcert.rexx 176
admmain.rexx 175
admmodtid.rexx 175
admpend.rexx 175
admpendall.rexx 176
admpendtid.rexx 175
cadisplay.rexx 159
cagetcert.rexx 159
cagorcvr.rexx 159
camain.rexx 158
camodify.rexx 159
careq.rexx 159
caretrieve.rexx 159
catmpl.rexx 158
installcert.rexx 159
summary 175
chains 425
challenge passphrase (field in end-user Web pages) 313
ChallengePassPhrase (named field in pkiserv.tmpl) 128
change requestor e-mail (action for certificate) 349
changes
summary xxiii
CISIZE statements 106
classroom courses, RACF xvi
clear, LDAP bind passwords in 420
client user ID 168
ClientName (named field in pkiserv.tmpl) 128
CMP (certificate management protocol), support for 379
determining the CA domain to route request to 385
enabling 68
error codes and messages returned 397
HTTP Server environment variables 392
PKIBody structure, fields supported 381
PKIHeader structure, fields supported 381
PKIMessage structure, fields supported 380
pkiserv.conf parameter to enable 68
setting up client to make requests 388
trace file, specifying 397
trace options, specifying 396
tracing 397
type cp message, fields supported 383
type cr message, fields supported 381
type error message, fields supported 385
type p10cr message, fields supported 383
CMP (certificate management protocol), support for *(continued)*
type rp message, fields supported 384
type rr message, fields supported 384
code samples
configuration directives 545, 547
configuration file 501
environment variables file 510
expiringmsg.form 254
httpd.conf 545
httpd2.conf 547
IKYCDB2 548
IKYCVSAM 551
IKYRVSAM 555
IKYSBIND 559
IKYSETUP 517
IKYVBKUP 560
IKYVREST 561
JCL to back up VSAM data sets 560
JCL to create VSAM data sets
not using RLS 551
using RLS 555
JCL to restore VSAM data sets 561
job to build DB2 package and plan for object store and ICL 559
job to create DB2 objects for object store and ICL 548
pendingmsg.form 254
pkiserv.conf 501
pkiserv.envars 510
pkiserv.tmpl
APPLICATION section 149
INSERT section 156
TEMPLATE section 153
PKISERVD 562
pkitpsamp.c 435
procedure to start PKI Services daemon 562
readymsg.form 253
recoverymsg.form 255
rejectmsg.form 253
renewcertmsg.form 254
code signing server certificate fields 145
codes returned from CMP functions 397
Common Data Security Architecture (CDSA) 3, 4
common name (field in end-user Web pages) 311
CommonName (named field in pkiserv.tmpl) 128
completed (status of certificate request) 338
compliance with IdenTrust 573
components
diagram 5
in message numbers 463
components of PKI Services 4
configurable section of IKYSETUP 30
configuration directives
example 545, 547
configuration file
example 501
for SSL traffic 91
pathname 507

configuration file *(continued)*
updating 79
for IdenTrust compliance 579
overview 58
steps 60
configuration, PKI Services
testing 117
configuring
system for PKI Services 27
CONSTANT subsection (in TEMPLATE section of pkiserv.tmpl) 139
CONTENT subsection
in APPLICATION section of pkiserv.tmpl 133
in TEMPLATE section of pkiserv.tmpl 138
CONTROL access for IRR.DIGTCERT.GENCERT 513, 515
core function 508
CORE subcomponent 508
country (field in end-user Web pages) 311
Country (named field in pkiserv.tmpl) 129
courses about RACF xvi
cp message type for CMP, fields supported 383
CPS in URI 66
CPS1 (parameter in pkiserv.conf) 66, 214
cr message type for CMP, fields supported 381
createcrls utility 358
CreateInterval (parameter in pkiserv.conf) 67
CreateOUValue (parameter in pkiserv.conf) 100
creating
VSAM data sets
using RLS 555
critical (marking of extension) 427
critical flag 71
CRL
allowing access to 88
creating immediately 358
distribution point, customizing 217
enabling support for large 223
entry extensions 429
extensions 429
larger than 32KB 69, 223
revoked certificates on 349
suspended certificates on 349
time interval between issuances 73
utility program for creation of 358
CRLDistDirPath (parameter in pkiserv.conf) 67, 222
CRLDistName (parameter in pkiserv.conf) 67, 221
CRLDistributionPoints (certificate extension) 225, 428
CRLDistSize (parameter in pkiserv.conf) 67, 221
CRLDistURI (parameter in pkiserv.conf) 68, 221
CRLDuration (parameter in pkiserv.conf) 68
CRLIDPExt (parameter in pkiserv.conf) 68
CRLNumber (CRL extension) 429
CRLReason (CRL entry extension) 429
cryptographic service provider (field in end-user Web pages) 314
cryptography
standards supported 7
CRYPTOZ class
profiles to allow PKI Services to generate key pairs 517
cryptoz_grp (variable in IKYSETUP) 39
CSECTs
IKY8B 450
IKYP0N 449
IKYP81 449
IKYP8A 449, 450
IKYP8B 449
IKYSCHDR 451
IKYSTART 451
IKYTIMER 452
CSF.SCSFMOD0 41
CSF.SCSFMOD1 41
csfkeys_profile (variable in IKYSETUP) 39
csfserv_profile (variable in IKYSETUP) 39
csfusers_grp (variable in IKYSETUP) 40
CSSM_TP_PassThrough
DBList 432
evidence 432
format 431
functions
CertGroupVerify 425
FreeEvidence 425
initial policy 432
parameters 431
performing certificate validation 434
purpose 431
return codes 433
CUSTOMERS
application name 150
customization
advanced 211
customizing
certificate templates file
minimal 160
end-user Web pages
minimal 160
pkiserv.tmpl
minimal 160

D

daemon
PKI Services, description 5
sample procedure for starting 562
starting 117
stopping 119
user ID
creating 511
PKISRVD 44
variable (user ID for PKI Services) 44
daemon (variable in IKYSETUP) 44
daemon user ID
authorizing for CA functions 29
creating 29
UID, IKYSETUP variable for 32
WEBSRV 47
daemon_uid (variable in IKYSETUP) 32
daemon, PKI Services
user ID 78
daily maintenance task
functions performed 81
specifying the days that it runs 74
specifying the time that it runs 75
specifying when it runs 81
specifying whether it runs at startup 75, 81
daily_Timer task,
specifying when it runs 81
data set name
certificate expiring message form 77
certificate ready message form 76
certificate reject message form 76
data sharing environment, for VSAM RLS 105
DB subcomponent 508
DB2 22
installing and configuring 25
use by PKI Services 4
DB2 database administrator
skills 16
DB2 objects
for object store and ICL, creating 548, 559
DB2 tables for ICL and ObjectStore
utility program to convert to from VSAM files 377
db2_repos (variable in IKYSETUP)
decision table 38
description 40
db2_subsys (variable in IKYSETUP)
description 40
DBPackage (parameter in pkiserv.conf) 61
DBSubsystem (parameter in pkiserv.conf) 61
DBType (parameter in pkiserv.conf) 61
debug flag 123
decision tables
db2_repos in IKYSETUP 38
key_backup in IKYSETUP 34
key_type in IKYSETUP 35
restrict_surrog in IKYSETUP 37
unix_sec in IKYSETUP 37
default
/etc/pkiserv 11
/usr/lpp/pkiserv 11
/var/pkiserv 11
binding information 421
CA private key size 39
CAring (SAF key ring) 44
certificate, data set for backup copy of 43
configuration file for z/OS HTTP Server 92
daemon user ID
PKI Services 44
Web server 47
data set
for copy of PKI Services certificate and private key 43
for copy of PKI Services certificate to assist backup process 44

default *(continued)*
data set *(continued)*
for copy of PKI Services RA certificate and private key 45
for Web server's root CA certificate for copying to file system 45
data set for backup copy of PKI Services certificate, private key 43
data set for backup copy of PKI Services RA certificate, private key 45
environment variables file 56
file locations 84
ICSF
profile to protect ICSF services 39
profile to protect PKI Services key 39
installation directory 11
key, data set for backup copy of 43
key, data set for backup copy of RA certificate 45
message level 459, 508
OMVSKERN (z/OS UNIX user ID) 39
PKI Services
administration group 45
configuration file 117
daemon user ID 44, 78
surrogate user ID 46
pkiserv.conf file values 60
primary and secondary extent allocations (in JCL) 104
RA certificate, data set for backup copy of 45
runtime directory 11
SAF key ring 78
sendmail location 57
STDOUT_LOGGING 508
surrogate user ID for PKI Services 46
time zone 118
UNIX user ID 39
variables directory 11
VSAM data set name
object store alternate index 62
Web server's daemon user ID 47
defining custom 261
delete (action for certificate) 349
delete (action on certificate request) 339
diagnostic messages, logging 509
diagram, PKI Services system 5
digital certificates
exporting certificates using R_PKIServ callable service 417
generating certificates using R_PKIServ callable service 417
retrieving certificates using R_PKIServ callable service 417
DIR parameter 57
directives
addtype 93
example 545, 547
exec 93
FastCGI 93
keyfile 93
log 94
normalmode 92, 93
pass 93
directives *(continued)*
protect 93
protection 93
redirect 93
sslclientauth 93
sslmode 92, 93
sslport 92, 93
SSLX500CARoots 93
SSLX500Host 93
SSLX500Password 93
SSLX500Port 93
SSLX500UserID 93
userId 92, 93
directories
/etc/pkiserv 11
/usr/lpp/pkiserv 11, 497
/var/pkiserv
description 11
for installation 11
for runtime 11
for variables 11
runtime 117
structure 497
var
setting up 84
directory server, LDAP
planning for 12
requirements 567
disability 601
disable (action for certificate) 349
distinguished name
e-mail address 311
for LDAP binding 99
LDAP administrator's 87, 92
qualifier 311
qualifiers
Email (deprecated) 129
EmailAddr 129
Mail 130
PostalCode 131
Street 132
distinguished name qualifier
field in end-user Web pages 311
distribution point ARL 225
distribution point CRLs
customizing 217
DN fields
mapping to LDAP attributes 568
DNQualifier (named field in pkiserv.tmpl) 129
documents
bibliography 607
configuring UNIX runtime environment 55
installing prerequisite products
ICSF 24
LDAP 22
OCSF 25
sendmail 24
z/OS HTTP Server 20
RACF administration 30
UNIX programmer 55
Domain component
field in end-user Web pages 311
domain name
fully qualified, for LDAP 87, 92, 99
domain name, alternate
field in end-user Web pages 312
domain to which a CMP request is routed, determining 385
DomainName (named field in pkiserv.tmpl) 129
DP CRLs
customizing 217
DSA
signature algorithm
updating 215

