Cryptographic Services PKI Services
Guide and Reference
Cryptographic Services PKI Services Guide and Reference
Twelfth edition, September 2009

This is a major revision of SA22-7693-10. This edition applies to Version 1 Release 11 of z/OS (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation
MHVRCFS, Mail Station P181
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrdfs@us.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
• Title and order number of this document
• Page number or topic related to your comment

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

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Tables ........................................................................ xi

Figures ...................................................................... xiii

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

Summary of changes .................................................... xxiii

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 ............................................... 6
Supported certificate types ........................................ 7
Supported certificate fields and extensions ....................... 8

Chapter 2. Planning your implementation ....................... 9
Installing PKI Services ............................................. 9
Requirements for sysplex support ................................ 9
Determining prerequisite products ................................ 10
   z/OS HTTP Server ........................................... 10
   Websphere Application Server (optional) .................. 10
   LDAP directory server ....................................... 10
   OCSF (optional) ................................................ 11
   ICSF (optional) ................................................. 11
   sendmail (optional) ............................................ 11
   OCEP (optional) ............................................... 11
Identifying skill requirements ...................................... 11
   Team members ................................................ 11
   Skills for setting up prerequisite products ................. 12
   Skills for setting up PKI Services ......................... 13
Creating an implementation plan ................................ 15
   Task roadmap for implementing PKI Services .......... 15

Chapter 3. Installing and configuring prerequisite products .... 17
Tasks to perform before setting up PKI Services ............... 17
   Installing and configuring the z/OS HTTP Server .......... 17
   Installing and configuring Websphere Application Server for z/OS 19
   Installing and configuring LDAP ........................... 19
   Installing and configuring ICSF (optional) ................. 21
Part 2. Configuring your system for PKI Services

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

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

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

Chapter 7. Updating z/OS HTTP Server configuration and starting the server
Steps for updating the z/OS HTTP Server configuration files
Steps for starting the z/OS HTTP Server

Chapter 8. Tailoring the PKI Services configuration file for LDAP
Excerpt of LDAP section
Storing information for encrypted passwords for your LDAP servers
Steps for tailoring the LDAP section of the configuration file

Chapter 9. Creating VSAM data sets
Planning VSAM storage requirements
Determining storage needs for ICL
Determining storage needs for the object store
(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
Tuning VSAM performance
(Optional) steps for adding VSAM buffer space
Back up and restoring the VSAM data sets
Steps for backing up the VSAM data sets
Steps for restoring the VSAM data sets

Chapter 10. Starting and stopping PKI Services
Steps for starting the PKI Services daemon
Part 3. Customizing PKI Services

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

Contents of the pkiserv.tmpl certificates templates file
  What are substitution variables? .......................... 102
  What are named fields? ............................... 103
  INSERT sections ..................................... 103
  The APPLICATION sections ............................. 110
  Templates that PKI Services provides .................. 112
  TEMPLATE sections .................................. 114
Summary of fields in certificate templates .................. 121
Examining the pkiserv.tmpl file .......................... 126
  Examining the APPLICATION section ................. 126
  Examining the TEMPLATE section .................... 130
  Examining the INSERT section ....................... 133
Relationship between CGIs and the pkiserv.tmpl file .... 135
Steps for performing minimal customization ............... 137
Steps for additional first-time customization ............ 138
Steps for retrofitting release changes into the PKI Services certificate templates .................. 142
Locating code for customizing end-user Web pages .......... 143
Steps for adding a new certificate template ............... 144
Changing the runtime user ID ........................... 145
  Steps for changing the runtime user ID for requesting certificates ............................... 146
  Steps for changing the runtime user ID for retrieving certificates ......................... 146
Customizing the OtherName field ........................ 147
  Steps for customizing the sample AltOther_<OID>INSERTs .................. 148

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

CGIs for administration Web pages ........................ 151
Customizing the administration Web pages ................ 153
Steps for customizing the administration Web pages .... 154
Changing the runtime behavior for accessing administration pages ............................... 154
  Steps for changing control of access to administration pages ................................. 155

Chapter 13. Implementing the Web application using Java server pages

Certificate templates files used with JSPs .................. 157
  Examining the pkitmpl.xml file ........................ 157
Roadmap for implementing the PKI Services Web application using JSPs ............................ 162
  Steps for preparing to implement the PKI Services Web application using JSPs ................. 162
  Giving Websphere users authorization to use PKI Services functions ........................... 163
  Allowing Websphere users to renew and revoke browser certificates ............................ 166
  Customizing the PKI Services Web application ........................................... 176
Directories for JSP files .................................. 182

Chapter 14. Advanced customization

Scaling for high volume installations ........................ 187
Using certificate policies ................................ 188
  Steps for creating the CertificatePolicies extension on a global basis ......................... 189
  Steps for creating the CertificatePolicies extension on a template basis ........................ 190
Updating the signature algorithm .......................... 191
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix E. Accessibility</td>
<td>509</td>
</tr>
<tr>
<td>Using assistive technologies</td>
<td>509</td>
</tr>
<tr>
<td>Keyboard navigation of the user interface</td>
<td>509</td>
</tr>
<tr>
<td>z/OS information</td>
<td>509</td>
</tr>
<tr>
<td>Notices</td>
<td>511</td>
</tr>
<tr>
<td>Policy for unsupported hardware</td>
<td>512</td>
</tr>
<tr>
<td>Programming interface information</td>
<td>513</td>
</tr>
<tr>
<td>Trademarks</td>
<td>513</td>
</tr>
<tr>
<td>Bibliography</td>
<td>515</td>
</tr>
<tr>
<td>Index</td>
<td>517</td>
</tr>
</tbody>
</table>
### Tables

1. Basic components of PKI Services and related products .......................... 4
2. Types of certificates you can request .............................................. 7
3. File system directory variables .......................................................... 9
4. Tasks and skills needed for installing prerequisite products ..................... 12
5. Roles, tasks, and skills for setting up PKI Services ................................ 13
6. Task roadmap for implementing PKI Services ....................................... 15
7. z/OS HTTP Server information you need to record .................................. 19
8. LDAP information you need to record .................................................. 20
9. IKYSETUP—Structure and divisions ..................................................... 28
10. IKYSETUP variables whose values must change ................................... 29
11. Decision table for key_backup ............................................................. 32
12. Decision table for key_type ............................................................... 33
13. Decision table for restrict_surrog ....................................................... 33
14. Decision table for unix_sec ............................................................... 34
15. IKYSETUP variables you might want to change depending on setup ............. 34
16. IKYSETUP variables you can optionally change ..................................... 37
17. Deciding which files to copy and change ............................................. 45
18. Information needed for updating the configuration file ............................ 52
19. LDAP information you need for tailoring LDAP configuration ..................... 73
20. Summary of configuration and usage of each Web server instance ................. 77
21. LDAP information you need for tailoring z/OS HTTP Server configuration .... 78
22. Information needed for updating the LDAP section of the configuration file .. 82
23. VSAM RLS information you need to record .......................................... 89
24. Structure and main divisions of the certificate template file (pkierv.tmp) ....... 101
25. Substitution variables ................................................................. 102
26. INSERTs that are common HTML for Web page content ......................... 104
27. Named fields in INSERT sections ....................................................... 105
28. Subsections of the APPLICATION sections ......................................... 111
29. Certificate templates PKI Services provides ...................................... 112
30. Names, aliases, and nicknames of certificate templates ........................... 115
31. Summary of subsections in certificate templates .................................... 121
32. Summary of fields for PKI browser certificate templates ......................... 122
33. Summary of fields for PKI server certificate templates ........................... 123
34. Summary of fields for SAF, SCEP, and PKI generated key certificate templates 125
35. CGI actions for end-user Web pages .................................................. 136
36. Location of code for various Web pages .............................................. 143
37. CGI actions for administrative Web pages .......................................... 152
38. Task roadmap for implementing the PKI Services Web application using the JSPs 162
39. JSP files in the Customers directory .................................................. 183
40. JSP files in the PKIServ directory ..................................................... 184
41. JSP files in the mod.inc directory ..................................................... 185
42. Task roadmap for creating multiple application domains ......................... 201
43. Task roadmap for adding a new CA domain ....................................... 207
44. Multiple CA domains: Worksheet #1 for recording progress adding new CA domains 208
45. Multiple CA domains: Worksheet #2 for planning your domain names ......... 209
46. Multiple CA domains: Worksheet #3 for planning your RACF identifiers, z/OS UNIX identifiers, and VSAM data set names .............................................. 210
47. IKYSETUP log information you need to record .................................... 213
48. Information you need to enable Simple Certificate Enrollment Protocol (SCEP) 223
49. Descriptions of variables for forms ................................................... 228
50. Summary of substitution variables in forms ....................................... 229
51. Files for the pkiexit.c exit routine .................................................... 235
52. Values of arguments for pre- and post-processing ................................ 236

© Copyright IBM Corp. 2001, 2009
### Table of Contents

- Values of arguments for pre- and post-processing ........................................ 240
- Package and class summary for JSP exit processing ...................................... 255
- Methods in class UserExit .................................................................................. 255
- Types of certificates you can request ............................................................... 273
- Summary of fields in end-user Web pages ......................................................... 276
- Summary of fields in the administration pages ................................................. 302
- Statuses of certificate requests .......................................................................... 302
- Summary of actions to perform on requests and required status ....................... 303
- Searches to display certificate requests ............................................................ 308
- Status of certificates ......................................................................................... 313
- Summary of actions to perform and required status to do so ......................... 314
- Searches to display certificates ........................................................................ 316
- Names, aliases, and nicknames of certificate templates .................................... 324
- Names, aliases, and nicknames of certificate templates .................................... 328
- List of valid field names for use in the preregistration record as input to the pkiprereg utility 332
- Information you need for locating your PKI Services certificates and key ring ..... 344
- Information you need for establishing PKI Services as an intermediate CA ....... 347
- Information you need for renewing your PKI Services certificate authority certificate .............................................................. 349
- Information you need for renewing your PKI Services RA certificate ............... 350
- Information you need for recovering a CA certificate profile ........................... 351
- Summary of access authorities required for PKI Services requests .................. 359
- Sequence of validation stages for PKITP certificate revocation checking ......... 370
- Summary of information about important files for PKITP ............................... 371
- PKI Services OCSF Trust Policy (PKITP) error codes ..................................... 375
- LOGREC data for PKI Services ....................................................................... 391
- Summary of information about important files ................................................. 403
- Meaning of fourth character in message number ............................................ 405
- Meaning of eighth character in message number ............................................ 405
- Files contained in subdirectories ....................................................................... 437
- Access required if you plan to have an administrator approve certificate requests .............................................................. 451
- Access required if you plan to use auto-approval ............................................ 451
- FACILITY class access needed for protecting administrative functions .......... 451
- Access PKISRVD needs to use RACF certificates ........................................... 453
- SMF event code and event code qualifier for PKI Services .............................. 485
- SMF data elements of the extended-length relocate section for PKI Services .... 485
- LDAP objectclasses and attributes that PKI Services sets ............................... 487
- Relationship of named fields to LDAP attributes and object identifiers ........... 488
- Tasks to perform to set up a Windows Vista system and the Internet Explorer browser to work with PKI Services .............................................................. 505
Figures

1. Component diagram of a typical PKI Services system ........................................... 6
2. Flowchart of the process of updating IKYSETUP ................................................. 41
3. Partial listing of the AltOther_1_2_3_4_6 sample INSERT showing the lines you are most likely to customize ........................................................... 151
4. A certreq_template tag ......................................................................................... 158
5. Configuring Websphere for application security .................................................. 163
6. Configuring Websphere for application security using SAF authorization .................. 164
7. Renewing or revoking a browser certificate ......................................................... 166
8. Websphere SSL certificate and key management page ........................................ 168
9. Websphere page for creating a new key store ...................................................... 169
10. Viewing the signer certificate .............................................................................. 169
11. Websphere new SSL configuration page .............................................................. 170
12. Websphere quality of protection settings page .................................................... 171
13. Defining a new port .............................................................................................. 172
14. Assigning a port to the application server ............................................................ 172
15. Assigning your new port to the application server .............................................. 173
16. Selecting Web container transport chains ........................................................... 174
17. Selecting a transport chain template .................................................................... 174
18. All existing transport chains ................................................................................ 175
19. Setting the SSL inbound channel properties ....................................................... 175
20. A portion of the JSP file notbefore.jsp, without customization ................................ 176
21. A portion of the JSP file notbefore.jsp, with customization ................................... 177
22. The Websphere Shared Libraries window ............................................................. 179
23. Message indicating that changes have been made to your local configuration......... 179
24. Window for specifying the EAR file ..................................................................... 180
25. Application properties page .................................................................................. 181
26. Shared library mapping for modules page .......................................................... 181
27. Adding the PKI Services shared library's location to the server's library path ........... 182
28. A sample CRLDistributionPoints extension for a certificate authority (CA) certificate 200
29. Illustration of two CA domains—one for employees and one for customers—administered by a single shared administrator who administers both domains ................................................... 205
30. Sample of readymsg.form .................................................................................... 226
31. Sample of rejectmsg.form ................................................................................... 227
32. Sample of expiringmsg.form ................................................................................. 227
33. Sample of renewcertmsg.form .............................................................................. 227
34. Sample of pendingmsg.form ................................................................................ 227
35. Sample of recoverymsg.form .............................................................................. 228
36. PKI Services end-user home page for certificate generation ............................... 275
37. The certificate popup window for installing the CA certificate ............................... 276
38. One-year SSL browser certificate request form .................................................... 282
39. Supplying the PKCS #10 certificate request for a server or device certificate .......... 283
40. Successful request displays transaction ID ............................................................. 284
41. Web page to retrieve your certificate .................................................................... 285
42. Successful request for a one-year PKI generated key certificate ............................. 285
43. Browser certificate installation Web page ............................................................ 286
44. Server certificate installation Web page ............................................................... 287
45. E-mail notification that your PKI generated key certificate is ready for pickup .......... 288
46. Web page for retrieving a PKI generated key certificate ........................................ 289
47. Window asking whether to open or save the PKCS #12 package .......................... 289
48. Popup window listing certificates ....................................................................... 290
49. Renew or revoke a certificate Web page ............................................................. 291
50. Web page requesting answers to security questions when you have forgotten the passphrase ...................................................................................... 293
51. Web page listing certificates that can be recovered ............................................. 294
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:
Preface

- 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 9 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 17 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 25 describes the tasks your team members need to perform to configure PKI Services.
- Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27 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 45 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 73 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 77 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 81 explains how the UNIX programmer updates the LDAP section of the PKI Services configuration file.
- Chapter 9, “Creating VSAM data sets,” on page 87 explains how the MVS programmer creates VSAM data sets.
- Chapter 10, “Starting and stopping PKI Services,” on page 95 explains how the MVS programmer starts and stops the PKI Services daemon.
- Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” on page 101 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 151 provides an overview of the CGI scripts and explains how to customize the administration Web pages.
- Chapter 14, “Advanced customization,” on page 187 explains how to use certificate policies, the signature algorithm, and the PKI exit.
- Chapter 16, “Using the end-user Web pages,” on page 273 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 299 shows the administration Web pages and explains how to process certificate requests and certificates.

Part 5, “Administering security for PKI Services,” on page 339 explains how to perform many RACF administration tasks needed for PKI Services, such as authorizing users, administering extensions, locating your PKI Services certificate and key ring, and so on.

Part 6, “Using the certificate validation service,” on page 365 describes how to setup and use the PKI Services Trust Policy (PKITP) plug-in for OCSF.

Part 7, “Troubleshooting,” on page 389 explains using logs and utilities:
- Chapter 21, “Using information from SYS1.LOGREC,” on page 391 describes SYS1.LOGREC — which is used to record unusual runtime events, such as an exception.
- Chapter 22, “Using information from the PKI Services logs,” on page 397 discusses using the PKI Services logs to debug problems and explains how to change logging options and display log options settings.
- Chapter 18, “Using PKI Services utilities,” on page 321 explains how to use PKI Services utilities: vosview displays the entries in the VSAM ObjectStore data set (request database), and iclview displays the entries in the issued certificate list (ICL).

Part 8, “Reference information,” on page 403 provides reference information including messages and important code samples.
- Chapter 23, “Messages,” on page 405 explains PKI Services messages.
- Chapter 24, “File directory structure,” on page 437 describes product and file system directories for PKI Services and files contained in them.
- Chapter 25, “The pkiserv.conf configuration file,” on page 441 provides a code sample of the pkiserv.conf configuration file.
- Chapter 26, “Environment variables,” on page 445 explains the pkiserv.envs environment variables file and provides a code sample.
- Chapter 27, “The IKYSETUP REXX exec,” on page 449 explains the contents of the IKYSETUP REXX exec that performs RACF administration and provides a code sample.
- Chapter 28, “Other code samples,” on page 471 provides additional code samples.

There are several appendixes, including the following:
- Appendix B, “Using a gskkyman key database for your certificate store,” on page 491 explains an alternative method for setting up your key database.
- Appendix C, “Configuring PKI Services as an IdenTrust™ certificate authority,” on page 493 explains configuring PKI Services to operate as an IdenTrust compliant certificate authority (CA).

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 CD-ROM and DVD online library collections. The collections include Softcopy Reader, which is a program that enables you to view the softcopy documents.

**SK3T-4269 z/OS Version 1 Release 11 Collection**

This collection contains the set of unlicensed documents for the current release of z/OS in both BookManager® and Portable Document Format (PDF) files. You can view or print the PDF files with an Adobe® reader.

**SK3T-4272 z/OS Security Server RACF Collection**

This softcopy collection contains the Security Server library for z/OS for multiple releases in both BookManager and Portable Document Format (PDF) files. It also contains z/OS software products documents that contain substantial RACF information. The collection does not contain licensed publications.

**SK3T-7876 IBM eServer™ zSeries Redbooks Collection**

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

**SK2T-2177 IBM Redbooks S/390 Collection**

This softcopy collection contains a set of documents called Redbooks that pertain to S/390® subject areas ranging from application development and enablement to hardware, networking, 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:

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

- **Enterprise systems security**
  For more information about security on the zSeries platform and z/OS, use this address:

- **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.

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

- **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](http://www.ibm.com/servers/eserver/zseries/zos/racf/) and select the "Downloads" topic from the navigation bar, or go to [ftp.software.ibm.com](http://ftp.software.ibm.com)/eserver/zseries/zos/racf/.
  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/ra

An announcement will be posted on RACF-L discussion list and on newsgroup ibm.servers.mvs.raf 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
Preface

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

Summary of changes
for SA22-7693-11
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 21 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 VSAM data sets”** 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 157 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 191 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 229 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 230 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 exits” describes the QRECOVER preprocessing and post-processing exits.

Chapter 15, “Customizing with installation exits” 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 exits” 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 23, “Messages” contains the following new messages:

- IKYC074I
- IKYI005I
- IKYI006I
- IKYI007I
- IKYK001I
- IKYK002I
- IKYK003I
• In Chapter 23, “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 25, “The pkiserv.conf configuration file,” on page 441 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 26, “Environment variables,” on page 445 documents the _PKISERV_ENABLE_JSP environment variable.

• Chapter 27, “The IKYSETUP REXX exec” describes setup to allow PKI Services to generate keys for certificates.

• Chapter 28, “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 225 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 23, “Messages” contains updated information for the following messages:
  – IKYC032I
  – IKYC037I
  – IKYC068I
  – IKYI002I
  – IKYP031E
  – IKYP035I
  – IKYP036I
  – IKYU015I

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 exits.”

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.

You may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

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

Summary of changes
for  SA22-7693-10
z/OS Version 1 Release 10

This document contains information previously presented in z/OS Cryptographic Services PKI Services Guide and Reference, SA22-7693-09, which supports z/OS Version 1 Release 9.

New information
• Chapter 1, “Introducing PKI Services” contains information about supported characters and adds RFC 4291 (IP Version 6 Addressing Architecture) to the list of supported standards.
• Chapter 6, “Tailoring the LDAP configuration for PKI Services” contains information about setting up an LDAP access control list (ACL) to allow users to access CRLs.
• Chapter 6, “Tailoring the LDAP configuration for PKI Services” contains information about setting up a secure connection with LDAP.
• In Chapter 6, “Tailoring the LDAP configuration for PKI Services,” Table 19 on page 73 indicates that the URL for the LDAP server can be specified with or without the preceding string “ldap://” or “ldaps://”.
• Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” documents a new substitution variable, %quicklink%, which includes the transaction ID in the link provided in the certificate ready notification e-mail so that the user does not have to key it in to pick up the certificate.
• Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” documents three new INSERTs.
• In Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” Table 27 on page 105 documents new support for IP version 6 addresses in the AltIPAddr field.
• Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs” describes how the value of the KeyUsage field determines whether a basic constraints extension is included a certificate request.
• In Chapter 16, “Using the end-user Web pages,” Table 57 on page 276 documents new support for IP version 6 addresses for the alternate name.
• In Chapter 18, “Using PKI Services utilities,” Table 67 on page 332 documents new support for IP version 6 addresses in the AltIPAddr field.
• In Chapter 19, “RACF administration for PKI Services,” “Steps for using encrypted passwords” on page 361 indicates that the URL for the LDAP server can be specified with or without the preceding string “ldap://” or “ldaps://”.

xxvi  z/OS V1R11.0 Cryptographic Services PKI Services Guide and Reference
Chapter 23, “Messages” contains the following new message:

– IKYP039E

In Chapter 25, “The pkiserv.conf configuration file,” three new OIDs are added.

In Appendix A, “LDAP directory server requirements,” Table 89 on page 488 and Table 88 on page 487 are updated to include three new distinguished name attribute types.

Appendix D, “Using the PKI Services Web application with Internet Explorer on Windows Vista systems,” on page 505 describes how to set up Internet Explorer and Microsoft® Windows® Vista to work with PKI Services.

Changed information

Chapter 2, “Planning your implementation” documents that OCSF is no longer required for PKI Services. It is now required only for the PKI Services Trust Policy (PKITP).

Chapter 3, “Installing and configuring prerequisite products” has been updated to reflect that OCSF is no longer required for PKI Services.

In Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” the sample certificate ready notification e-mail is updated to use the substitution variable %%quicklink%%.

In Chapter 16, “Using the end-user Web pages,” Table 57 on page 276 has been reorganized.

Chapter 23, “Messages” contains updated information for the following messages:

– IKYC004I
– IKYC005I
– IKYC007I
– IKYC008I
– IKYC030I
– IKYC056I
– IKYI002I
– IKYL001I
– IKYL002I
– IKYL003I
– IKYL004I

In Chapter 27, “The IKYSETUP REXX exec,” the IKYSETUP sample has been updated to support hardware-generated keys up to 4096 bits and to delete the function to allow the PKI Services daemon access to OCSF.

Moved information

The chapter describing the PKI Services utilities has moved from Part 7, “Troubleshooting,” on page 389 to Part 4, “Using PKI Services,” on page 271.

Deleted information

Information about migration paths that are no longer supported has been deleted.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Summary of changes
for SA22-7693-09
z/OS Version 1 Release 9
This document contains information previously presented in z/OS Cryptographic Services PKI Services Guide and Reference, SA22-77693-08, which supports z/OS Version 1 Release 8.

New information

- **Chapter 25, “The pkiserv.conf configuration file,” on page 441** - New keywords added for:
  - E-mail addresses of administrators who will receive e-mail notification on pending requests immediately
  - E-mail addresses of administrators who will receive e-mail notification on pending requests every day
  - Timeout value for the exit program
  - Data set containing requests pending for approval
  - Data set containing renewed certificate

- **Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” on page 101** - “TEMPLATE sections” on page 114 describes the <AUTORENEW> tag in the certificate template file.

- **Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” on page 101** - Samples added for new e-mail notification forms for administrator notification of pending requests (pendingmsg.form) and automatic renewal of certificates (renewcertmsg.form). Descriptions of new variables used in these forms added.

- **Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” on page 101** - “Setting up automatic renewal of certificates” on page 230 describes how to set up automatic renewal of certificates.

- **“Steps for additional first-time customization” on page 138** - has an additional step for setting up automatic renewal of certificates.

- **Chapter 15, “Customizing with installation exits,” on page 235** describes the preprocessing and postprocessing exits for automatic certificate renewal, and the exit timeout value that can be set in the pkiserv.conf configuration file. It describes scenarios that have been added to the sample exit shipped with PKI Services illustrating the use of the automatic renewal exits.

- **Chapter 26, “Environment variables,” on page 445** - New environment variable _PKISERV_VARDIR

- **Chapter 23, “Messages,” on page 405** contains the following new messages:
  - [IKYP035I] through [IKYP038I], inclusively
  - [IKY061I] through [IKY073I], inclusively

Changed information

- **Chapter 4, “Running IKYSETUP to perform RACF administration”** - the default value for the variable ra_label in Table 10 on page 29 is corrected.

- **Chapter 4, “Running IKYSETUP to perform RACF administration”** - the PRIVATE qualifier is deleted from the default values for the variables backup_dsn and ra_backup_dsn.

- **“Installing and configuring LDAP” on page 19** - the distinguished name (DN) for the LDAP administrator can be a RACF-style DN.

- **Table 18 on page 52** - Use of ExpireWarningTime and ExpiringMessageForm keywords for automatic certificate renewal.

- **Chapter 23, “Messages”** contains updated information for the following messages:
  - [IKYC005I]
Deleted information

- Information about migrating from z/OS V1R3 has been deleted, because this migration path is no longer supported.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.
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 9** 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 17** describes installing and configuring related products: the z/OS HTTP Server, LDAP, and optionally ICSF and OCSF.
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 Appendix C, “Configuring PKI Services as an IdenTrust™ certificate authority,” on page 493.

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

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       | 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
| Web application      | • Annotating the reason for an administrative action |
Table 1. Basic components of PKI Services and related products (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-user Web application</td>
<td>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.</td>
</tr>
<tr>
<td>Exit</td>
<td>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.</td>
</tr>
<tr>
<td>ICSF (optional)</td>
<td>Securely stores the PKI Services certificate authority's private signing key and key pairs that PKI Services generates for certificates.</td>
</tr>
<tr>
<td>LDAP</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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.
Introducing PKI Services

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:
  - Microsoft Internet Explorer version 5.x
  - Netscape Navigator and Netscape Communicator version 4.x
- Server certificates

Figure 1. Component diagram of a typical PKI Services system
Introducing PKI Services

- 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
- MD5 and SHA1 hash algorithms
- RFC 2560: Online Certificate Status Protocol - OCSP
- RFC 4291: IP Version 6 Addressing Architecture
- Cisco Systems’ Simple Certificate Enrollment Protocol (SCEP) (Internet draft: draft-nourse-scep-11.txt)

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

<table>
<thead>
<tr>
<th>Type of certificate</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-year PKI SSL browser certificate</td>
<td>End-user client authentication using SSL</td>
</tr>
<tr>
<td>One-year PKI S/MIME browser certificate</td>
<td>Browser-based e-mail encryption</td>
</tr>
<tr>
<td>One-year PKI generated key certificate</td>
<td>Generation of public and private keys by PKI Services</td>
</tr>
<tr>
<td>Two-year PKI browser certificate for authenticating to z/OS</td>
<td>End-user client authorization using SSL when logging onto z/OS</td>
</tr>
<tr>
<td>Two-year PKI Authenticode—code signing server certificate</td>
<td>Software signing</td>
</tr>
<tr>
<td>Two-year PKI Windows logon certificate</td>
<td>End-user client authentication for an Active Directory user logging in to a Windows desktop using a smart card</td>
</tr>
<tr>
<td>Five-year PKI SSL server certificate</td>
<td>SSL Web server certification</td>
</tr>
<tr>
<td>Five-year PKI IPSEC server (firewall) certificate</td>
<td>Firewall server identification and key exchange</td>
</tr>
<tr>
<td>Five-year PKI intermediate CA server certificate</td>
<td>Subordinate (non-self-signed) certificate-authority certification</td>
</tr>
<tr>
<td>Five-year SCEP certificate</td>
<td>Creation of a preregistration record for certificate requestors. (Certificate requestors using Simple Certificate Enrollment Protocol (SCEP) must be preregistered.)</td>
</tr>
</tbody>
</table>

Unlike other templates, this template is intended for administration use only.
### Table 2. Types of certificates you can request (continued)

<table>
<thead>
<tr>
<th>Type of certificate</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$-year PKI browser certificate for extensions demonstration</td>
<td>Demonstration of all extensions supported by PKI Services</td>
</tr>
<tr>
<td>One-year SAF browser certificate</td>
<td>End-user client authentication where the security product (RACF, not PKI Services) is the certificate provider</td>
</tr>
<tr>
<td><strong>Note:</strong> The certificate generated by this template can not be managed by the PKI Services administrator.</td>
<td></td>
</tr>
<tr>
<td>One-year SAF server certificate</td>
<td>Web server SSL certification where the security product (RACF, not PKI Services) is the certificate provider</td>
</tr>
<tr>
<td><strong>Note:</strong> The certificate generated by this template can not be managed by the PKI Services administrator.</td>
<td></td>
</tr>
</tbody>
</table>

**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.
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

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Default value or customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>variables-dir</td>
<td>The file system directory where PKI Services creates working files.</td>
<td>/var/pkiserv</td>
</tr>
<tr>
<td>install-dir</td>
<td>The file system directory where PKI Services is installed.</td>
<td>/usr/lpp/pkiserv</td>
</tr>
<tr>
<td>runtime-dir</td>
<td>The file system directory where PKI Services looks for configuration files.</td>
<td>/etc/pkiserv</td>
</tr>
</tbody>
</table>

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):

- All systems in the sysplex that run PKI Services must be at z/OS V1R4 or later.
- 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).

See "(Optional) preliminary steps for establishing VSAM RLS" on page 88 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.
Planning your implementation

**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)**

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 17.

**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 19.
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 23.

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 21.

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 22.

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 371.

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:

- ICSF programmer (optional)
- LDAP programmer
- MVS programmer
- OCEP programmer (optional)
- OCSF programmer (optional)
- RACF administrator
- UNIX programmer
- Web server programmer
Planning your implementation

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

<table>
<thead>
<tr>
<th>Role</th>
<th>Tasks</th>
<th>Required skills</th>
<th>Documented in:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSF programmer</td>
<td>(Optionally) installing and configuring ICSF (if not already done)</td>
<td>ICSF installation and configuration skills</td>
<td>• z/OS Cryptographic Services ICSF Administrators Guide</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• z/OS Cryptographic Services ICSF Application Programmer's Guide</td>
</tr>
<tr>
<td></td>
<td>(Optionally) setting up the ICSF PKCS #11 token data set (TKDS)</td>
<td>ICSF installation and configuration skills</td>
<td>• z/OS Cryptographic Services ICSF System Programmer's Guide</td>
</tr>
<tr>
<td>LDAP programmer</td>
<td>Installing and configuring LDAP (if not already done) and recording information</td>
<td>LDAP installation and configuration skills</td>
<td>• IBM Tivoli Directory Server Administration and Use for z/OS</td>
</tr>
<tr>
<td>OCEP programmer</td>
<td>(Optionally) Installing and configuring OCEP for use with PKITP</td>
<td>OCEP installation and configuration skills</td>
<td>• z/OS Integrated Security Services Open Cryptographic Enhanced Plug-ins Application Programming</td>
</tr>
<tr>
<td>OCSF programmer</td>
<td>(Optionally) Installing and configuring OCSF for use with PKITP</td>
<td>OCSF installation and configuration skills</td>
<td>• z/OS Open Cryptographic Services Facility Application Programming</td>
</tr>
<tr>
<td>UNIX programmer</td>
<td>(Optionally) Configuring sendmail if your installation is planning to send e-mail notifications to users about certificates</td>
<td>• Basic UNIX commands such as the <code>cp</code> (copy) command and <code>mkdir</code> (make directory) command • sendmail configuration skills</td>
<td>• z/OS Communications Server: IP Configuration Guide</td>
</tr>
</tbody>
</table>
Planning your implementation

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

<table>
<thead>
<tr>
<th>Role</th>
<th>Tasks</th>
<th>Required skills</th>
<th>Documented in:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web server programmer</td>
<td>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</td>
<td>z/OS HTTP Server installation and configuration skills</td>
<td>• z/OS HTTP Server Planning, Installing, and Using</td>
</tr>
<tr>
<td></td>
<td>If you are implementing the PKI Services Web application using Java server pages (JSPs): Installing and configuring Websphere Application Server</td>
<td>Websphere Application Server installation and configuration skills</td>
<td>• The information center for your release of Websphere Application Server, at <a href="http://www.ibm.com/">http://www.ibm.com/</a> software/webservers/ appserv/was/library</td>
</tr>
</tbody>
</table>

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.

See Chapter 3, “Installing and configuring prerequisite products,” on page 17 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

<table>
<thead>
<tr>
<th>Role</th>
<th>Tasks</th>
<th>Required skills</th>
<th>Documented in:</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP programmer</td>
<td>• Customizes LDAP configuration for PKI Services</td>
<td>• LDAP customization skills</td>
<td>IBM Tivoli Directory Server Administration and Use for z/OS</td>
</tr>
</tbody>
</table>
### Planning your implementation

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

<table>
<thead>
<tr>
<th>Role</th>
<th>Tasks</th>
<th>Required skills</th>
<th>Documented in:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS programmer</td>
<td>• Creates VSAM object store and ICL data sets and indexes</td>
<td>• Basic MVS skills</td>
<td>• <a href="#">z/OS MVS System Commands</a></td>
</tr>
<tr>
<td></td>
<td>• (Optionally) sets up VSAM RLS</td>
<td>– Editing a data set</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Starts the PKI Services daemon</td>
<td>– ISPF COPY command</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>– MVS console START command</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• JCL knowledge to change job card</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Basic browser and Web skills</td>
<td></td>
</tr>
<tr>
<td>RACF administrator</td>
<td>• Adds groups and user IDs</td>
<td>• RACF administration</td>
<td>• <a href="#">z/OS TSO/E REXX Reference</a></td>
</tr>
<tr>
<td></td>
<td>• Sets up access control</td>
<td>• REXX skills (for working with IKYSETUP REXX exec)</td>
<td>• <a href="#">z/OS UNIX System Services Planning</a></td>
</tr>
<tr>
<td></td>
<td>• Creates certificates</td>
<td>• RACF commands such as the following: ADDGROUP ADDSD ADDUSER RACDCERT RDEFINE PERMIT SETROPTS</td>
<td>• <a href="#">z/OS Security Server RACF Security Administrator's Guide</a></td>
</tr>
<tr>
<td></td>
<td>• Sets up daemon security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIX programmer</td>
<td>• Copies files</td>
<td>• Basic UNIX commands, such as the <code>cp</code> (copy) command</td>
<td>• <a href="#">z/OS UNIX System Services Command Reference</a></td>
</tr>
<tr>
<td></td>
<td>• (Optionally) customizes environment variables</td>
<td>• Getting superuser authority</td>
<td>• <a href="#">z/OS UNIX System Services Planning</a></td>
</tr>
<tr>
<td></td>
<td>• (Optionally) customizes (non-LDAP sections of) pkiserv.conf configuration file</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Sets up /var/pkiserv directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Updates the LDAP section of the pkiserv.conf configuration file</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 5. Roles, tasks, and skills for setting up PKI Services (continued)

<table>
<thead>
<tr>
<th>Role</th>
<th>Tasks</th>
<th>Required skills</th>
<th>Documented in:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web server programmer</td>
<td>• Helps set up PKI Services</td>
<td>• z/OS HTTP Server customization skills</td>
<td>z/OS HTTP Server Planning, Installing, and Using</td>
</tr>
<tr>
<td></td>
<td>– 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</td>
<td>• Websphere Application Server customization and administration skills</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– 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.</td>
<td>• Editing configuration files</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Customizes the PKI Services Web pages</td>
<td>• Customizing the Web pages</td>
<td></td>
</tr>
</tbody>
</table>

### 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>
<thead>
<tr>
<th>Subtask</th>
<th>Associated procedure (See ...)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing and configuring prerequisite products:</td>
<td>Chapter 3, “Installing and configuring prerequisite products”</td>
</tr>
<tr>
<td>• z/OS HTTP Server</td>
<td>• Steps for installing and configuring the z/OS HTTP Server to work with PKI Services</td>
</tr>
<tr>
<td>• OCSF (optional)</td>
<td>• Installing and configuring OCSF</td>
</tr>
<tr>
<td>• LDAP directory server</td>
<td>• Steps for installing and configuring LDAP</td>
</tr>
<tr>
<td>• ICSF (optional)</td>
<td>• Installing and configuring ICSF (optional)</td>
</tr>
<tr>
<td>• OCEP (optional)</td>
<td>• Configuring and getting started with PKITP</td>
</tr>
<tr>
<td>• sendmail (optional)</td>
<td>• Configuring sendmail (optional)</td>
</tr>
</tbody>
</table>

Configuring your system for PKI Services: Part 2, “Configuring your system for PKI Services”

---

Chapter 2. Planning your implementation 15
### Table 6. Task roadmap for implementing PKI Services (continued)

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Associated procedure (See ...)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• RACF</td>
<td>• Chapter 4, “Running IKYSETUP to perform RACF administration&quot;</td>
</tr>
<tr>
<td>• z/OS UNIX</td>
<td>• Chapter 5, “Configuring the UNIX runtime environment”</td>
</tr>
<tr>
<td>• LDAP configuration</td>
<td>• Chapter 6, “Tailoring the LDAP configuration for PKI Services&quot;</td>
</tr>
<tr>
<td>• z/OS HTTP Server</td>
<td>• Chapter 7, “Updating z/OS HTTP Server configuration and starting the server”</td>
</tr>
<tr>
<td>• LDAP</td>
<td>• Chapter 8, “Tailoring the PKI Services configuration file for LDAP&quot;</td>
</tr>
<tr>
<td>• VSAM</td>
<td>• Chapter 9, “Creating VSAM data sets”</td>
</tr>
<tr>
<td></td>
<td>• Chapter 10, “Starting and stopping PKI Services”</td>
</tr>
<tr>
<td>Customizing PKI Services:</td>
<td>Part 3, “Customizing PKI Services”</td>
</tr>
<tr>
<td>• Customizing end-user Web pages</td>
<td>• Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs”</td>
</tr>
<tr>
<td>• Customizing administration Web pages</td>
<td>• Chapter 12, “Customizing the administration Web pages if you use REXX CGI execs”</td>
</tr>
<tr>
<td>• Advanced customizing</td>
<td>• Chapter 14, “Advanced customization”</td>
</tr>
<tr>
<td>Testing PKI Services:</td>
<td>Part 4, “Using PKI Services”</td>
</tr>
<tr>
<td>• Using end-user Web pages</td>
<td>• Chapter 16, “Using the end-user Web pages”</td>
</tr>
<tr>
<td>• Using administration Web pages</td>
<td>• Chapter 17, “Using the administration Web pages”</td>
</tr>
<tr>
<td>Administering PKI Services:</td>
<td>Part 5, “Administering security for PKI Services”</td>
</tr>
<tr>
<td>• RACF</td>
<td>• Chapter 19, “RACF administration for PKI Services”</td>
</tr>
</tbody>
</table>
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 VSAM data sets”), 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)

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 19
3. “Installing and configuring LDAP” on page 19
4. “Installing and configuring ICSF (optional)” on page 21
5. “Configuring sendmail (optional)” on page 22

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 27 and Chapter 7, “Updating z/OS HTTP Server configuration and starting the server,” on page 77—assume that this is already done.)

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
## Installing and configuring prerequisites

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:

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

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

<table>
<thead>
<tr>
<th>z/OS HTTP Server information</th>
<th>Explanation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS HTTP Server fully qualified domain name</td>
<td>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.)</td>
<td></td>
</tr>
<tr>
<td>The full UNIX pathname of your httpd.conf configuration file</td>
<td>(The Web server programmer needs to know the full UNIX pathname when updating the httpd.conf configuration file to support PKI Services.)</td>
<td></td>
</tr>
</tbody>
</table>

## 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 157 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 Appendix A, “LDAP directory server requirements,” on page 487.

Before you begin:

1. You need LDAP programming skills to complete this procedure.
Installing and configuring prerequisites

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:

<table>
<thead>
<tr>
<th>If ...</th>
<th>Then...</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>You do not have LDAP installed and configured ...</td>
<td>Follow the instructions in the Administration section of IBM Tivoli Directory Server Administration and Use for z/OS.</td>
<td></td>
</tr>
<tr>
<td>You have LDAP installed and configured but not for the TDBM or LDBM backend ...</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>You have LDAP installed and configured for the TDBM or LDBM backend ...</td>
<td>Go to the next step.</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>LDAP information</th>
<th>Explanation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distinguished name</td>
<td>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, PROFILENAME=user, O=racfdb, C=us.</td>
<td></td>
</tr>
<tr>
<td>Distinguished name password</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>LDAP fully qualified domain name and port</td>
<td>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 19 for a definition of fully qualified domain name.</td>
<td></td>
</tr>
</tbody>
</table>
Installing and configuring prerequisites

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

<table>
<thead>
<tr>
<th>LDAP information</th>
<th>Explanation</th>
<th>Value</th>
</tr>
</thead>
</table>
| 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 preferred but not required.

- 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 27.)
- PKI Services can use ICSF's PKCS #11 token data set (TKDS) to store key pairs that PKI Services generates for certificate requests. If the TKDS is not set up PKI Services cannot generate key pairs.

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.
Installing and configuring prerequisites

If ICSF is not already installed and configured for PKA, do this by following the instructions in the [z/OS Cryptographic Services ICSF Administrator’s Guide](https://www.ibm.com). If you want PKI Services to generate key pairs for certificate requests, set up the TKDS by following the instructions in the [z/OS Cryptographic Services ICSF Writing PKCS #11 Applications](https://www.ibm.com).

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](https://www.ibm.com)

Follow the instructions in the [z/OS Communications Server: IP Configuration Guide](https://www.ibm.com) 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.

**Note:** Because PKI Services always provides the return e-mail address, you do not need to configure sendmail to provide it. This simplifies your setup.

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. Execute the following command:

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

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](https://www.ibm.com)
Installing and configuring prerequisites

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 371.

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

- Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27 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 45 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 73 explains how to update your LDAP configuration (performed earlier—see “Installing and configuring LDAP” on page 19) for PKI Services.

- Chapter 7, “Updating z/OS HTTP Server configuration and starting the server,” on page 77 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 81 explains how to update the LDAP section of the PKI Services configuration file.

- Chapter 9, “Creating VSAM data sets,” on page 87 explains how to create VSAM data sets.

- Chapter 10, “Starting and stopping PKI Services,” on page 95 explains how to start and stop the PKI Services daemon.
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
  - Giving administrators access to VSAM data sets
  - 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

Overview of IKYSETUP

IKYSETUP consists of several parts:

- Configurable section—This section assigns values to variables.
Running IKYSETUP

- A section that issues RACF commands to perform RACF administration tasks. (See "Actions IKYSETUP performs by issuing RACF commands" on page 449 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 43.)

**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>
<thead>
<tr>
<th>Configurable section—assigns values to variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Values you must change to customize (See Table 10 on page 29)</td>
</tr>
<tr>
<td>- Values you might change that are related to setup (See Table 15 on page 34)</td>
</tr>
<tr>
<td>- Values you can optionally change (See Table 16 on page 37)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issues RACF commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records information in the log data set</td>
</tr>
</tbody>
</table>

**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.
Running IKYSETUP

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 29
- Table 14 on page 34
- The rows of Table 15 on page 34 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 and Table 16 on page 37, 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 204.

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

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value and your company's information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca_dn</td>
<td>The CA’s distinguished name. (For a definition of distinguished name, see Table 8 on page 20)</td>
<td>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 20 for a definition of suffix.)</td>
<td>When you also set ca_domain: OU(‘ca_domain Human Resources Certificate Authority’)</td>
</tr>
<tr>
<td></td>
<td>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 and web.expires (in Table 16 on page 37) 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.</td>
<td></td>
<td>When you do not set ca_domain: OU(‘Human Resources Certificate Authority’)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>O(‘Your Company’)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C(‘Your Country 2 Letter Abbreviation’)</td>
</tr>
</tbody>
</table>
### Table 10. IKYSETUP variables whose values must change (continued)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value and your company’s information</th>
</tr>
</thead>
</table>
| 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 |
| 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. **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_dns        | The RA’s distinguished name for use with Simple Certificate Enrollment Protocol (SCEP). (For a definition of distinguished name, see Table 8 on page 20) 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_dns="". | No | CN('Registration Authority')
OU('Human Resources Certificate Authority')
O('Your Company')
C('Your Country 2 Letter Abbreviation') |
### Table 10. IKYSETUP variables whose values must change (continued)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value and your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ra_label</td>
<td>The certificate label of your RA certificate in RACF.</td>
<td>No</td>
<td>When you also set ca_domain: Local PKI RA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When you do not set ca_domain: Local PKI RA</td>
</tr>
<tr>
<td>surrog_uid</td>
<td>The UID associated with the surrogate user ID.</td>
<td>No</td>
<td>555</td>
</tr>
<tr>
<td>web_dn</td>
<td>Your Web server’s distinguished name. (For a definition of distinguished name, see Table 8 on page 20)</td>
<td>The value of the Web server’s common name (CN), which is your server’s symbol IP address. For example, <a href="http://www.YourCompany.com">www.YourCompany.com</a> must match your Web server’s fully qualified domain name.</td>
<td>CN(‘www.YourCompany.com’)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>O(‘Your Company’)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L(‘Your City’)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SP(‘Your Full State or Province Name’)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C(‘Your Country 2 Letter Abbreviation’)</td>
</tr>
<tr>
<td>web_ring</td>
<td>The name of the Web server’s SAF key ring.</td>
<td>httpd*.conf—KeyFile directive</td>
<td>SSLring</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1. The RACF administrator copies the fully qualified domain name from an earlier table: Table 7 on page 19.
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.)
Running IKYSETUP

Variables whose values might change depending on setup

To help in completing the next table of variables (see Table 15 on page 34) fill out the following four decision tables:

Deciding the value of key_backup

Use the following decision table to determine the value of key_backup in Table 15 on page 34. 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

<table>
<thead>
<tr>
<th>If ...</th>
<th>Then ...</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want to back up your CA’s certificate and private key to a passphrase encrypted data set ...</td>
<td>Do not change the default key_backup=1</td>
<td>When you use IKYSETUP, you need to enter a passphrase whose display is not inhibited—it appears on the screen in the clear.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You cannot backup PCICC keys (key_type=2).</td>
</tr>
<tr>
<td>You do not want to back up your CA’s certificate and private key to a passphrase encrypted data set ...</td>
<td>Set key_backup=0</td>
<td>—</td>
</tr>
</tbody>
</table>

Deciding the value of key_type

Use the following decision table to determine the value of key_type in Table 15 on page 34. The key_type variable determines whether you are using ICSF, PCICC, or DSA 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 21.)
### Table 12. Decision table for key_type

<table>
<thead>
<tr>
<th>If ...</th>
<th>Then ...</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want to use software cryptography and you want a key generated using the RSA algorithm ...</td>
<td>Do not change the default key_type=0</td>
<td>—</td>
</tr>
</tbody>
</table>
| 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 15 on page 34:  
  - 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 backup the private key when you select the 2 value.  
Review and possibly change the following additional variables in Table 15 on page 34:  
  - 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. |

### Deciding the value of restrict_surrog

Use the following decision table to determine the value of restrict_surrog in Table 15 on page 34. 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

<table>
<thead>
<tr>
<th>If ...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want to assign the RESTRICTED attribute to the surrogate user ID ...</td>
<td>Set restrict_surrog=1</td>
</tr>
<tr>
<td>You do not want to assign the RESTRICTED attribute to the surrogate user ID ...</td>
<td>Do not change the default restrict_surrog=0</td>
</tr>
</tbody>
</table>

### Deciding the value of unix_sec

Use the following decision table to determine the value of unix_sec in Table 15 on page 34. 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
Running IKYSETUP

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 <code>pwd()</code> 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**

<table>
<thead>
<tr>
<th>If ...</th>
<th>Then ...</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>You already have z/OS UNIX security set up ...</td>
<td>Set unix_sec=1</td>
<td></td>
</tr>
<tr>
<td>You do not have z/OS UNIX security set up and you do not want to set it up ...</td>
<td>Do not change the default of unix_sec=0</td>
<td></td>
</tr>
<tr>
<td>You do not have z/OS UNIX security set up and you want to set it up for the first time ...</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

Update the following table based on your answers in the preceding decision tables. 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 15. IKYSETUP variables you might want to change depending on setup**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value or your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpx_userid.</td>
<td>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.)</td>
<td>No</td>
<td>1(default for number of items) OMVSKERN</td>
</tr>
<tr>
<td>ca_keysize</td>
<td>The size in bits of the certificate-authority’s private key. The minimum key size is 512 bits. The maximum key size is 1024 bits for key_type=1, and key_type=3, and 4096 bits for key_type=0 and key_type=2.</td>
<td>No</td>
<td>1024</td>
</tr>
<tr>
<td>cryptoz_grp</td>
<td>The name of a RACF group for users authorized to use PKCS #11 tokens. Applies only if you set key_gen = 1.</td>
<td>No</td>
<td><strong>---</strong></td>
</tr>
</tbody>
</table>
### Running IKYSETUP

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value or your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td>csfkeys_profile</td>
<td>A profile to protect the PKI Services key in ICSF. (This is non-applicable if key_type=0 or key_type=3.) If you do not want IKYSETUP to create the profile, set csfkeys_profile=&quot;&quot;. <strong>Note:</strong> When RACF stores the private key in the PKDS, it generates the label as: ‘IRR.DIGTCERT.CERTIFAUTH.unique-time-stamp’</td>
<td>No</td>
<td>IRR.DIGTCERT.CERTIFAUTH.*</td>
</tr>
<tr>
<td>csfserv_profile</td>
<td>A profile to protect ICSF services. (This is non-applicable if key_type=0 or key_type=3.)</td>
<td>No</td>
<td>CSF*</td>
</tr>
<tr>
<td>csfusers_grp</td>
<td>A group of authorized ICSF service users. (This is non-applicable if key_type=0 or key_type=3.)</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>key_backup</td>
<td>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). <strong>Note:</strong> This value is ignored when key_type=2 is also specified. 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.</td>
<td>No</td>
<td>1 (yes)</td>
</tr>
<tr>
<td>key_gen</td>
<td>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.</td>
<td>No</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 15. IKYSETUP variables you might want to change depending on setup (continued)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value or your company's information</th>
</tr>
</thead>
</table>
| key_type      | Specifies whether PKI Services should use ICSF, PCICC, or DSA 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).  
If you are changing key_type to 1 or 2, 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. | 0 |
| pgmcnt1_dsn.  | A list in which pgmcnt1_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.SCSCFMOD0’  
• ‘CSF.SCSCFMOD1’ |
| 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) |
### Variables you can optionally change

Review the values of the following variables to determine if you want to change any of the defaults in the rightmost column. (You should probably change at least the values for ca_expires and web_expires.) If you decide to change any value, cross out the default in the rightmost column and record your company’s information.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value or your company’s information</th>
</tr>
</thead>
</table>
| 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. | No | "" |
|               |             |                      | Guideline: Do not change the default (null) value until you perform advanced customization. (See "Adding a new CA domain" on page 204.) |
### Running IKYSETUP

Table 16. IKYSETUP variables you can optionally change (continued)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value or your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ca.expires</code></td>
<td>The date the PKI Services CA certificate expires.</td>
<td>No</td>
<td>2020/01/01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The date format is <code>yyyy/mm/dd</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>You should update this value to the expiration date of your CA certificate.</td>
</tr>
<tr>
<td><code>ca_ring</code></td>
<td>The name of the PKI Services SAF key ring.</td>
<td><code>pkiserv.conf—SAF KeyRing</code> value</td>
<td>When you also set <code>ca_domain: CAring.ca_domain</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When you do not set <code>ca_domain: CAring</code></td>
</tr>
<tr>
<td><code>cacert_dsn</code></td>
<td>The data set that will contain the PKI Services certificate to assist the backup process.</td>
<td>No</td>
<td>When you also set <code>ca_domain: daemon.ca_domain.CACERT.DERBIN</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When you do not set <code>ca_domain: daemon.CACERT.DERBIN</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> <code>daemon</code> refers to the <code>daemon</code> variable in this table.</td>
</tr>
<tr>
<td><code>daemon</code></td>
<td>The PKI Services daemon user ID. If you also set <code>ca_domain</code>, you can choose to assign a unique user ID to the daemon for each CA domain. <strong>Example:</strong> For a <code>ca_domain</code> called BankA, you might choose user ID PKISRVD.</td>
<td><code>pkiserv.conf—SAF KeyRing</code> value</td>
<td>PKISRVD</td>
</tr>
<tr>
<td><code>export_dsn</code></td>
<td>The data set that will contain the Web server’s root CA certificate for copying to file system.</td>
<td>No</td>
<td>When you also set <code>ca_domain: daemon.ca_domain.WEBROOT.DERBIN</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When you do not set <code>ca_domain: daemon.WEBROOT.DERBIN</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> <code>daemon</code> refers to the <code>daemon</code> variable in this table.</td>
</tr>
<tr>
<td><code>log_dsn</code></td>
<td>The log data set name.</td>
<td>No</td>
<td>When you also set <code>ca_domain: your-id.ca_domain.IKYSETUP.LOG</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When you do not set <code>ca_domain: your-id.IKYSETUP.LOG</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Notes:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1. The <code>your-id</code> refers to the RACF ID of the person running IKYSETUP. (You do not need to add this; MVS adds this for you.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Changing the default is not suggested.</td>
</tr>
</tbody>
</table>
### Table 16. IKYSETUP variables you can optionally change  (continued)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value or your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pkigroup</strong></td>
<td>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 <code>ca_domain</code>, you can choose to assign a unique group name to the administration group for each CA domain. <strong>Example:</strong> For a <code>ca_domain</code> called BankA, you might choose group name PKIGRPA.</td>
<td>No</td>
<td>PKIGRP</td>
</tr>
<tr>
<td><strong>ra_backup_dsn</strong></td>
<td>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 <code>backup_dsn</code> value.</td>
<td>No</td>
<td>When you also set <code>ca_domain</code>: <code>'daemon.ca_domain.RAKEY.BACKUP.P12BIN'</code> When you do not set <code>ca_domain</code>: <code>'daemon.RAKEY.BACKUP.P12BIN'</code> <strong>Note:</strong> The <code>daemon</code> refers to the <code>daemon</code> variable in this table.</td>
</tr>
<tr>
<td><strong>signing_ca_label</strong></td>
<td>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.</td>
<td>No</td>
<td>***</td>
</tr>
<tr>
<td><strong>surrog</strong></td>
<td>The surrogate user ID for PKI Services. If you also set <code>ca_domain</code>, you can choose to assign a unique user ID as the surrogate user ID for each CA domain. <strong>Example:</strong> For a <code>ca_domain</code> called BankA, you might choose user ID PKISERVA. <strong>Note:</strong> This cannot be an existing user ID (because IKYSETUP creates the user ID with the NOPASSWORD attribute).</td>
<td>Surrogate user ID in <code>httpd*.conf</code></td>
<td>PKISERV</td>
</tr>
</tbody>
</table>
Running IKYSETUP

Table 16. IKYSETUP variables you can optionally change  (continued)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
<th>Referenced elsewhere</th>
<th>Default value or your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td>vsamhlq</td>
<td>The high-level qualifier of the VSAM data sets for PKI Services. Note: The RACF administrator gets this information from the MVS programmer.</td>
<td></td>
<td>Same as the daemon variable earlier in this table.</td>
</tr>
<tr>
<td>web_expires</td>
<td>The date the Web server certificate expires.</td>
<td>No</td>
<td>2020/01/01</td>
</tr>
<tr>
<td>web_label</td>
<td>The label for the Web server’s certificate.</td>
<td>No</td>
<td>SSL Cert</td>
</tr>
<tr>
<td>webserver</td>
<td>The Web server’s daemon user ID.</td>
<td>See Web server documentation.</td>
<td>WEBSRV</td>
</tr>
</tbody>
</table>

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:
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 29.

   The following example shows how to change the pkigroup_mem variables. (Remember that for pkigroup_mem.n, 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"
   ```

---

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 29.

   The following example shows how to change the pkigroup_mem variables. (Remember that for pkigroup_mem.n, 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 15 on page 34. The following example shows how to change the key_type variable.

   **Example:**
   
   ```
   key_type=1
   ```

4. Optionally update any variables you changed in Table 16 on page 37. The following example shows how to change the log_dsn variable.

   **Example:**
   
   ```
   log_dsn="PRIVATE.IKYSETUP.LOG"
   ```

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 43 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 449.) The bottom part provides a record of important information that you will need for later steps, such as 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 29, Table 15 on page 34, and Table 16 on page 37. 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.
Running IKYSETUP

(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 361.

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(25600000) threads(512))  
ADDUSER PKISERV nopassword omvs(uid(555)) name('PKI Srvs Surrogate')  
SETROPTS EGN GENERIC(DATASET)  
ADDSD 'PKISRVD.**' UACC(NONE)  
ADDGROUP PKIGRP OMVS(GID(655))

Allowing administrators to access PKI databases ...  
PERMIT 'PKISRVD.**' ID(PKIGRP) ACCESS(CONTROL)  
SETROPTS GENERIC(DATASET) REFRESH

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

Back up the CA certificate ...

Marking CA certificate as HIGHTRUST ...

Saving the CA certificate to a data set for OPUT ...

Creating the PKI Services keyring ...

Creating the Webserver SSL certificate and keyring ...

Creating the Webserver SSL certificate and keyring ...

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(PKISERV) ACCESS(READ)  
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(PKISERV) ACCESS(READ)  
Allowing the Webserver to switch identity to PKISERV ...

RDEFINE CLASSACT(SURROGAT)  
RDEFINE SURROGAT BPX.SRV.PKISERV  
PERMIT BPX.SRV.PKISERV CLASS(SURROGAT) ID(PKISRVD) ACCESS(READ)  
SETROPTS RACLIST(SURROGAT) REFRESH

Allowing the PKI Services daemon to generate key pairs ...

RDEFINE CRYPTOZ SO.PKISRVD.* UACC(NONE)  
RDEFINE CRYPTOZ USER.PKISRVD.* UACC(NONE)  
PERMIT SO.PKISRVD.* CLASS(CRYPTOZ) ID(PKISRVD) ACCESS (UPDATE)
Running IKYSETUP

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 PKISRVD.* STDATA(USER(PKISRVD))
SETROPTS CLASACT(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 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 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 28, “Other code samples,” on page 471.

<table>
<thead>
<tr>
<th>File</th>
<th>Purpose</th>
<th>Need to copy?</th>
<th>Need to change?</th>
</tr>
</thead>
<tbody>
<tr>
<td>expiringmsg.form</td>
<td>The form for an e-mail sent to a user when a certificate is going to expire.</td>
<td>Only if your company sends an e-mail notification to a user about a certificate that is going to expire</td>
<td>Guideline: Make no changes to this file until later. See “Customizing e-mail notifications sent to users” on page 225 for details about making changes.</td>
</tr>
<tr>
<td>pendingmsg.form</td>
<td>The form for an e-mail sent to an administrator when requests are pending approval</td>
<td>Only if your company sends an e-mail notification to an administrator about requests that are pending approval</td>
<td>Guideline: Make no changes to this file until later. See “Customizing e-mail notifications sent to users” on page 225 for details about making changes.</td>
</tr>
</tbody>
</table>
### Configuring the UNIX runtime environment

#### Table 17. Deciding which files to copy and change (continued)

<table>
<thead>
<tr>
<th>File</th>
<th>Purpose</th>
<th>Need to copy?</th>
<th>Need to change?</th>
</tr>
</thead>
<tbody>
<tr>
<td>pkiserv.conf</td>
<td>Configuration file. Contains various settings and values PKI Services needs.</td>
<td>Only if you are configuring PKI Services for the first time.</td>
<td>The UNIX programmer might need to change the <strong>LDAP</strong> section of this file. <strong>Guideline:</strong> Do not change it now but change it later when you perform &quot;Steps for tailoring the <strong>LDAP</strong> section of the configuration file&quot; on page 82.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The UNIX programmer needs to update the non-LDAP section of the pkiserv.conf configuration file if any of the following is true:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• You intend to run multiple instances of PKI Services in a sysplex.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• You intend to run multiple CA domains on a single z/OS image. (See &quot;Adding a new CA domain&quot; on page 204.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• You are customizing certificate revocation list (CRL) distribution point processing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• You are configuring PKI Services for the first time and do not intend to send e-mail notifications for certificate-related events.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• You intend to have certificates that are about to expire be automatically renewed and sent to their owners.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• You intend to have e-mail notifications sent to the PKI administrator when there are requests pending for approval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• You intend to allow certificate requestors to ask PKI Services to generate keys for certificates.</td>
</tr>
<tr>
<td>pkiserv.envars</td>
<td>The environment variables file.</td>
<td>Only if you are configuring PKI Services for the first time and the file needs changes.</td>
<td>UNIX programmer might have to update this file. See &quot;Optionally updating PKI Services environment variables&quot; on page 49.</td>
</tr>
</tbody>
</table>
Table 17. Deciding which files to copy and change (continued)

<table>
<thead>
<tr>
<th>File</th>
<th>Purpose</th>
<th>Need to copy?</th>
<th>Need to change?</th>
</tr>
</thead>
<tbody>
<tr>
<td>pkiserv.tmpl</td>
<td>The certificate templates file used with REXX CGI execs. It contains HTML-style code that builds the Web pages underlying certificate requests.</td>
<td>Only if you are configuring PKI Services for the first time and using the REXX CGI execs to implement the PKI Services Web application.</td>
<td><strong>Guideline:</strong> 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 107 for details about making changes.</td>
</tr>
<tr>
<td>PKIServ.xsd</td>
<td>The XML schema that defines the syntax of the XML certificate templates file pkitmpl.xml.</td>
<td>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.</td>
<td><strong>Rule:</strong> Do not make changes to this file.</td>
</tr>
<tr>
<td>pkitmpl.xml</td>
<td>The certificate templates file used with Java server pages (JSPs). It defines applications and certificates in XML.</td>
<td>Only if you are configuring PKI Services for the first time and using the JSPs to implement the PKI Services Web application.</td>
<td><strong>Guideline:</strong> Make no changes to this file until later. See Chapter 13, “Implementing the Web application using Java server pages,” on page 157 for details about making changes.</td>
</tr>
<tr>
<td>readymsg.form</td>
<td>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.</td>
<td>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.</td>
<td><strong>Guideline:</strong> Make no changes to this file until later. See “Customizing e-mail notifications sent to users” on page 225 for details about making changes.</td>
</tr>
<tr>
<td>rejectmsg.form</td>
<td>The form for an e-mail sent to a user when the PKI Services administrator has rejected a certificate request.</td>
<td>Only if your company sends an e-mail notification to a user after the PKI Services administrator has rejected a certificate request.</td>
<td><strong>Guideline:</strong> Make no changes to this file until later. See “Customizing e-mail notifications sent to users” on page 225 for details about making changes.</td>
</tr>
<tr>
<td>renewcertmsg.form</td>
<td>The form for an e-mail sent to a user when PKI Services has automatically renewed an expiring certificate.</td>
<td>Only if your company enables automatic renewal of certificates.</td>
<td><strong>Guideline:</strong> Make no changes to this file until later. See “Customizing e-mail notifications sent to users” on page 225 for details about making changes.</td>
</tr>
<tr>
<td>recoverymsg.form</td>
<td>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.</td>
<td>Only if your company allows users to request that PKI Services generate key pairs for certificate requests.</td>
<td><strong>Guideline:</strong> Make no changes to this file until later. See “Customizing e-mail notifications sent to users” on page 225 for details about making changes.</td>
</tr>
</tbody>
</table>

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
Configuring the UNIX runtime environment

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 9

- 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.envs). If any values need to
   change, copy this file by entering the following command:
   cp -p /install-dir/samples/pkiserv.envs 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 484.)

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 95.
Configuring the UNIX runtime environment

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.envvars).

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: See “Optionally updating PKI Services environment variables” on page 49 to determine if you need to update environment variables.

Perform the following steps to update PKI Services environment variables:

1. Examine the values in the environment variables file (by default, pkiserv.envvars) and update the file as necessary. (See “Environment variables in the environment variables file” on page 445 for a description of the environment variables and “The pkiserv.envvars environment variables file” on page 448 for a code sample of the environment variables file.)

   Note: If you did not install sendmail in its default location (/usr/sbin), you need to update the PATH environment variable.

2. Make any needed changes to PKISERVD, such as updating the pathname of the environment variables file (FN and DIR parameters). (See “PKISERVD sample procedure to start PKI Services daemon” on page 484 for a code sample of the PKISERVD procedure.)

3. _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. 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

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 the UNIX runtime environment

- Customizing certificate revocation list (CRL) distribution point processing. (See "Customizing distribution point CRLs" on page 192 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

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 70.)

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:
  
  \[
  \text{name=dotted-decimal}
  \]
  
  The following excerpt is from the OIDs section:
  
  ```
  [OIDs]
  ...
  MyPolicy=1.2.3.4
  ```

- **[ObjectStore]**
  The ObjectStore section specifies operational information for various files and data sets.
  
  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]
  $igAlg=sha-1WithRSAEncryption
  ```

- **[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
  ```
Configuring the UNIX runtime environment

[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 82.

(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 22 on page 82.) 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>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OID section</td>
<td>A registered Object ID identifying your organization's usage policy, for example: 1.2.3.4</td>
<td>If you are creating your own certificate policy, see &quot;Using certificate policies&quot; on page 188 for information on creating certificate policies. Otherwise, do not change this information.</td>
<td>1.2.3.4</td>
</tr>
<tr>
<td>MyPolicy</td>
<td>VSAM data set name for the ObjectStore base cluster. This is the request database. Each VSAM request record consists of a fixed header followed by a variable-length section.</td>
<td>For the high-level qualifier before the period, see the vsamhlq variable in Table 16 on page 37. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.</td>
<td>'pkisrvd.vsam.ost' Note that this begins with the VSAM high-level qualifier.</td>
</tr>
</tbody>
</table>

Table 18. Information needed for updating the configuration file
Table 18. Information needed for updating the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
<th>Guideline: If you are adding a new CA domain, insert the ca_domain value from Table 16 on page 37 as the second qualifier in the data set name. Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectTidDSN</td>
<td>VSAM data set name for the ObjectStore transaction ID (TID) alternate index.</td>
<td>For the high-level qualifier before the period, see the vsamhlq variable in Table 16 on page 37. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.</td>
<td>'pkisrvd.vsam.ost.path' Note that this begins with the VSAM high-level qualifier.</td>
<td></td>
</tr>
<tr>
<td>ObjectStatusDSN</td>
<td>VSAM data set name for the ObjectStore status alternate index.</td>
<td>For the high-level qualifier before the period, see the vsamhlq variable in Table 16 on page 37. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.</td>
<td>'pkisrvd.vsam.ost.status' Note that this begins with the VSAM high-level qualifier.</td>
<td>Guideline: If you are adding a new CA domain, insert the ca_domain value from Table 16 on page 37 as the second qualifier in the data set name. Example: 'pkisrvd.employee.vsam.ost.status'</td>
</tr>
<tr>
<td>ObjectRequestorDSN</td>
<td>VSAM data set name for the ObjectStore requestor alternate index.</td>
<td>For the high-level qualifier before the period, see the vsamhlq variable in Table 16 on page 37. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.</td>
<td>'pkisrvd.vsam.ost.requestor' Note that this begins with the VSAM high-level qualifier.</td>
<td>Guideline: If you are adding a new CA domain, insert the ca_domain value from Table 16 on page 37 as the second qualifier in the data set name. Example: 'pkisrvd.employee.vsam.ost.requestor'</td>
</tr>
</tbody>
</table>
Configuring the UNIX runtime environment

Table 18. Information needed for updating the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICLDSN</td>
<td>VSAM data set name for the ICL base cluster.</td>
<td>For the high-level qualifier before the period, see the vsamhlq variable in Table 16 on page 37</td>
<td>'pkisrvd.vsam.icl'</td>
</tr>
<tr>
<td></td>
<td>This 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.</td>
<td></td>
<td>Note that this begins with the VSAM high-level qualifier.</td>
</tr>
<tr>
<td></td>
<td><strong>Guideline:</strong> If you are adding a new CA domain, insert the ca_domain value from Table 16 on page 37 as the second qualifier in the data set name. <strong>Example:</strong> 'pkisrvd.employee.vsam.icl'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICLStatusDSN</td>
<td>VSAM data set name for ICL status alternate index.</td>
<td>For the high-level qualifier before the period, see the vsamhlq variable in Table 16 on page 37</td>
<td>'pkisrvd.vsam.icl.status'</td>
</tr>
<tr>
<td></td>
<td>The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.</td>
<td></td>
<td>Note that this begins with the VSAM high-level qualifier.</td>
</tr>
<tr>
<td></td>
<td><strong>Guideline:</strong> If you are adding a new CA domain, insert the ca_domain value from Table 16 on page 37 as the second qualifier in the data set name. <strong>Example:</strong> 'pkisrvd.employee.vsam.icl.status'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICLRequestorDSN</td>
<td>VSAM data set name for ICL requestor alternate index.</td>
<td>For the high-level qualifier before the period, see the vsamhlq variable in Table 16 on page 37</td>
<td>'pkisrvd.vsam.icl.requestor'</td>
</tr>
<tr>
<td></td>
<td>The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.</td>
<td></td>
<td>Note that this begins with the VSAM high-level qualifier.</td>
</tr>
<tr>
<td></td>
<td><strong>Guideline:</strong> If you are adding a new CA domain, insert the ca_domain value from Table 16 on page 37 as the second qualifier in the data set name. <strong>Example:</strong> 'pkisrvd.employee.vsam.icl.requestor'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RemoveCompletedReqs</td>
<td>Time period that completed certificate requests remain in the ObjectStore before automatic deletion. This is a number followed by d (days) or w (weeks).</td>
<td>UNIX programmer decides this value.</td>
<td>1w</td>
</tr>
</tbody>
</table>
### Configuring the UNIX runtime environment

**Table 18. Information needed for updating the configuration file (continued)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveInactiveReqs</td>
<td>Time period that incomplete, inactive certificate requests remain in the ObjectStore before automatic deletion. This is a number followed by d (days) or w (weeks).</td>
<td>UNIX programmer decides this value.</td>
<td>4w</td>
</tr>
<tr>
<td>RemoveExpiredCertsAndKeys</td>
<td>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.</td>
<td>UNIX programmer decides this value.</td>
<td>520w</td>
</tr>
<tr>
<td>RemoveExpiredCerts</td>
<td>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.</td>
<td>UNIX programmer decides this value.</td>
<td>0d</td>
</tr>
<tr>
<td>SharedVSAM</td>
<td>Indicates whether you intend to share a single copy of the PKI Services VSAM data sets among multiple images in a sysplex. This is T (True) or F (False).</td>
<td>UNIX programmer decides this value.</td>
<td>F</td>
</tr>
</tbody>
</table>

**CertPolicy section**
### Configuring the UNIX runtime environment

**Table 18. Information needed for updating the configuration file (continued)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdminNotifyNewn</td>
<td>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 <code>&lt;userid&gt;@&lt;system&gt;</code>.</td>
<td>UNIX programmer decides this value. Do not change this information until you set up administrator notification of requests pending approval.</td>
<td><a href="mailto:abigail@mycompany.com">abigail@mycompany.com</a></td>
</tr>
<tr>
<td>AdminNotifyReminder</td>
<td>The e-mail address to which reminder notifications of requests pending approval should be sent when PKI Services starts, and once a day thereafter. 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 <code>&lt;userid&gt;@&lt;system&gt;</code>.</td>
<td>UNIX programmer decides this value. Do not change this information until you set up administrator notification of requests pending approval.</td>
<td><a href="mailto:abigail@mycompany.com">abigail@mycompany.com</a></td>
</tr>
<tr>
<td>ARLDist</td>
<td>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.</td>
<td>UNIX programmer decides this value. Do not change this information until you perform advanced customization. See &quot;Creating a distribution point ARL&quot; on page 199 for more information.</td>
<td>$F$</td>
</tr>
<tr>
<td>CPS1</td>
<td>The Uniform Resource Identifier (URI) where your organization's Certification Practice Statement (CPS) is located. This is in the form: <a href="http://www.mycompany.com/cps.html">http://www.mycompany.com/cps.html</a></td>
<td>Do not change this information until you perform advanced customization. See &quot;Using certificate policies&quot; on page 188 for more information.</td>
<td><a href="http://www.mycompany.com/cps.html">http://www.mycompany.com/cps.html</a></td>
</tr>
</tbody>
</table>

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
### Configuring the UNIX runtime environment

#### Table 18. Information needed for updating the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateInterval</td>
<td>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).</td>
<td>UNIX programmer decides this value.</td>
<td>3m</td>
</tr>
<tr>
<td>CRLDistDirPath</td>
<td>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/.</td>
<td>UNIX programmer decides this value. Do not change this information until you perform advanced customization. See &quot;Customizing distribution point CRLs&quot; on page 192 for more information.</td>
<td>/var/pkiserv/</td>
</tr>
<tr>
<td>CRLDistName</td>
<td>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.</td>
<td>UNIX programmer decides this value. Do not change this information until you perform advanced customization. See &quot;Customizing distribution point CRLs&quot; on page 192 for more information.</td>
<td>CRL</td>
</tr>
<tr>
<td>CRLDistSize</td>
<td>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.</td>
<td>UNIX programmer decides this value. Do not change this information until you perform advanced customization. See &quot;Customizing distribution point CRLs&quot; on page 192 for more information.</td>
<td>500</td>
</tr>
</tbody>
</table>
## Configuring the UNIX runtime environment

### Table 18. Information needed for updating the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRLDistURI</td>
<td>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.</td>
<td>UNIX programmer decides this value. Do not change this information until you perform advanced customization. See &quot;Customizing distribution point CRLs&quot; on page 192 for more information.</td>
<td>—</td>
</tr>
<tr>
<td>CRLDuration</td>
<td>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).</td>
<td>UNIX programmer decides this value.</td>
<td>2d</td>
</tr>
<tr>
<td>EnableSCEP</td>
<td>Specifies whether Simple Certificate Enrollment Protocol (SCEP) is allowed. This is T (True) or F (False).</td>
<td>UNIX programmer decides this value. Do not change this information until you perform advanced customization. See &quot;Enabling Simple Certificate Enrollment Protocol (SCEP)&quot; on page 219 for more information.</td>
<td>F</td>
</tr>
</tbody>
</table>
## Table 18. Information needed for updating the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
</table>
| 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                                     |
| 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                                   |
### Configuring the UNIX runtime environment

**Table 18. Information needed for updating the configuration file (continued)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCSPType</td>
<td>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.</td>
<td>Change to basic if you want to enable the responder.</td>
<td>none</td>
</tr>
<tr>
<td>PolicyCritical</td>
<td>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.</td>
<td>UNIX programmer decides this value. Do not change this information until you perform advanced customization. See <a href="#">&quot;Using certificate policies&quot; on page 188</a> for more information.</td>
<td>F</td>
</tr>
<tr>
<td>PolicyRequired</td>
<td>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.</td>
<td>UNIX programmer decides this value. Do not change this information until you perform advanced customization. See <a href="#">&quot;Using certificate policies&quot; on page 188</a> for more information.</td>
<td>F</td>
</tr>
<tr>
<td>Parameter</td>
<td>Information needed</td>
<td>Where to get this information</td>
<td>Sample value or your customized value</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>PolicyName1</td>
<td>The name of the policy. (This is the same policy name used with the MyPolicy parameter of the OIDs section.)</td>
<td>Do not change this information until you perform advanced customization. See &quot;Using certificate policies&quot; on page 188 for more information.</td>
<td>MyPolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Policy1Org</td>
<td>This is the organization name for the CertificatePolicies extension. For example: International Business Machines, Inc.</td>
<td>Do not change this information until you perform advanced customization. See &quot;Using certificate policies&quot; on page 188 for more information.</td>
<td>My Company, Inc.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Policy1Notice1</td>
<td>The first company notice number.</td>
<td>Do not change this information until you perform advanced customization. See &quot;Using certificate policies&quot; on page 188 for more information.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Policy1Notice2</td>
<td>The second company notice number.</td>
<td>Do not change this information until you perform advanced customization. See &quot;Using certificate policies&quot; on page 188 for more information.</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring the UNIX runtime environment

**Table 18. Information needed for updating the configuration file (continued)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SigAlg1</td>
<td>The Object ID for the signature algorithm. If the certificate key type is RSA, the SigAlg1 algorithm value must be one of the following: * sha-1WithRSAEncryption (the default) * sha-256WithRSAEncryption * md-5WithRSAEncryption * md-2WithRSAEncryption If the certificate key type is DSA, the SigAlg1 algorithm value must be as follows: * id-dsa-with-sha1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Changing the default also requires adding a line in the OIDs section. See [&quot;Updating the signature algorithm&quot;](page 191) on page 191 for more information.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Guideline</strong>: The MD2 and MD5 hashes have been found to be vulnerable to attack. Avoid specifying md-5WithRSAEncryption and md-2WithRSAEncryption if possible.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TimeBetweenCRLs</td>
<td>How often a certificate revocation list should be created. This is a number followed by w (weeks), d (days), h (hours), m (minutes), or s (seconds). <strong>Note</strong>: If you change this value after PKI Services has been in operation and then restart PKI Services, the change does not take effect until after the next CRL is created.</td>
<td>UNIX programmer decides this value.</td>
<td>1d</td>
</tr>
</tbody>
</table>
### Configuring the UNIX runtime environment

Table 18. Information needed for updating the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserNoticeText1</td>
<td>A legal statement about certificate issuance and use. For example: Certificate for IBM internal use only</td>
<td>Do not change this information until you perform advanced customization. See &quot;Using certificate policies&quot; on page 188 for more information.</td>
<td>statement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you changed PolicyRequired=F to PolicyRequired=T, you need to replace the variable <code>statement</code> with your own value for this:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>________________________________________________________________________________________</td>
<td></td>
</tr>
</tbody>
</table>

**General section**

<table>
<thead>
<tr>
<th>ExitTimeout</th>
<th>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 will cancel 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.</th>
<th>UNIX programmer decides this value.</th>
<th>10s</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitialThreadCount</td>
<td>Number of threads (at least 2 and no more than 100) the PKI Services daemon should create at program initialization.</td>
<td>UNIX programmer decides this value.</td>
<td>10</td>
</tr>
</tbody>
</table>
### Configuring the UNIX runtime environment

**Table 18. Information needed for updating the configuration file (continued)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
</table>
| ReadyMessageForm      | The full pathname 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 16 on page 37 as the second qualifier in the pathname. **Example:** /etc/pkiserv/employees/readymsg.form

| RejectMessageForm    | The full pathname 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 16 on page 37 as the second qualifier in the pathname. **Example:** /etc/pkiserv/employees/rejectmsg.form
### Configuring the UNIX runtime environment

#### Table 18. Information needed for updating the configuration file  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpiringMessageForm</td>
<td>The full pathname 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.</td>
<td>UNIX programmer decides this value.</td>
<td>/etc/pkiserv/expiringmsg.form</td>
</tr>
<tr>
<td>AdminNotifyForm</td>
<td>The full pathname or data set name containing the 'request(s) pending for approval' message form. Defaults to no notification sent.</td>
<td>UNIX programmer decides this value.</td>
<td>AdminNotifyForm=/etc/pkiserv/pendingmsg.form</td>
</tr>
<tr>
<td>RenewCertForm</td>
<td>The full pathname or data set name containing the 'renewed certificate'. Defaults to no certificate sent.</td>
<td>UNIX programmer decides this value.</td>
<td>RenewCertForm=/etc/pkiserv/renewcertmsg.form</td>
</tr>
<tr>
<td>RecoverForm</td>
<td>The full pathname 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.</td>
<td>UNIX programmer decides this value.</td>
<td>RecoverForm=/etc/pkiserv/recoverymsg.form</td>
</tr>
</tbody>
</table>

**Guideline:** If you are adding a new CA domain, use the `ca_domain` value from [Table 16 on page 37](#) as the second qualifier in the pathname. **Example:** /etc/pkiserv/employees/expiringmsg.form

---

*SAF section*
Configuring the UNIX runtime environment

Table 18. Information needed for updating the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Sample value or your customized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyRing</td>
<td>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 (/).)</td>
<td>See the ca_ring and daemon values in Table 16 on page 37</td>
<td>PKISRVD/CAring</td>
</tr>
<tr>
<td>RA_label</td>
<td>The label of your PKI Services registration authority (RA) certificate.</td>
<td>See the ra_label value in Table 10 on page 29</td>
<td>Local PKI RA</td>
</tr>
<tr>
<td>TokenName</td>
<td>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</td>
<td>PKISRVD.PKIToken</td>
<td></td>
</tr>
</tbody>
</table>

LDAP section

For information about the LDAP section, see Table 22 on page 82

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

Note: Keep in mind that everything in the pkiserv.conf file—including section names, keys, and values—is case-sensitive.

1. If necessary, update the ObjectStore section:
   a. 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 18 on page 52

   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'
Configuring the UNIX runtime environment

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 90), you must make equivalent changes in pkiserv.conf.

b. If necessary, change 1w in the following line to the value in the RemoveCompletedReqs row in Table 18 on page 52:

RemoveCompletedReqs=1w

c. If necessary, change 4w in the following line to the value in the RemoveInactiveReqs row in Table 18 on page 52:

RemoveInactiveReqs=4w

d. If necessary, uncomment the following line and, optionally, change 26w to the value in the RemoveExpiredCerts row in Table 18 on page 52:

RemoveExpiredCerts=26w

e. If necessary, uncomment the following line and, optionally, change 520w to the value in the RemoveExpiredCertsAndKeys row in Table 18 on page 52:

RemoveExpiredCertsAndKeys=520w

f. If necessary, update the SharedVSAM lines:
   - 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:
     
     SharedVSAM=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 18 on page 52:

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 18 on page 52:
     
     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 18 on page 52:

TimeBetweenCRLs=1d

d. If necessary, change 2d in the following line to the value in the CRLDuration row in Table 18 on page 52:

CRLDuration=2d

e. If necessary, change F in the following line to the value in the PolicyRequired row in Table 18 on page 52:
Configuring the UNIX runtime environment

PolicyRequired=F

For more information on this parameter, see “Using certificate policies” on page 188.

f. If necessary, change F in the following line to the value in the PolicyCritical row in Table 18 on page 52
   PolicyCritical=F

   For more information on this parameter, see “Using certificate policies” on page 188.

   g. If necessary, change 120d in the following line to the value in the MaxSuspendDuration row in Table 18 on page 52
      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 195 to determine the values for Table 18 on page 52

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


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 18 on page 52
      InitialThreadCount=10

   b. If necessary, change 30s in the following line to the value in the ExitTimeout row in Table 18 on page 52
      ExitTimeout=30s

   c. 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 pathname in the following lines to the corresponding values in Table 18 on page 52
         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 pathname in the following lines to the corresponding value in Table 18 on page 52
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:

```bash
# 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:

```bash
# 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:

```bash
# 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:

```bash
# 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 18 on page 52:

   ```bash
   KeyRing=PKISRVD/CAring
   ```

   b. If you specified `EnableSCEP=T` in Step 2j on page 68, change `Local PKI RA` in the following line to the value in the ra_label row in Table 10 on page 29:

   ```bash
   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
# Configuring the UNIX runtime environment

- #11 token data set (TKDS), uncomment the following line and change
  `PKISRVD.PKIToken` to the value you chose in the `TokenName` row in Table 18 on page 52:

- `TokenName=PKISRVD.PKIToken`

## 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 25, “The `pkiserv.conf` configuration file,” on page 441, 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:

<table>
<thead>
<tr>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PKISRVD</code></td>
<td>Use your daemon value in Table 16 on page 37</td>
</tr>
<tr>
<td><code>'pkisrvd.webroot.derbin'</code></td>
<td>Use your export_dsn value in Table 16 on page 37</td>
</tr>
<tr>
<td><code>/var/pkiserv</code></td>
<td>Guideline: Use your ca_domain value from Table 16 on page 37 to qualify the directory location if you are adding a new CA domain. For example, <code>/var/pkiserv/employees</code>.</td>
</tr>
</tbody>
</table>

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:**
   ```bash
   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:**
   ```bash
   chmod 644 /var/pkiserv/cacert.der
   ```

4. Change the ownership of the file by entering the following command from the UNIX command line:
   **Example:**
   ```bash
   chown pkisrvd /var/pkiserv/*
   ```
Configuring the UNIX runtime environment
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 Appendix A, “LDAP directory server requirements,” on page 487 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 74.

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 74.

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 19.
- You need to know the following information from LDAP installation. Copy the information into the following table from (completed) Table 8 on page 20:

<table>
<thead>
<tr>
<th>LDAP information</th>
<th>Explanation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator's distinguished name</td>
<td>This is the distinguished name to use for LDAP binding. (For a definition of distinguished name, see Table 8 on page 20)</td>
<td>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 &quot;cn=Admin&quot;</td>
</tr>
<tr>
<td>Administrator password</td>
<td>This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example:</td>
<td>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)</td>
</tr>
<tr>
<td>LDAP fully qualified domain name and port</td>
<td>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 19 for a definition of fully qualified domain name. You can specify this address with or without the preceding string “ldap://” or “ldaps://”.</td>
<td></td>
</tr>
<tr>
<td>Suffix</td>
<td>(For a definition of suffix, see Table 8 on page 20) The suffix value is specified after the suffix keyword in the LDAP server configuration file. suffix “o=your-company,c=your-country-abbreviation”</td>
<td></td>
</tr>
</tbody>
</table>
Tailoring the LDAP configuration for PKI Services

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 19 on page 73.

```bash
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 81.

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.
Tailoring the LDAP configuration for PKI Services

- 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 16 on page 37](#). 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 17). 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>
<thead>
<tr>
<th>Server instance</th>
<th>Protocol</th>
<th>SSL</th>
<th>Server authentication</th>
<th>Client authentication</th>
<th>Port number</th>
</tr>
</thead>
<tbody>
<tr>
<td>First instance</td>
<td>HTTP</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>80</td>
</tr>
<tr>
<td>First instance</td>
<td>HTTPS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>443</td>
</tr>
<tr>
<td>Second instance</td>
<td>HTTPS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>1443</td>
</tr>
</tbody>
</table>

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 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 9.
Updating z/OS HTTP Server configuration and starting the server

- You need to know the following LDAP information. Record the information in the rightmost row of Table 21. 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>
<thead>
<tr>
<th>LDAP information</th>
<th>Explanation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator's distinguished name</td>
<td>This is the distinguished name to use for LDAP binding. (For a definition of distinguished name, see Table 8 on page 20) The LDAP administrator defines the administrator's distinguished name with the adminDN keyword in the LDAP server configuration file. For example, the value is &quot;cn=Admin&quot; in the following: adminDN &quot;cn=Admin&quot;</td>
<td></td>
</tr>
<tr>
<td>Administrator password</td>
<td>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)</td>
<td></td>
</tr>
<tr>
<td>LDAP fully qualified domain name</td>
<td>This is the IP address on which the LDAP server is listening, for example, for ldap.widgets.com. See Table 7 on page 19 for a definition of fully qualified domain name.</td>
<td></td>
</tr>
<tr>
<td>LDAP port</td>
<td>This is the port for LDAP, for example, 389 in ldap.widgets.com:389</td>
<td></td>
</tr>
</tbody>
</table>

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 9)
   a. Copy the keyfile, ssimode, ssiport, and normalmode directives as is, replacing any existing values.
   b. If your organization customized the value of web_ring (see Table 10 on page 29), 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 19.)

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 9.)

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 9.)

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 29), 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 21 on page 78.

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

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 19)
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 9)
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

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 19)

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'
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 45 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 95).