E

e-mail
applications 3
secure 3
e-mail address
for alternate name (field in end-user Web pages) 312
for distinguished name (field in end-user Web pages) 311
for notifications 314
e-mail notifications
AdminNotifyForm
updating 82
certificate expiration
when processed 82
certificate renewal
when processed 82
copying files for 56
customizing
overview 252
steps for 257
editing 257
environment variables, updating for 57
ExpireWarningTime
description 70
updating 80
ExpiringMessageForm
description 77
updating 82
expiringmsg.form
copying 56
description 53
forms for
expiringmsg.form 254
pendingmsg.form 254
readymsg.form 253
recoverymsg.form 255
rejectmsg.form 253
renewcertgmsg.form 254
NotifyEmail 131
pendingmsg.form
copying 56
description 53
pkiserv.conf
updating, overview 59
updating, steps 79
ReadyMessageForm
description 76
updating 82
readymsg.form
copying 56
description 55

e-mail notifications *(continued)*
RecoverForm
updating 82
recoverymsg.form
copying 56
description 55
RejectMessageForm
description 76
updating 82
rejectmsg.form
copying 56
description 55
RenewCertForm
updating 82
renewcertmsg.form
description 55
retrieving your certificate 321
updating
environment variables 57
ExpireWarningTime 80
expiringmsg.form 257
readymsg.form 257
recoverymsg.form 257
rejectmsg.form 257
variables 255
%%cadomain%% 255
%%dn%% 255
%%notafter%% 255
%%pendreqlist%% 255
%%printcert%% 255
%%quicklink%% 256
%%recoverylink%% 256
%%recoverylist%% 256
%%rejectreason%% 256
%%requestor%% 256
%%transactionid%% 256
EAR file for JSPs
updating 198
editing
IKYSETUP 48
Email (deprecated field in pkiserv.tmpl) 129
EmailAddr (named field in pkiserv.tmpl) 129
enable (action for certificate) 349
EnableCMP (parameter in pkiserv.conf) 68
EnableLargeCRLPosting (parameter in pkiserv.conf) 69
EnableSCEP (parameter in pkiserv.conf) 69
encrypted passwords for LDAP servers 101
BindProfile1
description 101
RACF administration 420
storing information for 97, 98
updating LDAP section of pkiserv.conf 98
encryption 215
end-user function 417
end-user functions
protecting 29, 511
end-user Web application
description 5
end-user Web pages
accessing 308
end-user Web pages *(continued)*
code locations 165
customizing 123
additional first-time 161
minimal 160
OtherName field for REXX CGI execs 171
fields 311
using 307
environment variables
_PKISERV_CA_DOMAIN 507
_PKISERV_CONFIG_PATH 507
_PKISERV_ENABLE_JSP 509
_PKISERV_EXIT 268, 273, 507
_PKISERV_MSG_LEVEL 508
_PKISERV_MSG_LOGGING 508
code sample file 510
description 507
file name
DIR parameter 57
FN parameter 57
HTTP Server, for CMP 392
in PKISERVD 562
TZ 57
updating
overview 57
steps 58
error codes returned from CMP functions 397
error message for CMP, fields supported 385
error message type for CMP, fields supported 385
error messages
list 463
error messages, logging 509
errorinfo substitution variable 124
EST5EDT 118
event code, SMF 565
examples
_PKISERV_CA_DOMAIN 507
_PKISERV_MSG_LEVEL 508
configuration directives 545, 547
configuration file 501
environment variables file 510
expiringmsg.form 254
httpd.conf 545
httpd2.conf 547
IKYCDB2 548
IKYCVSAM 551
IKYRVSAM 555
IKYSBIND 559
IKYSETUP 517
IKYVBKUP 560
IKYVREST 561
JCL
IKYCVSAM 551
IKYRVSAM 555
IKYVBKUP 560
IKYVREST 561
PKISERVD 563
log options settings 460
LOGREC data 452
named field 125
output from displaying log options settings 460
pendingmsg.form 254
examples *(continued)*
pkiserv.conf 501
pkiserv.envars 510
pkiserv.tmpl
APPLICATION section 149
INSERT section 156
TEMPLATE section 153
PKISERVD 562
pkitpsamp.c 435
procedure to start PKI Services daemon 562
readymsg.form 253
recoverymsg.form 255
rejectmsg.form 253
renewcertmsg.form 254
substitution variable 124
excerpt
pkiserv.conf
CertPolicy section 59
General section 60
LDAP section 97
ObjectStore section 59
OIDs section 59
SAF section 60
pkiserv.tmpl 148
exec directive 93
exit
_PKISERV_EXIT environment variable 507
environment variable 507
pathname 507
preprocessing
automatic renewal 270
scenarios
maintaining customized certificate repository 286
exit methods for JSPs 287
exit routine
arguments 273
post-processing 276, 278
EXPORT 280
GENRENEW 276
REQRENEW 278
REVOKE 282, 284
postprocessing
automatic renewal 271
preprocessing 277
EXPORT 279
GENCERT 275
GENRENEW 275
QRECOVER 283
REQRENEW 277
REVOKE 281
scenarios
allowing only selected users to request certificates 285
automatic renewal of certificates 272
recover lost passphrase 287
renewal only within 30 days of expiration 286
updating sample code 268
updating sample code for CGIs 273
using 267
exit, PKI Services
description of 5

ExitTimeout (parameter in pkiserv.conf) 74, 268
expired (status of certificate) 349
expired certificates
removing 81
ExpireWarningTime (parameter in pkiserv.conf) 70
expiring message form for certificate 77
ExpiringMessageForm (parameter in pkiserv.conf) 77
expiringmsg.form
code sample 254
copying 56
customizing 257
in samples directory 499
purpose 53
EXPORT
accesses required 419
description 286
parameters
post-processing 280
preprocessing 279
R_PKIServ function 512
return codes
post-processing 280
preprocessing 279
export_dsn (variable in IKYSETUP) 45
ExportCert class 299
extended key usage (field in end-user Web pages) 312
extensions
CertificatePolicies 72
custom 261
supported by PKI services 261
supported by PKI Services 8
supported by PKITP 428
X.509 version 3 standard 8
extensions demonstration
certificate
fields 144
extent allocations in IKYCVSAM 104
ExtKeyUsage (named field in pkiserv.tmpl) 129

F

FACILITY class profile
IRR.PROXY.DEFAULTS 421, 422
FACILITY class resource
IRR.RPKISERV.PKIADMIN 420
FAILURECONTENT subsection
in APPLICATION section of pkiserv.tmpl 133
FAILURECONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 142
FastCGI directive 93
field in end-user Web pages 312
fields
administration Web pages 338
end-user Web pages 311
in IKYSETUP REXX exec
change based on setup 34
change optionally 42
change required 31
modifiable by administrator 343
X.509 version 3 standard 8
file directory structure 497
file system
installation directory 11
runtime directory 11
subdirectories 497
files
CGIs
administrator Web pages 175
end-user Web pages 158
copying
for configuring PKI Services 55
default locations 461
exit 267
expiringmsg.form
code sample 254
copying 56
customizing 257
httpd.conf
code sample 545
updating 92
httpd1443.conf
source for 547
updating 93
httpd2.conf
code sample 547
copying from 93
IKYCVSAM
code sample 551
copying 106
extent allocations 104
updating 106
IKYRVSAM
code sample 555
copying 107
updating 107
IKYSETUP
code sample 517
running 29
IKYVBKUP
code sample 560
IKYVREST
code sample 561
JSPs 204
Makefile.pkiexit 267
pendingmsg.form
code sample 254
copying 56
pkiexit.c 267
pkiserv.conf
code sample 501
copying 56
updating 58, 98, 213, 214
pkiserv.envars
code sample 510
copying 56
updating 58
pkiserv.tmpl
additional customization 161
contents 123
copying 56
minimal customization 160
retrofitting changes 165
PKIServ.xsd
copying 56
description 179
PKISERVD
code sample 562
files *(continued)*
PKISERVD *(continued)*
updating 58
pkitmpl.xml
copying 56
description 179, 180
updating 198
PKITP 429
pkitp.h 498
pkitp.so 498
pkixgen.tmpl
description 179
readymsg.form
code sample 253
copying 56
customizing 257
recoverymsg.form
code sample 255
copying 56
customizing 257
rejectmsg.form
code sample 253
copying 56
customizing 257
renewcertmsg.form
code sample 254
FINDRECOVERCONTENT subsection in APPLICATION section of pkiserv.tmpl 133
firewall certificate
description 136
fields 145
five-year PKI intermediate CA certificate
description 136
fields 145
five-year PKI IPSEC server (firewall) certificate
description 136
fields 145
five-year PKI SSL server certificate
description 136
fields 145
five-year SCEP certificate
description 136
five-year SCEP preregistration
fields 147
FN parameter 57
forms for e-mail notifications
copying 56
customizing 252, 257
expiringmsg.form 254
pendingmsg.form 254
readymsg.form 253
recoverymsg.form 255
rejectmsg.form 253
renewcertmsg.form 254
variables 255
%%cadomain%% 255
%%dn%% 255
%%notafter%% 255
%%pendreqlist%% 255
%%printcert%% 255
%%quicklink%% 256
%%recoverylink%% 256
%%recoverylist%% 256
%%rejectreason%% 256
%%requestor%% 256

forms for e-mail notifications *(continued)*
variables *(continued)*
%%transactionid%% 256
FreeEvidence 425
fully qualified domain name
LDAP 87, 92

G

GENCERT
accesses required 419
exit routine scenario use 285, 286
parameters
post-processing 276
preprocessing 275
R_PKIServ function 512
return codes
post-processing 276
preprocessing 275
General section (of pkiserv.conf)
default values 74
description 60
excerpt 60
information needed 74
generated key pairs for certificates
setting up ICSF 23
TokenName keyword in configuration file 78
GENRENEW
accesses required 419
exit routine scenario use 286
parameters
post-processing 276
preprocessing 275
R_PKIServ function 512
return codes
post-processing 276
preprocessing 275
getCertificate method 299
getPassphrase method 299
getTransactionid method 299
GLD.SGLDLNK 41
global ARL (CA revocation list) 223
grace period, certificate suspension 71
groups
authorizing 404
deleting 404
GSK.SGSKLOAD 41
gskkyman 571

H

HIGHTRUST attribute 405
HoldInstructionCode (CRL entry extension) 429
host identity mapping 9
HostIdMap (named field in pkiserv.tmpl) 129
HostIdMappings
field (on end-user Web pages) 313
HostIdMappings extension
administering 404, 405
PKITP support 428
HTTP server
requirement for 12
HTTP Server
environment variables for CMP 392
installing and configuring 20
starting 94
updating configuration 91
HTTP Server configuration directives 545, 547
HTTP Server, z/OS
use by PKI Services 5
httpd.conf 545
application domains
adding 228
editing
to add application domains 228
httpd.envvars 273
httpd1443.conf 91
editing log directives in 94
httpd2.conf 547