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 361 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:

1. The Server line specifies the fully qualified domain and port of your LDAP server. If your file contains a Server line, PKI Services looks for the matching AuthName and AuthPwd lines and uses these values.
Tailoring the PKI Services configuration file for LDAP

2. The BindProfile parameter specifies the name of the LDAPBIND class profile. If your file does not contain a Server line but does contain a BindProfile line, PKI Services looks for the bind information in the LDAPBIND class profile. (If Server is present, PKI Services does not look for bind information in BindProfile, even if the value in Server 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 22 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 22. Information needed for updating the LDAP section of the configuration file

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Default value and your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumServers=</td>
<td>The number of available LDAP servers. These are replicas that can post certificates and CRLs.</td>
<td>From LDAP programmer</td>
<td>1</td>
</tr>
<tr>
<td>PostInterval=</td>
<td>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 &gt; 0.</td>
<td>UNIX programmer decides this. Specify a number followed by h (hours), m (minutes) or s (seconds). Example: 6m</td>
<td>5m</td>
</tr>
</tbody>
</table>

**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 22. Information needed for updating the LDAP section of the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Default value and your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server1</strong></td>
<td>You use this parameter only if you are storing LDAP passwords in the clear.</td>
<td>Copy this information from the earlier (completed) table.</td>
<td>myldapserver.mycompany.com:389</td>
</tr>
<tr>
<td></td>
<td>This parameter's value is the fully qualified domain name (domain name or IP address and port) for the first LDAP server.</td>
<td></td>
<td>Note: If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for each server.</td>
</tr>
<tr>
<td></td>
<td>If you are using a Secure Sockets Layer (SSL) session, the fully qualified domain name should be preceded by ldaps://.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AuthName1</strong></td>
<td>You use this parameter only if you are storing LDAP passwords in the clear.</td>
<td>Copy this information from the earlier (completed) table.</td>
<td>CN=root</td>
</tr>
<tr>
<td></td>
<td>This parameter’s value is the distinguished name to use for LDAP binding.</td>
<td></td>
<td>Notes:</td>
</tr>
<tr>
<td></td>
<td>(See Table 8 on page 20 for a definition of distinguished name.)</td>
<td></td>
<td>• If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for each server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• The default name of the LDAP server configuration file is ds.conf for the IBM Tivoli Directory Server for z/OS LDAP server.</td>
</tr>
</tbody>
</table>
### Tailoring the PKI Services configuration file for LDAP

#### Table 22. Information needed for updating the LDAP section of the configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Information needed</th>
<th>Where to get this information</th>
<th>Default value and your company’s information</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthPwd1=</td>
<td>You use this parameter only if you are storing LDAP passwords in the clear.</td>
<td>Copy this information from the earlier (completed) table.</td>
<td>root</td>
</tr>
<tr>
<td></td>
<td>This parameter’s value is the password to use for LDAP binding. The LDAP programmer sets this.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note:</td>
<td>Include this parameter, Server1, and AuthName1 only if you are storing the LDAP password in the clear.</td>
<td>Note: If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for each server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CreateOUValue=</td>
<td>Value to use for the OU attribute when creating LDAP entries under the objectclass organizationalUnit. (See Table 88 on page 487)</td>
<td>UNIX programmer decides this (after consulting with LDAP programmer)</td>
<td>Created by PKI Services</td>
</tr>
<tr>
<td></td>
<td>This is used only when no OU value is specified in the relative distinguished name.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RetryMissingSuffix=</td>
<td>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.</td>
<td>UNIX programmer decides this (after consulting with LDAP programmer)</td>
<td>T</td>
</tr>
</tbody>
</table>

For more information, see Using encrypted passwords for LDAP servers on page 361.
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 22 on page 82:
   ```
   NumServers=1
   ```

2. Optionally change 5m in the following line to the posting interval in Table 22 on page 82:
   ```
   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 362).
        ```
        # 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 22 on page 82:
        ```
        Server1=your-ldap-server-address:port
        ```
     b. Change `CN=root` in the following line to the value of the administrator distinguished name in Table 22 on page 82.
Tailoring the PKI Services configuration file for LDAP

AuthName1=CN=root

C. Change root in the following line to the value of the administrator password in Table 22 on page 82
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 22 on page 82
CreateOUValue=Created by PKI Services

6. If necessary, change T in the following line to the RetryMissingSuffix value in Table 22 on page 82
RetryMissingSuffix=T
Chapter 9. Creating VSAM data sets

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 92

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.

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 using a sysplex, 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 following guidelines to update the space allocation parameters for the DEFINE CLUSTER and DEFINE ALTERNATEINDEX statements. (For more information about IDCAMS, see [DFSMS Access Method Services for Catalogs](https://www.ibm.com-support.z抢zOSDFSMSAccessMethodServicesforCatalogs)).

Determining storage needs for ICL

The ICL maintains a permanent record for each certificate PKI Services issues. There is one ICL record for each issued certificate. Unless set up otherwise, the ICL
Creating VSAM data sets

An ICL record grows continuously over time as more certificates are issued. Assuming average-size certificates, one ICL record occupies 1024 bytes of storage. 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

The object store holds records to track active certificate requests. There is one object store record for each active certificate request and potentially another record to post the certificate to the LDAP directory. 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. Assuming average-size certificate requests, 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. This would mean the data set should be able to hold at least 41 000 concurrent certificate requests. If multiple volume support is used, you can double this amount.

If your anticipated needs differ 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 473 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.)

(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 ObjectStore 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 90 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 \textit{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>
<thead>
<tr>
<th>VSAM information you need to record</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of storage class for VSAM RLS</td>
<td></td>
</tr>
</tbody>
</table>

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.

\textbf{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 \textit{“(Optional) preliminary steps for establishing VSAM RLS” on page 88.}

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 \texttt{SYS1.SAMPLIB(IKYCVSAM)} to your JCL data set. (See \textit{“IKYCVSAM” on page 473} 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 \textit{Table 23} for example:
        \begin{verbatim}
        STORCLAS(VSAMRLS)
        \end{verbatim}
      - If you are running without a Parallel Sysplex, replace the \texttt{vvvvv} in the VOL statements with a \texttt{VOL=SER} suitable for your VSAM data sets.
Creating 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. These lines follow:
   SPANNED -
   CISIZE(1024) -

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 [1a].)

e. Update the primary and secondary extent allocations based on your anticipated future needs. (See "Planning VSAM storage requirements" on page 87 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 SharedVSAM=T in the pkiserv.conf configuration file. (See the SharedVSAM row in Table 18 on page 52.) 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 88.

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 ObjectStore 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 52.

e. Update the primary and secondary extent allocations based on your anticipated future needs. (See "Planning VSAM storage requirements" on page 87 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 484 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 89.)

**Example:**

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

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.config).

4. Optionally, you might need to adjust the numeric values for BUFNI and BUFND.
   **Tip:** The STATUS and REQUESTOR 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](https://www.ibm.com/redbooks/pdfs/sg246969.pdf)).

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

<table>
<thead>
<tr>
<th>Default value</th>
<th>Suggested value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectDSN='pkisrvd.vsam.ost'</td>
<td>ObjectDSN=DD:OST</td>
</tr>
<tr>
<td>ObjectTidDSN='pkisrvd.vsam.ost.path'</td>
<td>ObjectTidDSN=DD:TID</td>
</tr>
<tr>
<td>ObjectStatusDSN='pkisrvd.vsam.ost.status'</td>
<td>ObjectStatusDSN=DD:OSTAT</td>
</tr>
<tr>
<td>ObjectRequestorDSN='pkisrvd.vsam.ost.requestor'</td>
<td>ObjectRequestorDSN=DD:OREQ</td>
</tr>
<tr>
<td>ICLDSN='pkisrvd.vsam.icl'</td>
<td>ICLDSN=DD:ICL</td>
</tr>
<tr>
<td>ICLStatusDSN='pkisrvd.vsam.icl.status'</td>
<td>ICLStatusDSN=DD:ISTAT</td>
</tr>
<tr>
<td>ICLRequestorDSN='pkisrvd.vsam.icl.requestor'</td>
<td>ICLRequestorDSN=DD:IREQ</td>
</tr>
</tbody>
</table>

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](https://www.ibm.com/redbooks/pdfs/sg244746.pdf).

**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 481 for a code sample of this file.)

2. Update the JCL as directed in the instructions in the prolog of the sample JCL:
Creating VSAM data sets

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 VOLUME value that contains sufficient free space to contain a complete backup of both 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 95.

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 95.

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 483 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 95.

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 95.

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

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 77) 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 9.
• 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 204 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.
   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.
Starting and stopping 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 48), 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 29.

   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 97), 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 22 on page 82.
   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
Starting and stopping PKI Services
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 101 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 151 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 157 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 187 explains:
  - “Scaling for high volume installations” on page 187
  - “Using certificate policies” on page 188
  - “Updating the signature algorithm” on page 191
  - “Customizing distribution point CRLs” on page 192
  - “Creating a distribution point ARL” on page 199
  - “Adding an application domain” on page 200
  - “Adding a new CA domain” on page 204
  - “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 219
  - “Setting up automatic renewal of certificates” on page 230
  - Chapter 15, “Customizing with installation exits,” on page 235
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 157.

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 24:

<table>
<thead>
<tr>
<th>Table 24. Structure and main divisions of the certificate template file (pkiserv.tmpl)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A prolog section of comments explaining main sections, subsections, named fields, and substitution variables.</td>
</tr>
<tr>
<td>A DEBUG flag appears right after the prolog section. (You can change DEBUG=0 to DEBUG=1 to get CGI debugging information.)</td>
</tr>
</tbody>
</table>

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 275. For details, see "APPLICATION sections" on page 110.

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 114.

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 284.

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.

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:

\[\text{base64cert}\]

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 \([\text{base64cert}]\) substitution variable is meaningless before the certificate is retrieved. Therefore, in this case, the value of \([\text{base64cert}]\) would be the null string (an empty string).

<table>
<thead>
<tr>
<th>Table 25 summarizes valid substitution variables:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table 25. Substitution variables</strong></td>
</tr>
<tr>
<td><strong>Substitution variable</strong></td>
</tr>
<tr>
<td>altrawvalue</td>
</tr>
<tr>
<td>base64cert</td>
</tr>
<tr>
<td>browsertype</td>
</tr>
<tr>
<td>cadomain</td>
</tr>
<tr>
<td>errorinfo</td>
</tr>
<tr>
<td>iecert</td>
</tr>
<tr>
<td>keyid</td>
</tr>
<tr>
<td>optfield</td>
</tr>
<tr>
<td>p12cert</td>
</tr>
<tr>
<td>printablecert</td>
</tr>
</tbody>
</table>
Customizing the end-user Web pages using REXX CGIs

Table 25. Substitution variables (continued)

<table>
<thead>
<tr>
<th>Substitution variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>readonly</td>
<td>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 <code>readonly</code> when the INSERT is used for output purposes, such as when displaying request or certificate information.</td>
</tr>
<tr>
<td>requestor</td>
<td>The requestor's e-mail address, when the keys for the certificate were generated by PKI Services.</td>
</tr>
<tr>
<td>serialno</td>
<td>The serial number of a certificate.</td>
</tr>
<tr>
<td>tmplname</td>
<td>A certificate template name. This is primed from the HTML tag <code>&lt;SELECT NAME=&quot;Template&quot;&gt;</code> in the <code>&lt;APPLICATION NAME=PKISERV&gt;</code> section. The end user selects it on the first Web page.</td>
</tr>
<tr>
<td>transactionid</td>
<td>A unique value returned from a certificate request.</td>
</tr>
</tbody>
</table>

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 135 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 114) 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:

```xml
<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.
Customizing the end-user Web pages using REXX CGIs

The following example of an INSERT defines a certificate field.

Example:
```
<INSERT NAME=Country>
<p> Country [optfield] <BR>
<INPUT NAME="Country" TYPE="text" SIZE=2 maxlength="2">
<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 26.)
- INSERTs that are related to the certificate content. (See Table 27 on page 105.) These include:
  - X.509 fields (for example, OrgUnit)
  - Non-X.509 fields (for example UserId).

Table 26. INSERTs that are common HTML for Web page content. These INSERTS are for internal processing.

<table>
<thead>
<tr>
<th>INSERT NAME</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>-AdditionalHeadIE</td>
<td>ActiveX controls to enable Internet Explorer to generate a key pair</td>
</tr>
<tr>
<td>-ChallengePassphrase, ChallengePassPhrase2</td>
<td>HTML for a Web page that requests the passphrase that was specified when a certificate request was submitted</td>
</tr>
<tr>
<td>-RecoverEmail, -RecoverEmail2</td>
<td>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.</td>
</tr>
<tr>
<td>-requestok</td>
<td>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 284)</td>
</tr>
<tr>
<td>-requestbad</td>
<td>HTML for the Web page “Request was not successful”</td>
</tr>
<tr>
<td>-requestor</td>
<td>HTML for a Web page that requests the name of a certificate requestor.</td>
</tr>
</tbody>
</table>
### Customizing the end-user Web pages using REXX CGIs

#### Table 26. INSERTs that are common HTML for Web page content (continued). These INSERTS are for internal processing.

<table>
<thead>
<tr>
<th>INSERT NAME</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>-requestor2</td>
<td>HTML for a Web page that requests the e-mail address of a certificate requestor.</td>
</tr>
<tr>
<td>-renewkeysetIE</td>
<td>ActiveX controls to create a renewal certificate request using the original certificates key pair.</td>
</tr>
<tr>
<td>-renewkeysetNS</td>
<td>Netscape based browser script for renewal certificate requests.</td>
</tr>
<tr>
<td>-renewrevokeok</td>
<td>HTML for the Web page “Request submitted successfully” after a successful attempt to revoke a certificate. (See Figure 49 on page 291 for a sample of the Web page to renew or revoke a certificate.)</td>
</tr>
<tr>
<td>-renewrevokebad</td>
<td>HTML for the Web page “Request was not successful” after an unsuccessful attempt to renew or revoke a certificate. (See Figure 49 on page 291 for a sample of the Web page to renew or revoke a certificate.)</td>
</tr>
<tr>
<td>-returnp12cert</td>
<td>HTML for a Web page that displays a PKCS #12 package.</td>
</tr>
<tr>
<td>-preregok</td>
<td>HTML for the Web page “Preregistration successful” after a successful attempt to preregister a client for a certificate.</td>
</tr>
<tr>
<td>-returnpkcs10</td>
<td>This returns the server certificate in B64 format.</td>
</tr>
<tr>
<td>returnbrowsercertNS</td>
<td>This contains [base64cert], which is the base64 substitution variable.</td>
</tr>
<tr>
<td>returnbrowsercertIE</td>
<td>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 286 for a sample of this Web page.</td>
</tr>
</tbody>
</table>

#### Named fields in INSERT sections

Most of the following fields are X.509 fields. Table 27 summarizes the named fields in INSERT sections. (See Restrictions at the end of the table.)

#### Table 27. Named fields in INSERT sections

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AltDomain</td>
<td>The host name of the machine where a certificate will be installed. This is a text field of up to 100 characters. <strong>Note:</strong> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</td>
</tr>
<tr>
<td>AltEmail</td>
<td>The user's e-mail address, including the @ character and any periods (.). This is a text field of up to 100 characters. <strong>Note:</strong> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</td>
</tr>
</tbody>
</table>
### Table 27. Named fields in INSERT sections (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AltIPAddr</strong></td>
<td>The unique IP address that specifies the location of the server or device 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.</td>
</tr>
<tr>
<td></td>
<td>- For IP version 4, the IP address is in dotted decimal format; for example, 9.67.97.103.</td>
</tr>
<tr>
<td></td>
<td>- 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 0001:0002:0000:0000:0000:0003:0004.</td>
</tr>
<tr>
<td></td>
<td>- 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: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</td>
</tr>
<tr>
<td><strong>AltOther</strong></td>
<td>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 147.)</td>
</tr>
<tr>
<td><strong>AltURI</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>ChallengePassPhrase</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>ClientName</strong></td>
<td>Name of the person or device being preregistered. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Restriction:</strong> The first 32 characters of the name must be unique, irrespective of case, for each preregistered user.</td>
</tr>
<tr>
<td><strong>CommonName</strong></td>
<td>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, <a href="http://www.ibm.com">www.ibm.com</a>. CommonName is a text field of up to 64 characters.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</td>
</tr>
</tbody>
</table>
Table 27. Named fields in INSERT sections (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>The country where your organization is located. This is a 2-character text field.</td>
</tr>
<tr>
<td></td>
<td>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</td>
</tr>
<tr>
<td>DNQualifier</td>
<td>The subject's distinguished name qualifier. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>DomainName</td>
<td>The subject's domain name. It contains all the domain name components in the form &lt;domain component1&gt;.&lt;domain component2&gt;. ... &lt;domain componentn&gt;. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>Email</td>
<td>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.</td>
</tr>
<tr>
<td>EmailAddr</td>
<td>The e-mail address for the distinguished name. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>ExtKeyUsage</td>
<td>The intended purpose of the certificate. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>clientauth: Client side authentication</td>
</tr>
<tr>
<td></td>
<td>codesigning: Code signing</td>
</tr>
<tr>
<td></td>
<td>emailprotection: E-mail protection</td>
</tr>
<tr>
<td></td>
<td>mssmartcardlogon: Microsoft Smartcard logon</td>
</tr>
<tr>
<td></td>
<td>ocspsigning: OCSP response signing</td>
</tr>
<tr>
<td></td>
<td>serverauth: Server side authentication</td>
</tr>
<tr>
<td></td>
<td>timestamping: Digital timestamping.</td>
</tr>
<tr>
<td>HostIdMap</td>
<td>This is the user ID for authorization purposes, in an e-mail type of format:</td>
</tr>
<tr>
<td></td>
<td>subject-id@host-name</td>
</tr>
<tr>
<td></td>
<td>For example, this could be <a href="mailto:dsmith@ibm.com">dsmith@ibm.com</a>. This is a text field of up to 100 characters.</td>
</tr>
<tr>
<td></td>
<td>There are three ways to use %HostIdMap%:</td>
</tr>
<tr>
<td></td>
<td>• If you place it in the CONTENT section, the end user can specify the value (or values, because it can be repeated).</td>
</tr>
<tr>
<td></td>
<td>• You can also place it in the APPL section that the application provides. If you do so, it should have the following form:</td>
</tr>
<tr>
<td></td>
<td>%HostIdMap=%host-name%</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>• 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 &quot;Administering HostIdMappings extensions&quot; on page 342 for more information.</td>
</tr>
<tr>
<td>KeyProt</td>
<td>(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.</td>
</tr>
<tr>
<td>KeySize</td>
<td>The size of the keys (public key and private key) in bits, if they are to be generated by PKI Services. Valid values are 512–2048.</td>
</tr>
</tbody>
</table>
### Customizing the end-user Web pages using REXX CGIs

Table 27. Named fields in INSERT sections (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyUsage</td>
<td>The intended purpose of the certificate. Each possible value is shown here with its intended purpose and possible PKIX bits:</td>
</tr>
<tr>
<td></td>
<td><strong>KeyUsage value</strong></td>
</tr>
<tr>
<td></td>
<td>certsign</td>
</tr>
<tr>
<td></td>
<td>crlsign</td>
</tr>
<tr>
<td></td>
<td>dataencrypt,</td>
</tr>
<tr>
<td></td>
<td>dataencipherment,</td>
</tr>
<tr>
<td></td>
<td>dataencipherment,</td>
</tr>
<tr>
<td></td>
<td>or dataencipherment</td>
</tr>
<tr>
<td></td>
<td>digitalsig</td>
</tr>
<tr>
<td></td>
<td>digitalsignature</td>
</tr>
<tr>
<td></td>
<td>docsing or nonrepudiation</td>
</tr>
<tr>
<td></td>
<td>handshake</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>keyagree or keyagreement</td>
</tr>
<tr>
<td></td>
<td>keycertsign</td>
</tr>
<tr>
<td></td>
<td>keyencrypt,</td>
</tr>
<tr>
<td></td>
<td>keyencipherment,</td>
</tr>
<tr>
<td></td>
<td>or keyencipherment</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Label</th>
<th>The label assigned to the requested certificate. This is a text field of up to 32 characters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locality</td>
<td>The city or municipality where your organization is located, such as Pittsburgh or Paris.</td>
</tr>
<tr>
<td></td>
<td>This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value is one of the relative distinguished names that is saved in the subject's</td>
</tr>
<tr>
<td></td>
<td>distinguished name in the certificate.</td>
</tr>
<tr>
<td>Mail</td>
<td>The e-mail address for the distinguished name.</td>
</tr>
<tr>
<td></td>
<td>This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value is one of the relative distinguished names that is saved in the subject's</td>
</tr>
<tr>
<td></td>
<td>distinguished name in the certificate.</td>
</tr>
<tr>
<td>NotBefore</td>
<td>Number of days (0–30) before the certificate becomes valid.</td>
</tr>
<tr>
<td>NotAfter</td>
<td>Number of days (365–720) that the certificate is current. This is a period of time from</td>
</tr>
<tr>
<td></td>
<td>365 days (1 year) to 720 days (2 years).</td>
</tr>
<tr>
<td>NotifyEmail</td>
<td>The e-mail address for notification purposes. If automatic certificate renewal is in effect,</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Notes:</strong></td>
</tr>
<tr>
<td></td>
<td>1. When a certificate is created and posted to LDAP, the NotifyEmail value, if specified,</td>
</tr>
<tr>
<td></td>
<td>is posted as the MAIL attribute. If the MAIL attribute already exists in that directory entry,</td>
</tr>
<tr>
<td></td>
<td>its value is replaced by the new value. If both NotifyEmail and Email appear on one request,</td>
</tr>
<tr>
<td></td>
<td>they must have the same value.</td>
</tr>
<tr>
<td></td>
<td>2. If a certificate for which PKI Services generated the keys is renewed, the NotifyEmail</td>
</tr>
<tr>
<td></td>
<td>field is ignored, and the renewed certificate is sent to the requestor's e-mail address.</td>
</tr>
<tr>
<td>Org</td>
<td>Organization. The legally registered name (or trademark name, for example, IBM) of your</td>
</tr>
<tr>
<td></td>
<td>organization. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value is one of the relative distinguished names that is saved in the subject's</td>
</tr>
<tr>
<td></td>
<td>distinguished name in the certificate.</td>
</tr>
</tbody>
</table>
### Table 27. Named fields in INSERT sections (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrgUnit</td>
<td>The name of your division or department. This is a text field of up to 64 characters. <strong>Note:</strong> The value is one of the relative distinguished names that is saved in the subject’s distinguished name in the certificate.</td>
</tr>
<tr>
<td>OrgUnit2</td>
<td>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. <strong>Note:</strong> The value is one of the relative distinguished names that is saved in the subject’s distinguished name in the certificate.</td>
</tr>
<tr>
<td>PassPhrase1</td>
<td>The user decides this and enters 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.</td>
</tr>
<tr>
<td>PostalCode1</td>
<td>The zip code or postal code. This is a text field of up to 64 characters. <strong>Note:</strong> The value is one of the relative distinguished names that is saved in the subject’s distinguished name in the certificate.</td>
</tr>
</tbody>
</table>
Table 27. Named fields in INSERT sections (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SerialNumber</td>
<td>Serial number of the subject device. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>SignWith</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Examples:</strong></td>
</tr>
<tr>
<td></td>
<td>&quot;SAF:CERTAUTH/Local CA Cert&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;PKI:&quot;</td>
</tr>
<tr>
<td></td>
<td>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'.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>StateProv</td>
<td>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.</td>
</tr>
<tr>
<td>Street</td>
<td>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.</td>
</tr>
<tr>
<td>Title</td>
<td>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.</td>
</tr>
<tr>
<td>TransactionId</td>
<td>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.</td>
</tr>
<tr>
<td>Uid</td>
<td>The subject's login ID. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>UnstructAddr</td>
<td>Unstructured address of the subject device. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>UnstructName</td>
<td>Unstructured device name. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>UserId</td>
<td>The owning SAF user ID. This is a text field of up to 8 characters.</td>
</tr>
</tbody>
</table>

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:

```xml
<APPLICATION NAME="appl-name">...</APPLICATION>
```

Each application section begins with an application name definition.

Examples:

```xml
<APPLICATION NAME="PKISERV">
  This application contains support for all templates and functions.
</APPLICATION>
```
Customizing the end-user Web pages using REXX CGIs

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 28.

<table>
<thead>
<tr>
<th>Section or subsection</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTENT</td>
<td>HTML for the “PKI Services Certificate Generation Application” Web page. (For a sample of this Web page, see Figure 36 on page 275) This subsection should contain one or more named fields (see &quot;What are named fields?&quot; on page 103) 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.</td>
</tr>
<tr>
<td>FAILURECONTENT</td>
<td>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.</td>
</tr>
<tr>
<td>FINDRECOVERCONTENT</td>
<td>This subsection contains the information for recovering a certificate in the case where the user has forgotten the passphrase. <strong>Note:</strong> Only certificates whose keys were generated by PKI Services are recoverable.</td>
</tr>
<tr>
<td>RECONTENT</td>
<td>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 291) This subsection uses the substitution variable [printablecert], which contains the data extracted from the ICL entry. (See &quot;What are substitution variables?&quot; on page 102.)</td>
</tr>
<tr>
<td>RECONTENT2</td>
<td>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.</td>
</tr>
<tr>
<td>RESUCCESSCONTENT</td>
<td>Contains only the named field %%renewrevokeok%% (whose associated INSERT contains HTML for the Web page “Request submitted successfully”).</td>
</tr>
<tr>
<td>REFailureCONTENT</td>
<td>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.</td>
</tr>
<tr>
<td>RETRIEVECONTENT2</td>
<td>This subsection contains the HTML to allow the end user to retrieve a recovered certificate.</td>
</tr>
<tr>
<td>RETURNCERT</td>
<td>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.</td>
</tr>
</tbody>
</table>
Customizing the end-user Web pages using REXX CGIs

Table 28. Subsections of the APPLICATION sections (continued)

<table>
<thead>
<tr>
<th>Section or subsection</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMINSCOPE</td>
<td>This subsection is for an administration page. It contains only the named field %%%SelectCA_dom%%% to prompt the administrator to choose a domain. (For more information, see “Adding a new CA domain” on page 204.)</td>
</tr>
<tr>
<td>ADMINHEADER</td>
<td>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 154 for more information.</td>
</tr>
<tr>
<td>ADMINFOOTER</td>
<td>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 154 for more information.</td>
</tr>
</tbody>
</table>

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 282 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 29. Certificate templates PKI Services provides

<table>
<thead>
<tr>
<th>Certificate template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-year SAF server certificate</td>
<td>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.</td>
</tr>
<tr>
<td>One-year SAF browser certificate</td>
<td>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 27 on page 105 for descriptions of fields) because the certificate is stored in a RACF database. This certificate is auto-approved.</td>
</tr>
<tr>
<td>Certificate template</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>One-year PKI SSL browser certificate</td>
<td>This template allows end users to request a browser certificate that PKI Services generates. The end user enters the common name. (See Table 27 on page 105 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.</td>
</tr>
<tr>
<td>One-year PKI S/MIME browser certificate</td>
<td>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.</td>
</tr>
<tr>
<td>One-year PKI generated key certificate</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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 27 on page 105.) |
| 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 Internet Explorer based requests 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.                                           |
Customizing the end-user Web pages using REXX CGIs

Table 29. Certificate templates PKI Services provides (continued)

<table>
<thead>
<tr>
<th>Certificate template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Five-year PKI intermediate CA server certificate</td>
<td>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.</td>
</tr>
<tr>
<td>Five-year SCEP certificate—Preregistration</td>
<td>This template supports certificate preregistration for Simple Certificate Enrollment Protocol (SCEP) clients. The PassPhrase is required.</td>
</tr>
<tr>
<td>n-year PKI browser certificate for extensions demonstration</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 29 on page 112.

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 30. Names, aliases, and nicknames of certificate templates

<table>
<thead>
<tr>
<th>True name</th>
<th>Alias</th>
<th>Nickname</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Year PKI SSL Browser Certificate</td>
<td>PKI Browser Certificate</td>
<td>1YBSSL</td>
</tr>
<tr>
<td>1-Year PKI S/MIME Browser Certificate</td>
<td>PKI Browser Certificate</td>
<td>1YBSM</td>
</tr>
<tr>
<td>1-Year PKI Generated Key Certificate</td>
<td>PKI Key Certificate</td>
<td>1YKRC</td>
</tr>
<tr>
<td>2-Year PKI Browser Certificate For Authenticating To z/OS</td>
<td>PKI Browser Certificate</td>
<td>2YBZOS</td>
</tr>
<tr>
<td>2-Year PKI Authenticode—Code Signing Certificate</td>
<td>PKI Server Certificate</td>
<td>2YIACS</td>
</tr>
<tr>
<td>2-Year PKI Windows Logon Certificate</td>
<td>PKI Browser Certificate</td>
<td>2YBWl</td>
</tr>
<tr>
<td>5-Year PKI SSL Server Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSSSL</td>
</tr>
<tr>
<td>5-Year PKI IPSEC Server (Firewall) Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSIPS</td>
</tr>
<tr>
<td>5-Year PKI Intermediate CA Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSCA</td>
</tr>
<tr>
<td>5-Year SCEP Certificate - Preregistration</td>
<td>—</td>
<td>5YSCEPP</td>
</tr>
<tr>
<td>n-Year PKI Certificate for Extensions Demonstration</td>
<td>PKI Browser Certificate</td>
<td>SAMPLB</td>
</tr>
<tr>
<td>1-Year SAF Browser Certificate</td>
<td>SAF Browser Certificate</td>
<td>—</td>
</tr>
<tr>
<td>1-Year SAF Server Certificate</td>
<td>SAF Server Certificate</td>
<td>—</td>
</tr>
</tbody>
</table>

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 299.

See [Setting up automatic renewal of certificates” on page 230](#) 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
Customizing the end-user Web pages using REXX CGIs

- ADMINAPPROVE
- SUCCESSCONTENT
- FAILURECONTENT
- RETRIEVECONTENT
- RETURNCERT
- PREREPORT

<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 282 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> Common Name [optfield] 
<BR>
.INPUT NAME="CommonName" TYPE="text" SIZE=64 maxlength="64">
<INSERT>

<INSERT NAME=Requestor>
<p> Your name for tracking this request [optfield] <BR>
.INPUT NAME="Requestor" TYPE="text" SIZE=32 maxlength="32">
<INSERT>

Named fields in this subsection are optional if the named field contains more than one word within the %% delimiters (as in %%Requestor (optional)%%). The user need not supply a value for Requestor.

<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>...

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 27 on page 105, you can also include the following named fields in this subsection only.
Customizing the end-user Web pages using REXX CGIs

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:
- BasicConstraints (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

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 188 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 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 188 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:
Customizing the end-user Web pages using REXX CGIs

<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
- DNQualifier
- DomainName
- EmailAddr
- EndDate
- ExtKeyUsage
- HostIdMap (can repeat)
- KeyUsage
- Locality
- Mail
Customizing the end-user Web pages using REXX CGIs

- 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 27 on page 105)

**Example:**
```
<ADMINAPPROVE>
%%KeyUsage%%
%%CommonName%%
%%OrgUnit%%
%%Org%%
%%Country%%
%%HostIdMap%%
%%HostIdMap%%
%%HostIdMap%%
<ADMINAPPROVE>

<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 103 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 284.)

<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.
Customizing the end-user Web pages using REXX CGIs

In all of the templates included with PKI Services, <SUCCESSCONTENT> contains only the named field %%requestbad%%. (See "What are named fields?" on page 103 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 285. 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 103 for an explanation of named fields.)
- The title of the Web page (This appears in the banner of your browser. Figure 41 on page 285 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 102 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
Customizing the end-user Web pages using REXX CGIs

(SCEP) rules for approval of a SCEP request. (For details, see “Overview of certificate request processing for preregistered SCEP clients” on page 220.)

Example:

<PREREGISTER>
AuthenticatedClient=AutoApprove
SemiauthenticatedClient=AdminApprove
UnauthenticatedClient=Reject
SubsequentRequest=AutoApprove
RenewalRequest=AutoApprove
</PREREGISTER>

Summary of subsections contained in certificate templates
The following table summarizes the subsections that are present in the various certificate templates in the pkiserv.tmpl file (as it is shipped):

<table>
<thead>
<tr>
<th>Subsection (in TEMPLATE section)</th>
<th>1-year PKI SSL browser</th>
<th>1-year PKI SSL/S/MIME browser</th>
<th>1-year SAF browser</th>
<th>1-year SAF server</th>
<th>1-year PKI key</th>
<th>2-year PKI browser for z/OS</th>
<th>2-year PKI Authenticode signing server</th>
<th>2-year PKI Windows logon browser</th>
<th>5-year PKI SSL server</th>
<th>5-year PKI IPSEC server (firewall)</th>
<th>5-year PKI intermediate CA server</th>
<th>5-year SCEP pre-registration</th>
<th>n-year PKI browser extensions demo</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTENT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>APPL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CONSTANT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ADMINAPPROVE</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SUCCESSCONTENT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FAILURECONTENT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RETRIEVECONTENT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RETURNCERT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PREREGISTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 32 on page 122

Fields in the PKI server certificate templates
Table 33 on page 123

Fields in the SAF (browser and server), SCEP, and PKI generated key certificate templates
Table 34 on page 125

Table 32 on page 122, Table 33 on page 123, and Table 34 on page 125 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)
**Customizing the end-user Web pages using REXX CGIs**

Table 32. Summary of fields for PKI browser certificate templates

<table>
<thead>
<tr>
<th>Field name</th>
<th>One-year PKI SSL browser</th>
<th>One-year PKI S/MIME browser</th>
<th>Two-year PKI browser for z/OS</th>
<th>Two-year PKI Windows logon certificate</th>
<th>n-year PKI browser extensions demonstration</th>
</tr>
</thead>
<tbody>
<tr>
<td>AltDomain</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltEmail</td>
<td>Required</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltIPAddr</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltOther_OID</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Constant 1</td>
<td>Optional</td>
</tr>
<tr>
<td>AltURI</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Constant 2</td>
<td></td>
</tr>
<tr>
<td>AuthInfoAcc</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Constant: 1</td>
<td></td>
</tr>
<tr>
<td>CertPolicies</td>
<td>Constant: 1</td>
<td></td>
<td></td>
<td>Constant: 1</td>
<td></td>
</tr>
<tr>
<td>ClientName</td>
<td>Required</td>
<td>Constant 1</td>
<td>Optional</td>
<td>Constant: 2</td>
<td>Optional</td>
</tr>
<tr>
<td>CommonName</td>
<td>Required</td>
<td>Constant: 3</td>
<td>Optional</td>
<td>Constant: 3</td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Constant: 2</td>
<td></td>
</tr>
<tr>
<td>Critical</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Constant: 2</td>
<td></td>
</tr>
<tr>
<td>DomainName</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Constant: 2</td>
<td></td>
</tr>
<tr>
<td>DNQualifier</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Constant: 2</td>
<td></td>
</tr>
<tr>
<td>EmailAddr</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Constant: 2</td>
<td></td>
</tr>
<tr>
<td>ExtKeyUsage</td>
<td>Constant: clientauth</td>
<td>Constant: clientauth</td>
<td>Constants: clientauth and mssmartlogon</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>HostldMap</td>
<td>Application provides</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>KeySize</td>
<td></td>
<td></td>
<td></td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>KeyUsage</td>
<td>Constant: handshake</td>
<td>Constant: digitalSig</td>
<td>Required</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Locality</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>Mail (previously called Email)</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>NotAfter</td>
<td>Constant: 365</td>
<td>Constant: 730</td>
<td>Optional</td>
<td>Constant: 730</td>
<td>Optional</td>
</tr>
<tr>
<td>NotBefore</td>
<td>Constant: 0</td>
<td></td>
<td>Optional</td>
<td>Constant: 0</td>
<td>Optional</td>
</tr>
<tr>
<td>NotifyEmail</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Org</td>
<td>Constant: The Firm</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>OrgUnit</td>
<td>Constant: Class 1 Internet Certificate CA</td>
<td>Required</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OrgUnit2</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>PassPhrase</td>
<td>Required</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>PostalCode</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>PublicKey</td>
<td>Browser provided</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Requestor</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>SerialNumber</td>
<td>Optional</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>SignWith</td>
<td>Constant: PKI:</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>
Customizing the end-user Web pages using REXX CGIs

Table 32. Summary of fields for PKI browser certificate templates (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>One-year PKI SSL browser</th>
<th>One-year PKI S/MIME browser</th>
<th>Two-year PKI browser for z/OS</th>
<th>Two-year PKI Windows logon certificate</th>
<th>n-year PKI browser extensions demonstration</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateProv</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uid</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnstructAddr</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnstructName</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserId</td>
<td>Application provides</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.O CSP.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.

Table 33. Summary of fields for PKI server certificate templates

<table>
<thead>
<tr>
<th>Field name</th>
<th>Two-year PKI Authenticode code signing server</th>
<th>Five-year PKI SSL server</th>
<th>Five-year PKI IPSEC server (firewall)</th>
<th>Five-year PKI intermediate CA server</th>
</tr>
</thead>
<tbody>
<tr>
<td>AltDomain</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltEmail</td>
<td>Required Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltIPAddr</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltOther_OID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltURI</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AuthInfoAcc</td>
<td>Constant 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CertPolicies</td>
<td>Constant: 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClientName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CommonName</td>
<td>Constant: My Company Code Signing Certificate</td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical</td>
<td>Constant: ExtKeyUsage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNQualifier</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DomainName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EmailAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Customizing the end-user Web pages using REXX CGIs

Table 33. Summary of fields for PKI server certificate templates (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Two-year PKI Authenticode code signing server</th>
<th>Five-year PKI SSL server</th>
<th>Five-year PKI IPSEC server (firewall)</th>
<th>Five-year PKI intermediate CA server</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtKeyUsage</td>
<td>Constant: codesigning</td>
<td>Constant: serverauth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HostIdMap</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeySize</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeyUsage</td>
<td>Constants: digitalsig and docsing</td>
<td>Constant: handshake</td>
<td>Constants: handshake and dataencrypt</td>
<td>Constant: certsign</td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Locality</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mail</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NotAfter</td>
<td>Constant: 730</td>
<td>Constant: 1825</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NotBefore</td>
<td>Constant: 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NotifyEmail</td>
<td>Required</td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Org</td>
<td>Constant: The Firm</td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OrgUnit</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OrgUnit2</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PassPhrase</td>
<td>Required</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PostalCode</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PublicKey</td>
<td>Required</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requestor</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SerialNumber</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SignWith</td>
<td>Constant: PKI:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StateProv</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnstructAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnstructName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserId</td>
<td>Application provides</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
2. The HostIdMap value is formed by concatenating UserId with @host-name.
Table 34. Summary of fields for SAF, SCEP, and PKI generated key certificate templates

<table>
<thead>
<tr>
<th>Field name</th>
<th>One-year SAF server</th>
<th>One-year SAF browser</th>
<th>Five-year SCEP preregistration</th>
<th>One-year PKI generated key</th>
</tr>
</thead>
<tbody>
<tr>
<td>AltDomain</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltEmail</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltIPAddr</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltOther_OID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AltURI</td>
<td>Optional</td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AuthInfoAcc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CertPolicies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClientName</td>
<td>Optional</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CommonName</td>
<td>Optional</td>
<td>Constant: 1</td>
<td>Optional</td>
<td>Required</td>
</tr>
<tr>
<td>Country</td>
<td>Required</td>
<td>Constant: US</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Critical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EmailAddr</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>ExtKeyUsage</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>HostIdMap</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>KeySize</td>
<td></td>
<td></td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>KeyUsage</td>
<td></td>
<td></td>
<td>Optional</td>
<td>Constant: handshake</td>
</tr>
<tr>
<td>Label</td>
<td>Required</td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Locality</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mail (previously called Email)</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>NotAfter</td>
<td>Constant: 365</td>
<td>Constant: 1825</td>
<td>Constant: 365</td>
<td></td>
</tr>
<tr>
<td>NotBefore</td>
<td>Constant: 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NotifyEmail</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Org</td>
<td>Required</td>
<td>Constant: The Firm</td>
<td>Optional</td>
<td>Constant: The Firm</td>
</tr>
<tr>
<td>OrgUnit</td>
<td>Required</td>
<td>Constants: OrgUnit=SAF template certificate and OrgUnit=Nuts and Bolts Division</td>
<td>Optional</td>
<td>Constant: Class 1 Internet Certificate CA</td>
</tr>
<tr>
<td>OrgUnit2</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PassPhrase</td>
<td></td>
<td>Required</td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>PostalCode</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>PublicKey</td>
<td>Required</td>
<td>Browser provided: 4</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Requestor</td>
<td>Optional</td>
<td></td>
<td>Required</td>
<td></td>
</tr>
<tr>
<td>SerialNumber</td>
<td></td>
<td></td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>SignWith</td>
<td>Constant: SAF:CERAUTH/taca</td>
<td>Constant: PKI:</td>
<td></td>
<td>Constant: PKI:</td>
</tr>
</tbody>
</table>

Customizing the end-user Web pages using REXX CGIs
Table 34. Summary of fields for SAF, SCEP, and PKI generated key certificate templates (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>One-year SAF server</th>
<th>One-year SAF browser</th>
<th>Five-year SCEP preregistration</th>
<th>One-year PKI generated key</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateProv</td>
<td>Optional</td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td></td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnstructAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnstructName</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserId</td>
<td>Application provides</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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”
- “Examining the TEMPLATE section” on page 130
- “Examining the INSERT section” on page 133

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 128

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.)

```plaintext
# =====================================================================
# Application - PKISERV
# The installation should customize the CONTENT, ADMINHEADER
# ADMINFOOTER, and ADMINSCOPE subsections as appropriate
```

---

Customizing the end-user Web pages using REXX CGIs

---

126  z/OS V1R11.0 Cryptographic Services PKI Services Guide and Reference
Customizing the end-user Web pages using REXX CGIs

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 300 for a sample of that Web page.)
Customizing the end-user Web pages using REXX CGIs

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 304 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 275 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 204 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.
#
# ==cue
#
<APPLICATION NAME=CUSTOMERS> 1
<CONTENT> 2
<HTML><HEAD>
<TITLE> Customers Certificate Generation Application </TITLE>
%%-copyright%%
</HEAD>
<TITLE> Customers Certificate Generation Application </TITLE>
</CONTENT>
<RECONTENT>
<HTML><HEAD>
<TITLE> Customers Renew or Revoke a Browser Certificate </TITLE>
</HEAD>
<RECONTENT>
```

```
Customizing the end-user Web pages using REXX CGIs

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 275 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 Request certificate button.
5. 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.)
Customizing the end-user Web pages using REXX CGIs

6. 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”. (See Figure 49 on page 291 for a sample of that Web page.)

7. 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 284 for a sample of that Web page.)

8. 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”.

9. The FINDRECOVERCONTENT subsection displays security questions for users to answer when they want to recover a certificate and have forgotten the passphrase.

10. The RETRIEVECONTENT2 subsection contains the HTML to allow the end user to retrieve a recovered 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
#
# 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>
<NICKNAME=2YBZOS>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
%%-AdditionalHead[browsertype]%%
</HEAD>

<BODY>
<H1>2-Year PKI Browser Certificate For Authenticating To z/OS</H1>
<p>
<H2>Choose one of the following:</H2>
...
</p>
</BODY>
```

130 z/OS V1R11.0 Cryptographic Services PKI Services Guide and Reference
Customizing the end-user Web pages using REXX CGIs

Enter values for the following field(s)

Requestor (optional)
NotifyEmail (optional)
PassPhrase
PublicKey2[browsertype]

Pick up a previously issued certificate

Pick up a previously issued certificate

Please bookmark this page

Retrieve Your [tmplname]
Customizing the end-user Web pages using REXX CGIs

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 133. See list item 1 on page 133)

14. The FAILURECONTENT subsection contains the HTML to display when the certificate request is unsuccessful. This subsection contains the
Customizing the end-user Web pages using REXX CGIs

The -requestok INSERT 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 135].)

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:12769Be4-e730-4e5c-a2b1-21490a70c8a1"
   CODEBASE="xenroll.cab#Version=5,131,3659,0"
   id="certmgr"
>
</OBJECT>
<OBJECT
   classid="clsid:8B4e2049-217d-11da-b2a4-000e7b8b2b09"
   id="g_objWCF"
>
</OBJECT>
</INSERT>

<INSERT NAME=-requestok>
<html>
<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.
<form METHOD="GET" ACTION="/[application]/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>

<INSERT NAME=-requestbad>
<html>
<head>
<title>Web Based Certificate Generation Failure</title>
</head>
<body>
</html>
</INSERT>
```
Customizing the end-user Web pages using REXX CGIs

```html
</HEAD>
<BODY>
<H1> Request was not successful</H1>
<p> Please correct the problem or report the error to your Web admin person</p>
<pre>
[errorinfo]
</pre>
</BODY>
</HTML>

# X.509 fields (INSERTs) valid for certificate requests

```vbscript
function ValidPassPhrase(frm){
    if ("[optfield]" == "") {
        alert("Enter required field."); frm.PassPhrase.focus();
        return false;
    }
    if ("[optfield]" == "") {
        alert("Reenter the pass phrase."); frm.ConfirmPassPhrase.focus();
        return false;
    }
}
```
Customizing the end-user Web pages using REXX CGIs

if (frm.PassPhrase.value != frm.ConfirmPassPhrase.value) {
    alert("Passwords don't match. Reenter."); frm.PassPhrase.focus();
    return false;
} else {
    return true;
}
//-->
</SCRIPT>
</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 133]) 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 133])

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 [3])

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 following are the CGIs for the end-user Web pages (including their directories):

- /usr/lpp/pkiserv/PKIServ/public-cgi/camain.rexx
- /usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/catmpl.rexx
- /usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/careq.rexx
- /usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/caretrieve.rexx
- /usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx

Chapter 11. Customizing the end-user Web application if you use REXX CGI execs 135
Customizing the end-user Web pages using REXX CGIs

- /usr/lpp/pkiserv/ssl-cgi-bin/cagorcvr.rexx
- /usr/lpp/pkiserv/ssl-cgi-bin/cadisplay.rexx
- /usr/lpp/pkiserv/ssl-cgi-bin/camodify.rexx

The following table summarizes the actions the CGIs perform:

<table>
<thead>
<tr>
<th>CGI exec</th>
<th>Action</th>
<th>Sample Web page</th>
</tr>
</thead>
<tbody>
<tr>
<td>camain.rexx</td>
<td></td>
<td>See Figure 36 on page 275</td>
</tr>
<tr>
<td></td>
<td>- When user clicks the Request certificate button, this calls catmpl.rexx, passing it a parameter identifying the selected template.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- The user can click the Pick up certificate button to go directly to caretrieve.rexx (if the certificate is already requested).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- The user can click the Renew or revoke certificate button to go to cadisplay.rexx.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- An administrator can click the Go to administration page button to go to admmain.rexx. (See Table 37 on page 152 for more information about admmain.rexx.)</td>
<td></td>
</tr>
<tr>
<td>catmpl.rexx</td>
<td></td>
<td>See Figure 38 on page 282</td>
</tr>
<tr>
<td></td>
<td>- Displays Web page coded in the HTML under the CONTENT subsection (of a TEMPLATE section).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- When the user clicks the Submit certificate request button, this passes template and field name parameters to careq.rexx.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- When the user clicks the Retrieve your certificate button, this passes control to caretrieve.rexx.</td>
<td></td>
</tr>
<tr>
<td>careq.rexx</td>
<td></td>
<td>See Figure 40 on page 284</td>
</tr>
<tr>
<td></td>
<td>- Processes field names under the APPL subsection (of a TEMPLATE section).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Note: Depending on the template, this can be:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Userid only</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Userid and HostIdMap.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Processes hardcoded field names under the CONSTANT subsection (of a TEMPLATE section).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Depending on the results, displays Web page coded in the HTML under the SUCCESSCONTENT or FAILURECONTENT subsection (of a TEMPLATE section):</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- The SUCCESSCONTENT subsection includes a Continue button the user can click to continue to caretrieve.rexx.</td>
<td></td>
</tr>
<tr>
<td>caretrieve.rexx</td>
<td></td>
<td>See Figure 41 on page 285</td>
</tr>
<tr>
<td></td>
<td>- 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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- When the user clicks the Retrieve and install certificate button, this passes the transaction ID parameter to cagetcert.rexx.</td>
<td></td>
</tr>
<tr>
<td>cagetcert.rexx</td>
<td></td>
<td>See Figure 43 on page 286</td>
</tr>
<tr>
<td></td>
<td>- Displays Web page coded in the HTML under the RETURNCERT subsection (of a TEMPLATE section). This HTML determines which of the following forms to use when returning the certificate:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- as a base64-encoded certificate (for server certificates)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- as an ActiveX object (for Microsoft Internet Explorer browser certificates)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- as an application/x-x509-user-certificate MIME type (for Netscape browser certificates).</td>
<td></td>
</tr>
</tbody>
</table>
Customizing the end-user Web pages using REXX CGIs

Table 35. CGI actions for end-user Web pages (continued)

<table>
<thead>
<tr>
<th>CGI exec</th>
<th>Action</th>
<th>Sample Web page</th>
</tr>
</thead>
<tbody>
<tr>
<td>cagorcvr.rexx</td>
<td>Displays Web page coded in HTML under the FINDRECOVERCONTENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subsection of the APPLICATION section. This subsection displays security</td>
<td></td>
</tr>
<tr>
<td></td>
<td>questions for users to answer. The answers to these questions can be</td>
<td></td>
</tr>
<tr>
<td></td>
<td>used to recover a passphrase, which is used to recover a certificate.</td>
<td></td>
</tr>
<tr>
<td>cadisplay.rexx</td>
<td>Displays Web page coded in the HTML under the RECONTENT</td>
<td>See Figure 49 on</td>
</tr>
<tr>
<td></td>
<td>subsection (of the APPLICATION section).</td>
<td>page 291</td>
</tr>
<tr>
<td></td>
<td>For renewing a certificate, the user fills in the passphrase and clicks the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Renew button. For revoking a certificate, the user clicks the Revoke</td>
<td></td>
</tr>
<tr>
<td></td>
<td>button. Both actions call camodify.rexx.</td>
<td></td>
</tr>
<tr>
<td>camodify.rexx</td>
<td>Displays Web page coded in the HTML under the SUCCESSCONTENT</td>
<td>See Figure 40 on</td>
</tr>
<tr>
<td></td>
<td>subsection (of a TEMPLATE section) for a successful renewal. The</td>
<td>page 284</td>
</tr>
<tr>
<td></td>
<td>SUCCESSCONTENT subsection includes a Continue button the user can</td>
<td></td>
</tr>
<tr>
<td></td>
<td>click to call caretrieve.rexx.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Displays the Web page coded in HTML under the RESUCCESSCONTENT subsection (of the APPLICATION section) for a successful revocation.</td>
<td></td>
</tr>
</tbody>
</table>

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 pkiservtmpl 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:
Customizing the end-user Web pages using REXX CGIs

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 342.

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.

```xml
<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:**

```xml
<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:**

```xml
<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:**

```javascript
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:**

```xml
<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:**
Customizing the end-user Web pages using REXX CGIs

<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:**
- 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).
- 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 page 118 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>
<HTML><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>
</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>
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Failure</TITLE>
</HEAD>
<BODY>
<H1> Request was not successful</H1>
</P>
<pre>
[errorinfo]
</pre>
</BODY>
</HTML>
</INSERT>
```

9. If necessary, update the RETRIEVECONTENT subsection.

**Note:** See “Steps for changing the runtime user ID for retrieving certificates” on page 146 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 137.)
Customizing the end-user Web pages using REXX CGIs

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 pkiservtmpl 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/pkiservtmpl) and your existing PKI Services certificates template file (/etc/pkiserv/pkiservtmpl).

Perform the following steps to retrofit changes into the pkiservtmpl 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/pkiservtmpl /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/pkiservtmpl /etc/pkiserv/pkiservtmpl
   ```

3. Using a compare program of your choice, compare the two template files:
   ```
   /etc/pkiserv/pkiservtmpl
   /etc/pkiserv/pkiserv.backup
   ```

4. Edit the runtime copy of the templates file (/etc/pkiserv/pkiservtmpl). 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.
Customizing the end-user Web pages using REXX CGIs

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 36. Location of code for various Web pages

<table>
<thead>
<tr>
<th>Main header (and sample Web page if any)</th>
<th>Location of code in pkiserv.tmpl certificate templates file</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1-Year PKI S/MIME Browser Certificate”</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“1-Year PKI SSL Browser Certificate” (See Figure 38 on page 282)</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“2-Year PKI Browser Certificate For Authenticating To z/OS”</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“2-Year PKI Authenticode - Code Signing Certificate”</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“5-Year PKI Intermediate CA Certificate”</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“5-Year PKI IPSEC Server (Firewall) Certificate”</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“5-Year PKI SSL Server Certificate”</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“5-Year SCEP certificate - Preregistration”</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“n-Year PKI Certificate for Extensions Demonstration”</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>“Here’s Your Certificate. Cut and Paste it to a File”</td>
<td>INSERT section, -return10cert INSERT. (This is referenced in the RETURNCERT subsection of the TEMPLATE section of each certificate template.)</td>
</tr>
<tr>
<td>“Internet Explorer Certificate Install” (See Figure 43 on page 286)</td>
<td>INSERT section, returnbrowsercertIE INSERT</td>
</tr>
<tr>
<td>“Preregistration successful”</td>
<td>INSERT section, -preregok INSERT. (This is referenced in the SUCCESSCONTENT subsection of the TEMPLATE section of each certificate template.)</td>
</tr>
<tr>
<td>“PKI Services Certificate Generation Application” (See Figure 36 on page 273)</td>
<td>APPLICATION section, CONTENT subsection</td>
</tr>
<tr>
<td>“Renew or Revoke a Browser Certificate” (See Figure 49 on page 291)</td>
<td>APPLICATION section, RECONTENT subsection</td>
</tr>
</tbody>
</table>
| “Request submitted successfully” (For submitting a successful certificate request or renewal, see Figure 40 on page 284) | - 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.) |
Customizing the end-user Web pages using REXX CGIs

Table 36. Location of code for various Web pages (continued)

<table>
<thead>
<tr>
<th>Main header (and sample Web page if any)</th>
<th>Location of code in pkiserv.tmpl certificate templates file</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Request was not successful”</td>
<td>- For an unsuccessful certificate request:</td>
</tr>
<tr>
<td></td>
<td>INSERT section, -requestbad INSERT. (This is referenced in</td>
</tr>
<tr>
<td></td>
<td>the FAILURECONTENT subsection of the</td>
</tr>
<tr>
<td></td>
<td>TEMPLATE section of each certificate template.)</td>
</tr>
<tr>
<td></td>
<td>- For an unsuccessful certificate revocation</td>
</tr>
<tr>
<td></td>
<td>request: INSERT section, -renewrevokebad INSERT. (This is</td>
</tr>
<tr>
<td></td>
<td>referenced in the</td>
</tr>
<tr>
<td></td>
<td>REFAILURECONTENT subsection of the APPLICATION section.)</td>
</tr>
<tr>
<td>“Retrieve Your 1-Year PKI S/MIME Browser</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>Certificate”</td>
<td></td>
</tr>
<tr>
<td>“Retrieve Your 1-Year PKI SSL Browser</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>Certificate” (See <a href="#">Figure 41 on page 285</a>)</td>
<td></td>
</tr>
<tr>
<td>“Retrieve Your 2-Year PKI Browser</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>Certificate For Authenticating To z/OS”</td>
<td></td>
</tr>
<tr>
<td>“Retrieve Your 2-Year PKI Authenticode -</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>Code Signing Certificate”</td>
<td></td>
</tr>
<tr>
<td>“Retrieve Your 5-Year PKI Intermediate CA</td>
<td>TEMPLATE section RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>Certificate”</td>
<td></td>
</tr>
<tr>
<td>“Retrieve Your 5-Year PKI IPSEC Server</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>(Firewall) Certificate”</td>
<td></td>
</tr>
<tr>
<td>“Retrieve Your 5-Year PKI SSL Server</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>Certificate”</td>
<td></td>
</tr>
<tr>
<td>“Retrieve Your [tmplname]”</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>“Retrieve Your SAF Browser Certificate 1-Year”</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>“Retrieve Your SAF Server Certificate 1-Year”</td>
<td>TEMPLATE section, RETRIEVECONTENT subsection</td>
</tr>
<tr>
<td>“SAF Browser Certificate 1-Year (Auto</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>Approved)”</td>
<td></td>
</tr>
<tr>
<td>“SAF Server Certificate 1-Year (Auto</td>
<td>TEMPLATE section, CONTENT subsection</td>
</tr>
<tr>
<td>Approved)”</td>
<td></td>
</tr>
</tbody>
</table>

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 27 on page 105 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.
### Customizing the end-user Web pages using REXX CGIs

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 139.

---

### 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 19, “RACF administration for PKI Services,” on page 341 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
Customizing the end-user Web pages using REXX CGIs

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:**
   ```
Customizing the end-user Web pages using REXX CGIs

<H1> Retrieve Your [tmplname] </H1>

<H3>Please bookmark this page</H3>

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=retrievform 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=retrievform METHOD=POST ACTION="/PKIServ/ssl-cgi-bin/surrogatethough/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
#<FORM NAME=retrievform 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 pkiservtmpl 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.
Customizing the end-user Web pages using REXX CGIs

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 yyyymmd 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 151. 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 151. The following steps refer to the marked lines.

Perform the following steps to customize the AltOther_</ OID> INSERT using Figure 3 on page 151 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.
Customizing the end-user Web pages using REXX CGIs

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.
Customizing the end-user Web pages using REXX CGIs
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 157.

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.
Customizing the administration Web pages using REXX CGIs

Table 37 (which lists the CGI execs in logical order) summarizes the actions they perform:

Table 37. CGI actions for administrative Web pages

<table>
<thead>
<tr>
<th>CGI exec</th>
<th>Action</th>
<th>Sample Web page</th>
</tr>
</thead>
<tbody>
<tr>
<td>admmain.rexx</td>
<td>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 304.</td>
<td></td>
</tr>
<tr>
<td>admpend.rexx</td>
<td>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 309.</td>
<td></td>
</tr>
<tr>
<td>admpendtid.rexx</td>
<td>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 305.</td>
<td></td>
</tr>
<tr>
<td>admmodtid.rexx</td>
<td>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 307.</td>
<td></td>
</tr>
<tr>
<td>admicl.rexx</td>
<td>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 309.</td>
<td></td>
</tr>
<tr>
<td>admiclcert.rexx</td>
<td>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 315.</td>
<td></td>
</tr>
<tr>
<td>admacttid.rexx</td>
<td>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 306.</td>
<td></td>
</tr>
<tr>
<td>admacttid2.rexx</td>
<td>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 306.</td>
<td></td>
</tr>
</tbody>
</table>
Customizing the administration Web pages using REXX CGIs

Table 37. CGI actions for administrative Web pages (continued)

<table>
<thead>
<tr>
<th>CGI exec</th>
<th>Action</th>
<th>Sample Web page</th>
</tr>
</thead>
</table>
| 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 311  
  • For an example of the “Processing was not successful” Web page, see Figure 68 on page 312  
  • For an example of the “Processing partially successful” Web page, see Figure 69 on page 312 |

| 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” | — |

| admicall.rexx | After the administrator searches for certificates and admicall.rexx displays the results, the administrator clicks a button to revoke or delete selected certificates. This calls admicall.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.
Customizing the administration Web pages using REXX CGIs

**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 204.

### 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:**
   ```html
   <ADMINHEADER>
   <HTML><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:**
   ```html
   <ADMINFOOTER>
   <p>email: webmaster@company.com</p>
   </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 301 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 19, “RACF administration for PKI Services,” on page 341.)
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 101 and Chapter 12, “Customizing the administration Web pages if you use REXX CGI execs,” on page 151, 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 336) 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 336. 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.

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.
- 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.

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:
- AltOther
- DomainName
- DNQualifier
- Uid
- KeyUsage
- ExtKeyUsage
- NotBefore
- NotAfter
- Country
- Org
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 Validformfieldname 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 38.

Table 38. Task roadmap for implementing the PKI Services Web application using the JSPs

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Associated instructions (see . . .)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td>&quot;Steps for preparing to implement the PKI Services Web application using JSPs&quot;</td>
</tr>
<tr>
<td>Give Websphere users authorization to use PKI Services functions</td>
<td>&quot;Giving Websphere users authorization to use PKI Services functions&quot; on page 163</td>
</tr>
<tr>
<td>Set up a Websphere SSL configuration that uses client authentication, to allow Websphere users to renew and revoke browser certificates</td>
<td>&quot;Allowing Websphere users to renew and revoke browser certificates&quot; on page 166</td>
</tr>
<tr>
<td>Customize the Web application</td>
<td>&quot;Customizing the PKI Services Web application&quot; on page 176</td>
</tr>
<tr>
<td>• Update the template file.</td>
<td>&quot;Updating the template file&quot; on page 176</td>
</tr>
<tr>
<td>• (Optional) Modify the JSP files and update the EAR file with the modified JSP files.</td>
<td>&quot;(Optional) Modifying the JSP files and the EAR file&quot; on page 176</td>
</tr>
<tr>
<td>• Deploy the EAR file to a Websphere application server.</td>
<td>&quot;Deploying the EAR file to a Websphere application server&quot; on page 178</td>
</tr>
</tbody>
</table>

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. If it is not, select it and click Apply.

   b. Click External authorization providers, and verify that System Authorization Facility (SAF) authorization is selected, as shown in Figure 6 on page 164. If it is not, click System Authorization Facility (SAF) authorization, and then click Apply.
c. From the web page shown in Figure 6, 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 179 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 16
on page 37

Note: If your installation uses security domains the role name is qualified by
the security domain, for example:

RDEFINE EJBROLE securitydomain.SAUser 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 surrogate user to use the R_PKIServ functions
REQCERT, GENCERT, ADD, VERIFY, REVOKE, RESPOND, SCEPREQ,
REORENEW, 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 16 on page 37)

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. 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)`

   d. 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)`

   e. Refresh the in-storage profiles so that the changes you have made to the FACILITY class take effect:

   `SETR RACLST(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](image-url)
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 on page 168](#).
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 169.
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**.
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 168) The page shown in Figure 11 is displayed.

![Image of SSL configuration page](image)

**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 171 is displayed.
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 172**.

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 203 except in steps 3 and 4 you need to find the section that looks like:

```xml
<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.
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 13. Defining a new port**

Click **OK** and then **Save** to save this new port number.

**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 174.
b. Click **Web container transport chains**, and then click **New**. You should see a page like Figure 17.

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 336.

(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 182.

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.

```jsp
    <SELECT NAME="notbefore">
        <OPTION VALUE="0">
            <% if (initvalue_str.equalsIgnoreCase("0")) out.print(" SELECTED "); %> 0
        </OPTION>
        <OPTION VALUE="30">
            <% if (initvalue_str.equalsIgnoreCase("30")) out.print(" SELECTED "); %>
            30
        </OPTION>
    </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 177.
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
   ```

---

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
   ```

Chapter 13. Implementing the Web application using Java server pages 177
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:

   ```sh
   iconv -f iso8859-1 -t ibm-1047 footer.jsp > $HOME/footer.jsp.edit
   ```

7. Use **oedit** to edit the `.edit` file that you created:

   ```sh
   oedit $HOME/footer.jsp.edit
   ```

8. Use **iconv** to converted the edited file back to ASCII. For example:

   ```sh
   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:

   ```sh
   cd ..
   jar -uvf PKIServ_Web.war mod_inc/footer.jsp
   ```

10. Update the EAR file with the updated WAR file:

    ```sh
    jar -uvf PKIServ.ear PKIServ_Web.war
    ```

11. Make sure the PKIServ.ear file is publicly readable by issuing the **chmod** command.

    ```sh
    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/](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 179 opens.
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 180](#) opens.
3. Download the PKI Services EAR file and upload it to the z/OS file system.
   a. Click **Remote file system**, and click **Browse**.
   b. Click the icon for the z/OS system. The root directory is displayed. Continue selecting until you reach:
      ```bash
      /usr/lpp/pkiserv/pkijsp/PKIServ.ear
      ```
   c. Click **OK** and **Next**. Continue clicking **OK** and **Next** until you see a message saying that the EAR was installed and the message shown in Figure 23 on page 179 indicating that changes have been made to your local configuration.
   d. 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 181 opens.
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**
   b. Click **Shared Library References**. The window shown in Figure 26 opens.
   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. Again you see the message shown in Figure 23 on page 179 indicating that changes have been made to your local configuration. Click **Save** to save the changes to the master configuration.
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](image)

b. Again you see the message shown in Figure 23 on page 179 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 39 on page 183 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 40 on page 184 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, pkiservtmpl, 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 41 on page 185 lists the JSP files in this directory. “Customizing the PKI Services Web application” on page 176 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 203.)

Table 39. JSP files in the Customers directory. These files perform end-user functions.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pkimain.jsp</td>
<td>Starting point for all tasks available to the end user, such as requesting 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.</td>
</tr>
<tr>
<td>certrequest.jsp</td>
<td>Web page to request a certificate.</td>
</tr>
<tr>
<td>certretrieve.jsp</td>
<td>Web page to retrieve a certificate.</td>
</tr>
<tr>
<td>genericbad.jsp</td>
<td>Displays error information when processing fails.</td>
</tr>
<tr>
<td>genericok.jsp</td>
<td>Notifies the user of successful processing when no additional information (such as a transaction ID) needs to be returned to the user.</td>
</tr>
<tr>
<td>keygenrequestok.jsp</td>
<td>Notifies the user of a successful certificate request with PKI-generated keys, and includes the e-mail address where notification has been sent.</td>
</tr>
<tr>
<td>pkcs10retrieved.jsp</td>
<td>Returns a retrieved PKCS #10 certificate.</td>
</tr>
<tr>
<td>recoverbad.jsp</td>
<td>Displays information for an unsuccessful attempt to find certificates for which the private key is to be recovered.</td>
</tr>
<tr>
<td>recoverok.jsp</td>
<td>Displays results of a successful attempt to find certificates for which the private key is to be recovered.</td>
</tr>
</tbody>
</table>
Table 39. JSP files in the Customers directory (continued). These files perform end-user functions.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>renew_revoke.jsp</td>
<td>Displays a client-authenticated browser certificate for the user to renew, revoke or suspend.</td>
</tr>
<tr>
<td>requestok.jsp</td>
<td>Returns results (such as the transaction ID) of a successful certificate request.</td>
</tr>
<tr>
<td>requestbad.jsp</td>
<td>Returns error information for an unsuccessful certificate request.</td>
</tr>
<tr>
<td>retrievebad.jsp</td>
<td>Returns error information for an unsuccessful attempt to retrieve a certificate.</td>
</tr>
</tbody>
</table>

Table 40. JSP files in the PKIServ directory. These files perform administrator functions.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>actcertok.jsp</td>
<td>Returns results of a successful certificate action, such as deleting, revoking, suspending, or resuming a certificate.</td>
</tr>
<tr>
<td>actcertbad.jsp</td>
<td>Returns error information on an unsuccessful certificate action, such as deleting, revoking, suspending, or resuming a certificate.</td>
</tr>
<tr>
<td>actrequestok.jsp</td>
<td>Returns results of a successful action on a certificate request, such as approving, rejecting or deleting.</td>
</tr>
<tr>
<td>acttrequestbad.jsp</td>
<td>Returns error information on an unsuccessful action on a certificate request, such as approving, rejecting or deleting.</td>
</tr>
<tr>
<td>adminmain.jsp</td>
<td>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.</td>
</tr>
<tr>
<td>admhome.jsp</td>
<td>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 (adminmain.jsp).</td>
</tr>
<tr>
<td>approvewithmods.jsp</td>
<td>Displays the form for an administrator to approve a certificate request with modifications.</td>
</tr>
<tr>
<td>certdetailsbad.jsp</td>
<td>Displays error information for a failed attempt to display the details of a certificate.</td>
</tr>
<tr>
<td>certdetailsok.jsp</td>
<td>Displays details of a single certificate.</td>
</tr>
<tr>
<td>genericfailure.jsp</td>
<td>Displays error information when the requested administration processing fails.</td>
</tr>
<tr>
<td>modifybad.jsp</td>
<td>Displays error information when an approve with modifications action fails.</td>
</tr>
<tr>
<td>modifyok.jsp</td>
<td>Displays results of successful approve with modification action.</td>
</tr>
<tr>
<td>querycertok.jsp</td>
<td>Displays search results when administrator queries certificates.</td>
</tr>
</tbody>
</table>
Table 40. JSP files in the PKIServ directory (continued). These files perform administrator functions.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryreqok.jsp</td>
<td>Displays search results when administrator queries certificate requests.</td>
</tr>
<tr>
<td>reqdetailsok.jsp</td>
<td>Displays details of a single certificate request.</td>
</tr>
</tbody>
</table>

Table 41. JSP files in the mod.inc directory. These files are modifiable include files.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adminbottomnav.jsp</td>
<td>Bottom of page navigation appearing on administrator’s Web pages</td>
</tr>
<tr>
<td>adminfooter.jsp</td>
<td>Bottom footer (the last element) on administrator’s Web pages</td>
</tr>
<tr>
<td>altdomain.jsp</td>
<td>Form field for altdomain</td>
</tr>
<tr>
<td>altemail.jsp</td>
<td>Form field for altemail</td>
</tr>
<tr>
<td>altipaddr.jsp</td>
<td>Form field for altipaddr</td>
</tr>
<tr>
<td>altother_1_2_3_4_5.jsp</td>
<td>altother form field sample for OID 1.2.3.4.5</td>
</tr>
<tr>
<td>altother_1_2_3_4_6.jsp</td>
<td>altother form field sample for OID 1.2.3.4.6</td>
</tr>
<tr>
<td>altother_1_3_6_1_4_1_311_20_2_3.jsp</td>
<td>altother form field sample for OID 1.3.6.1.4.1.311.20.2.3</td>
</tr>
<tr>
<td>alturi.jsp</td>
<td>Form field for alturi</td>
</tr>
<tr>
<td>autorenew.jsp</td>
<td>Form field for autorenew</td>
</tr>
<tr>
<td>bottomnav.jsp</td>
<td>Bottom of page navigation on end user Web pages</td>
</tr>
<tr>
<td>certrow.jsp</td>
<td>Displays a row in a table about certificates, format tied to PKIServ/certdetailsok.jsp</td>
</tr>
<tr>
<td>challengepassphrase.jsp</td>
<td>Displays passphrase form field when user is prompted to match the passphrase previously entered</td>
</tr>
<tr>
<td>clientname.jsp</td>
<td>Form field for clientname</td>
</tr>
<tr>
<td>commonfunctions.jsp</td>
<td>Contains javascript functions that are commonly used, such as trim()</td>
</tr>
<tr>
<td>commonname.jsp</td>
<td>Form field for commonname</td>
</tr>
<tr>
<td>country.jsp</td>
<td>Form field for country</td>
</tr>
<tr>
<td>date.jsp</td>
<td>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</td>
</tr>
<tr>
<td>dnqualifier.jsp</td>
<td>Form field for distinguished name qualifier</td>
</tr>
<tr>
<td>domainname.jsp</td>
<td>Form field for domain name</td>
</tr>
<tr>
<td>emailaddr.jsp</td>
<td>Form field for emailaddr</td>
</tr>
<tr>
<td>extkeyusage.jsp</td>
<td>Form field for extended key usage</td>
</tr>
<tr>
<td>footer.jsp</td>
<td>Footer (last element) on end users pages</td>
</tr>
<tr>
<td>hostidmap.jsp</td>
<td>Form field for host id mapping</td>
</tr>
<tr>
<td>keysize.jsp</td>
<td>Form field for keysize</td>
</tr>
</tbody>
</table>
Table 41. JSP files in the `mod.inc` directory (continued). These files are modifiable include files.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyusage.jsp</td>
<td>Form field for key usage</td>
</tr>
<tr>
<td>label.jsp</td>
<td>Form field for label</td>
</tr>
<tr>
<td>locality.jsp</td>
<td>Form field for locality</td>
</tr>
<tr>
<td>mail.jsp</td>
<td>Form field for mail</td>
</tr>
<tr>
<td>notafter.jsp</td>
<td>Form field for the number of days the certificate should be valid (from the time of request)</td>
</tr>
<tr>
<td>notbefore.jsp</td>
<td>Form field for the number of days before the certificate should be valid</td>
</tr>
<tr>
<td>notifyemail.jsp</td>
<td>Form field for notifyemail</td>
</tr>
<tr>
<td>org.jsp</td>
<td>Form field for org</td>
</tr>
<tr>
<td>orgunit.jsp</td>
<td>Form field for orgunit</td>
</tr>
<tr>
<td>passphrase.jsp</td>
<td>Form field for user to enter passphrase</td>
</tr>
<tr>
<td>postalcode.jsp</td>
<td>Form field for postal code</td>
</tr>
<tr>
<td>publickey.jsp</td>
<td>“Public key” form field, text box for entering base64-encoded PKCS #10 certificate request</td>
</tr>
<tr>
<td>publickey_smartcard.jsp</td>
<td>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</td>
</tr>
<tr>
<td>recoveremail.jsp</td>
<td>recoveremail form field</td>
</tr>
<tr>
<td>requestor.jsp</td>
<td>Requestor form field (with prompt for name for tracking this request)</td>
</tr>
<tr>
<td>requestor2.jsp</td>
<td>Requestor form field (with prompt for e-mail address of requestor)</td>
</tr>
<tr>
<td>security.jsp</td>
<td>Displays a form field for entering a response to a security question</td>
</tr>
<tr>
<td>security2.jsp</td>
<td>Displays a form field for entering a response to a second security question</td>
</tr>
<tr>
<td>serialnumber.jsp</td>
<td>Form field for entering a certificate's serial number</td>
</tr>
<tr>
<td>signwith.jsp</td>
<td>Form field for selecting the signer of the requested certificate</td>
</tr>
<tr>
<td>stateprov.jsp</td>
<td>Form field for stateprov</td>
</tr>
<tr>
<td>street.jsp</td>
<td>Form field for street</td>
</tr>
<tr>
<td>title.jsp</td>
<td>Form field for title</td>
</tr>
<tr>
<td>transactionid.jsp</td>
<td>Form field for entering a certificate request's transaction ID</td>
</tr>
<tr>
<td>uid.jsp</td>
<td>Form field for uid</td>
</tr>
<tr>
<td>unstructaddr.jsp</td>
<td>Form field for unstructaddr</td>
</tr>
<tr>
<td>unstructname.jsp</td>
<td>Form field for unstructname</td>
</tr>
<tr>
<td>userid.jsp</td>
<td>Form field for userid</td>
</tr>
</tbody>
</table>
Chapter 14. Advanced customization

This topic describes the advanced customization procedures available for PKI Services. All are optional.

- “Scaling for high volume installations” on page 188
- “Using certificate policies” on page 191
- “Updating the signature algorithm” on page 192
- “Customizing distribution point CRLs” on page 192
- “Creating a distribution point ARL” on page 199
- “Adding an application domain” on page 200
- “Adding a new CA domain” on page 204
- “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 219
- “Setting up automatic renewal of certificates” on page 230
- Chapter 15, “Customizing with installation exits,” on page 235

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 192.
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 exits,” on page 235.) 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.
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](https://www.rfc-editor.org/rfc/rfc2459).)

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 189.

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 190.

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
Advanced customization

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:

   Example:

   [CertPolicy]
   PolicyName1=MyPolicy

6. If you want to add qualifiers, perform the following steps:

   a. Update the Policy1Org and Policy1Notice fields in the following example:

   Policy1Org=My Company, Inc
   Policy1Notice=1

   Policy1Org Your organization’s name, for example, International Business Machines, Inc.
**Advanced customization**

**Policy1Notice1 through Policy1Noticen**
Your notice numbers. (You might need more than one Policy1Notice 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.ibm.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 189 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:
If you want to make the CertPolicies extension critical, specify the following in the CONSTANT section:

```
%Critical=CertPolicies
```

**Rule:** The policy numbers in the template file must exist in the pkiserv.conf 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. By default, IKEYSETUP creates the CA certificate with an RSA key pair. The default value of the signature algorithm in the pkiserv.conf file is sha-1WithRSAEncryption. You can change the signature algorithm by changing the SigAlg1 value in the CertPolicy section of the pkiserv.conf configuration file.

If the 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)
- md-5WithRSAEncryption (OID value 1.2.840.113549.1.1.4)
- md-2WithRSAEncryption (OID value 1.2.840.113549.1.1.2)

If the certificate key type is DSA, specify the SigAlg1 algorithm value as follows:

- id-dsa-with-sha1 (OID value 1.2.840.10040.4.3)

**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.
Advanced customization

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 CertPolicy section.

2. If you want to use a signature algorithm other than SHA1, change sha-1WithRSAEncryption in the following line to the value for the signature algorithm that you want to use:
   
   SigAlg1=sha-1WithRSAEncryption

   • For SHA256, the value is sha-256WithRSAEncryption.
   • For MD5, the value is md-5WithRSAEncryption.
   • For MD2, the value is md-2WithRSAEncryption.
   • For DSA, the value is id-dsa-with-sha1.

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.
Advanced customization

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**
  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 199 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 195 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)](https://tools.ietf.org/html/rfc1738).
Determining CRLDistURI

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” on page 193.

This is an optional parameter. If you do not specify a CRLDistURI value, the URI format name will not be created. You can specify multiple entries for the CRLDistURI 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 with an n value of 0.

There are different ways to specify the value of CRLDistURI 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 LdapServern, where n is greater than zero.

Restriction: PKI Services provides syntax checking based only on valid values for the CRLDistURI 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 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” on page 195 for additional information.

Specifying an LDAP URI

For LDAP, there are two ways to indicate the CRLDistURI value. Choose either of the following two methods:

- Specify the protocol and the domain name (and the port, if needed). The value for CRLDistURI can be specified with or without a trailing slash.
  
  Example:
  
  CRLDistURI1=ldap://ldap.bankxyz.com:389/

- Specify the keyword LdapServern to have PKI Services build the CRLDistURI value for you based on a server identified by the Servern or BindProfilen 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 LdapServern 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 LdapServern keyword must be identified in one of the following ways:
   - The corresponding LDAP server must be identified by a Servern or BindProfilen 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 361.

Determining CRLDistDirPath
If the protocol for the URI you specified with CRLDistURI 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 193.
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: Before executing this procedure, be aware of the following restrictions:
• If running PKI Services in a sysplex, all instances of PKI Services must specify the same values for each of the above parameters.
• 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:

\[
\text{CRLDistSize} = \frac{E}{P}
\]

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:

\[
\text{CRLDistSize} = \frac{100}{0.10} = 1000
\]

The \( \text{CRLDistSize} \) in bytes can be roughly estimated to be 500 + (25 \times \text{number of CRL entries}). Using the example above, the average CRL size in bytes would be 500 + (25 \times 100) = 3000 bytes.

**Restriction:** A single CRL can not exceed 32K bytes in length. Therefore, you must limit its length. In addition, the longer the CRL, the longer it will take to process it.

**Rules:**
- a. The value of \( \text{CRLDistSize} \) 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.
   
   If you selected the 0 value for CRLDistSize, complete this step and then continue with Step 12.

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 198.)

   **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` 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 193.

You can specify multiple entries for the `CRLDistURI` parameter, using the parameters `CRLDistURI1`, `CRLDistURI2`, and so forth.

6. If necessary, update the value of `CRLDistURI` in the `CertPolicy` section of `pkiserv.conf` to your customized value or values.

7. If the protocol for the URI you specified with `CRLDistURI` in Step 5 is LDAP protocol, skip to Step 12.

8. If the protocol for the URI you specified with `CRLDistURI` in Step 5 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 195.

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 199.

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 `CRLDistURI`, 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.
Advanced customization

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 199.) 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.)
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 18 on page 52 and when the certificate contains the necessary OCSP responder information in the AuthInfoAccess extension. (See Template sections on page 114.)

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 200 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.
Advanced customization

```
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'
                }
              }
            }
          }
          [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 204.
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 42. Both are tasks for the Web server programmer.

Table 42. Task roadmap for creating multiple application domains

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Associated procedures (See ...)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update the PKI Services template file</td>
<td>“Steps for creating multiple application sections in the PKI Services template file”</td>
</tr>
<tr>
<td>Update the Web server configuration files</td>
<td>“Steps for adding application domains to the Web server configuration files” on page 202</td>
</tr>
</tbody>
</table>

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.
Advanced customization

4. Customize the content of the Web pages for this application by modifying the `<CONTENT>...<CONTENT>` subsection. (See "TEMPLATE sections" on page 114 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 201. 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 177

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 178. 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 200.)

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, VSAM data sets, 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.)

Advanced customization

Task overview

This topic includes a task roadmap that you can use to add a new CA domain. The roadmap includes several subtasks that are listed in as listed in Table 43 on page 207. It is intended to direct you to add a new CA domain after you have completed...
Advanced customization

all required tasks in Part 2, “Configuring your system for PKI Services,” on page 25. 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 VSAM files, CA certificate, key ring, and LDAP namespace. You also create a dedicated copy of the PKI Services configuration file (pkiserv.conf), templates file (pkiserv.tmp) 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, 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:
  - pkiservtmpl 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 205 you would add the following URL (to the HTTP configuration files) and the following configuration path (to the HTTP environment variables file).

<table>
<thead>
<tr>
<th>New application domain (URL)</th>
<th>New configuration path</th>
</tr>
</thead>
<tbody>
<tr>
<td>http://&lt;server-name&gt;/Employees/*</td>
<td>_PKISERV_CONFIG_PATH_EMPLOYEES=/etc/pkiserv/employees</td>
</tr>
</tbody>
</table>

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 25. 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 43 on page 207 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 11.)

To create a new CA domain, complete the subtasks in Table 43 on page 207. 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 25 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>
<thead>
<tr>
<th>Subtask</th>
<th>Team member</th>
<th>Associated procedure (See...)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Plan additional CA domains.</td>
<td>UNIX programmer</td>
<td>“Subtask 1: Steps for planning additional CA domains” on page 208.</td>
</tr>
<tr>
<td>2. Reconfigure your initial CA domain to allow it to coexist with other CA domains. (This is a one-time setup.)</td>
<td>UNIX programmer</td>
<td>“Subtask 2: Steps for reconfiguring your initial CA domain to allow it to coexist with other CA domains” on page 210.</td>
</tr>
<tr>
<td>3. Run IKYSETUP.</td>
<td>MVS programmer</td>
<td>“Subtask 3: Steps for running the IKYSETUP exec” on page 212.</td>
</tr>
<tr>
<td>4. Configure the UNIX environment.</td>
<td>UNIX programmer</td>
<td>“Subtask 4: Steps for configuring the UNIX environment” on page 214.</td>
</tr>
<tr>
<td>7. Allocate VSAM data sets.</td>
<td>MVS programmer</td>
<td>“Subtask 7: Steps for allocating VSAM data sets” on page 218.</td>
</tr>
</tbody>
</table>

**Recording your progress adding CA domains**

As you complete each subtasks for adding a new CA domain, use Table 44 on page 208 to mark your progress. The tasks correspond to the subtasks listed in the roadmap in Table 43.
Table 44. Multiple CA domains: Worksheet #1 for recording progress adding new CA domains

<table>
<thead>
<tr>
<th>Subtask</th>
<th>First CA domain</th>
<th>Second CA domain</th>
<th>Third CA domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Plan additional CA domains.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Reconfigure initial domain. (once only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Run IKYSETUP.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Configure the UNIX environment.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Update the PKI Services template file. (Perform this task only if you implement the Web application using REXX CGI execs.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Update the Web server configuration.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Allocate VSAM data sets.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Start PKI Services.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
   - Pathnames (in lower case)—
     /etc/pkiserv/employees/pkiserv.conf

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 45 and Table 46. 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.)

<table>
<thead>
<tr>
<th>Installation variable</th>
<th>Default directory name</th>
</tr>
</thead>
<tbody>
<tr>
<td>install-dir</td>
<td>/usr/lpp/pkiserv</td>
</tr>
<tr>
<td>runtime-dir</td>
<td>/etc/pkiserv</td>
</tr>
</tbody>
</table>

a. Fill in the values for new CA domains, administrative domains, and directories in Table 45. You can add your information in the blank lines below or you can modify or cross out the sample rows.

**Table 45. Multiple CA domains: Worksheet #2 for planning your domain names. Use row 2 for shared administrators, row 3 for dedicated administrators.**

<table>
<thead>
<tr>
<th>CA domain name (runtime directory)</th>
<th>Truncated CA domain name</th>
<th>Administrative domain name (runtime directory)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Customers (/etc/pkiserv)</td>
<td>CUSTOMER</td>
<td>PKIServ (/etc/pkiserv)</td>
</tr>
<tr>
<td>2. Employees (/etc/pkiserv/employees)</td>
<td>EMPLOYEE</td>
<td>PKIServ (/etc/pkiserv)</td>
</tr>
<tr>
<td>3. Employees (/etc/pkiserv/employees)</td>
<td>EMPLOYEE</td>
<td>AdmEmployees (/etc/pkiserv/employees)</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b. Fill in your RACF user IDs, groups, and VSAM data set qualifiers in Table 46 on page 210. You can add your information in the blank lines below or you can modify or cross out the sample rows.
Advanced customization

Table 46. Multiple CA domains: Worksheet #3 for planning your RACF identifiers, z/OS UNIX identifiers, and VSAM data set names. Use row 2 for shared administrators, row 3 for dedicated administrators.

<table>
<thead>
<tr>
<th>Daemon user ID (UID)</th>
<th>Surrogate user ID (UID)</th>
<th>PKI administration group name (GID)</th>
<th>VSAM data set qualifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. PKISRVD (554)</td>
<td>PKISERV (555)</td>
<td>PKIGRP (655)</td>
<td>PKISRVD.VSAM</td>
</tr>
<tr>
<td>2. PKISRVD (554)</td>
<td>PKISERV (555)</td>
<td>PKIGRP (655)</td>
<td>PKISRVD.EMPLOYEE.VSAM</td>
</tr>
<tr>
<td>3. PKIDEMP (556)</td>
<td>PKISEMP (557)</td>
<td>PKIGEMP (657)</td>
<td>PKISRVD.EMPLOYEE.VSAM</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 45 on page 209) 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 45 on page 209)
         
         Example:
         
         _PKISERV_CONFIG_PATH_CUSTOMERS=/etc/pkiserv
Advanced customization

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 45 on page 209)

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 16 on page 37

   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 RACLST(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:
Advanced customization

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 45 on page 209.) 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 29. 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 27. 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.
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 449. 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 16 on page 37).

   Table 47. IKYSETUP log information you need to record

<table>
<thead>
<tr>
<th>Name of the SAF key ring:</th>
</tr>
</thead>
</table>

   When you are done: You have customized and run the IKYSETUP exec for this CA domain. Record your progress in Table 44 on page 208.
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 45. 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 70.

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 45 on page 209.) 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 18 on page 52.)

   **ObjectStore** Qualify each VSAM data set name with the CA domain name.  
   **Example:** `ObjectDSN='pkisrvd.employee.vsam.ost'` (See “Subtask 7: Steps for allocating VSAM data sets” on page 218.)

   **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 195 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 47 on page 213.  
   **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 82.  
   Make sure the LDAP directory is configured with a suffix for this CA domain. (See the explanation for the `Suffix` variable in Table 19 on page 73.)
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 448.)
   **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 44 on page 208.

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 101. 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 214) 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.

<table>
<thead>
<tr>
<th>Default values for the Customers CA domain</th>
<th>New values for the Employees CA domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION=&quot;/Customers/...&quot; (in mixed case)</td>
<td>ACTION=&quot;/Employees/...&quot; (also in mixed case)</td>
</tr>
<tr>
<td>&lt;APPLICATION NAME=CUSTOMERS&gt; (in upper case)</td>
<td>&lt;APPLICATION NAME=EMPLOYEES&gt; (also in upper case)</td>
</tr>
</tbody>
</table>
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 45 on page 209. 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.

<table>
<thead>
<tr>
<th>Default value for the PKISERV administrative CA domain</th>
<th>New value for the new AdmEmployees administrative CA domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;APPLICATION NAME=PKISERV&gt; (in upper case)</td>
<td>&lt;APPLICATION NAME=ADMEMPLOYEES&gt; (also in upper case)</td>
</tr>
</tbody>
</table>

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:

```xml
<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:**

```xml
<insert name=SelectCADomain>
<p> Select the CA domain to work with
<select name="domain">
# 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>
```
It’s time to finish customizing the PKI Services template file (pkiserv.tmpl) for this CA domain. Record your progress in Table 44 on page 208. Continue to the next task.