I

ICL
base cluster
VSAM data set name for 63
certificates maintained in 348
creating DB2 objects for, steps 111
description 103
local DB2 subsystem, IKYSETUP variable for 40
repository, IKYSETUP variable 40
requestor
VSAM data set name for 63
sample job to create DB2 package and plan for 559
sample to create DB2 objects for 548
space considerations, using VSAM 104
status alternate index
VSAM data set name for 63
time period before automatic deletion
expired certificates 64, 65
ICL (issued certificate list)
configuration parameter for database implementation 61
converting VSAM files to DB2 tables 377
space considerations, using DB2 111
viewing 359
ICL data sets and indexes
creating 106
ICLDSN (parameter in pkiserv.conf) 63
ICLRequestorDSN (parameter in pkiserv.conf) 63
ICLStatusDSN (parameter in pkiserv.conf) 63
iclview utility 359
ICSF
authorizing PKI Services 29
default profile to protect services 39
installing and configuring 23
migrating CA certificate and private key to 514
migrating RA certificate and private key to 514
public key data set (PKDS) 23
requirement for 13
token data set (TKDS) 23
ICSF *(continued)*
use by PKI Services 5
ICSF programmer
installing and configuring ICSF 23
skills 14
tasks 14
IDCAMS 104
IdenTrust, configuring PKI Services for compliance
adding an intermediate CA 574
adjusting general settings 574
defining templates 575
introduction 573
roadmap 575
samples for IdenTrust compliance
browser certificate template 579
configuration file 579
server certificate template 581
steps
adjusting general settings 577
creating templates 577
modifying pkiserv.conf 576
system prerequisites 574
task overview 574
task roadmap 575
IDP extension 68
iecert substitution variable 124
IKY8B CSECT 450
IKYALLOC
library installed in 497
IKYCDB2
library installed in 497
sample 548
IKYCVSAM
copying 106
creating VSAM data sets 104
extent allocations 104
library installed in 497
sample 551
updating 106
IKYDDDEF
library installed in 497
IKYISMKD
library installed in 497
IKYMKDIR
library installed in 497
IKYP0N CSECT 449
IKYP81 CSECT 449
IKYP8A CSECT 449, 450
IKYP8B CSECT 449
IKYPKID 497
IKYPRTM 497
IKYRVSAM
copying 107
library installed in 497
sample 555
updating 107
IKYSBIND
library installed in 497
sample 559
IKYSCHDR CSECT 451
IKYSETUP REXX exec
actions 511
code sample 517
command to run 49
decision tables
for db2_repos 38

IKYSETUP REXX exec *(continued)*
decision tables *(continued)*
for key_backup 34
for restrict_surrog 37
for unix_sec 37
key_type 35
library installed in 497
parts 30
RACF administration 29
actions 511
steps for 47
sample log data set 50
structure and divisions 30
variables
backup_dsn 43
bpx_userid. 39
ca_dn 32
ca_domain 43
ca_expires 44
ca_exyears 44
ca_keysize 39
ca_label 32
ca_ring 44
cacert_dsn 44
changes based on setup 34
changes optional 42
changes required 31
cryptoz_grp 39
csfkeys_profile 39
csfserv_profile 39
csfusers_grp 40
daemon 44
daemon_uid 32
db2_repos 38, 40
db2_subsys 40
export_dsn 45
key_backup 34, 40
key_gen 40
key_type 35, 41
log_dsn 45
pgmcntl_dsn. 41
pki_gid 32
pkigroup 45
pkigroup_mem. 33
ra_backup_dsn 45
ra_dn 33
ra_label 33
restrict_surrog 37, 41
signing_ca_label 46
surrog 46
surrog_uid 33
unix_sec 37, 42
vsamhlq 46
web_dn 34
web_expires 46
web_exyears 47
web_label 47
web_ring 34
webserver 47
IKYSPROC 497
IKYSTART CSECT 451
IKYTIMER CSECT 452
IKYVBKUP
library installed in 497
sample 560
using 109
IKYVREST
library installed in 497
sample 561
using 110
implementation plan, creating 17
IMWEBSRV started procedure 95
include subdirectory 498
INETD 118
informational messages, logging 509
InitialThreadCount (parameter in pkiserv.conf) 74
inquiry access, authorizing users for 403, 404
INSERT sections of pkiserv.tmpl 125, 156
INSERTs
-AdditionalHeadIE 126
-ChallengePassphrase 126
-ChallengePassphrase2 126
-preregok 127
-RecoverEmail 126
-RecoverEmail2 126
-renewkeysetIE 127
-renewkeysetNS 127
-renewrevokebad 127
-renewrevokeok 127
-requestbad 126
-requestok 126
-requestor 126
-requestor2 127
-returnp12cert 127
AltDomain 127
AltEmail 127
AltIPAddr 128
AltOther 128
AltURI 128
ChallengePassPhrase 128
ClientName 128
CommonName 128
Country 129
DNQualifier 129
DomainName 129
Email (deprecated) 129
EmailAddr 129
ExtKeyUsage 129
HostIdMap 129
InstallCert 129
KeyProt 129
KeySize 130
KeyUsage 130
Label 130
Locality 130
Mail 130
NotAfter 130
NotBefore 130
NotifyEmail 131
Org 131
OrgUnit 131
OrgUnit2 131
PassPhrase 131
PostalCode 131
PublicKey 131
PublicKey2IE 132
PublicKey2NS 132
PublicKeyIE 131
PublicKeyNS 132
RecoverEmail 132
INSERTs *(continued)*
Requestor 132
Requestor2 132
returnbrowsercertIE 127
returnbrowsercertNS 127
returnpkcs10 127
Security*n* 132
SerialNumber 132
SignWith 132
StateProv 132
Street 132
Title 132
TransactionId 132
Uid 132
UnstructAddr 133
UserId 133
install_pkitp 429, 430
install-dir 11
installation directory 11
installation exit routine 267
InstallCert (named field in pkiserv.tmpl) 129
installcert.rexx 159
installing
PKI Services 11
skills 16
prerequisite products
skills 14
intermediate CA
certificate
description 136
description of template 136
fields 145
template description 136
establishing PKI Services as 409
intermediate certificate authority
establishing PKI Services as 408
Internet Explorer
configuring to trust PKI services 591
key protection field on end-user Web page 314
requesting a certificate 315
selecting a key size 316
setting up to run end-user Web application on Windows Vista 585
supported standard 7
verifying certificate installed correctly 322, 323
Internet Protocol Security standard (IPSEC) 3
interval
before certificate expiration 70
between certificate revocation lists 68, 73
certificate suspension grace period 71
scanning database for approved requests 67
warning message about certificate expiration 70
introduction to PKI Services 3
InvalidityDate (CRL entry extension) 429
IP address
AltIPAddr field 128
format 128

IP address *(continued)*
LDAP fully qualified domain name 87
IP address, alternate
field in end-user Web pages 312
IPSEC
certificate format 7
certificates 7
supported standard 3
IRR.DIGTCERT.ADD 419, 513
IRR.DIGTCERT.CERTIFAUTH.* 39
IRR.DIGTCERT.EXPORT 419
IRR.DIGTCERT.GENCERT 419, 513, 515
IRR.DIGTCERT.GENRENEW 419, 513
IRR.DIGTCERT.LIST 515
IRR.DIGTCERT.LISTRING 515
IRR.DIGTCERT.QRECOVER 419
IRR.DIGTCERT.REQCERT 419, 513
IRR.DIGTCERT.REQRENEW 419, 513
IRR.DIGTCERT.RESPOND 419
IRR.DIGTCERT.REVOKE 419
IRR.DIGTCERT.SCEPREQ 419
IRR.DIGTCERT.VERIFY 419
IRR.PROXY.DEFAULTS 98, 421, 422
IRR.PROXY.DEFAULTS profile in FACILITY class 421
IRR.RPKISERV 418
IRR.RPKISERV.PKIADMIN 420, 513
IRRSPX00 callable service
controlling the use 417
IRRSPX00 SAF callable service
description 5
exit routines 273
issued certificate list (ICL) 348
configuration parameter for database implementation 61
converting VSAM files to DB2 tables 377
description 103
space considerations, using DB2 111
viewing 359
IssuerAltName
certificate extension 428
IssuerAltName (CRL extension) 429
Issuing Distribution Point (IDP) extension 68
IssuingDistributionPoint (CRL extension) 429

J

Java server pages (JSPs)
deploying to a Websphere application server 200
description 180
directories for 204
environment variable for enabling 509
exit routine processing for 287
implementing the Web application using 179
TemplateTool utility 370
updating 198
XML template for 180, 198
JCL
backing up VSAM data sets 560
JCL *(continued)*
creating VSAM data sets
not using RLS 551
using RLS 555
example
IKYCVSAM 551
IKYRVSAM 555
IKYVBKUP 560
IKYVREST 561
PKISERVD 563
EXEC card, PARM= operand limitation 58
restoring VSAM data sets 561
VSAM data sets 560, 561
not using RLS 551
using RLS 555
JOB card 106
JSPs (Java server pages)
deploying to a Websphere application server 200
description 180
directories for 204
exit routine processing for 287
implementing the Web application using 179
TemplateTool utility 370
updating 198
XML template for 180, 198

K

key certificate
fields 147
key generation for certificates 259
key pairs for certificates
allowing PKI Services to generate 517
key pairs for certificates, generating
setting up ICSF 23
TokenName keyword in configuration file 78
key protection (field in end-user Web pages) 314
key ring
associating Web server and CA certificates with 29
creating 29
locating 406
key ring, SAF
IKYSETUP variable 44
key size (field in end-user Web pages) 313, 314
key usage (field in end-user Web pages) 313
key_backup (variable in IKYSETUP)
decision table 34
default value 40
description 40
key_gen (variable in IKYSETUP) 40
key_type (variable in IKYSETUP)
decision table 35
default value 41
description 41
keyboard 601
keyfile directive 93
keyid substitution variable 124
KeyProt (named field in pkiserv.tmpl) 129
KeyRing (parameter in pkiserv.conf) 78
keys for certificate requests
setting up PKI Services to generate 76, 259
KeySize (named field in pkiserv.tmpl) 130
KEYSMSTR class
activating 421
profile, defining 421
KeyUsage (certificate extension) 428
KeyUsage (named field in pkiserv.tmpl) 130