**Guideline:** Complete all subtasks for this new CA domain and ensure 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 executables, perform the steps in “Updating the Web server configuration if you use REXX CGI executables.”

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 218.

**Updating the Web server configuration if you use REXX CGI executables**

**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 202
- “Steps for updating the z/OS HTTP Server configuration files” on page 77.

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 45 on page 209 for domain name and directory) following the instructions in “Steps for adding application domains to the Web server configuration files.” 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 45 on page 209 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 45 on page 209)

   **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.
Advanced customization

Example:

_PKISERV_CONFIG_PATH_ADM_EMPLOYEES=/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 44 on page 208.

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: Steps for allocating VSAM data sets

Before you begin: This procedure requires you to be familiar with the information in Chapter 9, "Creating VSAM data sets," on page 87. You will find additional details about the following steps there.

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 PKISRVD.VSAM to include corresponding VSAM data set qualifier from Table 46 on page 210. 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 44 on page 208.

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 95. 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 45 on page 209.)

   **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 29.

   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 44 on page 208.

   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,
Advanced customization

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 pretrained to PKI Services, your CA. You can preregister SCEP clients in batches using the pkiprereg utility (see [Using the pkiprereg utility on page 331]) 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 a SCEP client on page 296].) 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.)
Advanced customization

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.
Advanced customization

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 and MD5 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 [KYP025I](KYP025I) Sample output:
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 305 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 48. Determine the values for these variables and record the information in the blank boxes:

Table 48. Information you need to enable Simple Certificate Enrollment Protocol (SCEP)

<table>
<thead>
<tr>
<th>Information needed</th>
<th>Where to find this information</th>
<th>Record your value here</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca_label—The label of your CA certificate in RACF.</td>
<td>See Table 10 on page 29</td>
<td></td>
</tr>
<tr>
<td>ra_label—The label of your RA certificate in RACF.</td>
<td>See Table 10 on page 29</td>
<td></td>
</tr>
<tr>
<td>ca_ring—The PKI Services SAF key ring.</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>ca_expires—The date the PKI Services CA certificate expires.</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>daemon—The user ID for the PKI daemon.</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>ra_backup_dsn—The name of the encrypted data set containing the backup copy of your new RA certificate and private key.</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>ra_dn—The RA's distinguished name.</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
</tbody>
</table>

Perform the following steps to enable PKI Services to process Simple Certificate Enrollment Protocol (SCEP) requests:
Advanced customization

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 52.)
   
   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) 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 221 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 101.)
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. 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.

Once a day, PKI Services checks the issued certificate list (ICL) for expiring certificates. (The `ExpireWarningTime` parameter (see the `CertPolicy` section in Table 18 on page 52) 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 `AdminNotifyNewn` 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 `AdminNotifyRemindern` keyword to specify one or more e-mail addresses of PKI administrators to be notified when a certificate is about to expire.
Advanced customization

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 138 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 48.)
- Customize the forms. (For details, see “Steps for customizing e-mail notification forms” on page 230.)
- 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 138.

For more information about the NotifyEmail field, see Table 27 on page 105. For information about fields on request forms, see Table 57 on page 276.

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
Advanced customization

Figure 31. Sample of rejectmsg.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.
Please contact the PKI Services administrator at 1-800-xxx-xxxx.
You will need the transaction ID listed below.

%%transactionid%%

Figure 32. Sample of expiringmsg.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 33. Sample of renewcertmsg.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%%

Figure 34. Sample of pendingmsg.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 35. Sample of requestmsg.form
Advanced customization

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>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%%cadomain%%</td>
<td>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.)</td>
</tr>
<tr>
<td>%%dn%%</td>
<td>The subject’s distinguished name. (This variable is valid in all the forms except the pending requests form, pendingmsg.form.)</td>
</tr>
<tr>
<td>%%printcert%%</td>
<td>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.)</td>
</tr>
<tr>
<td>%%notafter%%</td>
<td>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.)</td>
</tr>
<tr>
<td>%%pendreqlist%%</td>
<td>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.)</td>
</tr>
</tbody>
</table>
Advanced customization

Table 49. Descriptions of variables for forms (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%quicklink%</td>
<td>A link to a certificate that is ready for pick up. It contains the transaction ID, the &quot;&amp;&quot; 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=1jB6bwokkoQ2SHV%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B%2B&amp;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.</td>
</tr>
<tr>
<td>%recoverylink%</td>
<td>Part of the link to a certificate that can be recovered. It contains the serial number, the &quot;&amp;&quot; 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.</td>
</tr>
<tr>
<td>%recoverylist%</td>
<td>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.</td>
</tr>
<tr>
<td>%requestor%</td>
<td>The requestor of the certificate.</td>
</tr>
<tr>
<td>%transactionid%</td>
<td>The transaction ID (CertId) returned. (This variable is valid for the ready and reject forms only. It is ignored in the other forms.)</td>
</tr>
</tbody>
</table>

Table 50 summarizes which substitution variables are supported by which forms:

Table 50. Summary of substitution variables in forms. “X” indicates that the form supports the variable.

<table>
<thead>
<tr>
<th>Substitution variable</th>
<th>readymsg</th>
<th>rejectmsg</th>
<th>expiringmsg</th>
<th>pendingmsg</th>
<th>renewcertmsg</th>
<th>recoverymsg</th>
</tr>
</thead>
<tbody>
<tr>
<td>%cadomain%</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>X</td>
<td>(ignored)</td>
<td>(ignored)</td>
</tr>
<tr>
<td>%dn%</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>(ignored)</td>
<td>X</td>
<td>(ignored)</td>
</tr>
<tr>
<td>%printcert%</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>X</td>
<td>(ignored)</td>
</tr>
<tr>
<td>%notafter%</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>X</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
</tr>
<tr>
<td>%pendreqlist%</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>X</td>
<td>(ignored)</td>
<td>(ignored)</td>
</tr>
<tr>
<td>%quicklink%</td>
<td>X</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
</tr>
<tr>
<td>%recoverylink%</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>X</td>
</tr>
<tr>
<td>%recoverylist%</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>X</td>
</tr>
<tr>
<td>%requestor%</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>(ignored)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>%transactionid%</td>
<td>X</td>
<td>X</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
<td>(ignored)</td>
</tr>
</tbody>
</table>
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 48.)

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 47.
   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 225.
   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=1YSSL>
<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
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:

```xml
<tns:NotifyEmail optional="true" />
```

to

```xml
<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](https://www.ibm.com/support/knowledgecenter/SSEPGH_1.4.0/com.ibm.zos.v1r11.bsect documentation).  

### 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](https://www.ibm.com/support/knowledgecenter/SSEPGH_1.4.0/com.ibm.zos.v1r11.bsect documentation).) 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 95.)
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 47.
- 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 225.

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 47.
- 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 225.

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 16 on page 37 if you’re not sure what your daemon user ID is.)
Advanced customization

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 254. 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.
Chapter 15. Customizing with installation exits

PKI Services supports the use of installation exits 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, pkiexit.c, written in the C language. It is intended to demonstrate the power of the exit for the daemon and REXX CGI execs, and to provide a guide for you to write your own exit. 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 exits 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. Both files are, by default, located in: /usr/lpp/pkiserv/samples/.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pkiexit.c</td>
<td>Code sample for the exit (in the C programming language). You probably need to update the exit code before using it.</td>
</tr>
<tr>
<td>Makefile.pkiexit</td>
<td>Makefile for pkiexit.c.</td>
</tr>
</tbody>
</table>

Exit 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, it must be a UNIX executable residing in a file system, with appropriate permission assigned. The PKI Services daemon identifies the exit 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 is invoked by the PKI Services daemon using standard UNIX parameters (that is, argc and argv[]). The exit communicates its results back to the PKI Services daemon by way of a return code. The exit is called for preprocessing and post-processing before and after automatic certificate renewal processing. Unlike the PKI Services CGI exits, messages written to 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 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 can be implemented in the same program as the exits 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 pkserv.conf file specifies the maximum time PKI Services will wait for the exit to return. If ExitTimeout is not specified, PKI Services waits at most 30 seconds for the exit to return. If ExitTimeout is specified with a value greater than 1 hour, PKI Services waits 1 hour at the most for the exit to return.

### Steps for updating the exit code sample

To update the exit code sample, pkiexit.c, perform the following steps:

1. Copy the sample exit 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, pkiexit, by entering the following command:
   ```
   make
   ```

3. Move the executable 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 for pre- and post-processing

This exit 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 shortly after the PKI Services daemon is started, and then once a day after that.

**Table 52. Values of arguments for pre- and post-processing**

<table>
<thead>
<tr>
<th>Time of processing</th>
<th>Argument 1</th>
<th>Argument 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing</td>
<td>0</td>
<td>The function number in EBCDIC:</td>
</tr>
<tr>
<td>Post-processing</td>
<td>1</td>
<td>500</td>
</tr>
</tbody>
</table>
Automatic renewal—preprocessing

Purpose: Provide additional criteria for automatic renewal of certificates.

Arguments:

argument 3
The Base64-encoded original certificate.

Return codes:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Continue with the request for automatic renewal processing.</td>
</tr>
<tr>
<td>4</td>
<td>Disallow automatic renewal processing for the current request.</td>
</tr>
<tr>
<td>8</td>
<td>Disallow automatic renewal processing from now on.</td>
</tr>
</tbody>
</table>

Note: Any value other than 0, 4, and 8 is treated as 4.

STDOUT: Non-applicable.

Note: The automatic renewal exit cannot write to STDOUT or STDERR like other exits. 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:*

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal return.</td>
</tr>
</tbody>
</table>

*Note:* The return code is not checked from the post exit processing.

*STDOUT:* Non-applicable.

*Note:* The automatic renewal exit cannot write to STDOUT or STDERR like other existing exits. The output has to be written to a file.
Scenario for using the exit

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 pkiexit.c shipped with PKI Services.

The preprocessing exit 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 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 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 exists, it must be a UNIX executable residing in the file system, and it must have appropriate permission assigned. To specify the exit, the UNIX programmer sets the _PKISERV_EXIT environment variable in the Web server's
Steps for updating the exit code sample

To update the exit code sample, `pkiexit.c`, perform the following steps:

1. Copy the sample exit 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, `pkiexit`, by entering the following command:
   ```
   make
   ```

3. Move the executable 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 for pre- and post-processing

The exit 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>
<thead>
<tr>
<th>Time of processing</th>
<th>Argument 1</th>
<th>Argument 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing</td>
<td>0</td>
<td>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</td>
</tr>
<tr>
<td>Post-processing</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
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 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:**

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Continue with the request with possible modifications.</td>
</tr>
<tr>
<td>4</td>
<td>Continue with the request with possible modifications, but change it to require administrator approval.</td>
</tr>
<tr>
<td>8 - 49</td>
<td>Deny the request and return to the caller immediately.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal</td>
</tr>
</tbody>
</table>

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:**

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Continue with the request with possible modifications.</td>
</tr>
<tr>
<td>4</td>
<td>Continue with the request with possible modifications, but change it to not require administrator approval.</td>
</tr>
<tr>
<td>8 - 49</td>
<td>Deny the request and return to the caller immediately.</td>
</tr>
</tbody>
</table>

**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 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:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal</td>
</tr>
</tbody>
</table>

STDOUT: Optional replacement TransactionId.
EXTRACTION—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:**

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Continue with the export.</td>
</tr>
<tr>
<td>8 - 49</td>
<td>Deny the request and return to the caller immediately.</td>
</tr>
</tbody>
</table>

**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, 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:*

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal</td>
</tr>
</tbody>
</table>

*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:*

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Continue with the request.</td>
</tr>
<tr>
<td>8 - 49</td>
<td>Deny the request and return to the caller immediately.</td>
</tr>
</tbody>
</table>

*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:*

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal</td>
</tr>
</tbody>
</table>

*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 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:**

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Continue with the query.</td>
</tr>
<tr>
<td>4</td>
<td>Cannot determine the passphrase.</td>
</tr>
<tr>
<td>8 - 49</td>
<td>Deny the request and return to the caller immediately.</td>
</tr>
</tbody>
</table>

**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 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:**

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal</td>
</tr>
</tbody>
</table>

**STDOUT:** Non-applicable.
Scenarios for using the exit

The sample exit 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:

<table>
<thead>
<tr>
<th>Access Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>No access for either resource. The user is not permitted to request this type of certificate. The certificate request is denied.</td>
</tr>
<tr>
<td>READ to PROJ.MEMBER</td>
<td>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.)</td>
</tr>
<tr>
<td>UPDATE to PROJ.MEMBER</td>
<td>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.)</td>
</tr>
<tr>
<td>READ to PROJ.PARTNER</td>
<td>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.</td>
</tr>
<tr>
<td>UPDATE to PROJ.PARTNER</td>
<td>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.</td>
</tr>
</tbody>
</table>

The preprocessing exit 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 through argv in field-name=field-value pairs, and the subroutine looks for the Template= and UserId= in the input parameters.
- When the exit 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 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 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 \( n = \text{number of days left in the old certificate's validity period} + \text{number of days specified by NotAfter.} \)

The preprocessing exit 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 \((86400 \times 30)\) 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 \(\text{timeBeforeExp}/86400 + \text{requestPeriod}\).
• 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 calls are required:
• When the user requests the PKI generated key certificate, the preprocessing exit 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 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 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 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 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 54. Package and class summary for JSP exit processing

<table>
<thead>
<tr>
<th>Package</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.ibm.pki.web.exits</td>
<td>UserExit</td>
</tr>
<tr>
<td></td>
<td>ExportCert</td>
</tr>
<tr>
<td></td>
<td>QRrecover</td>
</tr>
<tr>
<td></td>
<td>RevokeCert</td>
</tr>
<tr>
<td></td>
<td>UserExitException</td>
</tr>
<tr>
<td>com.ibm.pki.rpkiserv</td>
<td>CertPlist</td>
</tr>
<tr>
<td></td>
<td>PkiCertificate</td>
</tr>
<tr>
<td></td>
<td>QrecoverResultsList</td>
</tr>
<tr>
<td></td>
<td>RpkiservException</td>
</tr>
</tbody>
</table>

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 55. Methods in class UserExit

<table>
<thead>
<tr>
<th>Method</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>preGenReqCert</td>
<td>Called before GENCERT or REQCERT requests</td>
</tr>
<tr>
<td>postGenReqCert</td>
<td>Called after GENCERT or REQCERT requests</td>
</tr>
<tr>
<td>preGenReqRenew</td>
<td>Called before GENRENEW or REQRENEW requests</td>
</tr>
<tr>
<td>postGenReqRenew</td>
<td>Called after GENRENEW or REQRENEW requests</td>
</tr>
<tr>
<td>preExport</td>
<td>Called before EXPORT requests</td>
</tr>
<tr>
<td>postExport</td>
<td>Called after EXPORT requests</td>
</tr>
<tr>
<td>preRevoke</td>
<td>Called before REVOKE requests</td>
</tr>
<tr>
<td>postRevoke</td>
<td>Called after REVOKE requests</td>
</tr>
<tr>
<td>preQRecover</td>
<td>Called before QRECOVER requests</td>
</tr>
<tr>
<td>postQRecover</td>
<td>Called after QRECOVER requests</td>
</tr>
</tbody>
</table>
preGenReqCert method

```java
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:**
<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (UserExit.SUCCESSFUL)</td>
<td>Continue with the request.</td>
</tr>
<tr>
<td>4 (UserExit.CHANGE_APPROVAL_STATUS)</td>
<td>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).</td>
</tr>
<tr>
<td>8 (UserExit.DENY_REQUEST) or greater</td>
<td>Deny the request.</td>
</tr>
</tbody>
</table>

**throws:**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserExitException</td>
<td>Handled the same as a return value of 8, and exception text is displayed on the resulting web page</td>
</tr>
</tbody>
</table>
postGenReqCert method

```java
public java.lang.String postGenReqCert(java.lang.String domain,
  com.ibm.pki rpkserv.CertPlist plist,
  int RACFrc,
  int RACFrscncode,
  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
- **RACFrscncode**
  - 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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (UserExit.SUCCESSFUL)</td>
<td>Continue with the request.</td>
</tr>
<tr>
<td>4 (UserExit.CHANGE_APPROVAL_STATUS)</td>
<td>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).</td>
</tr>
<tr>
<td>8 (UserExit.DENY_REQUEST) or greater</td>
<td>Deny the request.</td>
</tr>
</tbody>
</table>

throws:

<table>
<thead>
<tr>
<th>Exception</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserExitException</td>
<td>Handled the same as a return value of 8, and exception text is displayed on the resulting web page</td>
</tr>
</tbody>
</table>
postGenReqRenew method

definition

```java
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

```java
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:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (UserExit.SUCCESSFUL)</td>
<td>Continue with the EXPORT.</td>
</tr>
<tr>
<td>8 (UserExit.DENY_REQUEST)</td>
<td>or greater Deny the request.</td>
</tr>
</tbody>
</table>

**Throws:**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserExitException</td>
<td>Handled the same as a return value of 8, and exception text is displayed on the resulting web page</td>
</tr>
</tbody>
</table>
**postExport method**

```java
public void postExport(java.lang.String domain,
                        com.ibm.pki.rpkserv.ExportCert exportobject,
                        int RACFrc,
                        int RACFrsrcode)
```

**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**

```java
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:*

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (UserExit.SUCCESSFUL)</td>
<td>Continue with the REVOKE request.</td>
</tr>
<tr>
<td>8 (UserExit.DENY_REQUEST) or greater</td>
<td>Deny the request, don’t revoke.</td>
</tr>
</tbody>
</table>

*throws:*

<table>
<thead>
<tr>
<th>Exception</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserExitException</td>
<td>Handled the same as a return value of 8, and exception text is displayed on the resulting Web page</td>
</tr>
</tbody>
</table>
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**

```java
public int preQRecover(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:*

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (UserExit.SUCCESSFUL)</td>
<td>Continue with the request.</td>
</tr>
<tr>
<td>4 (UserExit.QRECOVER_PASSPHRASE_NOT_FOUND)</td>
<td>Cannot determine the passphrase.</td>
</tr>
<tr>
<td>8 (UserExit.DENY_REQUEST) or greater</td>
<td>Deny the request.</td>
</tr>
</tbody>
</table>
postQRecover method

```java
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

```java
public class ExportCert
    extends java.lang.Object

    ExportCert contains the parameters passed on an R_PKIServ EXPORT request.
```

Method summary

```java
    java.lang.String getPassphrase()
    java.lang.String getTransactionid()
    void setPassphrase(java.lang.String passphrase)
    void setTransactionid(java.lang.String transactionid)
```

Class QRecover

```java
public class QRecover
    extends java.lang.Object

    QRecover contains the parameters passed on an R_PKIServ QRECOVER request.
```

Method summary

```java
    java.lang.String getPassphrase()
    com.ibm.pki.rpkserv.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

```java
public class RevokeCert
    extends java.lang.Object

    RevokeCert contains the parameters passed on an R_PKIServ REVOKE request.
```

Method summary

```java
    int getReason()
    java.lang.String getSerial_number()
    void setReason(int reason)
    void setSerial_number(java.lang.String serial_number)
```

Class UserExitException

```java
public class UserExitException
    extends java.lang.Exception
```

266  z/OS V1R11.0 Cryptographic Services PKI Services Guide and Reference
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.

<table>
<thead>
<tr>
<th>Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserExitException(java.lang.String message)</td>
</tr>
<tr>
<td>UserExitException(java.lang.Throwable cause)</td>
</tr>
</tbody>
</table>

### Class CertPlist

Package: com.ibm.pki.rpkserv

```java
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_EMAILADOR
- 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_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_DNQUALIFIER
- CERTPLIST_UDID
- CERTPLIST_COMMONNAME
- CERTPLIST_DOMAINNAME
- CERTPLIST_EMAIL
- CERTPLIST_CLIENTNAME
- CERTPLIST_STARTDATE
- CERTPLIST_ENDDATE
- CERTPLIST_AUTORENEW
- CERTPLIST_KEYSIZE
### Method summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td><code>addValue(java.lang.String name, java.lang.String value)</code></td>
<td>Adds a value to the certPlist.</td>
</tr>
<tr>
<td>void</td>
<td><code>addValue(java.lang.String name, java.lang.String[] values)</code></td>
<td>Adds values to the certPlist.</td>
</tr>
<tr>
<td>java.util.Vector</td>
<td><code>getNames()</code></td>
<td>Gets the names of all name and value pairs in the certPlist.</td>
</tr>
<tr>
<td>java.util.Vector</td>
<td><code>getValues(java.lang.String name)</code></td>
<td>Gets the vector of string values for this name in the certPlist.</td>
</tr>
<tr>
<td>void</td>
<td><code>removeAllValues(java.lang.String name)</code></td>
<td>Deletes all of the values associated with a name in the certPlist.</td>
</tr>
<tr>
<td>void</td>
<td><code>removeValue(java.lang.String name, java.lang.String value)</code></td>
<td>Deletes one of the values associated with a name in the certPlist.</td>
</tr>
<tr>
<td>java.lang.String</td>
<td><code>toString()</code></td>
<td>Returns the string representation of this object.</td>
</tr>
</tbody>
</table>

### Class PkiCertificate

Package: com.ibm.pki.rpkiserv

```java
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:

<table>
<thead>
<tr>
<th>Integer</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PkiCertificate.PKIS_CERTIFICATE_TYPE_BASE64</td>
<td>Certificate is BASE64-encoded</td>
</tr>
<tr>
<td>PkiCertificate.PKIS_CERTIFICATE_TYPE_DER</td>
<td>Certificate is DER-encoded</td>
</tr>
<tr>
<td>PkiCertificate.PKIS_CERTIFICATE_TYPE_PKCS12</td>
<td>Certificate is DER-encoded PKCS #12</td>
</tr>
<tr>
<td>PkiCertificate.PKIS_CERTIFICATE_TYPE_PKCS7_CHAIN</td>
<td>Certificate is DER-encoded PKCS #7 chain</td>
</tr>
</tbody>
</table>
Method summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>public java.lang.String getBase64Encoded()</td>
<td>Certificate in Base64 format. Use this method if the certificate type is PKIS_CERTIFICATE_TYPE_BASE64.</td>
</tr>
<tr>
<td>public byte[] getDerEncoded()</td>
<td>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.</td>
</tr>
<tr>
<td>public int getType()</td>
<td>Returns the certificate type.</td>
</tr>
</tbody>
</table>

Class QrecoverResultsList

Package: com.ibm.pki.rpkiserv

public class QrecoverResultsList
extends java.lang.Object

Contains QRECOVER results from R_Pkiserv.

Method summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.String getIssuerDn()</td>
<td>Returns the issuer’s distinguished name.</td>
</tr>
<tr>
<td>java.lang.String getKeyId()</td>
<td>Returns the key Id.</td>
</tr>
<tr>
<td>java.lang.String getPassPhrase()</td>
<td>Returns the passphrase provided when the certificate request was made.</td>
</tr>
<tr>
<td>java.lang.String getSerialNum()</td>
<td>Returns the serial number.</td>
</tr>
<tr>
<td>java.lang.String getSubjectDn()</td>
<td>Returns the subject’s distinguished name.</td>
</tr>
<tr>
<td>java.lang.String getValidityDates()</td>
<td>Returns the validity period in local time.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Constructor summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>RpkiservException()</td>
</tr>
<tr>
<td>Constructor to create an empty RpkiservException object.</td>
</tr>
<tr>
<td>RpkiservException(java.lang.String exceptionText)</td>
</tr>
<tr>
<td>Constructor to create an RpkiservException object with only exception text.</td>
</tr>
</tbody>
</table>

----------------------- 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 273** 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 299** shows the administration Web pages and explains how to process certificate requests and certificates.

- **Chapter 18, “Using PKI Services utilities,” on page 321** explains using the PKI Services utilities.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iclview</td>
<td>A UNIX program that displays the entries in the VSAM issued certificate list (ICL) data set.</td>
</tr>
<tr>
<td>vosview</td>
<td>A UNIX program that displays the entries contained in the VSAM ObjectStore data set (request database)</td>
</tr>
<tr>
<td>pkiprereg</td>
<td>A UNIX program that creates Simple Certificate Enrollment Protocol (SCEP) preregistration records</td>
</tr>
<tr>
<td>TemplateTool</td>
<td>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.</td>
</tr>
</tbody>
</table>
Chapter 16. Using the end-user Web pages

This topic describes how the end user can use the PKI Services Web pages.

Note: 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.

By default, the end user can:
- 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

The following table lists the types of certificates you can request:

<table>
<thead>
<tr>
<th>Type of certificate</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-year PKI SSL browser certificate</td>
<td>End-user client authentication using SSL</td>
</tr>
<tr>
<td>One-year PKI S/MIME browser certificate</td>
<td>Browser-based e-mail encryption</td>
</tr>
<tr>
<td>One-year PKI generated key certificate</td>
<td>Generation of public and private keys by PKI Services</td>
</tr>
<tr>
<td>Two-year PKI browser certificate for authenticating to z/OS</td>
<td>End-user client authorization using SSL when logging onto z/OS</td>
</tr>
<tr>
<td>Two-year PKI Authenticode—code signing server certificate</td>
<td>Software signing</td>
</tr>
<tr>
<td>Two-year PKI Windows logon certificate</td>
<td>End-user client authentication for an Active Directory user logging in to a Windows desktop using a smart card</td>
</tr>
<tr>
<td>Five-year PKI SSL server certificate</td>
<td>SSL Web server certification</td>
</tr>
<tr>
<td>Five-year PKI IPSEC server (firewall) certificate</td>
<td>Firewall server identification and key exchange</td>
</tr>
<tr>
<td>Five-year PKI intermediate CA server certificate</td>
<td>Subordinate (non-self-signed) certificate-authority certification</td>
</tr>
<tr>
<td>Five-year SCEP certificate</td>
<td>Creation of a preregistration record for certificate requestors. (Certificate requestors using Simple Certificate Enrollment Protocol (SCEP) must be preregistered.)</td>
</tr>
<tr>
<td>n-year PKI browser certificate for extensions demonstration</td>
<td>Demonstration of all extensions supported by PKI Services</td>
</tr>
</tbody>
</table>

Unlike other templates, this template is intended for administration use only.
Using the end-user Web pages

Table 56. Types of certificates you can request  (continued)

<table>
<thead>
<tr>
<th>Type of certificate</th>
<th>Use</th>
</tr>
</thead>
</table>
| 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 can not 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 can not 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 56 on page 273 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:
### PKI Services Certificate Generation Application

**Install the CA certificate to enable SSL sessions for PKI Services**

**Choose one of the following:**

- **Request a new certificate using a model**
  
  ```
  Select the certificate template to use as a model: [Select Certificate]
  
  Request Certificate
  ```

- **Pick up a previously requested certificate**
  
  ```
  Enter the assigned transaction ID
  
  Select the certificate certain type: [Select Certificate]
  
  Pick up Certificate
  ```

- **Renew or revoke a previously issued browser certificate**
  
  ```
  Renew or Revoke Certificate
  ```

- **Recover a previously issued certificate whose key was generated by PKI Services**
  
  ```
  Enter the email address when the original certificate was requested
  
  Enter the same pass phrase as on the request form: [Enter Pass Phrase]
  
  Click here if you forgot the pass phrase
  
  Recover Certificate
  ```

- **Administrators click here**
  
  ```
  Go to Administration Page
  ```

*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 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 the OK button. The following is an example of the popup window you might see, depending on the CA certificate you have installed.
Using the end-user Web pages

Figure 37. The certificate popup window for installing the CA certificate

b. Click the **Install certificate** button. (This initiates a series of pop-ups in which you need to click **Next** buttons and finally the **Finish** button, 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 Vista system” on page 507.

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 57. Summary of fields in end-user Web pages*

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate</td>
<td></td>
</tr>
</tbody>
</table>

---

z/OS V1R11.0 Cryptographic Services PKI Services Guide and Reference
Using the end-user Web pages

Table 57. Summary of fields in end-user Web pages (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate fields related to Subject's Distinguished Name</td>
<td>Notes:</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Common name</td>
<td>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.</td>
</tr>
<tr>
<td>Country</td>
<td>The country where your organization is located. This is a 2-character text field.</td>
</tr>
<tr>
<td>Distinguished name qualifier</td>
<td>Specifies information to add to the subject distinguished name of an entry to make it unambiguous.</td>
</tr>
<tr>
<td>Domain component</td>
<td>One component of a domain name associated with the subject distinguished name. For example, the domain name <a href="http://www.ibm.com">www.ibm.com</a> is represented by 3 components: www, ibm and com.</td>
</tr>
<tr>
<td>E-mail address</td>
<td>E-mail address with attribute EMAIL for the distinguished name. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>Locality</td>
<td>The city or municipality where your organization is located, such as Pittsburgh or Paris. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>Mail</td>
<td>E-mail address with attribute MAIL for the distinguished name. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Restriction:</strong> If you specify a value for this parameter and for Notification e-mail address, the two values must be the same.</td>
</tr>
<tr>
<td>Organization</td>
<td>The legally registered name (or trademark name, for example, IBM) of your organization. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>Organizational unit</td>
<td>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.</td>
</tr>
<tr>
<td>Postal code</td>
<td>Your postal code or zip code. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>Serial number</td>
<td>Serial number of the subject device. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>State or Province</td>
<td>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.</td>
</tr>
<tr>
<td>Street</td>
<td>Your street address. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>Title</td>
<td>Your job title. This is a text field of up to 64 characters.</td>
</tr>
<tr>
<td>Unstructured address</td>
<td>The unstructured address of the subject device.</td>
</tr>
<tr>
<td>Unstructured name</td>
<td>The unstructured name of the subject device.</td>
</tr>
<tr>
<td>User ID</td>
<td>The system login name associated with the subject distinguished name.</td>
</tr>
</tbody>
</table>
### Table 57. Summary of fields in end-user Web pages (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not after (date)</td>
<td>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.</td>
</tr>
<tr>
<td>Not before (date)</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Certificate fields related to extensions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate domain name</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</td>
</tr>
<tr>
<td>Alternate e-mail address</td>
<td>E-mail address for alternate name, including the @ character and any periods (.). This is a text field of up to 100 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</td>
</tr>
<tr>
<td>Alternate IP address</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>• For IP version 4, the IP address is in dotted decimal format; for example, 9.67.97.103.</td>
</tr>
<tr>
<td></td>
<td>• 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 0001:0002:0003:0004:0005:0006:0007:0008.</td>
</tr>
<tr>
<td></td>
<td>• In a mixed IP version 4 and IP version 6 environment, the IP address can be expressed in the format xxx:xxx:xxx::dd.dd, where the xxx values are the hexadecimal values of the six high-order 16-bit pieces of the address, and the dd 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</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</td>
</tr>
<tr>
<td>Alternate other name</td>
<td>Additional identifier for the alternate name. See your PKI Services administrator for information about this field.</td>
</tr>
<tr>
<td>Alternate uniform resource identifier (URI)</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</td>
</tr>
<tr>
<td>Extended key usage</td>
<td>This indicates the intended purpose of the certificate. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• clientauth: Client side authentication</td>
</tr>
<tr>
<td></td>
<td>• codesigning: Code signing</td>
</tr>
<tr>
<td></td>
<td>• emailprotection: Email protection</td>
</tr>
<tr>
<td></td>
<td>• msssmartcardlogon: Smart card logon for Microsoft Windows users</td>
</tr>
<tr>
<td></td>
<td>• ocspsigning: OCSP response signing</td>
</tr>
<tr>
<td></td>
<td>• serverauth: Server side authentication</td>
</tr>
<tr>
<td></td>
<td>• timestamping: Digital timestamping</td>
</tr>
<tr>
<td>HostIdMapping</td>
<td>This is the user ID for authorization purposes in the format: subject-id@host-name</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <a href="mailto:DSmith@ibm.com">DSmith@ibm.com</a></td>
</tr>
<tr>
<td></td>
<td>This is a text field of up to 100 characters.</td>
</tr>
</tbody>
</table>
Table 57. Summary of fields in end-user Web pages (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key usage</td>
<td>The intended purpose of the certificate. Each possible value is shown here with its intended purpose and possible PKIX bits:</td>
</tr>
<tr>
<td>KeyUsage value</td>
<td>Intended purpose PKIX bits</td>
</tr>
<tr>
<td>certsign</td>
<td>Certificate and CRL signing KeyCertSign and cRLSign</td>
</tr>
<tr>
<td>crlsign</td>
<td>CRL signing cRLSign</td>
</tr>
<tr>
<td>dataencrypt, dataencipherment, or dataenciph</td>
<td>Data encryption dataEncipherment</td>
</tr>
<tr>
<td>digitalsig or digitalsignature</td>
<td>Authentication digitalSignature</td>
</tr>
<tr>
<td>docsign or nonrepudiation</td>
<td>Document signing nonRepudiation</td>
</tr>
<tr>
<td>handshake</td>
<td>Protocol handshaking (for example, SSL) digitalSignature and keyEncipherment</td>
</tr>
<tr>
<td>keyagree or keyagreement</td>
<td>Key agreement keyAgreement</td>
</tr>
<tr>
<td>keycertsign</td>
<td>Certificate signing keyCertSign</td>
</tr>
<tr>
<td>keyencrypt, keyencipherment, or keyenciph</td>
<td>Key transport keyEncipherment</td>
</tr>
</tbody>
</table>

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-----
MIIBiDCBgIIBADAZMRwczFQY0QOQDEe5Kb2hJbFElYwmpYzC5NzANBqk6hkiG
9wBAQEFAAAOB4wYkCq7YAEsCT1cJHAGPq160fAjAL+X56s5nmgwq02V0030Yu
/mEnQ7n796ke+zJbmDco5tWk140Yf9ZyB5biURMjFLzHa44V0dEvfnun80H2pw
w1X0IZcCIZymSadurlyOk646pgiyIPMQS/t0ttG45UsK0i0J4V4f7ps+i1a6
t+cCAwEAaAwMC4GCs0g51b3DQDj9jJ9fQyDVR0DB8YEFAltKoBvRnFqDAC0
lojH1rNwRC9MA0QG6CqG01b3DQDeBBQoA4GhA1bCvPyvppXi3HmpK2P0YB8nsz
AjrRoaEyHr1iW1OR6vwhqKcLxa9hToQi21cdcBRpfV7w56UfdCOCxMn4a534Tk3
SWYdOiHxp98hHiB3EruwDw+57Fv3JHU3FwU61FLFvAjBv+351eWQymoR6mE5W
CathpmmGFQsDE5E
-----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 283.

PKI Services internal use fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge passphrase</td>
<td>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.</td>
</tr>
<tr>
<td>KeySize</td>
<td>The size of the key pair (public key and private key) that you want PKI Services to generate for you.</td>
</tr>
<tr>
<td>Label</td>
<td>The label assigned to the requested certificate. This is a text field of up to 32 characters. This field applies only to SAF certificates.</td>
</tr>
</tbody>
</table>
### Table 57. Summary of fields in end-user Web pages (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification e-mail address</td>
<td>E-mail address for notification purposes. This is a text field of up to 64 characters. <strong>Note:</strong> If you specify a value for this parameter and for Mail, the two values must be the same.</td>
</tr>
<tr>
<td>Passphrase</td>
<td>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.)</td>
</tr>
<tr>
<td>Requestor's name</td>
<td>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. <strong>Note:</strong> For a PKI generated key certificate, the requestor name needs to be in the form of an e-mail address.</td>
</tr>
<tr>
<td>Transaction ID</td>
<td>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.</td>
</tr>
</tbody>
</table>

#### Browser-specific fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptographic service provider</td>
<td>(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.</td>
</tr>
<tr>
<td>Key protection</td>
<td>(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.)</td>
</tr>
<tr>
<td>Key size</td>
<td>(This is for the Netscape browser 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.</td>
</tr>
</tbody>
</table>

### Steps for requesting a new certificate

To request a new certificate, first go to the PKI Services home page. (See Figure 36 on page 275.)

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 a SCEP client" on page 296.

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 282 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
Using the end-user Web pages

- 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 the Request certificate button. 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 Netscape 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 282 appears.
1-Year SSL Browser Certificate

Choose one of the following:

- **Request a New Certificate**
  
  Enter values for the following field(s):
  
  Your name for tracking this request (optional)
  
  Email address for distinguished name (optional)
  
  Common Name
  
  Email address for notification purposes (optional)
  
  Pass phrase for securing this request. You will need to supply this value when retrieving your certificate
  
  Reenter your pass phrase to confirm
  
  Select the following key information
  
  Cryptographic Service Provider [Gemplus GemSAFE Card CSP v1.0]
  
  Enable strong private key protection? [No]
  
  Submit certificate request Clear

- **Pick Up a Previously Issued Certificate**
  
  Retrieve your certificate

---

**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 57 on page 276 for descriptions of fields.) If you are using Netscape, 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 283.
5-Year PKI SSL Server Certificate

Choose one of the following:

- Request a New Certificate

  Enter values for the following fields:

  Your name for identifying this request (Optional)

  Email address for distinguished-name (Optional)

  Common Name (Optional)

  Organizational Unit (Optional)

  Organizational Unit (Optional)

  Organization (Optional)

  Street address (Optional)

  Locality (Optional)

  State or Province (Optional)

  Zip code or postal code (Optional)

  Country (Optional)

  Email address for alternate name (Optional)

  Domain name for alternate name (Optional)

  Uniform Resource Identifier for alternate name (Optional)

  IP address for alternate name in dotted decimal form (Optional)

  Email address for notification purpose (Optional)

  Password for securing this request. You will need to supply this value when retrieving your certificate

  Re-enter your pass phrase to confirm

  Base64 encoded PKCS#10 certificate request

  Submit certificate request Close

- Pick Up a Previously Issued Certificate

  Retrieve your certificate

  email_support@your-company.com

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 57 on page 276.

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 the Submit certificate request button. 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.
   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 285, you might have to fill in the transaction ID on that Web page.)

   Transaction ID:

   b. Click the Continue button. This displays the following Web page:
Using the end-user Web pages

Retrieve Your 1-Year PKI SSL Browser Certificate

Please bookmark this page

Since your certificate may not have been issued yet, we recommend that you create a bookmark to this location so that when you bookmark the browser will display your transaction ID. This is the easiest way to check your status.

Enter the assigned transaction ID

If you specified a pass phrase when submitting the certificate request, type it here, exactly as you typed it on the request form

Retrieve and Install Certificate

Home page

e-mail: webmaster@your-company.com

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 286) 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

Request submitted successfully

A link to pick up the certificate was sent to the specified requestor's e-mail address at lewallon@us.ibm.com.

Home page

e-mail: webmaster@your-company.com

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 285. (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 275 and “Steps for retrieving your certificate from the PKI Services home page” on page 287.)

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 288.)

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 285)

2. If you entered a passphrase when requesting your certificate, enter the passphrase.

3. Click the Retrieve and install certificate button. If you are using Netscape, 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 287 shows an example of the Web page.)

**Internet Explorer certificate install**

Click "Install Certificate" to store your new certificate into your browser.

**Figure 43. Browser certificate installation Web page**
4. Click the Install certificate button. 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 Netscape, click the Security button, 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 284)

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 the Pick up certificate button on the PKI Services home page. (See Figure 36 on page 275) This displays the Web page that Figure 41 on page 285 shows.

2. Enter your passphrase (this is the challenge passphrase) if you specified one when requesting your certificate.

3. Click the Retrieve and install certificate button. If you are using Netscape, 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 286 shows. (This is for a browser certificate. For a server certificate, Figure 44 shows an example of the Web page.)

4. Click the Install certificate button. If the certificate installs successfully, you get a popup window that says Your new certificate installed successfully.
Using the end-user Web pages

5. Check that your certificate installed correctly:
   - For Netscape, click the **Security** button, 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 275). 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:
   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%2F2SHV++++++
   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 289 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**.
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.

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:
Using the end-user Web pages

1. On the PKI Services home page (see Figure 36 on page 275), click the Renew or revoke certificate button. This displays a popup window with a list of certificates, such as the following figure shows:

![Popup window listing certificates](image)

*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 291 shows an example of the Web page you might see, depending on the certificate you are renewing:
Renew or Revoke a Browser Certificate

Here is the certificate you selected:

| Requestor: Gumby | Created: 2007/04/04 |
| Status: Active | Modified: 2007/04/04 |
| Template: 1-Year PKI SSL Browser Certificate |
| Serial #: 5 |
| Subject: MAIL=ymc@us.ibm.com,CN=Gumby,OU=Class 1 Internet Certificate CA,O=The Firm |
| Issuer: OU=Master CA,O=IBM,C=US |
| Validity: 2007/04/04 00:00:00 - 2008/04/02 23:59:59 |
| Usage: handshake(digitalSignature, keyEncipherment) |
| Extended Usage: client auth |

If this is the correct certificate, choose one of the following:

- Renew the above certificate
  - Email address for notification purposes (optional)
  - Pass phrase for securing this request. You will need to supply this value when retrieving your certificate
  - Reenter your pass phrase to confirm
  - Renew

- Revoke the above certificate
  - Revoke

- Suspend the above certificate
  - Suspend

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 the Renew or revoke certificate button as in Step 1 on page 290.

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.
Using the end-user Web pages

4. Click the Renew button.

5. If the renewal request is successful, this displays a Web page that says Request submitted successfully and displays the transaction ID. Click the Continue button on this Web page.

6. This takes you to the Web page from which you retrieve your certificate. (See Figure 41 on page 285 for an example of this Web page and Steps for retrieving your certificate from the bookmarked Web page on page 286 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 275), click Renew or revoke certificate. This displays a popup window with a list of certificates, as in Figure 48 on page 290.

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 289.)

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.
Using the end-user Web pages

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 275), 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 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.

Recover your certificate

Security questions – answer the following with the same answers you provided in the original request if you forget the pass phrase.

Enter the e-mail address when the original certificate was requested

<table>
<thead>
<tr>
<th>What's the intended use of this certificate?</th>
</tr>
</thead>
<tbody>
<tr>
<td>What's the name of your elementary school?</td>
</tr>
</tbody>
</table>

Recover Certificate

<table>
<thead>
<tr>
<th>Home Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>email: <a href="mailto:webmaster@your-company.com">webmaster@your-company.com</a></td>
</tr>
</tbody>
</table>

Figure 50. Web page requesting answers to security questions when you have forgotten the passphrase
Using the end-user Web pages

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.

Certificate(s) to be recovered

The following issued certificates were requested by the specified email address lewallen@us.ibm.com with the specified pass phrase or security answers.

A note has been sent to the above address. Use the supplied link from the note to recover the one you need.

Show Pass phrase

| Serial #: 0000000000000000 |
| Subject: CN=Nancy Lewallen,OU=Class 1 Internet Certificate CA,O=The Firm |
| Validity: 2008/10/12 00:00:00 - 2008/10/12 00:00:00 |

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 51. Web page listing certificates that can be recovered

Figure 52. Web page showing the passphrase for a certificate to be recovered

3. Open the e-mail you were sent. [Figure 53 on page 295] shows a sample e-mail that lists one certificate eligible for recovery. Click on the link for the certificate that you want to recover.
Using the end-user Web pages

4. The link takes you to the Web page shown in Figure 54.

Retrieve your recovered certificate

Key ID 51427A4B98F57A03E

Serial number 000000000000000D

Enter the requestor’s email address

Enter the same pass phrase as on the request form

Retrieve Certificate

Home Page

email: webmaster@your-company.com

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 296.
Using the end-user Web pages

![File Download]

**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 a SCEP client**

These steps are performed by the PKI administrator using the end-user Web pages. To preregister a SCEP client, first go to the PKI Services home page. (See Figure 36 on page 275.)

Perform the following steps to complete a preregistration request and preregister a 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 the **Request certificate** button. A form where you fill in information is displayed.

3. The preregistration form appears. (See Figure 56 on page 297.) 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.
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 the Submit certificate request button.

If your preregistration request is successful, you see a page like the one shown in Figure 57, which tells you your temporary transaction ID.

---

**Preregistration successful**

Here's the temporary transaction ID so you may locate the preregistration record: 1j5NqaAnB1q=VkndWBef3kI=

**Press 'Preregister' to preregister another client using the same template.**

---

Figure 57. Successful preregistration request

---

Chapter 16. Using the end-user Web pages
Using the end-user Web pages

6. (Optional) Click the Examine Preregistration Record button.

7. (Optional) Click the Preregister button 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 the Home Page button) or return to the administration home page (by clicking the Administration Home Page button).
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 a SCEP client" on page 296.)
- 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 304) and a button for each end-user home page (see the default “PKI Administrators Start Page” in Figure 58 on page 300).
Using the administration Web pages

PKI Administrators Start Page

Install the CA certificate to enable SSL sessions for PKI Services

Choose one of the following:

Manage existing certificates and certificate requests

Go to the Customers' home page

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 the OK button. The following is an example of the popup window you might see, depending on the CA certificate you have installed:
Using the administration Web pages

b. Click the **Install certificate** button. (This initiates a series of pop-ups in which you need to click **Next** buttons and finally the **Finish** button, 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 Vista system” on page 507.

3. Click the **Go to administration page** button.

4. You will be prompted to authenticate, as shown in the following figure. Provide the necessary information:

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**.

---

*Figure 59. The certificate popup window for installing the CA certificate*

*Figure 60. Entering your user ID and password*
Using the administration Web pages

This calls up the “PKI Services Administration” Web page. (See Figure 61 on page 304)

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 on page 301 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 58. Summary of fields in the administration pages

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recent activity</td>
<td>This specifies a time range for searches. Possible values include:</td>
</tr>
<tr>
<td></td>
<td>• Not selected</td>
</tr>
<tr>
<td></td>
<td>• Within the past day</td>
</tr>
<tr>
<td></td>
<td>• Within the past week</td>
</tr>
<tr>
<td></td>
<td>• Within the past month</td>
</tr>
<tr>
<td></td>
<td>• Within the past six months</td>
</tr>
<tr>
<td>Requestor name</td>
<td>The name of the person requesting the certificate, as it appears in the</td>
</tr>
<tr>
<td></td>
<td>common name field of the certificate request form.</td>
</tr>
<tr>
<td>Serial number</td>
<td>PKI Services assigns this number to a certificate when you approve it.</td>
</tr>
<tr>
<td>Transaction ID</td>
<td>PKI Services assigns this number to a request when a user requests it.</td>
</tr>
<tr>
<td></td>
<td>This is a text field of up to 56 characters.</td>
</tr>
</tbody>
</table>

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 59. Statuses of certificate requests

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pending Approval</td>
<td>The request requires administrative approval. No action has been taken on the request yet.</td>
</tr>
<tr>
<td>Approved</td>
<td>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.</td>
</tr>
</tbody>
</table>
Using the administration Web pages

Table 59. Statuses of certificate requests (continued)

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>The certificate has been issued and the requestor has retrieved it. This is a final state.</td>
</tr>
<tr>
<td>Preregistered</td>
<td>The certificate request is from a preregistered Simple Certificate Enrollment Protocol (SCEP) client.</td>
</tr>
<tr>
<td>Rejected</td>
<td>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).</td>
</tr>
<tr>
<td>Rejected, User notified</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 60. Summary of actions to perform on requests and required status

<table>
<thead>
<tr>
<th>Action</th>
<th>Required status of request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approve</td>
<td>Pending Approval</td>
</tr>
<tr>
<td>Approve with modifications</td>
<td>Pending Approval</td>
</tr>
<tr>
<td>Reject</td>
<td>Pending Approval</td>
</tr>
<tr>
<td>Delete</td>
<td>All statuses (Pending Approval, Approved, Completed, Preregistered, Rejected, or Rejected, Notified)</td>
</tr>
</tbody>
</table>

Using the PKI Services administration home page

Figure 61 on page 304 shows the “PKI Services Administration” home page.
Using the administration Web pages

PKI Services Administration

Choose one of the following:

- **Work with a single certificate request**

  Enter the Transaction ID:
  
  ![Process Request]

- **Work with a single issued certificate**

  Enter the Serial Number:
  
  ![Process Certificate]

- **Specify search criteria for certificates and certificate requests**

  Certificate Requests
  - Show all requests
  - Show requests pending approval
  - Show approved requests
  - Show completed requests
  - Show rejected requests
  - Show rejections in which the client has been notified
  - Show preregistered requests

  Issued Certificates
  - Show all issued certificates
  - Show revoked certificates
  - Show suspended certificates
  - Show expired certificates
  - Show active certificates (not expired, not revoked, not suspended)
  - Show disabled certificates (suspended or revoked, not expired)
  - Show active, automatic renewal enabled certificates
  - Show active, automatic renewal disabled certificates
  - Show active, not renewable certificates

  Additional search criteria (Optional)
  - Requestor's name
  - Show recent activity only (Not Selected)
  - Show certificates that will expire (Not Selected)

  ![Find Certificates or Certificate Requests]

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 the **Process request** button. This displays the “Single Request” Web page shown in the example figure (Figure 62 on page 305).
Using the administration Web pages

**Single Request**

| Requestor: | W. M. Fortey | Created: | 2006/06/06 |
| Status: | Pending Approval | Modified: | 2006/06/06 |
| Transaction Id: | 5G9jzj2IGN2SHV+------- | Passphrase: | PICK9FAN |
| Template: | 1-Year PKI SSL Browser Certificate | Notify Email: | wmf@unfortey.com |

**Subject:** MAIL=wmfortey@wmfortey.com,CN=W. M. Fortey,OU=Class 1 Internet Certificate CA,O=The Firm

**Issuer:** OU=Master CA,O=IBM,C=US

**Validity:** 2006/06/06 00:00:00 - 2007/06/05 23:59:59

**Usage:** handshake, digitalSignature, keyEncipherment

**Extended Usage:** clientAuth

**Fingerprints:**

**Action to take:**

- Action Comment (Optional)

  - Approve Request As Is
  - Approve Request with Modifications
  - Reject Request
  - Delete Request

---

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 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 buttons on the “Single Request” Web page to process the request.
   - The Approve the request as is button
   - The Approve the request with modifications button
   - The Reject request button
   - The Delete request button
Using the administration Web pages

The buttons 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 buttons in the preceding list appear.
- If the administrator has already processed the request or if the request is Preregistered, only the Delete request button appears.

a. When you click the Approve the request as is button 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”)

```
Processing successful

Request with transaction ID 1jLjSxMOCqQU/whsWfE9Fr+ is successfully approved.