L

label (field in end-user Web pages) 314
Label (named field in pkiserv.tmpl) 130
LargeCRLPostPath (parameter in pkiserv.conf) 71
LDAP 88
administrator's distinguished name
description 87, 92
administrator's password
description 87, 92
allowing access to CRLs 88
attributes
mapped to DN fields 568
mapped to object identifiers 568
PKI Services requires 567
authorization to create certificates and CRLs 88
backend 22
bind passwords
encrypted 420
in the clear 420
configuring 21
directory server requirements 567
distinguished name
administrator's 87, 92
for binding 99
domain name
description 87, 92
Server1 parameter 99
encrypted passwords
BindProfile1 description 101
LDAPBIND class profile 98
RACF administration tasks for 420
storing information for 97
updating LDAP section of pkiserv.conf 98
establishing secure connection with 88
fully qualified domain name
description 87, 92
for LDAP server 99
Server1 parameter 99
installing 21
IP address
for LDAP server 99
IP address and port 99
objectclasses required by PKI Services 567
OU attribute 100, 102

LDAP *(continued)*
password
administrator's 87, 92
encrypted 420
for binding 100
in the clear 420
PKI Services objectclasses and attributes requirements 567
port
description 87, 92
for LDAP server 99
post interval 98
profile name 101
retrying post requests 100
servers available (number of) 98
subcomponent for message logging 508
suffix, description 88
tailoring configuration for PKI Services 87
tailoring pkiserv.conf 97
TDBM DB2 backend 22
time interval for scanning for items to post 98
use by PKI Services 5
version supported 7
LDAP programmer
skills 15, 16
tasks
configuration, tailoring LDAP 87
configuring LDAP 21
CRLs, allowing access to 88
installing LDAP 21
LDAP configuration, tailoring 87
schema.user.ldif, updating 88
setting up LDAP ACL for accessing CRLs 88
setting up LDAP ACL for creating certificates 88
summary 15, 16
tailoring LDAP configuration 87
updating schema.user.ldif 88
LDAP section (of pkiserv.conf)
default value 98
description 60
excerpt 97
information needed 98
tailoring 98
LDAP server
requirement for 12
LDAP server configuration file
adminPW 87, 92
LDAPBIND class
displaying information about 422
profile
creating 421
specifying name when configuring 98
LDBM backend 13
ldif2tdbm load utility 87, 92
legal statement about certificate issuance and use 73
lib 498
libraries
AIEALNKE 497
APROCLIB 497
ASAMPLIB 497
libraries *(continued)*
PROCLIB 497
SAMPLIB 497
SIEALNKE 497
load libraries 41
load utility (ldif2tdbm) 87, 92
local PKI certificate authority 32
Local PKI RA (default RA certificate label) 78
local PKI registration authority 33
locality (field in end-user Web pages) 311
Locality (named field in pkiserv.tmpl) 130
log data set
from running IKYSETUP 50
log data set name
IKY SETUP variable 45
log directives in httpd1443.conf, editing 94
log options
changing 459
displaying 460
log_dsn (variable in IKYSETUP) 45
logging message level 508
LOGREC
description 449
sample data 452
logs
changing options for 459
IKYSETUP data set sample 50
using information from 455

M

mail (field in end-user Web pages) 311
Mail (named field in pkiserv.tmpl) 130
mainframe
education xviii
maintenance task, daily
functions performed 81
specifying the days that it runs 74
specifying the time that it runs 75
specifying when it runs 81
specifying whether it runs at startup 75, 81
MaintRunDays (parameter in pkiserv.conf) 74
MaintRunTime (parameter in pkiserv.conf) 75
Makefile.pkiexit 267, 268, 273
Makefile.pkitpsamp 429
mapping
DN fields to LDAP attributes 568
host identity 9
MaxSuspendDuration (parameter in pkiserv.conf) 71
members
connecting
to new group 404
to RACF group 403
deleting from group 404
deleting from RACF group 403
message form
certificate expiring 77
certificate ready 76
certificate rejected 76
message levels
_PKISERV_MSG_LEVEL 459
for logging 508
logging
diagnostic 509
error 509
informational 509
severe 509
verbose diagnostic 509
warning 509
message logging
CORE subcomponent 508
DB subcomponent 508
LDAP subcomponent 508
PKID subcomponent 508
POLICY subcomponent 508
SAF subcomponent 508
TPOLICY subcomponent 508
message numbers
components identified 463
message types 463
messages returned from CMP functions 397
methods, exit, for JSPs 287
Microsoft Internet Explorer
key protection field on end-user Web page 314
requesting a certificate 315
selecting a key size 316
supported standard 7
verifying certificate installed correctly 322, 323
Microsoft Windows
installing CAPICOM 586
installing PKI Services CA certificate on 592
setting up to run end-user Web application 585
MODIFY command
change log options 459
display logging options 460
stop PKI Services daemon 119
Mozilla-based browser
key size field on end-user Web page 314
requesting a certificate 315
selecting a key size 316
verifying certificate installed correctly 322, 323
MVS programmer
installation of PKI Services 11
skills 16
tasks
creating VSAM data sets 104
enabling VSAM data sets for RLS 107
establishing RLS, preliminary steps 105
RLS, enabling VSAM data sets for 107
RLS, preliminary steps for establishing 105
starting PKI Services daemon 117
stopping PKI Services daemon 119
VSAM data sets, creating 104

MyPolicy (parameter in pkiserv.conf) 60, 213

N

n-year PKI certificate for extensions
demonstration
description 136
name
certificate templates
alias names 137
nicknames 137
short names 137
table summarizing 137
true names 137
field (in end-user Web pages) 314
named fields (in pkiserv.tmpl) 125
nickname
certificate template 137
normal operating mode of z/OS HTTP Server 515
normalmode directive 92, 93
not after date (field in end-user Web pages) 312
not before date (field in end-user Web pages) 312
NotAfter (named field in pkiserv.tmpl) 130
NotBefore (named field in pkiserv.tmpl) 130
notice, legal, for certificate 73
Notices 603
notification e-mail address (field in end-user Web pages) 314
notification forms
copying 56
customizing 252
notifications
customizing 257
retrieving your certificate 321
NotifyEmail
deleting 161
must match MAIL 326
NotifyEmail (named field in pkiserv.tmpl)
description 131
notifying users
forms for
pendingmsg.form 254
renewcertmsg.form 254
NumServers (parameter in pkiserv.conf) 98

O

Object ID
for policy 72
object identifiers
mapping to LDAP attributes 568
object store
ample job to create DB2 package and plan for 559
configuration parameter for database implementation 61
creating DB2 objects for, steps 111
description 103, 355
enabled for sysplex 65
object store *(continued)*
local DB2 subsystem, IKYSETUP variable 40
repository, IKYSETUP variable 40
sample to create DB2 objects for 548
space considerations, using DB2 111
space considerations, using VSAM 105
time period before automatic deletion
inactive requests 64
unsuccessful requests 64
time period before automatic deletion from
completed requests 64
incomplete requests 64
VSAM data set names
requestor alternate index 62
status alternate index 62
TID alternate index 62
object store data set records
viewing 372
objectclasses
LDAP, that PKI Services requires 567
ObjectDSN (parameter in pkiserv.conf) 61
ObjectRequestorDSN (parameter in pkiserv.conf) 62
ObjectStatusDSN (parameter in pkiserv.conf) 62
ObjectStore
converting VSAM files to DB2 tables 377
DB subcomponent for message logging 508
section of pkiserv.conf
default value 61
description 59
excerpt 59
information needed 61
ObjectTidDSN (parameter in pkiserv.conf) 62
OCEP
configuring for PKITP 429
data library (DL) 432
installing and configuring 26
optional installation 13
programmer
skills 15
Trust Policy 425, 427, 432
OCSF
configuring 25
installing 25
programmer
installing and configuring OCSF 25
skills 15
requirement for 13
Trust Policy
module 430
overview 425
plug-in 425
OCSF Trust Policy
API
CSSM_TP_PassThrough 430
OCSPType (parameter in pkiserv.conf) 71
OIDs section (of pkiserv.conf)
default value 60
description 59
excerpt 59
information needed 60
OMVSKERN 39
one-year PKI generated key certificate
description 135
fields 147
one-year PKI S/MIME browser certificate
description 135
fields 144
one-year PKI SSL browser certificate
description 135
fields 144
one-year SAF browser certificate
description 135
fields 147
one-year SAF server certificate
description 135
fields 147
optfield substitution variable 124
Org (named field in pkiserv.tmpl) 131
organization (field in end-user Web pages) 311
organization name
for CertificatePolicies extension 72
organizational unit (field in end-user Web pages) 311
organizationalUnit objectclass 100
OrgUnit (named field in pkiserv.tmpl) 131
OrgUnit2 (named field in pkiserv.tmpl) 131
other name, alternate
field in end-user Web pages 312
OtherName (field in end-user Web pages)
customizing for REXX CGI execs 170
OU attribute 100, 102
overview of PKI Services 3

P

p10cr message type for CMP, fields supported 383
p12cert substitution variable 124
Parallel Sysplex support
prerequisites 11
requirements 11
parameters
changing 272
validating 272
pass directive 93
passphrase
recovering 272
passphrase (field in end-user Web pages) 314
PassPhrase (named field in pkiserv.tmpl) 131
passwords
binding 420
encrypted, for LDAP servers
LDAPBIND class profile 98
RACF administration for 420
for LDAP binding 100
for LDAP servers, encrypted
storing information for 97