You may continue to approve/reject/delete more request(s) by clicking the button below:

Process More Request(s)
Administration Home Page

Figure 63. Processing successful Web page
```

From these Web pages, you can then click the Process more request(s) button to return to the PKI Services administration home page (Figure 61 on page 304).

b. When you click the Approve the request with modifications button, this displays a “Modify and Approve Request” Web page similar to the one shown in Figure 65 on page 307.

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.
```

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.)
On the "Modify and Approve Request" Web page, you can change the following fields.

- Common name, unless noted as in Figure 64 on page 306
- Organizational unit (this can be multiple fields), unless noted as in Figure 64 on page 306
- Organization, unless noted as in Figure 64 on page 306
- e-mail address

**Figure 65. Modifying the request Web page**

![Modify and Approve Request](image)
Using the administration Web pages

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 the Approve with specified modifications button; 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 275.)

C. When you click the Reject request button, this displays a Web page that informs you that “Processing is successful” or that “Processing is not successful”. From these Web pages, click the Process more request(s) button to return to the PKI Services administration home page. (See Figure 61 on page 304.)

d. When you click the Delete request button, this displays a Web page that informs you that “Processing is successful” or that “Processing is not successful”. On these Web pages, click the Process more request(s) button to return to the PKI Services administration home page. (See Figure 61 on page 304.)

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>
<thead>
<tr>
<th>Search criteria</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show all requests</td>
<td>Displays all certificate requests (all statuses (Pending Approval, Approved, Completed, Preregistered, Rejected, or Rejected, User Notified).</td>
</tr>
<tr>
<td>Show requests pending approval</td>
<td>Displays only certificate requests whose status is Pending Approval.</td>
</tr>
<tr>
<td>Show approved requests</td>
<td>Displays certificate requests whose status is Approved or Completed.</td>
</tr>
<tr>
<td>Show completed requests</td>
<td>Displays certificate requests whose status is Completed.</td>
</tr>
<tr>
<td>Show preregistered requests</td>
<td>Display certificate requests whose status is Preregistered.</td>
</tr>
<tr>
<td>Show all rejected requests</td>
<td>Displays certificate requests whose status is Rejected, or Rejected, User Notified.</td>
</tr>
<tr>
<td>Show rejections in which the client has been notified</td>
<td>Displays certificate requests whose status is Rejected, User Notified.</td>
</tr>
</tbody>
</table>
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 304), select one of the searches by clicking the appropriate radio button under “Certificate Requests”. (Table 61 on page 308 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 button. This displays the following Web page:

![Certificate Requests](image)

**Certificate Requests**

The following certificate requests matched the search criteria specified:

<table>
<thead>
<tr>
<th>Requestor</th>
<th>Certificate Request Information</th>
<th>Status</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>W. M. Forsey</td>
<td>Trans ID: 1H50jH4GQzS9y3V7+++++++ Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=<a href="mailto:wmforsey@wmforsey.com">wmforsey@wmforsey.com</a>,CN=W. M. Forsey,OU=Class 1 Internet Certificate CA,O=The Firm</td>
<td>Completed</td>
<td>Created: 2006/06/06 Modified: 2006/06/06</td>
</tr>
<tr>
<td>Jean Forsey</td>
<td>Trans ID: 1H50jH7QK1L2S9y3V7+++++++ Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=<a href="mailto:jeanforsey@wmforsey.com">jeanforsey@wmforsey.com</a>,CN=Jean Forsey,OU=Class 1 Internet Certificate CA,O=The Firm</td>
<td>Pending Approval</td>
<td>Created: 2006/06/06 Modified: 2006/06/06</td>
</tr>
<tr>
<td>scep client 1234</td>
<td>Trans ID: 1H50jH11d9r2S9y3V7+++++++ Template: 5-Year SCEP Certificate - Preregistration Subject: SCEP client 1234</td>
<td>Peregistered</td>
<td>Created: 2006/06/06 Modified: 2006/06/06</td>
</tr>
</tbody>
</table>

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](image)

**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
Using the administration Web pages

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 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 305.
   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 action buttons 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 buttons appear only if certificate requests
         are pending approval. Otherwise, only the Delete button 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 action buttons below the comment field to perform that
      action on all listed requests. The action buttons 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 buttons appear only if certificate requests
         are pending approval. Otherwise, only the Delete button appears.

         **Tip:** If you select the Show all requests radio button (see Figure 61 on page
         304) and click the Approve button 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 the
   Respecify your search criteria Web page button to return to the PKI Services
   administration home page (see Figure 61 on page 304) or the Home page
   button to return to the PKI Services home page (see Figure 36 on page 275).

4. After you click an action button, the next Web page is one of the following:
Using the administration Web pages

- “Processing successful” (see Figure 67)
- “Processing was not successful” (see Figure 68 on page 312)
- “Processing partially successful” (see Figure 69 on page 312)

If “Processing was not successful”, you can click on the transaction ID to display the “Single Request” Web page; see Figure 62 on page 305.

Processing can be unsuccessful because requests do not have the status required for the action you selected; see Table 60 on page 303.

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 305.

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.

---

**Processing successful**

Requests with specified transaction IDs are successfully approved.

You may continue to approve/reject/delete more request(s) by clicking the button below:

- Process More Request(s)
- Administration Home Page
- Home Page

emai1_wehmaster@your-company.com

*Figure 67. Request processing was successful Web page*
Using the administration Web pages

5. After approving requests as appropriate, you can:
   • Click Process more request(s) to return to Figure 66 on page 309
   • Click Administration home page to return to Figure 61 on page 304
   • Click Home page to return to Figure 36 on page 275

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 62. Status of certificates**

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>The certificate has not yet expired, has not been revoked, and is not currently suspended.</td>
</tr>
<tr>
<td>Active, AutoRenew</td>
<td>The certificate is eligible for automatic renewal and this capability is enabled.</td>
</tr>
<tr>
<td>Active, AutoRenewDisabled</td>
<td>The certificate is eligible for automatic renewal but this capability is disabled.</td>
</tr>
<tr>
<td>Active, NotRenewable</td>
<td>The certificate cannot be renewed.</td>
</tr>
<tr>
<td>Expired</td>
<td>The certificate’s validity period expired while it was active.</td>
</tr>
<tr>
<td>Revoked</td>
<td>The certificate has not expired but it has been revoked. Such certificates are published on the next certificate revocation list (CRL).</td>
</tr>
<tr>
<td>Revoked, Expired</td>
<td>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.</td>
</tr>
<tr>
<td>Suspended</td>
<td>The certificate has not expired but it is currently suspended. Such certificates are published on the next certificate revocation list (CRL).</td>
</tr>
</tbody>
</table>

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:
Using the administration Web pages

Table 63. Summary of actions to perform and required status to do so

<table>
<thead>
<tr>
<th>Action</th>
<th>Required status of certificate</th>
<th>Who performs action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Renew</td>
<td>Active</td>
<td>End user</td>
</tr>
<tr>
<td>Resume</td>
<td>Suspended</td>
<td>Administrator</td>
</tr>
<tr>
<td>Revoke</td>
<td>Active or Suspended</td>
<td>End user or administrator</td>
</tr>
<tr>
<td>Suspend</td>
<td>Active</td>
<td>End user or administrator</td>
</tr>
<tr>
<td>Delete</td>
<td>All statuses (Active, Expired, Suspended, Revoked, or Revoked, Expired)</td>
<td>Administrator</td>
</tr>
<tr>
<td>Enable automatic renewal</td>
<td>Active or Active,AutoRenewDisabled</td>
<td>Administrator</td>
</tr>
<tr>
<td>Disable automatic renewal</td>
<td>Active or Active,AutoRenew</td>
<td>Administrator</td>
</tr>
<tr>
<td>Change requestor e-mail</td>
<td>All statuses</td>
<td>Administrator</td>
</tr>
</tbody>
</table>

**Note:** This action applies only to certificates for which PKI Services generated the key pair.

**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 304), 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:
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 buttons 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...
Using the administration Web pages

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,AutoRenewDisabled**.

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 64. Searches to display certificates**

<table>
<thead>
<tr>
<th>Searches</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show all issued certificates.</td>
<td>Displays all certificates (can be any status).</td>
</tr>
<tr>
<td>Show revoked certificates.</td>
<td>Displays certificates whose status is <strong>Revoked</strong> or <strong>Revoked, Expired</strong>.</td>
</tr>
<tr>
<td>Show suspended certificates.</td>
<td>Displays certificates whose status is <strong>Suspended</strong>.</td>
</tr>
<tr>
<td>Show expired certificates.</td>
<td>Displays certificates whose status is <strong>Expired</strong> or <strong>Revoked, Expired</strong>.</td>
</tr>
<tr>
<td>Show active certificates (not expired, not revoked, not suspended).</td>
<td>Displays certificates whose status is <strong>Active</strong>.</td>
</tr>
<tr>
<td>Show disabled certificates (suspended or revoked, not expired).</td>
<td>Displays certificate requests whose status is <strong>Suspended or Revoked</strong>.</td>
</tr>
<tr>
<td>Show certificates with automatic renewal enabled.</td>
<td>Displays certificates whose status is <strong>Active,AutoRenew</strong>.</td>
</tr>
<tr>
<td>Show certificates with automatic renewal disabled.</td>
<td>Displays certificates whose status is <strong>Active,AutoRenewDisabled</strong>.</td>
</tr>
<tr>
<td>Show certificates that cannot be renewed.</td>
<td>Displays certificates whose status is <strong>Active,NotRenewable</strong>.</td>
</tr>
</tbody>
</table>

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 304), select one of the searches by clicking the appropriate radio button 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 the **Find certificates or certificate requests** button. This displays the following Web page.

### Issued Certificates

The following issued certificates matched the search criteria specified:

<table>
<thead>
<tr>
<th>All</th>
<th>Requestor</th>
<th>Certificate Information</th>
<th>Status</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
<td>Jan Renew</td>
<td>Serial #: 19002 Template: 1-Year PKI SSL Browser Certificate Subject: CN=Jan Renew,OU=Class 1 Internet Certificate, CA,O=The Firm</td>
<td>Active</td>
<td>Created: 2023/07/17, Modified: 2023/07/22</td>
</tr>
<tr>
<td>☑</td>
<td>W. M. Forsey</td>
<td>Serial #: 19022 Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=<a href="mailto:W.mforsey@sacompany.com">W.mforsey@sacompany.com</a>, CN=W. M. Forsey, OU=Class 1 Internet Certificate, CA,O=The Firm</td>
<td>Active</td>
<td>Created: 2023/07/21, Modified: 2023/07/22</td>
</tr>
<tr>
<td>☑</td>
<td>W. M. Forsey Jr.</td>
<td>Serial #: 19024 Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=<a href="mailto:W.mforseyjr@sacompany.com">W.mforseyjr@sacompany.com</a>, CN=W. M. Forsey Jr, OU=Class 1 Internet Certificate, CA,O=The Firm</td>
<td>Active</td>
<td>Created: 2023/07/21, Modified: 2023/07/22</td>
</tr>
<tr>
<td>☑</td>
<td>Doolie Dorbie</td>
<td>Serial #: 19025 Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=<a href="mailto:Doolie@sacompany.com">Doolie@sacompany.com</a>, CN=Doolie Dorbie, OU=Class 1 Internet Certificate, CA,O=The Firm</td>
<td>Active</td>
<td>Created: 2023/07/21, Modified: 2023/07/22</td>
</tr>
</tbody>
</table>

**Choose one of the following:**

- Click on a serial number to see more information or to revoke or delete certificates individually
- **Select and take action against multiple requests at once**

**Action Comment (Optional):**

- Revoking all selected active or suspended certificates
- Suspend all selected active certificates
- Delete all selected certificates

![Figure 71. Processing certificates using searches](image)

**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
Using the administration Web pages

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 315.

b. From the single certificate Web page, you can perform the steps in “Steps for processing a single certificate” on page 314, starting with Step 2 on page 315.

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 action buttons below the comment field to perform that action on all listed requests. The action buttons 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** buttons appear only when your search matches at least one certificate whose status is **Active**.
2) The **Resume** button 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 304) or **Home page** to return to the PKI Services home page (see Figure 36 on page 275).

4. After you click an action, the next Web page tells you:
   - “Processing was successful” (see Figure 72 on page 319)
Using the administration Web pages

- “Processing was not successful” (see Figure 73)
- “Processing partially successful” (see Figure 74 on page 320)

If “Processing was not successful”, you can click on a serial number to display the “Single Certificate” Web page; see Figure 70 on page 315. Processing can be unsuccessful because certificates do not have the status required for the action you selected; see Table 63 on page 314.

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 315. 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.

**Processing successful**

Certificate with specified serial numbers are successfully revoked.

You may continue to revoke/delete more certificate(s) by clicking the button below:

- Process More Certificate(s)
- Administration Home Page
- Home Page

email: webmaster@your-company.com

Figure 72. Processing of certificate was successful Web page

**Processing was not successful**

EY10021 EAF Service IG800200 Returned EAF DC = 8 ECAF DC = 8 ECIF RN = 72
The status(s) of the certificate(s) have been changed by other processes.

The following certificates could not be revoked because of a state change. Click on the links below for more information:

- Serial No.: JDD0021

You may continue to revoke/delete/suspend/resume more certificate(s) by clicking the button below:

- Process More Certificate(s)
- Administration Home Page
- Home Page

email: webmaster@your-company.com

Figure 73. Request processing was not successful Web page
Using the administration Web pages

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 275)

Relationship between certificate requests and matching certificates

PKI Services maintains two databases:

- The request database (RDB), also called the ObjectStore
- 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.

iclview A UNIX program that displays the entries in the VSAM issued certificate list (ICL) data set.

vosview A UNIX program that displays the entries contained in the VSAM ObjectStore data set (request database)

pkiprereg A UNIX program that creates Simple Certificate Enrollment Protocol (SCEP) preregistration records

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.
Using the iclview utility

Purpose

The iclview program displays the entries contained in a VSAM issued certificate list (ICL) data set. Each VSAM 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 the PKI Services PATH, LIBPATH, and NLSPATH environment variables with the appropriate pkiserv directory before you run iclview. Once you have updated the pkiserv.envars file, you can run iclview from the UNIX command line.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>You must add ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>/install-dir/pkiserv/bin</td>
</tr>
<tr>
<td>LIBPATH</td>
<td>/install-dir/pkiserv/lib</td>
</tr>
<tr>
<td>NLSPATH</td>
<td>/install-dir/pkiserv/lib/nls/msg/%L/%N</td>
</tr>
</tbody>
</table>

The default directory for install-dir is /usr/lpp.

Format

iclview {-d vsam-dataset-name [-r] | -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 execute the iclview utility command with no parameters.

-d vsam-dataset-name
   Specifies the MVS data set name of the VSAM issued certificate list (ICL).

   Notes:
   1. Make sure to include the escape character, which is a backslash (\), before the quotation marks enclosing the MVS data set name. For example, see \\pkisrvd.vsam.icl' in Examples on page 323.
   2. The -d and -c options are mutually exclusive.

-r
   Indicates to open in VSAM record-level sharing (RLS) mode the VSAM data set specified with the -d option.

-c
   Indicates to retrieve the data set name and RLS information from the pkiserv.conf configuration file.

   Notes:
   1. The -c and -d options are mutually exclusive.
   2. When you also specify the -D option, you must use the -p option to specify the CA domain configuration directory if not /etc/pkiserv.

-p path
   Specifies the directory where the pkiserv.conf configuration file resides. If not specified, the directory defaults to /etc/pkiserv.

-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 VSAM ICL data set 'PKISRVD.VSAM.ICL', passing the certificate to a utility called dumpasn1, use the following:

iclview -d 'pkisrvd.vsam.icl' -s 'dumpasn1 -'

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:

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
  Revoked at 2002-12-31 10:31:02
  Revocation Reason: Temporarily suspended
  Additional flags Set: NotRenewable
  KeyID: 12FD6897EE1F987DC9CA1440B62CCD1C0A9BB

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 114 for more information.)

Table 65. Names, aliases, and nicknames of certificate templates

<table>
<thead>
<tr>
<th>True name</th>
<th>Alias</th>
<th>Nickname</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Year PKI SSL Browser Certificate</td>
<td>PKI Browser Certificate</td>
<td>1YBSSL</td>
</tr>
<tr>
<td>1-Year PKI S/MIME Browser Certificate</td>
<td>PKI Browser Certificate</td>
<td>1YBSM</td>
</tr>
<tr>
<td>1-Year PKI Generated Key Certificate</td>
<td>PKI Key Certificate</td>
<td>1YKRC</td>
</tr>
<tr>
<td>2-Year PKI Browser Certificate For Authenticating To z/OS</td>
<td>PKI Browser Certificate</td>
<td>2YBZOS</td>
</tr>
<tr>
<td>2-Year PKI Authenticode—Code Signing Certificate</td>
<td>PKI Server Certificate</td>
<td>2YIACS</td>
</tr>
<tr>
<td>2-Year PKI Windows Logon Certificate</td>
<td>PKI Browser Certificate</td>
<td>2YBWL</td>
</tr>
<tr>
<td>5-Year PKI SSL Server Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSSSL</td>
</tr>
<tr>
<td>5-Year PKI IPSEC Server (Firewall) Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSIPS</td>
</tr>
<tr>
<td>5-Year PKI Intermediate CA Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSCA</td>
</tr>
<tr>
<td>5-Year SCEP Certificate - Preregistration</td>
<td>—</td>
<td>5YSCEPP</td>
</tr>
<tr>
<td>n-Year PKI Certificate for Extensions Demonstration</td>
<td>PKI Browser Certificate</td>
<td>SAMPLB</td>
</tr>
<tr>
<td>1-Year SAF Browser Certificate</td>
<td>SAF Browser Certificate</td>
<td>—</td>
</tr>
<tr>
<td>1-Year SAF Server Certificate</td>
<td>SAF Server Certificate</td>
<td>—</td>
</tr>
</tbody>
</table>

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.
vosview

Using the vosview utility

Purpose

The vosview program displays the entries contained in a VSAM ObjectStore data set (the request database). Each VSAM 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 the PKI Services PATH, LIBPATH, and NLSPATH environment variables with the appropriate pkiserv directory before you run vosview. Once you have updated the pkiserv.envars file, you can run vosview from the UNIX command line.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>You must add ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>/install-dir/pkiserv/bin</td>
</tr>
<tr>
<td>LIBPATH</td>
<td>/install-dir/pkiserv/lib</td>
</tr>
<tr>
<td>NLSPATH</td>
<td>/install-dir/pkiserv/lib/nls/msg/%L/%N</td>
</tr>
</tbody>
</table>

The default directory for install-dir is /usr/lpp.

Format

vosview {-d vsam-dataset-name [-r] | -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 execute the vosview utility command with no parameters.

-d vsam-dataset-name

Specifies the MVS data set name of the VSAM ObjectStore.

Notes:
1. Make sure to include the escape character, which is a backslash (\), before the quotation marks enclosing the MVS data set name. For example, see "\'pkisrvd.vsam.ost\'" in "Examples" on page 327.
2. The -d and -c options are mutually exclusive.

-r

Indicates to open in record-level sharing (RLS) mode the VSAM data set specified with the -d option.

-c

Indicates to retrieve the data set name and RLS information from the pkiserv.conf configuration file.

Notes:
1. The -c and -d options are mutually exclusive.
2. When you also specify the -D option, you must use the -p option to specify the CA domain configuration directory if not /etc/pkiserv.

-p path

When used with the -c option, specifies the directory where the pkiserv.conf configuration file resides. If not specified, the directory defaults to /etc/pkiserv.

-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 VSAM ObjectStore data set 'PKISRVD.VSAM.OST', passing the request data to a utility called dumpasn1, use the following:

vosview -d 'PKISRVD.VSAM.OST' -s 'dumpasn1 -'

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 114 for more information.)

table 66. Names, aliases, and nicknames of certificate templates

<table>
<thead>
<tr>
<th>True name</th>
<th>Alias</th>
<th>Nickname</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Year PKI SSL Browser Certificate</td>
<td>PKI Browser Certificate</td>
<td>1YBSSL</td>
</tr>
<tr>
<td>1-Year PKI S/MIME Browser Certificate</td>
<td>PKI Browser Certificate</td>
<td>1YBSM</td>
</tr>
<tr>
<td>1-Year PKI Generated Key Certificate</td>
<td>PKI Key Certificate</td>
<td>1YKRC</td>
</tr>
<tr>
<td>2-Year PKI Browser Certificate For Authenticating To z/OS</td>
<td>PKI Browser Certificate</td>
<td>2YBZOS</td>
</tr>
<tr>
<td>2-Year PKI Authenticode—Code Signing Certificate</td>
<td>PKI Server Certificate</td>
<td>2YIACS</td>
</tr>
<tr>
<td>2-Year PKI Windows Logon Certificate</td>
<td>PKI Browser Certificate</td>
<td>2YBWl</td>
</tr>
<tr>
<td>5-Year PKI SSL Server Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSSSL</td>
</tr>
<tr>
<td>5-Year PKI IPSEC Server (Firewall) Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSIPS</td>
</tr>
<tr>
<td>5-Year PKI Intermediate CA Certificate</td>
<td>PKI Server Certificate</td>
<td>5YSaca</td>
</tr>
<tr>
<td>5-Year SCEP Certificate - Preregistration</td>
<td>—</td>
<td>5YSCEPP</td>
</tr>
<tr>
<td>n-Year PKI Certificate for Extensions Demonstration</td>
<td>PKI Browser Certificate</td>
<td>SAMPLB</td>
</tr>
<tr>
<td>1-Year SAF Browser Certificate</td>
<td>SAF Browser Certificate</td>
<td>—</td>
</tr>
<tr>
<td>1-Year SAF Server Certificate</td>
<td>SAF Server Certificate</td>
<td>—</td>
</tr>
</tbody>
</table>

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. |
Using the pkiprereg utility

Purpose

The pkiprereg program creates Simple Certificate Enrollment Protocol (SCEP) preregistration records in batch.

Format

pkiprereg {-m mode [-t tmpl-file] -s SCEP-file [-o out-file] [-d domain] [-h | -?]}

Path setup

Update the PKI Services PATH, LIBPATH, and NLSPATH environment variables with the appropriate pkiserv directory before you run pkiprereg. Once you have updated the pkiserv.envars file, you can run pkiprereg from the UNIX command line.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>You must add ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>/install-dir/pkiserv/bin</td>
</tr>
<tr>
<td>LIBPATH</td>
<td>/install-dir/pkiserv/lib</td>
</tr>
<tr>
<td>NLSPATH</td>
<td>/install-dir/pkiserv/lib/nls/msg/%L/%N</td>
</tr>
</tbody>
</table>

The default directory for install-dir is /usr/lpp.

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.

<table>
<thead>
<tr>
<th>Mode name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>verify</td>
<td>Checks the data file for format errors but does not load the records into PKI Services.</td>
</tr>
<tr>
<td>generate</td>
<td>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.</td>
</tr>
<tr>
<td>load</td>
<td>Calls PKI Services to load the preregistration records.</td>
</tr>
<tr>
<td>remove</td>
<td>Calls PKI Services to remove the preregistration records.</td>
</tr>
</tbody>
</table>

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

-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.
pkipurereg

- `SCEP-file`
  Specifies the pathname of the file containing the preregistration data that is input for all modes.

- `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.

- `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 `pkipurereg` command.

**Input**

The input for the `pkipurereg` 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 `pkipurereg` `verify` and `load` modes, the Template and `ClientName` `name=value` pairs are required. All other `name=value` pairs are optional.
- In `pkipurereg` `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 67](#).
- 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>
<thead>
<tr>
<th>Field name</th>
<th>Maximum length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template</td>
<td>8 characters</td>
<td>The nickname of the preregistration template to be used.</td>
</tr>
<tr>
<td>Field name</td>
<td>Maximum length</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ClientName</td>
<td>64 characters</td>
<td>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.</td>
</tr>
<tr>
<td>PassPhrase</td>
<td>32 characters</td>
<td>The password to be communicated to the requestor. The &quot;*&quot; value can be used as a placeholder when running in <code>generate</code> mode.</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>64 characters</td>
<td>Serial number of the subject device.</td>
</tr>
<tr>
<td>UnstructAddr</td>
<td>64 characters</td>
<td>Unstructured address of the subject device.</td>
</tr>
<tr>
<td>EmailAddr</td>
<td>64 characters</td>
<td>E-mail address for the <code>EMAIL=</code> attribute for subject distinguished name.</td>
</tr>
<tr>
<td>Mail</td>
<td>64 characters</td>
<td>E-mail address for the <code>MAIL=</code> attribute for subject distinguished name.</td>
</tr>
<tr>
<td>DNQualifier</td>
<td>64 characters</td>
<td>Specifies information to add to the subject's distinguished name to make it unambiguous.</td>
</tr>
<tr>
<td>Uid</td>
<td>64 characters</td>
<td>The system login name associated with the subject.</td>
</tr>
<tr>
<td>Title</td>
<td>64 characters</td>
<td>Title for subject distinguished name.</td>
</tr>
<tr>
<td>DomainName</td>
<td>64 characters</td>
<td>One component of a domain name. For example, domain name <code>www.ibm.com</code> is represented by 3 components: <code>www</code>, <code>ibm</code> and <code>com</code>.</td>
</tr>
<tr>
<td>OrgUnit</td>
<td>64 characters</td>
<td>Organizational unit for subject distinguished name.</td>
</tr>
<tr>
<td>Org</td>
<td>64 characters</td>
<td>Organization for subject distinguished name.</td>
</tr>
<tr>
<td>Street</td>
<td>64 characters</td>
<td>Street for subject distinguished name.</td>
</tr>
<tr>
<td>Locality</td>
<td>64 characters</td>
<td>Locality for subject distinguished name.</td>
</tr>
<tr>
<td>StateProv</td>
<td>64 characters</td>
<td>State or province for subject distinguished name.</td>
</tr>
<tr>
<td>PostalCode</td>
<td>64 characters</td>
<td>Postal code for subject distinguished name.</td>
</tr>
<tr>
<td>Country</td>
<td>2 characters</td>
<td>Country abbreviation for subject distinguished name.</td>
</tr>
</tbody>
</table>
Table 67. List of valid field names for use in the preregistration record as input to the pkiprereg utility (continued)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Maximum length</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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:x: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: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](https://www.ibm.com).  

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 SYSCEPP 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 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, `pkinserv.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, `pkinserv.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.

### 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, `pkserv.envvars`.) Once you have updated these variables, you can run `TemplateTool` from the shell script or from Java.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>You must add ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td><code>/install-dir/pkserv/bin</code></td>
</tr>
<tr>
<td></td>
<td>The directory containing the <code>java</code> command.</td>
</tr>
<tr>
<td>CLASSPATH</td>
<td><code>/install-dir/pkserv/lib/pki_xml.jar</code></td>
</tr>
<tr>
<td></td>
<td>The default directory for <code>install-dir</code> is <code>/usr/lpp</code>.</td>
</tr>
</tbody>
</table>
TemplateTool

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.
Part 5. Administering security for PKI Services

This part explains how to administer security for PKI Services.

- Chapter 19, “RACF administration for PKI Services,” on page 341 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 341
- “Authorizing users for inquiry access” on page 341
- “Administering HostIdMappings extensions” on page 342
- “Locating your PKI Services certificates and key ring” on page 344
- “Establishing PKI Services as an intermediate CA” on page 346
- “Renewing your PKI Services CA and RA certificates” on page 348
- “Recovering a CA certificate profile” on page 351
- “Retiring and replacing the PKI Services CA private key” on page 354
- “R_PKIServ (IRRSPX00) callable service” on page 357
- “Using encrypted passwords for LDAP servers” on page 361
Chapter 19. 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 342
- "Locating your PKI Services certificates and key ring" on page 344
- "Establishing PKI Services as an intermediate CA" on page 346
- "Renewing your PKI Services CA and RA certificates" on page 348
- "Recovering a CA certificate profile" on page 351
- "Retiring and replacing the PKI Services CA private key" on page 354
- "R_PKIServ (IRRSPX00) callable service" on page 357
- "Using encrypted passwords for LDAP servers" on page 361.

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 16 on page 37 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.
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 16 on page 37.)

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)`

Administrating 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.
RACF administration for PKI Services

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 68. Information you need for locating your PKI Services certificates and key ring

<table>
<thead>
<tr>
<th>Information needed</th>
<th>Where to find this information</th>
<th>Record your value here</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca_label—The label of your CA certificate in RACF</td>
<td>See Table 10 on page 29</td>
<td></td>
</tr>
<tr>
<td>ra_label—The label of your RA certificate in RACF</td>
<td>See Table 10 on page 29</td>
<td></td>
</tr>
<tr>
<td>ca_ring—The PKI Services SAF key ring</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>daemon—The user ID for the PKI Services daemon</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>log_dsn—The data set name of the IKYSETUP log</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>cacert_dsn—The data set name of your CA certificate as exported from RACF</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
</tbody>
</table>

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 68.) 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: 2Q1JmZmDhZmjgdQm4GQ5NfSYUDDwUBA
   Status: HIGHTRUST
   Start Date: 2001/06/04 23:00:00
   End Date: 2020/01/01 22:59:59
b. Alternately, locate the CA certificate using its certificate label. (Get the label name from \texttt{ca\_label} in Table 68 on page 344.) Display its information by entering the following RACF command from a TSO command prompt.\[ \text{RACDCERT CERTAUTH LIST(LABEL('ca\_label'))} \]

The RACDCERT CERTAUTH LIST command produces the same output as the RACDCERT CHECKCERT (shown in Step \text{1a}) with the addition of information about any ring associations. For example:

\textbf{Sample output:}

\begin{verbatim}
Ring Associations:
  Ring Owner: PKISRVD
  Ring: >CAring<
\end{verbatim}

c. Examine the CA certificate information. If you are diagnosing errors, note the following:

\begin{itemize}
  \item The first line must indicate that this is a CERTAUTH certificate.
  \item Label must match your \texttt{ca\_label} value (as in the preceding table).
  \item If Serial Number is not equal to 00, this indicates that the certificate has been renewed or was issued by another certificate authority.
  \item If Issuer's Name differs from Subject's Name, this indicates that the certificate was issued by another certificate authority.
  \item Subject's Name must match the original value recorded for the PKI Services SUBJECTSDN in the IKYSETUP log.
  \item Private Key Type and Private Key Size must be present.
  \item Private Key Type indicates whether the key is a non-ICSF key (software key) or an ICSF key (ICSF or PCICC key).
  \item If Ring Associations are listed, ensure that an association is displayed for the daemon user ID as ring owner and your \texttt{ca\_ring} value (from Table 68 on page 344) as ring name.
\end{itemize}

2. Locate the CA key ring and examine its information.

a. Get the ring name from \texttt{ca\_ring} in Table 68 on page 344 and display its information by executing the following RACF command from a TSO command prompt:

\[ \text{RACDCERT ID(daemon) LISTRING(ca\_ring)} \]

\textbf{Sample output:}

\begin{verbatim}
Digital ring information for user PKISRVD:

\begin{tabular}{lcccc}
Certificate Label Name & Cert Owner & USAGE & DEFAULT \\
\hline
Local PKI CA & CERTAUTH & PERSONAL & YES \\
Local PKI RA & PKISRVD & PERSONAL & NO \\
\end{tabular}
\end{verbatim}
RACF administration for PKI Services

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 68 on page 344 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: 2QiJmZmHmZpmdOg4G7Qn9sYUDDwUBA
   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 Certificate 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>
<thead>
<tr>
<th>Information needed</th>
<th>Where to find this information</th>
<th>Record your value here</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacert_dsn—The data set name of the new PKI Services CA certificate.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ca_label—The label of your CA certificate in RACF</td>
<td>See Table 69 on page 29</td>
<td></td>
</tr>
<tr>
<td>export_dsn—The data set name of the root CA certificate as exported from RACF.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>temp_dsn—The name of the temporary data set to contain your new certificate request and returned certificate.</td>
<td>You decide this based on local data set naming conventions.</td>
<td></td>
</tr>
</tbody>
</table>

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).
RACF administration for PKI Services

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 70.

   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 349
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>
<thead>
<tr>
<th>Information needed</th>
<th>Where to find this information</th>
<th>Record your value here</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacert_dsn—The data set name of your renewed CA certificate as exported from RACF. (This data set is needed for recovery.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ca_label—The label of your CA certificate in RACF</td>
<td>See Table 10 on page 29</td>
<td></td>
</tr>
<tr>
<td>temp_dsn—The temporary data set to contain your new certificate request and returned certificate.</td>
<td>You decide this based on local data set naming conventions.</td>
<td></td>
</tr>
</tbody>
</table>

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 70.

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 71. Information you need for renewing your PKI Services RA certificate

<table>
<thead>
<tr>
<th>Information needed</th>
<th>Where to find this information</th>
<th>Record your value here</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca_label—The label of your CA certificate in RACF</td>
<td>See Table 10 on page 29</td>
<td></td>
</tr>
<tr>
<td>daemon—The user ID of the PKI Services daemon.</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>racert_dsn—The name of the data set to contain your new certificate request.</td>
<td>You decide this based on local data set naming conventions.</td>
<td></td>
</tr>
<tr>
<td>ra_label—The label of your RA certificate in RACF.</td>
<td>See Table 10 on page 29</td>
<td></td>
</tr>
</tbody>
</table>

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.
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-encoded data set that has PKCS #12 format. If the CA certificate profile in RACF is accidentally deleted, you can recover it from the backup data set.

Steps for recovering a CA certificate profile

**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>
<thead>
<tr>
<th>Information needed</th>
<th>Where to find this information</th>
<th>Record your value here</th>
</tr>
</thead>
<tbody>
<tr>
<td>backup_dsn—The name of the data set containing the backup copy of your original CA certificate and its private key</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>cacert_dsn—The data set name of your CA certificate as exported from RACF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ca_label—The label of your CA certificate in RACF</td>
<td>See Table 10 on page 29</td>
<td></td>
</tr>
<tr>
<td>ca_ring—The PKI Services SAF key ring</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>daemon—The user ID for the PKI Services daemon</td>
<td>See Table 16 on page 37</td>
<td></td>
</tr>
<tr>
<td>your-passphrase—The passphrase you used when backing up the private key</td>
<td>You specified this when running IKYSETUP.</td>
<td></td>
</tr>
</tbody>
</table>

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 348.

3. Stop and restart PKI Services.
2. Perform the following steps to update the RACF profile with the serial number of the last certificate PKI Services issued. (You need to restore the certificate serial number incrementer value that is stored in the profile, because otherwise PKI Services resumes issuing certificates starting from serial number 1.)

   a. Execute the following command from the UNIX command line to run the iclview utility:

      iclview -d \'pkisrvd.vsam.icl\'

      Record the serial number displayed (in hex) of the last certificate listed:

      | Serial number (in hex) of last certificate: |
      |---------------------------------------------|

   b. To determine your CA certificate's profile name, you need the data set, cacert_dsn, that contains your CA certificate. If you don't have a copy, create one with this command:

      RACDCERT CERTAUTH EXPORT(LABEL('ca_label')) DSN(cacert_dsn)

      Then issue the following command to perform an unsuccessful ADD:

      RACDCERT CERTAUTH ADD(cacert_dsn) WITHLABEL('*** Bad Label ***')

      The unsuccessful ADD displays an error message including the profile name. For example:

      IRRD109I The certificate cannot be added. Profile 0A.OU=Human¢Resources¢Certificate¢Authority.O=IBM.C=US is already defined.

      Note: The profile name of a certificate is in the format of serial number.issuer's distinguished name. The issuer's distinguished name uses the character X'4A', (which is a “¢” on US English code pages) in place of a blank character. The profile name does not contain the subject's distinguished name.

      Record the profile name:

      | Profile name: |
      |---------------|

   c. Create the ICHEINTY ALTER job shown in Figure 75 on page 353 in your own JCL data set, replacing the highlighted values based on the information you recorded in the previous steps. Make sure that all blank characters have the value X'4A'. You can turn on the HEX ON option to display the hexadecimal values when you edit the job using the ISPF editor.

   d. Submit the job and make sure that its return code is 0.
RACF administration for PKI Services

Figure 75. Sample JCL data set for restoring the certificate serial number incrementer value (Part 1 of 2)
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.
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 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. 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('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('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('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](https://www.ibm.com) 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.
RACF administration for PKI Services

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 73. Summary of access authorities required for PKI Services requests

<table>
<thead>
<tr>
<th>Request</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPORT</td>
<td>• IRR.DIGTCERT.EXPORT</td>
</tr>
<tr>
<td></td>
<td>- READ access if PassPhrase is specified or if CertID is specified as</td>
</tr>
<tr>
<td></td>
<td>PKICACERT.</td>
</tr>
<tr>
<td></td>
<td>- UPDATE access if the PassPhrase parameter is not specified with</td>
</tr>
<tr>
<td></td>
<td>IRRSPX00.</td>
</tr>
<tr>
<td></td>
<td>- CONTROL access if you want to export a PKCS #7 certificate.</td>
</tr>
<tr>
<td>GENCERT</td>
<td>• IRR.DIGTCERT.GENCERT — CONTROL access</td>
</tr>
<tr>
<td></td>
<td>• IRR.DIGTCERT.ADD</td>
</tr>
<tr>
<td></td>
<td>- UPDATE access if any hostIdMapping information is specified in the</td>
</tr>
<tr>
<td></td>
<td>certificate request parameter list or the UserId field in the</td>
</tr>
<tr>
<td></td>
<td>certificate request parameter list indicates the certificate is</td>
</tr>
<tr>
<td></td>
<td>being requested for another user other than the caller</td>
</tr>
<tr>
<td></td>
<td>- READ access otherwise</td>
</tr>
<tr>
<td>GENRENEW</td>
<td>• IRR.DIGTCERT.GENRENEW — READ access</td>
</tr>
<tr>
<td></td>
<td>• IRR.DIGTCERT.GENCERT — CONTROL access</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: It is assumed that the calling application has already verified the input certificate using the VERIFY function.</td>
</tr>
<tr>
<td>QRECOVER</td>
<td>• IRR.DIGTCERT.QRECOVER — READ access</td>
</tr>
<tr>
<td>REQCERT</td>
<td>• IRR.DIGTCERT.REQCERT — READ access</td>
</tr>
<tr>
<td>REQRENEW</td>
<td>• IRR.DIGTCERT.REQRENEW — READ access</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: It is assumed that the calling application has already verified the input certificate using the VERIFY function.</td>
</tr>
<tr>
<td>RESPOND</td>
<td>• IRR.DIGTCERT.RESPOND — READ access</td>
</tr>
</tbody>
</table>
RACF administration for PKI Services

Table 73. Summary of access authorities required for PKI Services requests (continued)

<table>
<thead>
<tr>
<th>Request</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>REVOKE</td>
<td>• IRR.DIGTCERT.REVOKE — READ access</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> It is assumed that the calling application has already verified the target certificate using the VERIFY function.</td>
</tr>
<tr>
<td>SCEPREQ</td>
<td>• IRR.DIGTCERT.SCEPREQ — READ access</td>
</tr>
<tr>
<td>VERIFY</td>
<td>• IRR.DIGTCERT.VERIFY — READ access</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> It is assumed that the calling application has already verified that the end user possesses the private key that correlates to the input certificate.</td>
</tr>
</tbody>
</table>

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.
- 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 (PREREJISTER, 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
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 357. 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 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 85.)