passwords *(continued)*
for LDAP servers, encrypted *(continued)*
updating LDAP section of pkiserv.conf 98
LDAP
encrypted 420
in the clear 420
LDAP administrator's 87, 92
RACF administration for 420
path name
certificate expiring message form 77
certificate ready message form 76
certificate reject message form 76
renewed certificate message form 77
request(s) pending for approval message form 77
requests meeting criteria for recovery message form 78
pathname
configuration file 507
exit program 507
pending approval (status of certificate request) 338
pendingmsg.form
code sample 254
copying 56
in samples directory 499
purpose 53
pgmcntl_dsn. (variable in IKYSETUP) 41
PKCS #10
browser certificate format 7
certificate request 313
SCEP certificate request 246
server certificate format 7
PKCS #10 certificate request message for CMP, fields supported 383
PKCS #11 TKDS (token data set) 23
PKDS (public key data set) 23
PKI (public key infrastructure)
defined 4
PKI Authenticode code signing server certificate
fields 145
PKI browser certificate for authenticating to z/OS
description 135
fields 144
PKI certificate for extensions demonstration
description 136
PKI extensions demonstration certificate
fields 144
PKI generated key certificate
description 135
PKI intermediate CA certificate
description 136
fields 145
PKI IPSEC server (firewall) certificate
description 136
fields 145
PKI S/MIME browser certificate
description 135
fields 144
PKI Services
daemon user ID
PKISRVD 44
starting 117
stopping 117, 119
updating
certificate templates file 161
using
administration Web pages 335
Web pages
using 335
PKI Services administration
processing certificate requests
selected 346
selected certificate requests 346
PKI Services daemon
starting 117, 562
PKI Services daemon user ID
creating 511
PKI SSL browser certificate
description 135
fields 144
PKI SSL server certificate
description 136
fields 145
PKI Windows logon certificate
fields 144
pki_gid (variable in IKYSETUP) 32
PKIBody structure for CMP, fields supported 381
PKICEnroll.dll
signing 594
PkiCertificate class 301
PKID
subcomponent for message logging 508
pkiexit.c
description 267
scenarios
allowing only selected users to request certificates 285
automatic renewal of certificates 272
maintaining customized certificate repository 286
recover lost passphrase 287
renewal only within 30 days of expiration 286
updating sample code 268
updating sample code for CGIs 273
pkigroup (variable in IKYSETUP) 45
pkigroup_mem. (variable in IKYSETUP) 33
PKIGRP 45
PKIHeader structure for CMP, fields supported 381
pkijsp subdirectory 498
PKIMessage structure for CMP, fields supported 380
pkiprereg utility 363
PKISERV
application name 133, 149
runtime user ID 168
surrogate user ID 46, 512
z/OS HTTP Server operating modes required 515
PKIServ subdirectory 498
pkiserv.conf
CertPolicy section
default values 65
description 59
excerpt 59
information needed 65
changing
signature algorithm 216
code sample 501
copying 55
distribution point CRLs, customizing 220
editing
for configuring PKI Services 60
to change signature algorithm 216
to create CertificatePolicies extension 213, 214
to customize distribution point CRLs 220
to test configuration 117
General section
default values 74
description 60
excerpt 60
information needed 74
LDAP section
default value 98
description 60
excerpt 97
information needed 98
ObjectStore section
default value 61
description 59
excerpt 59
information needed 61
OIDs section
default value 60
description 59
excerpt 59
information needed 60
parameters
AdminNotifyForm 77
AdminNotifyNew 65
AdminNotifyReminder 66
ARLDist 66, 222
AuthName1 99
AuthPwd1 100
BindProfile1 101
CertValidityConstraint 66
CPS1 66, 214
CreateInterval 67
CreateOUValue 100
CRLDistDirPath 67, 222
CRLDistName 67, 221
CRLDistSize 67, 221
CRLDistURI 68, 221
CRLDuration 68
CRLIDPExt 68
DBPackage 61
DBSubsystem 61
DBType 61
EnableCMP 68
EnableLargeCRLPosting 69
EnableSCEP 69
ExitTimeout 74
ExpireWarningTime 70

pkiserv.conf *(continued)*
parameters *(continued)*
ExpiringMessageForm 77
ICLDSN 63
ICLRequestorDSN 63
ICLStatusDSN 63
InitialThreadCount 74
KeyRing 78
LargeCRLPostPath 71
MaintRunDays 74
MaintRunTime 75
MaxSuspendDuration 71
MyPolicy 60, 213
NumServers 98
ObjectDSN 61
ObjectRequestorDSN 62
ObjectStatusDSN 62
ObjectTidDSN 62
OCSPType 71
Policy1Notice1 72, 214
Policy1Notice2 73
Policy1Org 72, 214
PolicyCritical 71, 213
PolicyName1 72, 213
PolicyRequired 72, 213, 214
PostInterval 98
RA_label 78
ReadyMessageForm 76
RecoverForm 78
RejectMessageForm 76
RemoveCompletedReqs 64
RemoveExpiredCerts 65
RemoveExpiredCertsAndKeys 64
RemoveInactiveReqs 64
RenewCertForm 77
RetryMissingSuffix 100
RunMaintAtStart 75
Server1 99
SharedPLEX 65
SigAlg1 73, 217
TimeBetweenCRLs 73
TokenName 78
UserNoticeText1 73, 214
purpose 54
SAF section
default value 78
description 60
excerpt 60
information needed 78
sections
CertPolicy section 59
General section 60
LDAP section 60
ObjectStore section 59
OID section 59
SAF section 60
signature algorithm
changing 216
steps for updating LDAP section 101
updating 79
overview 58
steps 60
pkiserv.envars
code sample 510
purpose 55
updating 57
pkiserv.tmpl
APPLICATION section 149
ADMINFOOTER subsection 177
ADMINHEADER subsection 177
ADMINSCOPE subsection 134, 177
application sections
adding 227
APPLICATION sections
ADMINFOOTER subsection 134
ADMINHEADER subsection 134
CONTENT subsection 133
FAILURECONTENT subsection 133
FINDRECOVERCONTENT subsection 133
RECONTENT subsection 134
RECONTENT2 subsection 134
REFAILURECONTENT subsection 134
RENEWEDCERT subsection 134
RESUCCESSCONTENT subsection 134
RETRIEVECONTENT2 subsection 134
RETURNCERT subsection 134
subsections 133
certificate templates 134
changing
to add application sections 227
copying 55
customizing
customization, additional first-time 161
customizing the OtherName field 171
minimally 160
retrofitting release changes 165
debug flag 123
description 123
editing
administration Web pages 178
end-user Web pages 160
to add application sections 227
INSERT sections 125, 156
INSERTs
-AdditionalHeadIE 126
-ChallengePassphrase 126
-ChallengePassphrase2 126
-preregok 127
-RecoverEmail 126
-RecoverEmail2 126
-renewkeysetIE 127
-renewkeysetNS 127
-renewrevokebad 127
-renewrevokeok 127
-requestbad 126
-requestok 126
-requestor 126
-requestor2 127
-returnp12cert 127
AltDomain 127
AltEmail 127
AltIPAddr 128
AltOther 128
AltURI 128
ChallengePassPhrase 128
pkiserv.tmpl *(continued)*
INSERTs *(continued)*
ClientName 128
CommonName 128
Country 129
DNQualifier 129
DomainName 129
Email (deprecated) 129
EmailAddr 129
ExtKeyUsage 129
HostIdMap 129
InstallCert 129
KeyProt 129
KeySize 130
KeyUsage 130
Label 130
Locality 130
Mail 130
NotAfter 130
NotBefore 130
NotifyEmail 131
Org 131
OrgUnit 131
OrgUnit2 131
PassPhrase 131
PostalCode 131
PublicKey 131
PublicKey2IE 132
PublicKey2NS 132
PublicKeyIE 131
PublicKeyNS 132
RecoverEmail 132
Requestor 132
Requestor2 132
returnbrowsercertIE 127
returnbrowsercertNS 127
returnpkcs10 127
Security*n* 132
SerialNumber 132
SignWith 132
StateProv 132
Street 132
Title 132
TransactionId 132
Uid 132
UnstructAddr 133
UserId 133
named fields 125
%%-renewrevokebad%% 134
%%-renewrevokeok%% 134
%%-requestok%% 142
%%returnp12cert%% 134
%%SelectCADomain%% 134
purpose 55
sections 123
substitution variables 124
TEMPLATE section 153
ADMINAPPROVE subsection 140
APPL subsection 138
CONSTANT subsection 139
CONTENT subsection 138
FAILURECONTENT subsection 142
PREREGISTER subsection 143
RETRIEVECONTENT subsection 142
RETURNCERT subsection 143

pkiserv.tmpl *(continued)*
TEMPLATE section *(continued)*
subsections 138
SUCCESSCONTENT
subsection 141
updating
customization, additional
first-time 161
customizing the OtherName
field 171
minimally 160
retrofitting release changes 165
PKIServ.xsd
in samples directory 499
PKISERVD
code sample 562
in PROCLIB 497
updating environment variables 57
PKISRVD
PKI Services daemon user ID 44, 78
pkitmpl.xml
description 179
in samples directory 499
purpose 55
pkitmpl.xml file
description 180
updating 198
PKITP
API called
CSSM_TP_PassThrough 430
certificate extensions supported 428
certificate policies supported 427
configuring 429
files 429
overview 425
PKI Services Trust Policy plug-in for
OCSF 425
pkitp_ivp 429, 430
pkitp.h 429
pkitp.so 429
pkitpsamp.c
description and directory 429
editing 434
sample code 435
PKIX standard 3, 4
PKIXEnroll.dll
signing 594
pkixgen.tmpl
creating 371
description 179
planning
for PKI Services 11
policy
notice number 72
Object ID for 72
usage 60
POLICY
subcomponent for message
logging 508
Policy1Notice1 (parameter in
pkiserv.conf) 72, 214
Policy1Notice2 (parameter in
pkiserv.conf) 73
Policy1Org (parameter in
pkiserv.conf) 72, 214
PolicyCritical (parameter in
pkiserv.conf) 71, 213, 427
PolicyName1 (parameter in
pkiserv.conf) 72, 213
PolicyRequired (parameter in
pkiserv.conf) 72, 213, 214, 427
ports
1443, port 91
443, port 91
80, port 91
for HTTP traffic 91
for SSL traffic 91
LDAP 87, 92
post requests
retrying for LDAP 100
post-processing
exit routine 273
EXPORT 280
GENCERT 276
GENRENEW 276
REQCERT 278
REVOKE 282, 284
postal code
DN field supported 568
field in end-user Web pages 311
PostalCode (named field in
pkiserv.tmpl) 131
postcerts utility 368
postExport method 294
postGenReqCert method 290
postGenReqRenew method 292
posting to LDAP
frequency 98
PostInterval (parameter in
pkiserv.conf) 98
postprocessing
automatic renewal 271
postQRecover method 298
postRevoke method 296
preExport method 293
preGenReqCert method 289
preGenReqRenew method 291
preprocessing
automatic renewal 270
exit routine 273
EXPORT 279
GENCERT 275
GENRENEW 275
QRECOVER 283
REQCERT 277
REVOKE 281
preQRecover method 297
PREREGISTER subsection (in TEMPLATE
section of pkiserv.tmpl) 143
preregistered (status of certificate
request) 338
preregistered requests
deleting 344
processing
single 340
preregistration
certificate for SCEP 136
enabling in pkiserv.conf 69
preregistration certificate
certificate
fields 147
prerequisite products
ICSF 13
prerequisite products *(continued)*
installing
skills 14
installing and configuring 19
LDAP server 12
OCEP 13
OCSF 13
planning for 12
Websphere Application Server 12
z/OS HTTP Server 12
prerequisites
for sysplex support 11
preRevoke method 295
printablecert substitution variable 124
private key
backing up 29
creating 29, 513
migrating to ICSF 29
storing
in ICSF 23
private key of PKI Services
rekeying 414
replacing 414
retiring 414
rollover 414
problems, diagnosing 449
PROCLIB 497
product libraries 497
products related to PKI Services 4
products, prerequisite
installing and configuring 19
profile
CA certificate, recovering 413
FACILITY class
IRR.PROXY.DEFAULTS 421, 422
IRR.DIGTCERT.ADD 419
IRR.DIGTCERT.EXPORT 419
IRR.DIGTCERT.GENCERT 419
IRR.DIGTCERT.GENRENEW 419
IRR.DIGTCERT.QRECOVER 419
IRR.DIGTCERT.REQCERT 419
IRR.DIGTCERT.REQRENEW 419
IRR.DIGTCERT.RESPOND 419
IRR.DIGTCERT.REVOKE 419
IRR.DIGTCERT.SCEPREQ 419
IRR.DIGTCERT.VERIFY 419
IRR.RPKISERV.PKIADMIN 420
KEYSMSTR class
LDAP.BINDPW.KEY 421
LDAPBIND class profile
defining 421
protect directive 93
PROTECTED attribute 512
protection directive 93
protocols
supported in PKI Services 7
province (field in end-user Web
pages) 311
public key cryptography
standards supported 7
public key data set (PKDS) 23
Public Key Infrastructure for X.509 3
publications
on CD-ROM and DVD xvi
softcopy xvi
PublicKey (named field in
pkiserv.tmpl) 131