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(0E23528875DECFAC)
   ```

   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:

<table>
<thead>
<tr>
<th>CLASS</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>FACILITY</td>
<td>IRR.PROXY.DEFAULTS</td>
</tr>
</tbody>
</table>

PROXY INFORMATION
LDAPHOST= LDAP://SOME.LDAP.HOST:389
BINDDN= CN=LDAP ADMINISTRATOR,OU=POUGHKEEPSIE,O=IBM,C=US
BINDPW= YES
RACF administration for PKI Services
Part 6. Using the certificate validation service

This part explains how to implement the PKI Services Trust Policy (PKITP) plug-in for OCSF.

- Chapter 20, “PKI Services Trust Policy (PKITP),” on page 367 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.
Chapter 20. 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.
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.)

Figure 76. Examples of organizations, certificates, and chains
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 "$y" (As Figure 76 on page 368 shows, "$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 188.)

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 policies 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 policies. 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 74 on page 370.
Table 74. Sequence of validation stages for PKITP certificate revocation checking

<table>
<thead>
<tr>
<th>Validation stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCSP responder</td>
<td>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.</td>
</tr>
<tr>
<td>DP CRL, using the URI format</td>
<td>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.</td>
</tr>
<tr>
<td>DP CRL, using the distinguished-name format</td>
<td>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.</td>
</tr>
<tr>
<td>Global revocation list</td>
<td>The trust policy uses the global CRL to find revocation status information for the certificate.</td>
</tr>
</tbody>
</table>

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.
- **CRLDistributionPoints**: See "Checking certificate status with PKITP" on page 369.
- **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>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Source location (default)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makefile.pkitpsamp</td>
<td>Makefile for pkitpsamp.c.</td>
<td>/usr/lpp/pkiserv/samples/</td>
</tr>
<tr>
<td>install_pkitp</td>
<td>Program that registers the PKI Services Trust Policy plug-in with OCSF.</td>
<td>/usr/lpp/pkiserv/bin</td>
</tr>
<tr>
<td>pkitp_ivp</td>
<td>This program verifies that the plug-in installed successfully.</td>
<td>/usr/lpp/pkiserv/bin</td>
</tr>
<tr>
<td>pkitp.h</td>
<td>Contains #define definitions for applications calling the PKI Services OCSF Trust Policy.</td>
<td>/usr/lpp/pkiserv/include/</td>
</tr>
<tr>
<td>pkitp.so</td>
<td>This is the OCSF Trust Policy plug-in for PKI Services.</td>
<td>/usr/lpp/pkiserv/lib</td>
</tr>
<tr>
<td>pkitpsamp.c</td>
<td>Sample application program (in the C language) to call the PKI Trust Policy plug-in.</td>
<td>/usr/lpp/pkiserv/samples</td>
</tr>
</tbody>
</table>

**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 22, 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:
   ```bash
   /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:

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>addin directory?</td>
<td>/usr/lpp/pkiserv/lib</td>
</tr>
<tr>
<td>addin filename?</td>
<td>pkitp.so</td>
</tr>
<tr>
<td>action? [install I uninstall]</td>
<td>install</td>
</tr>
</tbody>
</table>

   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.
   ```bash
   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-F0BE0FB8F9}`. (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; /*0LIC*/
    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:

```c
typedef struct tp_initialpolicy {
   /* initial-policy-set */
   uint32 NumberOfPolicyIdentifiers;  // number of application specific
   CSSM_OID_PTR PolicyIdentifiers;    // address of array of policy OIDs
   /* 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.)

```c
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 76 lists the error codes that are unique to PKI Services OCSF Trust Policy (PKITP).

<table>
<thead>
<tr>
<th>Decimal value</th>
<th>Error description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8002</td>
<td>Certificate policies violation.</td>
</tr>
<tr>
<td>8003</td>
<td>Incorrect certificate distinguished name chaining.</td>
</tr>
<tr>
<td>8004</td>
<td>Certificate encoding error. Subject name missing.</td>
</tr>
<tr>
<td>8006</td>
<td>Incorrect certificate BasicConstraints extension—cA flag off in signing certificate.</td>
</tr>
<tr>
<td>8008</td>
<td>Incorrect certificate KeyUsage extension—keyCertSign flag off in signing certificate.</td>
</tr>
<tr>
<td>8010</td>
<td>Unsupported AltName form in certificate.</td>
</tr>
<tr>
<td>8013</td>
<td>Certificate or CRL encoding error. Signature algorithm mismatch.</td>
</tr>
<tr>
<td>8014</td>
<td>Certificate encoding error. Incorrect version.</td>
</tr>
<tr>
<td>8015</td>
<td>CRL encoding error. Incorrect version.</td>
</tr>
<tr>
<td>8016</td>
<td>Unsupported critical extension in certificate.</td>
</tr>
<tr>
<td>8017</td>
<td>Unsupported critical extension in CRL.</td>
</tr>
<tr>
<td>8018</td>
<td>Unsupported critical entry extension in CRL.</td>
</tr>
<tr>
<td>8019</td>
<td>Certificate encoding error. Duplicate extension.</td>
</tr>
<tr>
<td>8020</td>
<td>CRL encoding error. Duplicate extension.</td>
</tr>
<tr>
<td>8021</td>
<td>Certificate signature failed verification.</td>
</tr>
<tr>
<td>8022</td>
<td>CRL signature failed verification.</td>
</tr>
<tr>
<td>8023</td>
<td>Incorrect date range in certificate or CRL. NotAfter earlier than NotBefore.</td>
</tr>
<tr>
<td>8024</td>
<td>Certificate’s date range is in the future.</td>
</tr>
<tr>
<td>8025</td>
<td>Certificate has expired.</td>
</tr>
<tr>
<td>8026</td>
<td>CRL’s date range is in the future.</td>
</tr>
<tr>
<td>8027</td>
<td>CRL has expired.</td>
</tr>
<tr>
<td>8028</td>
<td>DBList incorrect, no LDAPDL DBs or non-LDAPDL specified.</td>
</tr>
<tr>
<td>8029</td>
<td>CRL not found.</td>
</tr>
<tr>
<td>8030</td>
<td>Certificate is revoked.</td>
</tr>
<tr>
<td>8031</td>
<td>Unable to build certificate chain.</td>
</tr>
<tr>
<td>8033</td>
<td>Certificate not trusted.</td>
</tr>
<tr>
<td>8034</td>
<td>Incorrect CRLDistributionPoints extension in certificate.</td>
</tr>
<tr>
<td>8051</td>
<td>Unexpected status code returned from accessing LDAPDL.</td>
</tr>
<tr>
<td>8052</td>
<td>Unexpected status code returned from accessing OCEPDL.</td>
</tr>
<tr>
<td>8053</td>
<td>DBList incorrect, no OCEPDL DB or DB empty.</td>
</tr>
</tbody>
</table>
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 373), 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 377 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 377), 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.
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
#define INITIALExplicitPolicy FALSE
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,0x03,0x02,0x01}; // 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.

#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;
#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

CSSM_TP_PASS_THROUGH

Chapter 20. PKI Services Trust Policy (PKITP)
LDAPUSERPASSWORD*; // password

#define USECRLS 1

#define NUM_POLICIES 2

#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

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 rc;
    rc = connectTP(keyring,NUM_LDAPS, &datasources, &tphandle); //D1C
    if (rc == 0) {
        CSSM_TP_PassThrough
    }
}
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);

int connectTP(char * ringname, int number_ldap, CSSM_DL_DB_LIST * datasources_ptr, CSSM_TP_HANDLE * tphandle_ptr) {
  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_MODULEINFO* 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) {
    printf("connectTP unable to obtain memory: line %d
",__LINE__); // @D3A
    return -1; // @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)
  {
    print("Failed CSSM_Init: line %d
",CSSM_GetError()->error,LINE__);
    return -1;
  }

  memset(voidptr,0,(sizeof(CSSM_DL_DB_HANDLE)+(number_ldap +1))); // zero it
  datasources_ptr->DLDBHandle = (CSSM_DL_DBHANDLE *)voidptr;
  if (CSSM_Init(&cssm_version, &memoryFuncs, NULL) != CSSM_OK)
  {
    print("Failed CSSM_Init: line %d
",CSSM_GetError()->error,LINE__);
    return -1;
  }

  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_ATTRIB);
    if (!moduleInfoPtr)
    {
      print("Failed to get module info: line %d
",LINE__);
      return -1;
    }
  }

  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);
}
printf("Failed CSSM_GetModuleInfo: %d, line %d\n",CSSM_GetError()->error, __LINE__); return -1;
}

LDAP_dlhandle = CSSM_ModuleAttach((CSSM_GUID*)&LDAPDL_GUID, &moduleInfoPtr->Version, AmemoryFuncs, 0, 0, 0, NULL, 0, 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 specific data structure: LDAP BIND PARAMS
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 PARAMS
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

// end of for 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, AmemoryFuncs, 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[0].DLHandle)
{
  if ((status = CSSM_ModuleDetach(datasources_ptr->DLDBHandle[0].DLHandle)) != 0)
  {
    printf("Failed CSSM_ModuleDetach: %d, line %d\n", status, __LINE__);
    // we continue trying to close other stuff
    datasources_ptr->DLDBHandle[0].DLHandle = 0; // clear handle
  }
#endif //@D1A

#!/****************************************************************************
// Say goodbye to DCEP
#****************************************************************************/
status = CSSM_DL_DbClose(datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1]);
if (status != 0)
{
  printf("Failed CSSM_DL_DbClose %d, line %d\n", status, __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", status, __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", status, __LINE__);
    // we continue trying to close other stuff
  }
}
return;

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) // failed to open cert file
      printf("File %s could not be opened\n", certFile[i]);
    if (i > 0) // if we've read any certs before this
      freeCertGroup(certGroupPtr); // free alloc'd storage
    if (!inFile) // if cert file is not an error
      printf("File %s could not be opened\n", certFile[i]);
    if (i > 0) // if we've read any certs before this
      freeCertGroup(certGroupPtr); // free alloc'd storage
    return(CSSM_FAIL);
    // if file is not an error
    seek(inFile, 0L, SEEK_END);
    certSize = ftell(inFile);
    rewind(inFile);
    /* Read in certificate data*/
    certArray[i].Length = certSize;
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; //@D1A
    #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.
    // CHandle, PolicyIdentifiers and NumberoPolicyIdentifiers
    // (not to be confused with fields of same name in TP_INITIALPOLICY structure),
    // AnchorCerts and NumberoAnchorCerts.
    // 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);
}

Chapter 20. PKI Services Trust Policy (PKITP) 385
// 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;
}
CSSM_TP_PassThrough

| End of Programming Interface information |

Chapter 20. PKI Services Trust Policy (PKITP)  387
CSSM_TP_PassThrough
Part 7. Troubleshooting

This part explains using logs for troubleshooting, including the following:

- Chapter 21, “Using information from SYS1.LOGREC,” on page 391 discusses SYS1.LOGREC, which is used to record unusual runtime events, such as an exception.
- Chapter 22, “Using information from the PKI Services logs,” on page 397 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 322 and “Using the vosview utility” on page 326.
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>
<thead>
<tr>
<th>CSECT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IKYP0N</td>
<td>Issued when an ABEND occurs in the one of the CSECTs running on the Monitor Thread.</td>
</tr>
<tr>
<td>IKYP81</td>
<td>Primary symptom string:</td>
</tr>
<tr>
<td>IKYP8A</td>
<td>Component ID (PIDS): 5752XXPKI</td>
</tr>
<tr>
<td>IKYP8B</td>
<td>Load module:                   IKYPKID#L</td>
</tr>
<tr>
<td>CSECT:</td>
<td>CSECT: IKYP0N, IKYP81, IKYP8A, or IKYP8B</td>
</tr>
<tr>
<td>Recovery routine:</td>
<td>ESTEXIT</td>
</tr>
<tr>
<td>Error Information:</td>
<td>Consists of an abend code and reason code:</td>
</tr>
<tr>
<td>Abend code:</td>
<td>The character S followed by 4 hexadecimal digits or the character U followed by 4 decimal digits.</td>
</tr>
<tr>
<td>Reason code:</td>
<td>8 hexadecimal digits.</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2001, 2009
### Table 77. LOGREC data for PKI Services (continued)

<table>
<thead>
<tr>
<th>CSECT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IKYP8A</td>
<td>Issued when an exception is caught in the service thread routine IKYP8A01 or in the services thread request routine IKYP8A02.</td>
</tr>
<tr>
<td></td>
<td><strong>Primary symptom string:</strong></td>
</tr>
<tr>
<td></td>
<td>Component ID (PIDS): 5752XXPKI</td>
</tr>
<tr>
<td></td>
<td>Load module: IKYPKID#L</td>
</tr>
<tr>
<td></td>
<td>CSECT: IKYP8A</td>
</tr>
<tr>
<td></td>
<td>Failing routine: IKYP8A01 or IKYP8A02</td>
</tr>
<tr>
<td></td>
<td>Error information: Consists of <em>either</em> an abend code and a reason code <em>or</em> a facility ID and a message number.</td>
</tr>
<tr>
<td></td>
<td><strong>Abend code:</strong> If present, either the character <em>U</em> followed by 4 decimal digits or the character <em>S</em> followed by 3 hexadecimal digits.</td>
</tr>
<tr>
<td></td>
<td><strong>Reason code:</strong> If present, 8 hexadecimal digits.</td>
</tr>
<tr>
<td></td>
<td><strong>Facility ID:</strong> If present, 3 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Message number:</strong> If present, 8 hexadecimal digits.</td>
</tr>
<tr>
<td></td>
<td><strong>Secondary symptom string:</strong></td>
</tr>
<tr>
<td></td>
<td>USER The user ID of the requestor.</td>
</tr>
<tr>
<td></td>
<td>FUNC A function code of 8 hexadecimal digits.</td>
</tr>
<tr>
<td></td>
<td>IKYP8B Issued when an ABEND occurs in the PC routine (or helper routines).</td>
</tr>
<tr>
<td></td>
<td><strong>Primary symptom string:</strong></td>
</tr>
<tr>
<td></td>
<td>Component ID (PIDS): 5752XXPKI</td>
</tr>
<tr>
<td></td>
<td>Load module: IKYPKID#L</td>
</tr>
<tr>
<td></td>
<td>CSECT: IKYP8B</td>
</tr>
<tr>
<td></td>
<td>Recovery routine: ARREXIT</td>
</tr>
<tr>
<td></td>
<td>Error information: Consists of an abend code and a reason code.</td>
</tr>
<tr>
<td></td>
<td><strong>Abend code:</strong> The character <em>S</em> followed by 4 hexadecimal digits or the character <em>U</em> followed by 4 decimal digits.</td>
</tr>
<tr>
<td></td>
<td><strong>Reason code:</strong> 8 hexadecimal digits.</td>
</tr>
</tbody>
</table>
### Using information from SYS1.LOGREC

**Table 77. LOGREC data for PKI Services (continued)**

<table>
<thead>
<tr>
<th>CSECT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IKYSCHDR</td>
<td>Issued from the <strong>dispatcher</strong> function when an exception is caught while creating and posting a CRL to LDAP.</td>
</tr>
</tbody>
</table>

**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 DISPATCH.

| 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.
## Using information from SYS1.LOGREC

### Table 77. LOGREC data for PKI Services (continued)

<table>
<thead>
<tr>
<th>CSECT</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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:
Using information from SYS1.LOGREC

TYPE: SYMPTOM RECORD  REPORT: SOFTWARE EDIT REPORT  DAY YEAR
SCP: VS 2 REL 3  REPORT DATE: 221 01
MODEL: 9672  ERROR DATE: 221 01
SERIAL: 048288  HH MM SS.TH
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  6422253  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 4CB31800 00000000 0001221F 19051602 |<C..........|
+010 FF0482BB 96720000 ..BHO...

Figure 77. Sample LOGREC data (Part 1 of 2)
### Figure 77. Sample LOGREC data (Part 2 of 2)
Chapter 22. 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 484 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 78 on page 398** shows this.
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 79 on page 399 shows this:
Using information from the PKI Services logs

Figure 79. Selecting a file to view

Figure 80 on page 400 shows the messages contained in the file:
Using information from the PKI Services logs

Figure 80. 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 23, "Messages," on page 405.) 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 IKYC0261 Deleting inactive object 37. L
Last changed at 2001/07/10 17:08:39
Using information from the PKI Services logs

_DIRECTORY_ subcomponents and message levels

The _DIRECTORY_ environment variable specifies the subcomponent and message level for logging messages.

The subcomponents are listed below:

<table>
<thead>
<tr>
<th>Subcomponent</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>The wildcard character (represents all subcomponents)</td>
</tr>
<tr>
<td>CORE</td>
<td>The core functions of PKI Services that are not specific to the other subcomponents</td>
</tr>
<tr>
<td>DB</td>
<td>Activity related to the request or issued certificate VSAM data stores</td>
</tr>
<tr>
<td>LDAP</td>
<td>LDAP posting operations</td>
</tr>
<tr>
<td>PKID</td>
<td>The PKI Services daemon address setup and infrastructure</td>
</tr>
<tr>
<td>POLICY</td>
<td>Certificate creation and revocation policy processing</td>
</tr>
<tr>
<td>SAF</td>
<td>SAF key ring, OCEP, and R_dataLib calls</td>
</tr>
<tr>
<td>TPOLICY</td>
<td>Trust policy plug-in processing</td>
</tr>
</tbody>
</table>

The message levels are listed hierarchically below:

<table>
<thead>
<tr>
<th>Debug level</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>This indicates logging only severe messages.</td>
</tr>
<tr>
<td>E</td>
<td>This indicates logging severe and error messages.</td>
</tr>
<tr>
<td>W</td>
<td>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.</td>
</tr>
<tr>
<td>I</td>
<td>This indicates logging severe, error, warning, and informational messages.</td>
</tr>
<tr>
<td>D</td>
<td>This indicates logging severe, error, warning, informational, and diagnostic messages.</td>
</tr>
<tr>
<td>V</td>
<td>This indicates logging all messages, including verbose diagnostic messages. This is very verbose.</td>
</tr>
</tbody>
</table>

**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 49.)

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...]

Chapter 22. Using information from the PKI Services logs 401
Using information from the PKI Services logs

**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:  B4 6F EB 2E E0 96 9D 05 32 84 E3 2E D0 3C 10 6F 98 F5 43 3C
    MD5:   C2 15 14 AC 12 B1 59 46 09 F6 35 ED FB B6 CF 31
```

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 222.
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 23, “Messages,” on page 405** explains PKI Services messages.
- **Chapter 24, “File directory structure,” on page 437** describes product and file system directories for PKI Services and files contained in them.
- **Chapter 26, “Environment variables,” on page 445** explains the `pkiserv.envars` environment variables file and provides a code sample.
- **Chapter 27, “The IKYSETUP REXX exec,” on page 449** explains the contents of the `IKYSETUP REXX` exec that performs RACF administration and provides a code sample.
- **Chapter 28, “Other code samples,” on page 471** provides additional code samples. **Table 78** 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 29, “SMF recording,” on page 485** describes the content of the System Management Facility (SMF) record that is generated by PKI Services.

**Table 78. Summary of information about important files**

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Source location (default)</th>
<th>For code sample...</th>
</tr>
</thead>
<tbody>
<tr>
<td>httpd.conf</td>
<td>Contain z/OS HTTP Server directives.</td>
<td>/usr/lpp/pkiserv/samples/</td>
<td>See page 471</td>
</tr>
<tr>
<td>httpd2.conf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IKYCVSAM</td>
<td>Sample IDCAMS JCL to create VSAM data sets (regardless of whether you are</td>
<td>SYS1.SAMPLIB</td>
<td>See page 473</td>
</tr>
<tr>
<td></td>
<td>using a sysplex or non-sysplex).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IKYRVSAM</td>
<td>Sample IDCAMS JCL to add VSAM record-level sharing (RLS) support.</td>
<td>SYS1.SAMPLIB</td>
<td>See page 477</td>
</tr>
<tr>
<td></td>
<td>IKYRVSAM reallocates your VSAM data sets in preparation for sharing in a</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>sysplex.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IKYSETUP</td>
<td>REXX exec to set up RACF profiles.</td>
<td>SYS1.SAMPLIB</td>
<td>See page 449</td>
</tr>
<tr>
<td>pkiserv.conf</td>
<td>PKI Services configuration file.</td>
<td>/usr/lpp/pkiserv/samples/</td>
<td>See page 441</td>
</tr>
<tr>
<td></td>
<td>(You copy this file to the runtime directory, /etc/pkiserv.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PKISERVD</td>
<td>Sample procedure to start PKI Services daemon.</td>
<td>SYS1.PROCLIB</td>
<td>See page 484</td>
</tr>
<tr>
<td>pkiserv.envars</td>
<td>PKI Services environment variables file.</td>
<td>/usr/lpp/pkiserv/samples/</td>
<td>See page 448</td>
</tr>
<tr>
<td></td>
<td>(You might need to copy this file to the runtime directory, /etc/pkiserv.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pkiserv.tmpl</td>
<td>PKI Services certificate template file for REXX CGI execs.</td>
<td>/usr/lpp/pkiserv/samples/</td>
<td>Not provided.</td>
</tr>
<tr>
<td></td>
<td>(You copy this file to the runtime directory, /etc/pkiserv.)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 78. Summary of information about important files (continued)

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Source location (default)</th>
<th>For code sample...</th>
</tr>
</thead>
<tbody>
<tr>
<td>pkitmpl.xml</td>
<td>PKI Services certificate template file for Java server pages.</td>
<td>/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)</td>
<td>Not provided.</td>
</tr>
<tr>
<td>pkitmpl.xml</td>
<td>PKI Services certificate template file for Java server pages.</td>
<td>/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)</td>
<td>Not provided.</td>
</tr>
<tr>
<td>pkixgen.tmpl</td>
<td>PKI Services certificate template file for the daemon's use when you implement the Web application using Java server pages.</td>
<td>You create this file from pkitmpl.xml using the TemplateTool utility.</td>
<td>Not provided.</td>
</tr>
</tbody>
</table>
Chapter 23. 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 79. Meaning of fourth character in message number

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
<th>Component producing messages</th>
<th>Where messages appear</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>CORE</td>
<td>Core subcomponent</td>
<td>PKI Services log</td>
</tr>
<tr>
<td>D</td>
<td>DB</td>
<td>Database accessing subcomponent</td>
<td>PKI Services log</td>
</tr>
<tr>
<td>I</td>
<td>INTERFACE</td>
<td>PKISERV CGIs or JSPs</td>
<td>In the user’s Web browser window</td>
</tr>
<tr>
<td>L</td>
<td>LDAP</td>
<td>LDAP bind subcomponent</td>
<td>PKI Services log</td>
</tr>
<tr>
<td>O</td>
<td>POLICY</td>
<td>Certificate creation and revocation policy subcomponent</td>
<td>PKI Services log</td>
</tr>
<tr>
<td>P</td>
<td>PKID</td>
<td>PKI Services daemon address space controller</td>
<td>PKI Services log</td>
</tr>
<tr>
<td>S</td>
<td>SAF</td>
<td>SAF interfacing subcomponent</td>
<td>PKI Services log</td>
</tr>
<tr>
<td>U</td>
<td>UTILITY</td>
<td>Utility programs</td>
<td>UNIX standard error (stderr)</td>
</tr>
</tbody>
</table>

Characters five through seven are numeric. The eighth character is the message type:

Table 80. Meaning of eighth character in message number

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
<th>Action required</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Informational (status message)</td>
<td>No action required</td>
</tr>
<tr>
<td>E</td>
<td>Eventual action</td>
<td>Possible problem that might require eventual action</td>
</tr>
<tr>
<td>A</td>
<td>Action required</td>
<td>Problem that requires immediate attention</td>
</tr>
</tbody>
</table>

For information about setting messages options using environment variables, see 446.

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...
Messages

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 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 81 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 22, "Using information from the PKI Services logs," on page 397.

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 52.

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 52.

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 52.

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 52

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 82.

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. 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.
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 YYYYYMMDD 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.

System programmer 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 302.

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.
Messages

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.


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.

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 52

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 52

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 52
IKYC037I Bad RemoveExpiredCerts or RemoveExpiredCertsAndKeys value in pkiserv.conf file

Explanation: PKI Services is initializing and 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 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 52.

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 52.

IKYC040I Bad CRLDistURI value in pkiserv.conf file

Explanation: PKI Services is attempting to purge the issued certificate list (ICL). 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 52.

IKYC041I Bad CRLDistURI value in pkiserv.conf file, exceeds the number of LDAP servers: LdapServer

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 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 52.

IKYC042I Bad CRLDistURI format in pkiserv.conf file: CRLDistURI

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 52.

IKYC043I Error nnnn in creating HFS 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
Messages

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 52.

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 52.

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. For more information, see "Enabling Simple Certificate Enrollment Protocol (SCEP)" on page 219.

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 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, 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 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 219.

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 219.

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, 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 219.

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 and 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 219.
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 219.

IKYC051I Error 0xnnnn 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 0xnnnn 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 219.

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 219.

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 219.

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 219.

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 219.
Messages

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 219. 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.

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 219.

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 key-pair and a 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 219.

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 PREREREGISTER 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.
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 219.

IKYC061I Can not generate a certificate or a request with automatic renewal

Explanation: PKI Services is trying to process a GENCERT/REQCERT or GENCERT/REQCERT 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.


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 GENCERT/REQCERT or GENRENEW/REQRENEW with nickname: template-nickname

Explanation: PKI Services is processing a GENCERT/REQCERT or GENRENEW/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
Messages

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.

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 52.

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 401.

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 setup 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 52.

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 52.

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
## Messages

A data set has been created. If the data set name is not specified in the pki\_serv.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 pki\_serv.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 (pki\_serv.conf), including their defaults, see [(Optional) Steps for updating the configuration file](#). See also ["Steps for creating the VSAM object store and ICL data sets and indexes" on page 52. See also ["Steps for creating the VSAM object store and ICL data sets and indexes"](#page 52].

### 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**

**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 I revoked) because of a state change.
12. Incorrect (CertId I Serial Number) specified.
13. The status of the (request I certificate) has been changed by another process.
14. (CertIds I 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 IBM 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-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 is 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,” on page 101 and Chapter 12, “Customizing the administration Web pages if you use REXX CGI execs,” on page 151.

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 is 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.
Messages

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 166.

IKYK001I Unexpected PKCS#11 function-name return code 0xnnnn. 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 0xnnnn . 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'0000': ICSF is not active or is not configured for PKCS #11 services.

X'0001': 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'0001', 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.


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,0=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 82.

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.
System programmer response: Diagnose the problem indicated by the return code. For LDAP_SERVER_D0WN, 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 IKYO009I is also displayed, report that information as well. For more information, see "Steps for tailoring the LDAP section of the configuration file" on page 82.

**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.DEFUALTS 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 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 "Steps for updating the configuration file" on page 52.

**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.*
Messages

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.

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.

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. However,
certificate request management functions are still
available through the R_PKIServ callable service.

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 19, "RACF
administration for PKI Services," on page 341.

For more information, see "Installing and configuring
ICSF (optional)" on page 211 "z/OS Cryptographic
Services ICSF System Programmer’s Guide" and z/OS

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
cmdnmand specifying a correct log parameter. For more
information, see "Changing logging options" on page
401

Routing code: 2
Descriptor code: 5

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 52

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.

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: 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 97 and "Changing logging
options" on page 401.

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 97 and "Stopping the PKI
Services daemon" on page 97.

Routing code: 2
Descriptor code: 6

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.

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

Routing code: 2
Descriptor code: 6
Messages

the PKI Services address space non-swappable. The attempt was unsuccessful. The SYSEVENT TRANSWAP error code is displayed.

System action: PKI Services stops.


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 0xnnnn

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 _erro2().

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 22, “Using information from the PKI Services logs,” on page 397.

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 95 and “Stopping the PKI Services daemon” on page 97.
Routing code: 2
Descriptor code: 6

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

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 0xnnnn
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.
Routing code: 2
Descriptor code: 6

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.
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}

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 401.

Routing code: 2
Descriptor code: 6

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 348 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.

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 19, “RACF administration for PKI Services,” on page 341 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.
Chapter 23. Messages

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 can not 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 19, “RACF administration for PKI Services,” on page 341. 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 95.

IKYP030I  CRL APPROACHING MAXIMUM SIZE

Explanation:  PKI Services is creating CRLs as a part of CRL interval processing. CRLs are stored in the request database before being published to the LDAP directory. At least one CRL has been determined to be rather large in size, approaching the VSAM record size limit of 32K bytes. CRL resizing is highly suggested.

System action:  PKI Services CRL processing continues. If the CRLs are all less than the VSAM size limit, then 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 VSAM record size limit, PKI Services CRL processing will be unsuccessful. The CRLs in question will not be published to the LDAP directory. When this happens you will also receive message IKYC010I with the error code description Record too Large.

System programmer response:  It is imperative that you correct the situation immediately to prevent the CRL from exceeding the VSAM record size limit. If you are not yet using distribution point CRLs, then you need to 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, then you must decrease the value specified for the CRLDistSize directive. Make the appropriate changes. Then, save the configuration file. 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 52.

Note:  The action stated above 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.

IKYP031E  [RSA | DSA] 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:

1. 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.
2. 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)
- 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)
Messages

Correct the configuration values, and restart PKI Services. For more information, see "Updating the digitalSignature algorithm" on page 191.

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 encrypting 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 219. 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

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: 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 19, "RACF administration for PKI Services," on page 341. To determine if your SCEP configuration requires a CA certificate or a CA/RA combination, see "Installing and configuring ICSF (optional)" on page 21.

Routing code: 2
Descriptor code: 6
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 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 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 19 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 82.

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) data set 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

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
Messages

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 19, “RACF administration for PKI Services,” on page 341 and z/OS Security Server RACF Security Administrator’s Guide.

IKYS004I  Error 0xnnnn 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.

IKYS005I  Error 0xnnnn 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 Open 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 52.

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 19, “RACF administration for PKI Services,” on page 341.
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 52.

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 27 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.
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 344 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

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 Cryptographic Services System SSL Programming. Make corrections as needed. Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see z/OS Security Server RACF Security Administrator's Guide and Chapter 19, "RACF administration for PKI Services," on page 341.

IKYS018I Error 0xnnnn 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 to 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 IKY1002I.

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 341.

System programmer response: See message IKY1002I for information. For more information, see "Using the pkiprereg utility" on page 331.
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 IKYI002.

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 -I 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 331.

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 -I 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 331.

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 -I 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 331.

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 331.

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 -I 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 331.

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 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 -I 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 331.

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.
Messages

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 “Using PKI Services utilities,” on page 321.

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 "Using PKI Services utilities,” on page 321.

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 "Using PKI Services utilities,” on page 321.

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 preregule 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 331.
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.
Chapter 24. 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
  - IKYCVSAM
  - IKYDDDEF
  - IKYISMKD
  - IKYMKDIR
  - IKYRVSAM
  - IKYSETUP
  - IKYVBKUP
  - IKYVREST
- PROCLIB/PROCLIB
  - IKYSPROC with alias PKISERVD
- SIEALNKE/AIEALNKE
  - IKYPKID - The PKI Services daemon
  - IKYPRTM - The Resource Termination Manager for the daemon

File system directory and subdirectories

Additionally, unless you change the default, SMP/E installs PKI Services into the file system directory `/usr/lpp/pkiserv`. The following table describes the directory structure and contents:

### Table 81. Files contained in subdirectories

<table>
<thead>
<tr>
<th>Subdirectory</th>
<th>Contains...</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>Utilities executables:</td>
</tr>
<tr>
<td></td>
<td>- iclview — Utility for viewing issued certificate list (certificate database). (For more information, see &quot;Using the iclview utility&quot; on page 322.)</td>
</tr>
<tr>
<td></td>
<td>- pkiprereg — Utility for creating preregistration records in batch for Simple Certificate Enrollment Protocol (SCEP) clients, see &quot;Using the pkiprereg utility&quot; on page 331.)</td>
</tr>
<tr>
<td></td>
<td>- pktp_install — Program to register the PKI Services Trust Policy plug-in with OCSF. (For more information, see &quot;Configuring and getting started with PKITP&quot; on page 371.)</td>
</tr>
<tr>
<td></td>
<td>- pktp.ivp — Program to verify that the PKI Services Trust Policy plug-in installed successfully. (For more information, see &quot;Configuring and getting started with PKITP&quot; on page 371.)</td>
</tr>
<tr>
<td></td>
<td>- vosview — Utility for viewing VSAM object store (request database). (For more information, see &quot;Using the vosview utility&quot; on page 326.)</td>
</tr>
<tr>
<td></td>
<td>- TemplateTool</td>
</tr>
<tr>
<td>include</td>
<td>C header files:</td>
</tr>
<tr>
<td></td>
<td>- pktp.h — C language header file for writing application programs that use the PKI Trust Policy Plug-in. (For more information, see &quot;Files for PKITP&quot; on page 371.)</td>
</tr>
</tbody>
</table>
### File directory structure

**Table 81. Files contained in subdirectories (continued)**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lib</strong></td>
<td>Loadable files:</td>
</tr>
<tr>
<td></td>
<td>- <em>pkitp.so</em> — OCSF Trust Policy plug-in for PKI Services. (For more information, see <a href="#">Files for PKITP</a> on page 371.)</td>
</tr>
<tr>
<td></td>
<td>- *<em>.dll</em> — Dynamic link libraries (DLLs) that the PKI Services daemon uses.</td>
</tr>
<tr>
<td></td>
<td>- <em>nls/msg/En_US.IBM-1047/</em>.cat* — The PKI Services message catalogs. (These message catalogs are also symbolically linked in the <code>/usr/lpp/pkiserv/lib/nls/msg/C</code> directory as well as the <code>/usr/lib/nls/msg/En_US.IBM-1047</code> and <code>/usr/lib/nls/msg/C</code> directories.)</td>
</tr>
<tr>
<td></td>
<td>- *<em>.jar</em> — Java archives.</td>
</tr>
<tr>
<td></td>
<td>- <em>libpksJNI.so</em> — JNI shared object.</td>
</tr>
<tr>
<td><strong>pkijsp</strong></td>
<td>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 176.)</td>
</tr>
<tr>
<td><strong>PKIServ</strong></td>
<td>CGIs that make up the PKIServ Web application. (For information about CGIs, see “Relationship between CGIs and the pkiserv.tmpl file” on page 135 and Table 37 on page 152.)</td>
</tr>
</tbody>
</table>

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 81. Files contained in subdirectories (continued)

<table>
<thead>
<tr>
<th>samples</th>
<th>Various sample files, including:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• expiringmsg.form—The e-mail message sent to a user as notification about an certificate that will expire</td>
</tr>
<tr>
<td></td>
<td>• httpd.conf—Contains z/OS HTTP Server directives. (For a code sample, see [z/OS HTTP Server configuration directives](page 471).)</td>
</tr>
<tr>
<td></td>
<td>• httpd2.conf—Contains z/OS HTTP Server directives for the second Web server. (For a code sample, see [z/OS HTTP Server configuration directives](page 471).)</td>
</tr>
<tr>
<td></td>
<td>• 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). (For a code sample, see [The pkiserv.envvars environment variables file](page 448).)</td>
</tr>
<tr>
<td></td>
<td>• Makefile.pkiexit—The makefile for the PKI Services exit. (For more information, see Chapter 15, “Customizing with installation exits,” on page 235.)</td>
</tr>
<tr>
<td></td>
<td>• 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](page 371).)</td>
</tr>
<tr>
<td></td>
<td>• pendingmsg.form—The e-mail message sent to an administrator listing pending requests.</td>
</tr>
<tr>
<td></td>
<td>• pkiexit.c—The sample PKI Services exit, which PKI Services provides. (For more information, see Chapter 15, “Customizing with installation exits,” on page 235.)</td>
</tr>
<tr>
<td></td>
<td>• pkiserv.envvars—The PKI Services environment variables file. (For more information, see &quot;Optionally updating PKI Services environment variables&quot; on page 49 and [The pkiserv.envvars environment variables file](page 448).)</td>
</tr>
<tr>
<td></td>
<td>• pkiserv.tmpl—The PKI Services certificate templates file used with the REXX CGI execs. (For more information, see Chapter 11, &quot;Customizing the end-user Web application if you use REXX CGI execs,&quot; on page 101.)</td>
</tr>
<tr>
<td></td>
<td>• pkiserv.conf—The PKI Services configuration file. (For more information, see [&quot;Optional] Steps for updating the configuration file&quot; on page 52 and Chapter 25, “The pkiserv.conf configuration file,” on page 441.)</td>
</tr>
<tr>
<td></td>
<td>• 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 157.)</td>
</tr>
<tr>
<td></td>
<td>• 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 157.)</td>
</tr>
<tr>
<td></td>
<td>• pkitpsamp.c—Sample application to call the PKI Trust Policy plug-in. (For more information, see [Files for PKITP](page 371) and [Building the sample application to invoke the certificate validation service](page 376).)</td>
</tr>
<tr>
<td></td>
<td>• readymsg.form—The e-mail message sent to a user as notification a certificate is ready for retrieval</td>
</tr>
<tr>
<td></td>
<td>• recovermsg.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</td>
</tr>
<tr>
<td></td>
<td>• rejectmsg.form—The e-mail message sent to a user as notification a request for a certificate has been rejected</td>
</tr>
<tr>
<td></td>
<td>• renewcertmsg.form—The e-mail message sent to a user as notification that PKI Services has automatically renewed an expiring certificate</td>
</tr>
</tbody>
</table>
File directory structure
Chapter 25. 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 52 and Table 18 on page 52.