PublicKey2IE (named field in pkiserv.tmpl) 132
PublicKey2NS (named field in pkiserv.tmpl) 132
PublicKeyIE (named field in pkiserv.tmpl) 131
PublicKeyNS (named field in pkiserv.tmpl) 132

Q

QRECOVER
  accesses required 419
  parameters
    preprocessing 283
  R_PKIServ function 512
  return codes
    post-processing 284
    preprocessing 283
QRecover class 299
QrecoverResultsList class 302

R

R_PKIServ callable service
  administrative functions 419
  controlling the use 417
  description 5
  end-user functions 417
  FACILITY class resources that protect 418, 420
  protected by FACILITY class resources 512
R_PKIServ SAF callable service
  exit routines 273
RA
  distinguished name, IKYSETUP variable 33
RA certificate
  creating 513
    using IKYSETUP 29
  label, IKYSETUP variable for 33
  locating 406
RA certificate and private key
  backing up 514
ra_backup_dsn (variable in IKYSETUP) 45
ra_dn (variable in IKYSETUP) 33
RA_label (parameter in pkiserv.conf) 78
ra_label (variable in IKYSETUP) 33
RACF
  administering PKI Services 403
  authorizing
    READ access 404
    users for inquiry access 403
  connecting members
    to group 403
    to new group 404
  deleting groups 404
  deleting members from group 403, 404
  setting up PKI Services 29
  use by PKI Services 5
RACF administration
  for PKI Services, ongoing 403
RACF administration *(continued)*
  for setting up PKI Services using IKYSETUP 29
    steps for 47
  using IKYSETUP 511
RACF administrator
  ongoing administration for PKI Services 403
  running IKYSETUP
    overview 29
    steps 49
  skills 16
  tasks 16
    IKYSETUP, running 29
    ongoing administration for PKI Services 403
    performed by IKYSETUP 511
    running IKYSETUP 29
    setting up PKI Services using IKYSETUP 47
RACF group
  providing access 29
RDB (request database) 355
RDN attribute 567
READ access
  authorizing 404
  IRR.DIGTCERT.GENCERT 515
  IRR.DIGTCERT.GENRENEW 513
  IRR.DIGTCERT.LIST 515
  IRR.DIGTCERT.LISTRING 515
  IRR.DIGTCERT.REQCERT 513
  IRR.DIGTCERT.REQRENEW 513
  IRR.RPKISERV.PKIADMIN 513
readonly substitution variable 125
ready message form for certificate 76
ReadyMessageForm (parameter in pkiserv.conf) 76
readymsg.form
  code sample 253
  copying 56
  customizing 257
  in samples directory 499
  purpose 55
recent activity (field in administration Web pages) 338
RECONTENT subsection (in APPLICATION sections of pkiserv.tmpl) 134
RECONTENT2 subsection (in APPLICATION sections of pkiserv.tmpl) 134
record level sharing (VSAM RLS) 105
RecoverEmail (named field in pkiserv.tmpl) 132
RecoverForm (parameter in pkiserv.conf) 78
recoverymsg.form
  code sample 255
  copying 56
  customizing 257
  purpose 55
recoverytmsg.form
  in samples directory 499
redirect directive 93
REFAILURECONTENT subsection (in APPLICATION section of pkiserv.tmpl) 134
registration authority (CA)
  certificate
    creating 29
reject (action on certificate request) 339
reject message form for certificate 76
rejected (status of certificate request) 338
rejected, user notified (status of certificate request) 338
RejectMessageForm (parameter in pkiserv.conf) 76
rejectmsg.form
  code sample 253
  copying 56
  customizing 257
  in samples directory 499
  purpose 55
relationship between certificate requests and certificates 355
relocate section
  variable data for type 80 SMF records 565
RemoveCompletedReqs (parameter in pkiserv.conf) 64
RemoveExpiredCerts (parameter in pkiserv.conf) 65
RemoveExpiredCertsAndKeys (parameter in pkiserv.conf) 64
RemoveInactiveReqs (parameter in pkiserv.conf) 64
renew (action for certificate) 349
Renew or revoke a browser certificate Web page 134
renewal of certificate, automatic
  exit routine processing for 267
renewal of certificates, automatic 137
RenewCertForm (parameter in pkiserv.conf) 77
renewcertmsg.form
  code sample 254
  in samples directory 499
  purpose 55
RENEWEDCERT subsection (in APPLICATION sections of pkiserv.tmpl) 134
REQCERT
  accesses required 419
  exit routine scenario use 285, 286
  parameters
    post-processing 278
    preprocessing 277
  R_PKIServ function 512
  return codes
    post-processing 278
    preprocessing 277
REQRENEW
  accesses required 419
  exit routine scenario use 286
  parameters
    post-processing 278
    preprocessing 277
  R_PKIServ function 512
  return codes
    post-processing 278
    preprocessing 277
request database
  description 103
request database (RDB) 355

request database records
viewing 372
requestor
alternate index
VSAM data set name for 62
Requestor (named field in pkiserv.tmpl) 132
requestor e-mail, changing for certificate 351
requestor name (field in administration Web pages) 338
requestor substitution variable 125
requestor's name (field in end-user Web pages) 314
Requestor2 (named field in pkiserv.tmpl) 132
requirements
LDAP directory server 567
prerequisite products 12
skills 13
sysplex support 11
RESPOND
accesses required 419
R_PKIServ function 512
restoring from backup
VSAM data sets
sample JCL 561
restrict_surrog (variable in IKYSETUP)
decision table 37
default value 41
description 41
RESTRICTED attribute 512
RESUCCESSCONTENT subsection (in APPLICATION sections of pkiserv.tmpl) 134
resume (action for certificate) 349
RETRIEVECONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 142
RETRIEVECONTENT2 subsection (in APPLICATION sections of pkiserv.tmpl) 134
RetryMissingSuffix (parameter in pkiserv.conf) 100
return codes
CSSM_TP_PassThrough 433
EXPORT
post-processing 280
preprocessing 279
GENCERT
post-processing 276
preprocessing 275
GENRENEW
post-processing 276
preprocessing 275
QRECOVER
post-processing 284
preprocessing 283
recording 449
REQCERT
post-processing 278
preprocessing 277
REQRENEW
post-processing 278
preprocessing 277
REVOKE
post-processing 282
return codes *(continued)*
REVOKE *(continued)*
preprocessing 281
returnbrowsercertIE 127
returnbrowsercertNS 127
RETURNCERT subsection (in APPLICATION sections of pkiserv.tmpl) 134
RETURNCERT subsection (in TEMPLATE section of pkiserv.tmpl) 143
returnpkcs10 127
revocation list, certificate authority (ARL) 223, 225
revocation request message for CMP, fields supported 384
revocation response message for CMP, fields supported 384
REVOKE
accesses required 419
parameters
post-processing 282, 284
preprocessing 281
R_PKIServ function 512
return codes
post-processing 282
preprocessing 281
revoke (action for certificate) 349
RevokeCert class 299
revoked (status of certificate) 349
revoked, expired (status of certificate) 349
RFC 4210, support for 379
RFC 4211, support for 379
RLS
enabling VSAM data sets for 107
MVS programmer task 16
preliminary steps for establishing 105
setting up, preliminary steps 105
RLS, VSAM record-level sharing 555
roadmaps
configuring PKI Services for IdenTrust compliance 575
for implementing PKI Services 17
roles of team members 14
rollover, PKI Services private key 414
rp message type for CMP, fields supported 384
RpkiservException class 302
rr message type for CMP, fields supported 384
RSA
signature algorithm
updating 215
RunMaintAtStart (parameter in pkiserv.conf) 75
runtime directory 117
runtime environment
configuring 53
runtime user ID
changing 168
for requesting certificates 168
for retrieving certificates 169
runtime-dir 11

S

S/MIME
certificate format 7
description of certificate 135
fields of certificate 144
supported standard 3
use of certificate 7
SAF
browser certificate
description 135
fields 147
key ring
creating 29, 513
KeyRing parameter 78
section (of pkiserv.conf)
default value 78
description 60
excerpt 60
parameter description 78
server certificate
description 135
fields 147
subcomponent for message logging 508
SAF key ring
IKYSETUP variable 44
SAMPLB 137
samples
_PKISERV_CA_DOMAIN 507
_PKISERV_MSG_LEVEL 508
browser certificate template 579
configuration directives 545, 547
configuration file 501
configuration file for IdenTrust compliance 579
directives 545, 547
environment variables file 510
expiringmsg.form 254
httpd.conf 545
httpd2.conf 547
IKYCDB2 548
IKYCVSAM 551
IKYRVSAM 555
IKYSBIND 559
IKYSETUP 517
IKYVBKUP 560
IKYVREST 561
JCL
IKYCVSAM 551
IKYRVSAM 555
IKYVBKUP 560
IKYVREST 561
PKISERVD 563
log data set from IKYSETUP 50
LOGREC data 452
pendingmsg.form 254
pkiserv.conf 501
pkiserv.envars 510
pkiserv.tmpl
APPLICATION section 149
INSERT section 156
TEMPLATE section 153
PKISERVD sample procedure 562
pkitpsamp.c 435
readymsg.form 253
recoverymsg.form 255
rejectmsg.form 253

samples *(continued)*
renewcertmsg.form 254
server certificate template for IdenTrust compliance 581
samples subdirectory 499
SAMPLIB 497
scenarios
exit routine
allowing only selected users to request certificates 285
maintaining customized certificate repository 286
providing customized TITLE 285
recover lost passphrase 287
renewal only within 30 days of expiration 286
SCEP certificates
requesting 331
SCEP preregistration certificate
fields 147
SCEPREQ
accesses required 419
sections of pkiserv.conf
CertPolicy section 59
General section 60
LDAP section 60
ObjectStore section 59
OIDs section 59
SAF section 60
secure
e-mail 3
secure connection with LDAP, establishing 88
Secure Multipurpose Internet Mail Extensions (S./MIME) 3
Secure Sockets Layer (SSL) 3, 7, 88
security topics for RACF
classroom courses xvi
Security*n* (named field in pkiserv.tmpl) 132
sendmail
configuring 24
not using default 57
requirement for 13
serial number
field in administration Web pages 338
field in end-user Web pages 311
serialno substitution variable 125
SerialNumber (named field in pkiserv.tmpl) 132
SERVAUTH class 405
server certificates
aliases 137
five-year PKI intermediate CA certificate 136
five-year PKI IPSEC server (firewall) certificate 136
five-year PKI SSL server certificate 136
generating 29
IdenTrust compliant template 581
installing 321, 322
one-year SAF server certificate 135
retrieving 321, 322
supported types 7
Server1 (parameter in pkiserv.conf) 99
setPassphrase method 299
setting up
PKI Services 27
var directory 84
setting up an ACL 88
settings
contained in pkiserv.conf 54
displaying log options 460
IKYP025I displays 485
log options, displaying 460
setTransactionid method 299
severe messages, logging 509
SharedPLEX (parameter in pkiserv.conf) 65
SharedVSAM (parameter in pkiserv.conf) 65
sharing control data sets (SHCDS) 105
SHCDS 105
shortcut keys 601
SIEALNKE 497
SigAlg1 (parameter in pkiserv.conf) 73, 217
signature algorithm
updating 215
signing key 23
signing_ca_label (variable in IKYSETUP) 46
SignWith (named field in pkiserv.tmpl) 132
Simple Certificate Enrollment Protocol (SCEP)
certificate
description 136
description of template 136
template description 136
certificate fields 147
enabling 246
enabling in pkiserv.conf 69
fingerprint checking 249
overview 246
preregistration 247, 248
overview 246
RA certificate
creating using IKYSETUP 29
record 247
request processing, overview 247
request states 247
steps for enabling 250
variables 247, 248
single certificate, processing 350
single request, processing 340
skill requirements 13
skills
DB2 database administratorr 16
ICSF programmer 14
installing PKI Services 16
installing prerequisite products 14
LDAP programmer 15, 16
MVS programmer 16
OCEP programmer 15
OCSF programmer 15
RACF administrator 16
UNIX programmer 15, 16
Web server programmer 15, 17
smart card, certificate for 136
smart cards 3
SMP/E 11
SMS base configuration 106
space considerations
for ICL, using DB2 111
for ICL, using VSAM 104
for object store, using DB2 111
for object store, using VSAM 105
for VSAM data sets 104
SPANNED statements 106
specifying expiration 42
square brackets (in substitution variables) 124
SSL
certificate
creating 29, 516
use 7
delivering certificates through 3
supported standards 7
two modes 91
with client authentication operating mode of z/OS HTTP Server 516
without client authentication operating mode of z/OS HTTP Server 516
SSL (Secure Sockets Layer) 88
sslclientauth directive 93
sslmode directive 92, 93
sslport directive 92, 93
SSLring 34
SSLX500CARoots directive 93
SSLX500Host directive 93
SSLX500Password directive 93
SSLX500Port directive 93
SSLX500UserID directive 93
standards
certificate extensions supported 8
LDAP 7
public key cryptography, supported 7
started procedure for PKI Services
associating user ID with 29
starting
PKI Services 117
PKI Services daemon 117
state (field in end-user Web pages) 311
StateProv (named field in pkiserv.tmpl) 132
status
alternate index
VSAM data set name for 62
statuses
certificate requests 338
certificates 348
STDERR_LOGGING 508
STDOUT 272
EXPORT
preprocessing 279
GENCERT
post-processing 276
preprocessing 275
GENRENEW
post-processing 276
preprocessing 275
REQCERT
post-processing 278
preprocessing 277
REQRENEW
post-processing 278

STDOUT *(continued)*
REQRENEW *(continued)*
preprocessing 277
STDOUT_LOGGING 508
steps
access to administration pages, changing 178
accessing
administration home page 335
end-user Web pages 308
adding
application sections 227
adding new certificate template 167
administering HostIdMappings extensions 405
administration Web pages
changing access to 178
customizing 177
approving single request 340
authorizing users for inquiry access 404
bind passwords encrypted for LDAP
IRR.PROXY.DEFAULTS profile 422
LDAPBIND class 421
building sample application 434
CA certificate
renewing 411
CA certificate profile, recovering 413
CA certificate, locating 406
certificate templates file
customization, additional first-time 161
customization, minimal 160
customizing the OtherName field 171
retrofitting release changes 165
certificates, locating 406
changing
administration Web pages 177
end-user Web pages 160, 161, 165
environment variables 58
fields in requests 340, 342
pkiserv.conf configuration file 60
runtime user ID for requesting certificates 168
runtime user ID for retrieving certificates 169
signature algorithm 216
changing PKI Services from a self-signed CA to an intermediate CA 409
configuration file, updating 60, 79
configuring
ICSF 23
LDAP 21
PKITP 430
z/OS HTTP Server 20
copying files
expiringmsg.form 56
pendingmsg.form 56
pkiserv.conf 56
pkiserv.envars 56
pkiserv.tmpl 56
readymsg.form 56
recoverymsg.form 56
rejectmsg.form 56
steps *(continued)*
creating
application sections 227
CertificatePolicies extension 213
ICL data sets 106
VSAM object store 106
customizing
administration Web pages 177
distribution point CRLs 220
e-mail notifications 257
end-user Web pages 160, 161, 165
pkiserv.tmpl 160, 161, 165
deleting
multiple certificates 352
selected certificates 353
single certificate 350
single request 340
e-mail notifications, customizing 257
enabling Simple Certificate Enrollment Protocol (SCEP) 250
encrypted passwords for LDAP servers 421
environment variables, updating 58
establishing your CA and RA certificates 513
gskkyman for certificate store 571
HostIdMappings extensions, administering 405
ICL data sets, creating 106
IdenTrust compliance
adjusting general settings 577
creating templates 577
modifying pkiserv.conf 576
IKYSETUP, using 47
inquiry access, authorizing users for 404
installing
ICSF 23
LDAP 21
z/OS HTTP Server 20
intermediate certificate authority, changing PKI Services to 409
invoking the certificate validation service 434
key ring, locating 406
LDAP
schema.user.ldif, updating 88
section of PKI Services configuration file, tailoring 98
updating schema.user.ldif 88
LDAP bind passwords, encrypted
IRR.PROXY.DEFAULTS profile 422
LDAPBIND class 421
LDAP servers, encrypted passwords for 421
locating
CA certificate 406
key ring 406
PKI Services certificates 406
RA certificate 406
modifying single request 340
passwords, encrypted LDAP binding
IRR.PROXY.DEFAULTS profile 422
LDAPBIND class 421
steps *(continued)*
performing RACF administration using IKYSETUP 47
PKI Services certificate authority certificate, renewing 411
PKI Services certificates, locating 406
PKI Services daemon
starting 117
stopping 119
PKI Services RA certificate, renewing 412
pkiserv.conf
copying 55
updating 60, 79
pkiserv.tmpl
copying 56
customization, additional first-time 161
customization, minimal 160
customizing the OtherName field 171
retrofitting release changes 165
PKITP, configuring 430
pkitpsamp.c 434
preregistering a SCEP client 331
processing
multiple certificates 352
multiple requests through searches 344
selected certificates 353
selected requests 346
single certificate 350
single request 340
RA certificate
renewing 412
RA certificate, locating 406
RACF administration using IKYSETUP 47
recovering a CA certificate profile 413
rejecting single request 340
renewing
certificate 324
PKI Services certificate authority certificate 411
PKI Services RA certificate 412
requesting a certificate 314
retrieving a certificate
from bookmarked Web page 321
from PKI Services home page 322
retrofitting changes into certificate templates 165
revoking
certificate (by user) 327
multiple certificates 352
selected certificates 353
single certificate 350
RLS
enabling VSAM data sets for 107
preliminary steps for establishing 105
running IKYSETUP 47
searching for requests 344
sendmail configuration, testing 25
setting up the var directory 84
Simple Certificate Enrollment Protocol (SCEP) enabling 250

steps *(continued)*
starting
PKI Services 117
PKI Services daemon 117
z/OS HTTP Server 94
stopping PKI Services daemon 119
suspending
certificate (by user) 327
tailoring LDAP section of PKI Services configuration file 98
testing sendmail configuration 25
updating
configuration file 60, 79
environment variables 58
exit code sample for CGIs 273
exit routine code sample 268
LDAP section of pkiserv.conf 101
pkiexit.c 268
pkiexit.c for CGIs 273
pkiserv.conf 60, 79
signature algorithm 216
single request 340
z/OS HTTP Server configuration files 91, 92
user ID for requesting certificates, changing 168
user ID for retrieving certificates, changing 169
using
gskkyman 571
IKYSETUP 47
using encrypted passwords for LDAP servers 421
creating IRR.PROXY.DEFAULTS profile 422
creating LDAPBIND class profile 421
var directory, setting up 84
viewing Web pages 117
VSAM object store, creating 106
z/OS HTTP Server configuration files, updating 91
STOP command 119
stopping
PKI Services 117, 119
storage needs
for ICL, using DB2 111
for ICL, using VSAM 104
for object store, using DB2 111
for object store, using VSAM 105
STORCLAS statements 106
store
creating 106
storing
certificate requests 355
certificate revocation lists 5
certificates 5
private key
in ICSF 5
street (field in end-user Web pages) 311
Street (named field in pkiserv.tmpl) 132
subcomponent level
for logging 508
subdirectory
ActiveX 497
bin 498
include 498
subdirectory *(continued)*
lib 498
pkiijsp 498
PKIServ 498
samples 499
SubjectAltName (certificate extension) 428
SubjectKeyIdentifier (certificate extension) 428
subordinate certificate authority
using PKI Services as 408
subsections in certificate templates, summary 143
substitution variables
base64cert 124, 127
browsertype 124
cadomain 124
errorinfo 124
iecert 124
keyid 124
optfield 124
p12cert 124
pkiserv.tmpl 124
printablecert 124
readonly 125
requestor 125
serialno 125
tmplname 125
transactionid 125
SUCCESSCONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 141
suffix
LDAP 88
summary of changes xxiii
surrog (variable in IKYSETUP) 46
surrog_uid (variable in IKYSETUP) 33
surrogate operation
setting up 29, 516
surrogate user ID 168
creating 29
IKYSETUP variable 46
PKISERV 46, 512
UID, IKYSETUP variable 33
suspend (action for certificate) 349
suspended (status of certificate) 349
SYS1.CSSLIB 41
SYS1.LINKLIB 41
SYS1.LOGREC 449
SYS1.SAMPLIB(IKYSETUP) 29
SYSOUT
records, contents 458
viewing information 455
sysplex support
daemon, PKI Services, starting 117
PKI Services daemon, starting 117
pkiserv.conf
updating, overview 58
updating, steps for 79
prerequisites 11
requirements 11
RLS
enabling VSAM data sets for 107
preliminary steps for establishing 105
SharedPLEX
description 65
sysplex support *(continued)*
SharedPLEX *(continued)*
updating 80
starting PKI Services daemon 117
updating pkiserv.conf 58
updating SharedPLEX 80
system architecture diagram 5

T

task roadmaps
configuring PKI Services for IdenTrust compliance 575
for implementing PKI Services 17
tasks
(noun, gerund phrase)
steps 589
allowing Websphere users to renew and revoke browser certificates
steps 189
application domain, creating
steps 229
authorization to PKI Services functions, giving to users
steps 185
automatic certificate renewal, setting up
steps 258
backing up the VSAM data sets
steps 109
CA certificate, setting the expiration date
steps 42
CAPICOM, installing on Windows system
steps 586
certificate for CMP requester, setting up in RACF database
steps 388
configuring Internet Explorer to trust PKI Services
steps 591
CRL, enabling support for large
steps 224
custom extension, adding to a certificate template
steps 262
EAR file, updating
steps 199
encryption of keys on CMP requests using key ring, setting up PKI Services for
steps 390
ICL, converting from VSAM to DB2
steps 113
ICL, creating DB2 objects for
steps 111
installer program for ActiveX controls, building
steps 595
installing PKI Services ActiveX program
steps 587
installing the PKI Services CA certificate on a Windows Vista system
steps 592

tasks *(continued)*
JSP files, deploying to a Websphere application server
steps 200
object store, converting from VSAM to DB2
steps 113
object store, creating DB2 objects for
steps 111
PKI Services ActiveX programs, signing
steps 594
PKI Services Web application, implementing using the JSPs
roadmap 184
preparing to implement the PKI Services Web application using the JSPs
steps 185
preparing Windows and Internet Explorer for use with end-user Web application
roadmap 585
recovering a certificate whose keys were generated by PKI Services
steps 328
restoring the VSAM data sets
steps 110
retrieving a PKI generated key certificate
steps 323
setting up key generation for certificate requests
steps 259
Web server certificate, setting the expiration date
steps 42
TCPIP.SEZALOAD 41
TDBM 22
specifying password as entry 87, 92
TDBM backend 13
team members 14
TEMPLATE section of pkiserv.tmpl
ADMINAPPROVE subsection 140
APPL subsection 138
CONSTANT subsection 139
CONTENT subsection 138
examining contents 153
FAILURECONTENT subsection 142
PREREGISTER subsection 143
RETRIEVECONTENT subsection 142
RETURNCERT subsection 143
subsections 138
SUCCESSCONTENT subsection 141
templates
adding 167
converting CGI file to XML file 370
customizing
additional first-time changes 161
minimal 160
OtherName field 171
retrofitting release changes 165
description 180
for JSPs 179
updating 198
XML 180, 198
templates *(continued)*
validating with TemplateTool utility 370
TemplateTool utility 370
threads
created at initialization 74
TID
alternate index
VSAM data set name for 62
time interval
before certificate expiration 70
between certificate revocation lists 68, 73
certificate suspension grace period 71
for scanning for items to post 98
scanning database for approved requests 67
warning message about certificate expiration 70
time period
in ICL before automatic deletion 64, 65
in object store before automatic deletion 64
time zone
default 118
TimeBetweenCRLs (parameter in pkiserv.conf) 73
timeout value
for PKI Services exit 74
Title 132
title (field in end-user Web pages) 311
TKDS
time period before automatic deletion expired certificates 64
TKDS (token data set) 23
specifying a token in for PKI Services 78
TLS (Transport Layer Security) 88
tmplname substitution variable 125
token data set
time period before automatic deletion expired certificates 64
token data set (TKDS) 23
specifying a token in for PKI Services 78
TokenName (parameter in pkiserv.conf) 78
TPOLICY
subcomponent for message logging 508
tracing CMP
environment variables 396
transaction ID
field in administration Web pages 338
field in end-user Web pages 314
TransactionId (named field in pkiserv.tmpl) 132
transactionid substitution variable 125
Transport Layer Security (TLS) 88
true name of certificate templates 137
trust level for PKI Services 405
Trust Policy
API called
CSSM_TP_PassThrough 430
Trust Policy *(continued)*
overview 425
two-year PKI Authenticode code signing server certificate
fields 145
two-year PKI Authenticode–code signing server certificate
description 136
two-year PKI browser certificate for authenticating to z/OS
description 135
fields 144
two-year PKI Windows logon certificate
description 136
fields 144
type 80 SMF record
table of event codes and qualifiers 565
table of relocate section variable data 565
TZ environment variable 57

U

Uid (named field in pkiserv.tmpl) 132
unencrypted, LDAP bind passwords 420
Uniform Resource Identifier (URI)
Certification Practice Statement 66
uniform resource identifier (URI), alternate
field in end-user Web pages 312
UNIX programmer
skill planning 15, 16
task planning 15, 16
tasks
configuring sendmail 24
configuring UNIX runtime environment 53
LDAP section of pkiserv.conf, updating 97
runtime environment, configuring 53
sendmail, configuring 24
UNIX runtime environment, configuring 53
updating LDAP section of pkiserv.conf 97
UNIX runtime environment
configuring 53
unix_sec (variable in IKYSETUP)
decision table 37
default value 42
description 42
UnstructAddr (named field in pkiserv.tmpl) 133
unstructured address
field in end-user Web pages 312
unstructured name
field in end-user Web pages 312
UPDATE access
IRR.DIGTCERT.ADD 513
IRR.RPKISERV.PKIADMIN 513
updating
certificate templates file
minimal 160
retrofitting changes 165

updating *(continued)*
pkiserv.tmpl
minimal 160
URI
containing CPS 66
URI, alternate 312
usage policy 60
user ID
associating with PKI Services started procedure 29, 511
changing
for retrieving certificates 169
requesting certificates 168
field in end-user Web pages 312
PKI Services daemon 78
runtime
changing 168
user ID, surrogate
creating 29
user notifications
copying files 56
customizing forms 257
UserExit class 288
UserExitException class 299
UserId (named field in pkiserv.tmpl) 133
userId directive 92, 93
UserNoticeText1 (parameter in pkiserv.conf) 73, 214
userPassword attribute 87, 92
uses of PKI Services 3
utilities 357
createcrls 358
iclview 359
pkiprereg 363
postcerts 368
subdirectory 498
TemplateTool 370
vosview 372
vsam2db2 377

V

var directory
setting up 84
variables
in IKYSETUP REXX exec
change based on setup 34
change optionally 42
change required 31
configurable section 30
in notification forms
%%cadomain%% 255
%%dn%% 255
%%notafter%% 255
%%pendreqlist%% 255
%%printcert%% 255
%%quicklink%% 256
%%recoverylink%% 256
%%recoverylist%% 256
%%rejectreason%% 256
%%requestor%% 256
%%transactionid%% 256
variables-dir 11
verbose diagnostic messages, logging 509
VERIFY
accesses required 419
VERIFY *(continued)*
R_PKIServ function 512
virtual private network (VPN) devices 3
Vista
installing CAPICOM 586
VOL statements 106
vosview
output 374
vosview utility 372
VPN devices 3
VSAM
data set name for ICL data 63
data set name for ICL requestor alternate index 63
data set name for ICL status alternate index 63
data set name for object store base cluster 61
data set name for object store requestor alternate index 62
data set name for object store status alternate index 62
data set name for object store TID alternate index 62
RLS
enabling data sets for 107
preliminary steps for establishing 105
VSAM data sets
backing up 109
sample JCL 560
creating 103
not using RLS 551
using RLS 106, 107, 555
giving administrators access to 29
restoring from backup 110
sample JCL 561
RLS, enabling for 107
VSAM files for ICL and ObjectStore
utility program to convert to DB2 tables 377
VSAM ICL data set records
viewing 359
VSAM object store data set records
viewing 372
VSAM object store, creating 106
vsam2db2 utility 377
vsamhlq (variable in IKYSETUP) 46

W

warning message before certificate expiration 70
warning messages, logging 509
Web pages
accessing 308, 335
Web pages, administration
customizing 175
Web pages, end user
customizing 123
fields 311
using 307
Web server
daemon user ID, IKYSETUP variable 47
distinguished name, IKYSETUP variable 34
Web server *(continued)*
SAF key ring, IKYSETUP variable for 34
Web server certificate 42
expiration date, IKYSETUP variable 46
label, IKYSETUP variable 47
life span, IKYSETUP variable 47
Web server programmer
skills 15, 17
tasks 15, 17
configuring z/OS HTTP Server 19
installing z/OS HTTP Server 19
starting z/OS HTTP Server 94
updating z/OS HTTP Server configuration files 91
z/OS HTTP Server, installing and configuring 19
z/OS HTTP Server, starting 94
z/OS HTTP Server, updating configuration files 91
updating the z/OS HTTP Server configuration files 91
Web server programmer
installing and configuring z/OS HTTP Server 19
web_dn (variable in IKYSETUP) 34
web_expires (variable in IKYSETUP) 46
changing value of 42
web_exyears (variable in IKYSETUP) 47
changing value of 42
web_label (variable in IKYSETUP) 47
web_ring (variable in IKYSETUP) 34
webserver (variable in IKYSETUP) 47
Websphere application server
deploying JSP file to 200
Websphere Application Server
installing and configuring 21
requirement for 12
use by PKI Services 5
Websphere users
authorizing to renew and revoke browser certificates 189
authorizing to use PKI Services functions 185
WEBSRV user ID 47
Windows
installing CAPICOM 586
installing PKI Services CA certificate on 592
setting up to run end-user Web application 585
Windows logon certificate
description 136

X

X.509v3 certificates 8
XML template file
description 180, 198

Z

z/OS
  PKI browser certificate for
    authenticating to
      description 135
      fields 144
  PKI Windows logon certificate
      fields 144
z/OS Basic Skills information
  center xviii
z/OS HTTP Server
  configuration files
      updating 91
  environment variables for CMP 392
  installing and configuring 20
  operating modes PKISERV
    requires 515
  requirement for 12
  setting up for surrogate
    operation 29, 516
  starting 94
  updating configuration 91
  use by PKI Services 5
z/OS product libraries
  AIEALNKE 497
  APROCLIB 497
  ASAMPLIB 497
  PROCLIB 497
  SAMPLIB 497
  SIEALNKE 497
z/OS UNIX level security 37
zip code (field in end-user Web
  pages) 311



Product Number: 5694-A01

Printed in USA

SA22-7693-13