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/.

```plaintext
# Licensed Materials - Property of IBM
# 5694-A01
# Copyright IBM Corp. 2001, 2009
# Status = HKY7760

[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-256WithRSAEncryption=1.2.840.113549.1.1.11
sha-1WithRSAEncryption=1.2.840.113549.1.1.5
md-5WithRSAEncryption=1.2.840.113549.1.1.4
md-2WithRSAEncryption=1.2.840.113549.1.1.2
id-dsa-with-shal=1.2.840.10040.4.3

MyPolicy=1.2.3.4

[ObjectStore]
# 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.requestor'

# How 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 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

# Are the VSAM data sets shared in a sysplex with other instances
# of PKI Services. True (T) or False (F)
SharedVSAM=F

[CertPolicy]
SigAlg1=sha-1WithRSAEncryption
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.,
pkiserv.conf

#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/

# 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
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

[SAF]
KeyRing=PKISRVD/CAring

# 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 26. 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.envvars. (See "The pkiserv.envvars environment variables file" on page 448.) 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 484.

Environment variables in the environment variables file

The environment variables contained in pkiserv.envvars 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 HTTP Server environment variables file. The z/OS HTTP Server environment variables file is usually
Environment variables

/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:

<table>
<thead>
<tr>
<th>Subcomponent</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>The wildcard character (represents all subcomponents)</td>
</tr>
<tr>
<td>CORE</td>
<td>The core functions of PKI Services that are not specific to the other subcomponents</td>
</tr>
<tr>
<td>DB</td>
<td>Activity related to the request or issued certificate VSAM data stores</td>
</tr>
<tr>
<td>LDAP</td>
<td>LDAP posting operations</td>
</tr>
<tr>
<td>PKID</td>
<td>The PKI Services daemon address setup and infrastructure</td>
</tr>
<tr>
<td>POLICY</td>
<td>Certificate creation and revocation policy processing</td>
</tr>
<tr>
<td>SAF</td>
<td>SAF key ring, OCEP, and R_dataLib calls</td>
</tr>
<tr>
<td>TPOLICY</td>
<td>Trust policy plug-in processing</td>
</tr>
</tbody>
</table>

The message levels are listed hierarchically:

<table>
<thead>
<tr>
<th>Debug level</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>This indicates logging only severe messages.</td>
</tr>
<tr>
<td>E</td>
<td>This indicates logging severe and error messages.</td>
</tr>
<tr>
<td>W</td>
<td>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.</td>
</tr>
<tr>
<td>I</td>
<td>This indicates logging severe, error, warning, and informational messages.</td>
</tr>
<tr>
<td>D</td>
<td>This indicates logging severe, error, warning, informational, and diagnostic messages.</td>
</tr>
<tr>
<td>V</td>
<td>This indicates logging all messages, including verbose diagnostic messages. This is very verbose.</td>
</tr>
</tbody>
</table>

**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 49.) 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, 2009
# Status = HKY7760
#
# Language and Path configurations
#
LANG=En_US.IBM-1047
PATH=/usr/sbin
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 27. 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. 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.

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
- 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:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPORT</td>
<td>Retrieves (exports) a previously requested certificate, or retrieves (exports) the PKI Services registration authority (RA) certificate or the certificate authority (CA) certificate.</td>
</tr>
<tr>
<td>GENCERT</td>
<td>Generates an auto-approved certificate.</td>
</tr>
<tr>
<td>GENRENEW</td>
<td>Generates an auto-approved renewal certificate. (The request submitted is automatically approved.)</td>
</tr>
<tr>
<td>QRECOVER</td>
<td>Lists certificates whose key pairs were generated by PKI Services under a requestor's e-mail address and passphrase.</td>
</tr>
<tr>
<td>REQCERT</td>
<td>Requests a certificate that an administrator must approve before it is created.</td>
</tr>
<tr>
<td>REQRENEW</td>
<td>Requests certificate renewal. The administrator needs to approve the request before the certificate is renewed.</td>
</tr>
<tr>
<td>RESPOND</td>
<td>Invokes the PKI OCSP responder.</td>
</tr>
<tr>
<td>REVOKE</td>
<td>Revokes a certificate that was previously issued.</td>
</tr>
<tr>
<td>SCEPREQ</td>
<td>Generates a certificate request using Simple Certificate Enrollment Protocol (SCEP).</td>
</tr>
<tr>
<td>VERIFY</td>
<td>Confirms that a given user certificate was issued by this certificate authority and, if so, returns the certificate fields.</td>
</tr>
</tbody>
</table>

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 82. Access required if you plan to have an administrator approve certificate requests

<table>
<thead>
<tr>
<th>Resource</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR.DIGTCERT.REQCERT</td>
<td>READ</td>
</tr>
<tr>
<td>IRR.DIGTCERT.REQRENEW</td>
<td>READ</td>
</tr>
</tbody>
</table>

If your clients can request certificates that are auto-approved without action by an administrator, PKISERV needs the following access:

Table 83. Access required if you plan to use auto-approval

<table>
<thead>
<tr>
<th>Resource</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR.DIGTCERT.ADD</td>
<td>UPDATE</td>
</tr>
<tr>
<td>IRR.DIGTCERT.GENCERT</td>
<td>CONTROL</td>
</tr>
<tr>
<td>IRR.DIGTCERT.GENRENEW</td>
<td>READ</td>
</tr>
</tbody>
</table>

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 84. FACILITY class access needed for protecting administrative functions

<table>
<thead>
<tr>
<th>Resource</th>
<th>Access</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR.RPKISERV.PKIADMIN[.ca_domain]</td>
<td>READ</td>
<td>For list and query operations</td>
</tr>
<tr>
<td></td>
<td>UPDATE</td>
<td>To act on certificate requests, preregistration requests, and issued certificates</td>
</tr>
</tbody>
</table>

To grant user ID ADMID authority to administer PKI Services, use the following RACF TSO commands:

**Example:**

```racf
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 the 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 85. 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>
<thead>
<tr>
<th>Table 85. Access PKISRVD needs to use RACF certificates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resource</strong></td>
</tr>
<tr>
<td>IRR.DIGTCERT.GENCERT</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>IRR.DIGTCERT.LISTRING</td>
</tr>
<tr>
<td>IRR.DIGTCERT.LIST</td>
</tr>
</tbody>
</table>

---

**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:
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') SUBJECTDN(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('pkisrdv.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
/* REXX */
/*********************************************************************/
/**/ /* DESCRIPTIVE NAME: PKI Services RACF setup CLIST */
/**/ /* Licensed Materials - Property of IBM */
/**/ /* 5694-A01 */
/**/ /* Copyright IBM Corp. 2001, 2009 */
/**/ /* Status - HKY7760 */
/**/ /*01* EXTERNAL CLASSIFICATION: OTHER */
/**/ /*01* END OF EXTERNAL CLASSIFICATION: */
/**/ /*01* 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. */
/**/ /*01* 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: */
/**/ /* $11=PK154 , HKY7708, 020514, JWS: PKI Services */
/**/ /* $01=MG01103, HKY7708, 020813, BRW: */
/**/ /* $01=MG01405, HKY7708, 021114, BRW: */
/**/ /* $12=PK156 , HKY7720, 040204, TCG: PKI Services VI */
/**/ /* $13=PK156 , HKY7720, 040130, MSF: PKI Services VI */
/**/ /* $14=PK157 , HKY7730, 050301, TCG: PKI Services VII */
```
IKYSETUP

/* $02-MG06477, HKY7730, 051018, TCG: Incorrectly formed command */
/* $01-MG07496, HKY7730, 060607, SHN: Misc updates */
/* $15-PKIS9 , HKY7740, 060607, SHN: Removing private from dsn */
/* $L6-PKIS10J, HKY7750, 070529, RDC: Remove DCSF setup */
/* $L7-RAODS, HKY7750, 070726, RDC: 4096 Bit Updates */
/* $L8-RAODS, 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 */

/**/

/* 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 @L4A*/
/* to the PKI daemons' keying, @D2A*/
/* C - Added CA domain qualifier to RA cert backup dataset @L5A*/
/* when run with CA Domain support. @L5A*/
/* - Added logic and doc to prevent usage of 2048 bit DSA keys for the CA certificate, @D3A*/
/* - Added CA domain qualifier to RA label when run with CA Domain support. @D3A*/
/* C - Removed 'private' dataset qualifier from all dataset names. @L5A*/
/* C - Remove RACF Profile setup for daemon access to OCSF @L5A*/
/* C - Prevent usage of keysize greater than 4096 for key type 2 (PCICC). @L5A*/
/* A - Key generation support @L5A*/
/* - Distinguish datasets needed to be saved and downloaded @L5A*/
/* C - Misleading comment about keysize supported for a keytype of 0. @D3A*/
/* C - Fixed error in cryptoz permit commands @D3A*/
/* - Added commands to define generic dataset profile for daemon user ID. @D3A*/
/* - Added conditional logic to not issue permit commands for DIGTCERT profiles when Web Server certificate and key ring is not being defined/created. @D3A*/
/* */
/**/ 

/*******************************************************************************/
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_de="" 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 */

ra_label = STRIP(ca_domain "Local PKI RA") /* Label for RA Certificate */
if (ra_label = "") then /* If no RA Label ... */
ra_dn="" /* */
else /* */
ra_dn= "CN('Registration Authority')", ca_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')"

web_ring = "SSLring" /* SAF keyring for web server */
IKYSETUP

/*******************************************************************/
/* 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 - Use ICSF? */
/* */
/* There are four possible choices for generation and */
/* 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 key using software then store the key in */
/* ICSF. (Option 1) */
/* */
/* - Generate the key through ICSF using the PCI cryptographic */
/* coprocessor (PCICC) then store the key in ICSF. (Option 2) */
/* */
/* - Generate the DSA key using software and retain it as a */
/* software key. This key cannot be saved in ICSF. (Option 3) */
/* */
/* Notes: */
/* */
/* - For options 1 and 2, ICSF must be configured for PKA */
/* support and running. Additionally, for option 2, a PCICC */
/* must be present and operational. */
/* */
/* - Options 0 and 2 are the only way to generate a private key */
/* larger than 1024 bits. */
/* */
/* - If option 2 is selected, the certificate and private key */
/* will not be backed up by this exec. */
/* */
/* - 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 or 3. */
The IKYSETUP REXX exec

Chapter 27. The IKYSETUP REXX exec 459
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' /*@L2C*/

****************************************************************/
/Question 5 - 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. */

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 */
/* cryptoz_profile_so = 'SO.'||daemon||'.*' /*@L9A*/
cryptoz_profile_user = 'USER.'||daemon||'.*' /*@L9A*/
cryptoz_grp = '' /*@L9A*/

/*******************************************************************************/
/* 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... */
    log_dsn="IKYSETUP.LOG" /* Under your ID */
else /* Else use CA Domain */
    log_dsn-ca_domain_trunc]]."IKYSETUP.LOG" /* CA Domain qualified */

/*******************************************************************************/
/* 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 */

web_label = "SSL Cert" /* Label for web server cert */
ca_expires ="2020/01/01" /* date the CA certificate for certificate authority should expire */
web_expires ="2020/01/01" /* date the certificate for web server SSL should expire */
if (ca_domain = "") then /* If no CA Domain... */
   ca_ring="CAring" /* keyring name for PKI Srvs */
else /* Else use CA Domain */
   ca_ring="CAring."||ca_domain /* CA Domain qualified */

/*******************************************************************/
/* You can select the size (in bits) of your CA's private key. */
/* The range is 512-4096. The default is 1024. Note, if you */
/* wish the key size to be greater than 1024, you must select */
/* key_type 0 or 2 above. */
/*******************************************************************/
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... */
   backup_dsn = "'"||daemon||".KEY.BACKUP.P12BIN'" /* */
else /* Else use CA Domain */
   backup_dsn = "'"||daemon||"."||ca_domain_trunc||", "
                  "KEY.BACKUP.P12BIN'" /* */
   ca_domain qualify backup dsn

/*******************************************************************/
/* Data set to contain the exported copy of the CA certificate */
/* (DER encoded). This is to assist the backup process. */
/*******************************************************************/
if (ca_domain = "") then /* If no CA Domain... */
   cacert_dsn = "'"||daemon||".CACERT.DERBIN' /* */
else /* Else use CA Domain */
   cacert_dsn = "'"||daemon||"."||ca_domain_trunc||", "
                 "CACERT.DERBIN'" /* */
   ca_domain qualify export dsn

/*******************************************************************/
/* 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. */
/*******************************************************************/
if (ca_domain = "") then /* If no CA Domain... */
   export_dsn = "'"||daemon||".WEBROOT.DERBIN' /* */
else /* Else use CA Domain */
   export_dsn = "'"||daemon||"."||ca_domain_trunc||", "
                 "WEBROOT.DERBIN'" /* */
   ca_domain qualify export dsn

/*******************************************************************/
/* 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... */
   ra_backup_dsn = "'"||daemon||".RAKEY.BACKUP.P12BIN' /* */
else /* Else use CA Domain */
   ra_backup_dsn = "'"||daemon||"."||ca_domain_trunc||", "
                   "RAKEY.BACKUP.P12BIN'" /* */
   ca_domain qualify RA backup dsn

/*******************************************************************/
/* 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 */
IKYSETUP

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) DSOORG(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
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"

/**************************************************************/
/* Maximum key size is 4096 for key type 0 (Software) and key */
/* type 2 (PCICC). Maximum key size is 1024 for key type 1 (ICSF) */
/* and key type 3 (DSA). If invalid key size is chosen, we will */
/* issue message and end execution. */
/**************************************************************/
if ca_keysize > 4096 then do
call logsay "Key size cannot be greater than 4096"
return 8
end
else /* @L8C */
   if (key_type=1 | key_type = 3) & ca_keysize > 1024 then do
      call logsay "Key size cannot be greater than 1024 for key type",
      key_type
      return 8
   end
end
/**************************************************************/
/* 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')"
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*/
if pkigroup ^= '' then do
  call tsoserv "ADDGROUP " pkigroup "OMVS(GID(''pki_gid''))"
do j = 1 to pkigroup_mem.0
  call tsoserv "CONNECT" pkigroup_mem.i "GROUP(''pki_gid'')"
end

/*================================================================******
* 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 databases ...'
call tsoserv "PERMIT 'vsamhlq.**' ID(pkigroup) ACCESS(CONTROL)"
call tsoserv "SETROPTS GENERIC(DATASET) REFRESH"

/*================================================================******
* 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')"
  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'
  call tsoserv certcmd
end /* pkigroup ^= '' */

/*================================================================******
* 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('daemon') ID(daemon)'
call tsoserv 'RACDCERT ADDRING('daemon') ID(daemon)'
  'RACDCERT ID('daemon') CONNECT(CERTAUTH),
  'LABEL('daemon')",
  '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)"
  if web_dn ^= '' then do
    call logsay2 'Creating the Webserver SSL certificate and keyring ...'
call tsoserv 'RACDCERT GENCERT ID(webserver) SIGNWITH(CERTAUTH),
  'LABEL('web_label')",
  'RING('ca_ring') USAGE(PERSONAL) DEFAULT) *
if web_dn ^= '' then do
  call tsoserv 'RACDCERT ADDRING('web_ring') ID(webserver)'
call tsoserv 'RACDCERT ADDRING('web_ring') ID(webserver)'
  'RACDCERT ID('webserver') CONNECT(CERTAUTH),
  'LABEL('web_label')"
  if (web_label ^= '') then /*@L4A*/
    call tsoserv 'RACDCERT ID('daemon') CONNECT(LABEL('web_label'))",
      'RING('ca_ring') USAGE(PERSONAL)"
      if web_label ^= '' then do
        call tsoserv 'RACDCERT ADDRING('web_ring') ID(webserver)'
call tsoserv 'RACDCERT ADDRING('web_ring') ID(webserver)'
      if web_label ^= '' then do
        call tsoserv 'RACDCERT ADDRING('web_ring') ID(webserver)'
call tsoserv 'RACDCERT ADDRING('web_ring') ID(webserver)'
          'RACDCERT ID('webserver') CONNECT(CERTAUTH),
          'LABEL('web_label')"
          if web_label ^= '' then do
            call tsoserv 'RACDCERT ADDRING('web_ring') ID(webserver)'
call tsoserv 'RACDCERT ADDRING('web_ring') ID(webserver)'
        end /* web_dn ^= '' */
      end /* web_label ^= '' */
    end /* web_label ^= '' */
  end /* web_dn ^= '' */
end /* pkigroup ^= '' */
if web_ring ^= "" then
    call tsoserv "RACICERT 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, i.e. 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 = "RACICERT CERTAUTH EXPORT(LABEL("ca_label"))",
        " DSN("export_dsn") FORMAT(CERTDER)" /*@L9A*/
    call tsoserv certcmd
end
else /*@L9A*/
    /* Need to manually 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)"
IKYSETUP

* 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 "DEFINE FACILITY IRR.DIGTCERT.GENCERT"
call tsoserv "DEFINE FACILITY IRR.DIGTCERT.LIST"
call tsoserv "PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY)",
* "ID("daemon") ACCESS(CONTROL)"
call tsoserv "PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)",
* "ID("daemon") ACCESS(READ)"
call tsoserv "PERMIT IRR.DIGTCERT.LISTRING 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 ...
* ACCESS(READ)"
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" ...
* ACCESS(READ)"
call tsoserv "SETROPTS CLASSACT(SURROGAT)"
call tsoserv "DEFINE SURROGAT BPX.SRV."surrog"
call tsoserv "PERMIT BPX.SRV."surrog" CLASS(SURROGAT)",
* "ID("webserver") ACCESS(READ)"
call tsoserv "SETROPTS RACLIST(SURROGAT) REFRESH"

if (key_type=1 | key_type=2) then do
/*****************************/
/* Allow the daemon to use ICSF */
/*****************************/
call logsay2 "Allowing the PKI Services daemon to use ICSF ...
* ACCESS(READ)"
call tsoserv "DEFINE CSFKSERV profile "UACC(NONE)"
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 (key_gen=1) then do
/*****************************/
/* Connect to ICSF */
/*****************************/
call tsoserv "CONNECT daemon "GROUP(" csfusers_grp ")"
call tsoserv "CONNECT surrog "GROUP(" csfusers_grp ")"
end
/* Allow the daemon to generate key pairs */
call logsay2 "Allowing the PKI Services daemon to generate key pairs ..."
if cryptoz_profile_so ^= '' & cryptoz_profile_user ^= '' then do
    call tsoserv "SETROPTS CLASSACT(CRYPTOZ)"
call tsoserv "SETROPTS GENERIC(CRYPTOZ)"
call tsoserv "SETROPTS RACLIST(CRYPTOZ)"
call tsoserv "RDEFINE CRYPTOZ", cryptoz_profile_so "UACC(NONE)"
call tsoserv "RDEFINE CRYPTOZ", cryptoz_profile_user "UACC(NONE)"
call tsoserv "PERMIT" cryptoz_profile_so "CLASS(CRYPTOZ)", * ID("daemon") ACCESS(UPDATE)"
call tsoserv "PERMIT" cryptoz_profile_user "CLASS(CRYPTOZ)", * ID("daemon") ACCESS(CONTROL)"
call tsoserv "SETROPTS RACLIST(CRYPTOZ) REFRESH"
call tsoserv "SETROPTS GENERIC(CRYPTOZ) REFRESH"
end
if cryptoz_grp ^= '' then do
    call tsoserv "CONNECT" daemon "GROUP(" cryptoz_grp ")"
end
end

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 */
call logsay2 "Allowing "surrog" to request certificate functions ..."
call tsoserv "SETR GENERIC(FACILITY)"
profilenametemp = "IRR.RPKISERV.**"
if (ca_domain = "") then /* @L4A*/
    profilenametemp = "IRR.RPKISERV.*."||ca_domain_trunc /* @L4A*/
call tsoserv "RDEFINE FACILITY "||profilenametemp /* @L4C*/
call tsoserv "PERMIT "||profilenametemp||" CLASS(FACILITY)", * ID("surrog") ACCESS(CONTROL)" /* @L4C*/

/* The administrative functions of PKI Services are protected */
call logsay2 "Creating the profile to protect PKI Admin functions ..."
if (ca_domain = "") then /* @L4A*/
    profilenametemp = "IRR.RPKISERV.PKIADMIN" /* @L4A*/
else /* @L4A*/
    profilenametemp = "IRR.RPKISERV.PKIADMIN."||ca_domain_trunc /* @L4A*/
call tsoserv "RDEFINE FACILITY "||profilenametemp /* @L4C*/
call tsoserv "PERMIT "||profilenametemp||" CLASS(FACILITY)", * ID("surrog") ACCESS(CONTROL)" /* @L4C*/
IKYSETUP

/* ID("pkigroup") ACCESS(UPDATE) */ /* BLAC */
call tsoserv "PERMIT ":[profname]" CLASS(FACILITY)", /* ID("surrog") ACCESS(NONE) */ /* BLAC */
call tsoserv "SETROPTS RACLIST(FACILITY) REFRESH"
/*****************************/
/* Done. Now write to the log */
/*****************************/
upper daemon vsamhql export_dsn
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
call logsay " "
Chapter 27. The IKYSETUP REXX exec 467
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*/
call logsay, if web_dn ^= "" then do /*2@L4A*/
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," call logsay "qualified domain name"
end /*2@L4A*/
else /*2@L4A*/
call logsay, "Webserver certificate and keyring not created. You must add the CA', "certificate as a 'trusted root' manually"
call logsay " *
call logsay, if web_dn ^= "" then do /*2@L4A*/
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," call logsay "qualified domain name"
end /*2@L4A*/
else /*2@L4A*/
call logsay, "Webserver certificate and keyring not created. You must add the CA', "certificate as a 'trusted root' manually"
call logsay " *
call logsay, if log_dsn ^= "" then do /*2@L4A*/
x = OUTTRAP(MSGS.) 'EXECIO' logdata.0 'DISKM IKYLOGDD (FINIS STEM LOGDATA.' 'FREE FI(IKYLOGDD)' x=OUTTRAP('OFF')
say "Commands complete. Results written to log data set" log_dsn
end /*2@L4A*/
/**************************************************************/ /* Exit */
/**************************************************************/
say 'The IKYSETUP EXEC has completed.' Exit max_return_code
/**************************************************************/ /* tsoserv - echo rc and commands and track highest rc */
/**************************************************************/
tzoserv: 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 /*2@L4A*/
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 /*2@L4A*/
    end /*2@L4A*/
    max_return_code= max(max_return_code,return_code)
    return return_code
return 0
/**************************************************************/ /* logsay - echo messages to the terminal and logdata stem */
IKYSETUP

/***************************************************************************/
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
temp 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
                    end
                end
            end
        end
    end
dn= ""
do i = 1 to 7
    if dns.i ^= "" then
        if dn ^= "" then
            dn= dn || " ," || dns.i
        else
            dn= dn || " " || dns.i
    end
end
return 0

Chapter 27. The IKYSETUP REXX exec 469
IKYSETUP
Chapter 28. 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."
- 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 473.
- 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 477.
- IKYVBKUP, which is sample JCL to back up the PKI Services VSAM data sets using the DFSMSdss DUMP utility. (See "IKYVBKUP" on page 481.
- 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 483.
- PKISERVD, which is a sample procedure to start PKI Services daemon. (See "PKISERVD sample procedure to start PKI Services daemon" on page 484.

Note: Other important programs are contained in other chapters:

- pkiserv.envs (the PKI Services environment variables file). See The pkiserv.envs environment variables file" on page 448.

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. 2001,2002
# Status = HKY7708
# Change-Activity:
# $L1=PKIS4 , HKY7708, 020429, JWS: PKI Services
# Change Descriptions:
# A - Multiple application support
# For a secure system, set the default User ID to %CLIENT%
UserId %CLIENT%

sslmode on
sslport 443
Normalmode on
Protection PublicUser {

© Copyright IBM Corp. 2001, 2009
471
Other code samples

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/*

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.
Other code samples

# Change Descriptions: #
# A - Multiple application support OLI A #
#---------------------------------------------------------------#

# For a secure system, set the default User ID to %%CLIENT%%
UserId %%CLIENT%%

# SSL support using a SAF keyring
keyfile SSLeRi ng SA F #
# OR
# May use a gskkyman key database instead of SAF keyring
#key file /etc/key.kdb

sslmode on
sslport 1443
Normalmode off
SSLCle ntAuth strong
SSLX500C A roots local_and_x500
SSLX500Host <ldap-server-name>
SSLX500Port <ldap-port-number>
SSLX500User ID <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

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/*

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 477.)
Other code samples

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 */
/** (C) Copyright IBM Corp. 2001, 2006 */
/** Status = H0Y7730 */
/** */
/** *****************************************************************************/
/** 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 */
/** 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(vvvvv) 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(vvvvv) statements from the DEFALTDX step. */
/** */
/** c) Remove all the SPANNED and CISIZE statements. */
/** */
/** If not using VSAM RLS, change all occurrences of vvvvv 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: */
/** */
/** $1L=PKIS1 H0Y7707 020314 PDJWS1: VSAM RLS 0L1A */
/** $1P=MG000719 H0Y7707 020416 PDJWS1: VSAM RLS 2 0P1A */
/** $2L=MG01176 H0Y7708 020826 PDJWS1: VSAM scaling 0L2A */
/** $2P=MG01346 H0Y7708 021022 PDJWS1: JCL errors 0P2A */
/** $3L=PKIS7 H0Y7730 050228 P0TG1: Multi-CA Support 0L3A */
Other code samples

```plaintext
C: Added STORCLAS instructions, LOG. Removed VOLUME DDs
D: Removed FILE(VOLUME) statements
C: Added more alt indexes and changed allocation parms
C: Removed VOL keywords from ALTERNATEINDEX statements.
C: Removed DD statements from BLDINDEX step. Added
IEBGENER step to remove hardcoded binary zeros
C: Updated prolog with CA Domain information
C: Updated DEFALTIDX step to use PKISRVD.VSAM.OST.AIX.IX
  dataset name instead of the former dataset name of
  PKISRVD.VSAM.AIX.IX.
-------------------------------------------------------------------
// Delete existing clusters, paths, alt indexes
-------------------------------------------------------------------
//DELCLUST EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  //SYSIN DD *
DELE
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(1024) -
  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(1024) -
  SPANNED) -
```
Other code samples

```plaintext
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=F8,LRECL=80,BLKSIZE=640),SPACE=(TRK,(1,1))
//SYSSN DD *
//GENERATE MAXFLDS=4,MAXLITS=80
//RECORD FIELD=(20,X'00000000000000000000000000000000',,1), 
// FIELD=(20,X'00000000000000000000000000000000',,21), 
// FIELD=(20,X'00000000000000000000000000000000',,41), 
// FIELD=(20,X'00000000000000000000000000000000',,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.OST)
/
//--------------*/
// 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))
```

476 z/OS V1R11.0 Cryptographic Services PKI Services Guide and Reference
**Other code samples**

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.
Other code samples

**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.

```plaintext
//IKYRVSAM JOB <job card parameters>

//*********************************************************************
//* SAMP:    IKYRVSAM *
//**
//** Licensed Materials - Property of IBM
//** 5604-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 RENAMES step. If you are using multiple CA Domains, IBM recommends using the first eight characters 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 RENAMES 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 RENAMES 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:

```
Other code samples

```
/* $L2=PKIS7 HKY7730 050228 PDTOGI: Multi-CA Support */
/* Change Description: */
/* C: Removed SPANNED, CISIZE, and FILE(VOLUME) statements */
/* C: Added more alt indexes and changed allocation parms */
/* C: Removed DD statements from BLDINDEX step */
/* C: Correct souce dataset names for ICL alternate indexes */
/* C: Updated prolog with CA Domain information */
/* C: Updated RENAMEDS step to use PKISRVD.VSAM.OST.AIX.IX */
/* dataset name and conditionally handle the former dataset */
/* - Updated DEFALTDX step to use PKISRVD.VSAM.OST.AIX.IX */
/* - Updated DEFKSDS to have a line continuation character */
/* - after the CYL parameters. */
/*-------------------------------------------------------------------*/
/* 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=(B,LE)
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
DEFINE CLUSTER - (NAME(PKISRVD.VSAM.OST) - STORELASS(class-name) -
```
Chapter 28. Other code samples  479
Other code samples

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 INDATASET(PKISRVD.OLDVSAM.OST) -
OUTDATASET(PKISRVD.VSAM.OST)
REPRO INDATASET(PKISRVD.OLDVSAM.ICL) -
OUTDATASET(PKISRVD.VSAM.ICL)

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))

/*
/* Repr 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
*/
DEFALTIDX 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.OST.STATUS) -
RELATE(PKISRVD.VSAM.OST) -
TRK(5,1) -
KEYS(40 4) ) -
DATA -
(NAME(PKISRVD.VSAM.OST.STATUS.DA)) -
INDEX -
(NAME(PKISRVD.VSAM.OST.STATUS.IX))
DEFINE PATH -
(NAME(PKISRVD.VSAM.OST.STATUS.PATH) -
PATHENTRY(PKISRVD.VSAM.OST.STATUS))
/*-------------------------------------------------------------------*
/* 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)

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 *
This sample JCL may be used to backup the VSAM data sets that 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 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:
$LO=PKIS11M HKY7760 080828 PDTCG1: VSAM Backup job

Change Description:
A000000-999999 New sample written for z/OS release 11

Delete existing backup dataset

Perform a DFSMS/dss Dump of the two PKI Services VSAM clusters

Backup1 EXEC PGM=ADRS2SSU,COND=(8,LT)
BACKUPDS DD DSN=PKISRVD.VSAM.BACKUP,DISP=(NEW,CATLG,DELETE),SPACE=(CYL,(xxx,yyy)),VOL=SER=(vvvvv)
SYSPRINT DD SYSOUT=*
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 used in the backup jcl (IKYVBKUP).
//*****************************************************************************
// Change Activity: $L0=PKIS11M HKY7760 080828 PTDGC1: 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, -
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 pkserv.envs. PKISERVD contains FN (file name) and DIR (directory) parameters, to specify the path name of the environment variables file. You can make any needed changes in PKISERVD, such as updating this path name.

Guideline: By default, the path name for the pkserv.envs environment variables file is /usr/lpp/pkserv/samples/pkserv.envs. 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/pkserv/pkserv.envs.

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.

```hll
PKISERVD PROC REGSIZE=256M, X
    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 pkserv.envs. PKISERVD contains FN (file name) and DIR (directory) parameters, to specify the path name of the environment variables file. You can make any needed changes in PKISERVD, such as updating this path name.

Guideline: By default, the path name for the pkserv.envs environment variables file is /usr/lpp/pkserv/samples/pkserv.envs. 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/pkserv/pkserv.envs.

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.

```
Chapter 29. 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 86 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 86. SMF event code and event code qualifier for PKI Services

<table>
<thead>
<tr>
<th>Event Dec(Hex)</th>
<th>Command</th>
<th>Code qualifier Dec(Hex)</th>
<th>Description</th>
<th>Relocate type sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>79(4F)</td>
<td>CRL publication</td>
<td>0(0)</td>
<td>Successful publication of revocation information</td>
<td>318, 319, 366, 379, 380, 381, 382, 383, 384, 385, 387</td>
</tr>
<tr>
<td>85(55)</td>
<td>SUCCRNEW</td>
<td>0(0)</td>
<td>Successful AutoRenew</td>
<td>318, 319, 341, 342, 346, 358, 363, 373, 391, 408</td>
</tr>
</tbody>
</table>

Relocate section variable data

Table 87 describes the variable data elements of the extended-length relocate section.

Table 87. SMF data elements of the extended-length relocate section for PKI Services

<table>
<thead>
<tr>
<th>Data type (SMF80TP2) Dec(Hex)</th>
<th>Data length (SMF80DL2)</th>
<th>Format</th>
<th>Audited by event code</th>
<th>Description (SMF80DA2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>318(13E)</td>
<td>1–255</td>
<td>EBCDIC</td>
<td>66, 67, 69, 72, 74, 79</td>
<td>Certificate or CRL serial number</td>
</tr>
<tr>
<td>319(13F)</td>
<td>1–255</td>
<td>EBCDIC</td>
<td>66, 67, 69, 72, 74, 79</td>
<td>Certificate or CRL issuer’s distinguished name</td>
</tr>
<tr>
<td>366(16E)</td>
<td>4</td>
<td>Binary</td>
<td>74</td>
<td>Certificate revocation reason</td>
</tr>
<tr>
<td>379(17B)</td>
<td>1–255</td>
<td>EBCDIC</td>
<td>79</td>
<td>CRL issuing distribution point DN</td>
</tr>
<tr>
<td>380(17C)</td>
<td>10</td>
<td>EBCDIC</td>
<td>79</td>
<td>CRL’s date of issue</td>
</tr>
<tr>
<td>381(17D)</td>
<td>8</td>
<td>EBCDIC</td>
<td>79</td>
<td>CRL’s time of issue</td>
</tr>
<tr>
<td>382(17E)</td>
<td>10</td>
<td>EBCDIC</td>
<td>79</td>
<td>CRL’s expiration date</td>
</tr>
<tr>
<td>383(17F)</td>
<td>8</td>
<td>EBCDIC</td>
<td>79</td>
<td>CRL’s expiration time</td>
</tr>
<tr>
<td>384(180)</td>
<td>10</td>
<td>EBCDIC</td>
<td>79</td>
<td>CRL’s date of publish</td>
</tr>
<tr>
<td>385(181)</td>
<td>8</td>
<td>EBCDIC</td>
<td>79</td>
<td>CRL’s time of publish</td>
</tr>
<tr>
<td>387(183)</td>
<td>1–1024</td>
<td>EBCDIC</td>
<td>79</td>
<td>CRL’s issuing distribution point URI</td>
</tr>
<tr>
<td>408(198)</td>
<td>256</td>
<td>EBCDIC</td>
<td>85</td>
<td>AutoRenew Exit path name</td>
</tr>
</tbody>
</table>
## Appendix A. 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 88. LDAP objectclasses and attributes that PKI Services sets

<table>
<thead>
<tr>
<th>End-entity or branch node?</th>
<th>Visible RDN attribute</th>
<th>Objectclasses used</th>
<th>Additional attributes set (other than visible RDN attribute)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a branch node</td>
<td>C=</td>
<td>country</td>
<td>—</td>
</tr>
<tr>
<td>Creating a branch node</td>
<td>L=</td>
<td>locality</td>
<td>—</td>
</tr>
<tr>
<td>Creating a branch node</td>
<td>O=</td>
<td>organization</td>
<td>—</td>
</tr>
<tr>
<td>Creating a branch node</td>
<td>OU=</td>
<td>organizationalUnit</td>
<td>—</td>
</tr>
<tr>
<td>Creating a branch node</td>
<td>DC=</td>
<td>domain</td>
<td>none</td>
</tr>
<tr>
<td>Creating a branch node</td>
<td>Any supported value other than the preceding</td>
<td>organizationalUnit, and extensibleObject</td>
<td>ou (the ou value from CreateOUValue in the LDAP section of pkiserv.conf file)</td>
</tr>
<tr>
<td>Creating a user end-entity</td>
<td>unstructuredName or unstructAddress</td>
<td>account, pkiUser, cEPDevice, and extensibleObject</td>
<td>userCertificate, and uid (hardcoded to NoUid)</td>
</tr>
<tr>
<td>Creating a user end-entity</td>
<td>serialNumber</td>
<td>account, pkiUser, pKCS10Device, and extensibleObject</td>
<td>userCertificate, and uid (hardcoded to NoUid)</td>
</tr>
<tr>
<td>Creating a user end-entity</td>
<td>DC</td>
<td>domain pkiUser, , and extensibleObject</td>
<td>userCertificate</td>
</tr>
<tr>
<td>Creating a user end-entity</td>
<td>dnQualifier</td>
<td>account, pkiUser, uniquelyQualifiedObject, and extensibleObject</td>
<td>userCertificate, and uid (hardcoded to NoUid)</td>
</tr>
<tr>
<td>Creating a user end-entity</td>
<td>UID</td>
<td>account, pkiUser, and extensibleObject</td>
<td>userCertificate</td>
</tr>
<tr>
<td>Creating a user end-entity</td>
<td>Any supported value other than unstructuredName, unstructAddress, serialNumber, DC, dnQualifier and UID</td>
<td>account, pkiUser, and extensibleObject</td>
<td>userCertificate, and uid (hardcoded to NoUid)</td>
</tr>
<tr>
<td>Creating a CA end-entity</td>
<td>O=</td>
<td>organization, and pkiCA</td>
<td>cACertificate, certificaterevocationlist, and authorityrevocationlist</td>
</tr>
<tr>
<td>Creating a CA end-entity</td>
<td>OU=</td>
<td>organizationalUnit, and pkiCA</td>
<td>cACertificate, certificaterevocationlist, and authorityrevocationlist</td>
</tr>
<tr>
<td>Creating a CA end-entity</td>
<td>DC</td>
<td>domain, pkiCA and extensibleObject</td>
<td>cACertificate, certificaterevocationlist, and authorityrevocationlist</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2001, 2009
### LDAP directory server requirements

#### Table 88. LDAP objectclasses and attributes that PKI Services sets (continued)

<table>
<thead>
<tr>
<th>End-entity or branch node?</th>
<th>Visible RDN attribute</th>
<th>Objectclasses used</th>
<th>Additional attributes set (other than visible RDN attribute)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a CA end-entity</td>
<td>dnQualifier</td>
<td>account,</td>
<td>cACertificate, certificateRevocationList, and authorityRevocationList, and uid (hardcoded to NoUid)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uniquelyQualifiedObject, pkiCA, and extensibleObject</td>
<td></td>
</tr>
<tr>
<td>Creating a CA end-entity</td>
<td>UID</td>
<td>account,</td>
<td>cACertificate, certificateRevocationList, and authorityRevocationList</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pkiCA, and extensibleObject</td>
<td></td>
</tr>
<tr>
<td>Creating a CA end-entity</td>
<td>Any supported value other than O, OU, DC, dnQualifier and UID</td>
<td>account, pkiCA, and extensibleObject</td>
<td>cACertificate, certificateRevocationList, and authorityRevocationList and uid (hardcoded to NoUid)</td>
</tr>
<tr>
<td>User end-entity that already exists</td>
<td>unstructuredName or unstructAddress</td>
<td>pkiUser, cEPDevice</td>
<td>userCertificate</td>
</tr>
<tr>
<td>User end-entity that already exists</td>
<td>serialNumber</td>
<td>pkiUser, pKCS10Device</td>
<td>userCertificate</td>
</tr>
<tr>
<td>User end-entity that already exists</td>
<td>Any supported value other than unstructuredName, unstructAddress, and serialNumber</td>
<td>pkiUser</td>
<td>userCertificate</td>
</tr>
<tr>
<td>CA end-entity that already exists</td>
<td>Any supported value</td>
<td>pkiCA</td>
<td>cACertificate, certificateRevocationList, and authorityRevocationList</td>
</tr>
<tr>
<td>Creating a distribution point CRL end-entity</td>
<td>CN=</td>
<td>commonName and cRLDistributionPoint</td>
<td>certificateRevocationList</td>
</tr>
<tr>
<td>Distribution point CRL end-entity that already exists</td>
<td>Any supported value</td>
<td>cRLDistributionPoint</td>
<td>certificateRevocationList</td>
</tr>
</tbody>
</table>

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](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 89 indicates.

#### Table 89. Relationship of named fields to LDAP attributes and object identifiers

<table>
<thead>
<tr>
<th>Named field</th>
<th>Visible RDN attribute</th>
<th>OID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommonName</td>
<td>CN</td>
<td>2.5.4.3</td>
</tr>
<tr>
<td>Title</td>
<td>TITLE</td>
<td>2.5.4.12</td>
</tr>
<tr>
<td>OrgUnit</td>
<td>OU</td>
<td>2.5.4.11</td>
</tr>
<tr>
<td>Org</td>
<td>O</td>
<td>2.5.4.10</td>
</tr>
<tr>
<td>Locality</td>
<td>L</td>
<td>2.5.4.7</td>
</tr>
<tr>
<td>StateProv</td>
<td>ST</td>
<td>2.5.4.8</td>
</tr>
<tr>
<td>Country</td>
<td>C</td>
<td>2.5.4.6</td>
</tr>
<tr>
<td>PostalCode</td>
<td>POSTALCODE</td>
<td>2.5.4.17</td>
</tr>
<tr>
<td>Street</td>
<td>STREET</td>
<td>2.5.4.9</td>
</tr>
</tbody>
</table>
### Table 89. Relationship of named fields to LDAP attributes and object identifiers (continued)

<table>
<thead>
<tr>
<th>Named field</th>
<th>Visible RDN attribute</th>
<th>OID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email(^1)</td>
<td>MAIL</td>
<td>0.9.2342.19200300.100.1.3</td>
</tr>
<tr>
<td>Mail</td>
<td>MAIL(^2)</td>
<td>0.9.2342.19200300.100.1.3</td>
</tr>
<tr>
<td>EmailAddr</td>
<td>EMAIL</td>
<td>1.2.840.113549.1.9.1</td>
</tr>
<tr>
<td>UnstructName</td>
<td>UNSTRUCTUREDNAME</td>
<td>1.2.840.113549.1.9.2</td>
</tr>
<tr>
<td>UnstructAddr</td>
<td>UNSTRUCTUREDADDRESS</td>
<td>1.2.840.113549.1.9.8</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>SERIALNUMBER</td>
<td>2.5.4.5</td>
</tr>
<tr>
<td>DNQualifier</td>
<td>DNQUALIFIER</td>
<td>2.5.4.46</td>
</tr>
<tr>
<td>DomainName</td>
<td>DC</td>
<td>0.9.2342.19200300.100.1.25</td>
</tr>
<tr>
<td>Uid</td>
<td>UID</td>
<td>0.9.2342.19200300.100.1.1</td>
</tr>
</tbody>
</table>

\(^1\) 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).
LDAP directory server requirements
Appendix B. 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
   ```
Using gskkyman

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?”.


15. Exit gskkyman.

16. You can optionally remove the /etc/cert.arm file.
Appendix C. 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.


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.


Overview of configuring z/OS PKI Services as a CA

This topic describes the configuration required to 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

**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 495.)
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 495 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 494.)

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
<table>
<thead>
<tr>
<th>Subtask</th>
<th>Associated instructions (see ...)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you have not already done so, install and configure PKI Services as a self-signed certificate authority (CA).</td>
<td>Follow the instructions in [Part 1, &quot;Planning,&quot; on page 1, Part 2, &quot;Configuring your system for PKI Services,&quot; on page 25, and Part 3, &quot;Customizing PKI Services,&quot; on page 99.]</td>
<td>Remember to store your PKI Services CA signing key in Integrated Cryptographic Service Facility (ICSF).</td>
</tr>
<tr>
<td>Establish PKI Services as an intermediate certificate authority under the IdenTrust root.</td>
<td>Follow the instructions in [Establishing PKI Services as an intermediate CA on page 346 and Part 5, &quot;Administering security for PKI Services,&quot; on page 339.]</td>
<td>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”.</td>
</tr>
<tr>
<td>Modify the PKI Services configuration file (pkiserv.conf).</td>
<td>&quot;Steps to modify pkiserv.conf for different certificate types&quot; on page 497</td>
<td>Make sure your certificate policies are in accordance with IT-PKI.</td>
</tr>
<tr>
<td>Create IdenTrust specific certificate templates in the PKI Services certificate templates file (pkiserv.tmpl).</td>
<td>&quot;Steps to create IdenTrust specific certificate templates&quot; on page 497</td>
<td>Make sure your certificate templates are in accordance with IT-PKI.</td>
</tr>
<tr>
<td>Stop and restart PKI Services to activate your changes.</td>
<td>Follow the instructions in [Chapter 10, &quot;Starting and stopping PKI Services,&quot; on page 95.]</td>
<td></td>
</tr>
</tbody>
</table>

**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 499.

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 and UserNoticeText (the CertPolicy directives) copied in Step 2 on page 496.

6. Change the value of the PolicyName directive to match the name defined in Step 4 on page 496.

7. Change the value of the UserNoticeText directive as needed for this policy.

8. Change the policy number in the directives’ name (the nn in PolicyName and UserNoticeText) 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 496 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 499.
- For more information about creating directives for certificate policies, see “Using certificate policies” on page 188 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 499 and the sample server certificate template in “Sample server certificate template for IdenTrust compliance” on page 502.
- For details about creating certificate templates, see Chapter 11, “Customizing the end-user Web application if you use REXX CGI execs,” on page 101.
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.

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 499
  
  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 499
  
  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 502
  
  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>
<HTML><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>
<li><h3>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="return ValidateEntry(this)"/>

 # 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="return ValidateEntry(this)"/>

 # 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)"/>

Appendix C. Configuring PKI Services as an IdenTrust™ certificate authority 499
Enter values for the following field(s)

```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;
}
```

#-- End user input fields and validation Javascript -----------

Pick Up a Previously Issued Certificate

```html
<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>

%%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:%%

%%requestok%%

%%requestbad%%

%%retrievecontent%%

<title>Web Based PKIX Certificate Generation Application Pg 3</title>
```

500 z/OS V1R11.0 Cryptographic Services PKI Services Guide and Reference
<H1> Retrieve Your [tmplname]/H1>

Please bookmark this page</h3>

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)) {
   
   return true;
}
else
   return false;
}
//-->

/%-TransactionId%%
%

%/ChallengePassPhrase (optional)%%

#-- End user input fields and validation Javascript -----------

<p>

</FORM>

To check that your certificate installed properly, follow the procedure below:

Netscape V6 - 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.

Netscape V4 - Click the Security button, then Certificates->Yours. Your certificate should appear in the list. Select it then click Verify.

Internet Explorer V5 - Click Tools->Internet Options, then Content, Certificates. Your certificate should appear in the Personal list. Click Advanced to see additional information.

<FORM METHOD=GET ACTION="/[application]/public-cgi/camain.rexx">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">

<INPUT TYPE="submit" VALUE="Home page">

</FORM>

%%-pagefooter%%

</BODY>

</HTML>

%%returnbrowsercert[browsertype]%%

</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>
#<HTML><HEAD>
#<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
#%copyright%
</HEAD>
</BODY>
</HTML>
</TITLE> 4-Year Identrus EE Server Signing Certificate</H1>
<p>Choose one of the following:</p>
<h3><li>Request a New Certificate</li></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]">
</form> 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;
  }
  return false;
}
</script>

%Requestor (Optional)%
%CommonName%
%OrgUnit%
Appendix C. Configuring PKI Services as an IdenTrust™ certificate authority 503
<!--
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">
</p>
</form>

%%-pagefooter%%
</body>
</html>

</retrievcontent>
<returncert>
  %%-returnpkcs10cert%%
</returncert>
</template>
Appendix D. Using the PKI Services Web application with Internet Explorer on Windows Vista systems

In order to use the PKI Services end-user Web application through the Internet Explorer browser on a Microsoft Windows Vista system, you must first set up the Vista system to work with PKI Services. This topic describes the tasks that you need to perform.

Note: You do not need to perform these tasks for browsers other than Internet Explorer.

If you are running z/OS Version 1 Release 7, 8, or 9, you also need to update your template file, pkiservtmpl. For instructions, go to http://www.ibm.com/servers/eserver/zseries/zos/pki/vistafix.html.

Table 90. Tasks to perform to set up a Windows Vista system and the Internet Explorer browser to work with PKI Services

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Associated instructions (see . . .)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install CAPICOM on the Microsoft Windows Vista system.</td>
<td>&quot;Installing CAPICOM on a Microsoft Windows Vista System&quot;</td>
</tr>
<tr>
<td>Configure Internet Explorer to trust PKI Services.</td>
<td>&quot;Configuring Internet Explorer to trust PKI Services&quot; on page 506</td>
</tr>
<tr>
<td>Install the z/OS PKI Services certificate authority (CA) certificate on the Microsoft Windows Vista system.</td>
<td>&quot;Installing the PKI Services CA certificate on a Microsoft Windows Vista system&quot; on page 507</td>
</tr>
</tbody>
</table>

Installing CAPICOM on a Microsoft Windows Vista System

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 is required to support 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 for Vista. Doing this causes more software than required by z/OS PKI Services to be installed on the Microsoft Windows Vista system.
  - Install the CAPICOM subset of the Microsoft Platform SDK. This approach installs CAPICOM as well as software development samples on the Microsoft Windows Vista 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 Vista 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 the Continue button, and allow the browser to activate any necessary ActiveX controls to perform the verification.
Steps for installing CAPICOM on a Microsoft Windows Vista system

Before you begin: You must run as the system administrator on the Microsoft Windows Vista system on which you are installing CAPICOM.

Perform the following steps to install CAPICOM on a Microsoft Windows Vista system.

1. Determine which method of installation to use:
   - Install the full Microsoft Platform SDK for Vista.
   - 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 for Vista, 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 for Vista, 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 Vista system.

When you are done, you have installed CAPICOM on a Microsoft Windows Vista system. Continue to the next task, "Configuring Internet Explorer to trust PKI Services."

Configuring Internet Explorer to trust PKI Services

In order to request or administer certificates using the PKI Services end-user Web application through Internet Explorer on a Windows Vista system, you must add the PKI Services system to the list of trusted sites recognized by Internet Explorer.

Steps for configuring Internet Explorer to trust PKI Services

Perform the following steps to configure Internet Explorer to trust PKI Services.

1. From the Microsoft Windows Vista 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 labelled "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 the Sites button to see the "Trusted Sites" panel.
6. In the area labelled “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 the Add button to add this site to the list of trusted sites.

7. Leave the box labelled “Require server verification (https:) for all sites in this zone” checked. Click the Close button to close the “Trusted Sites” panel and return to the “Security” tab of the “Internet Options” panel.

8. Click the Apply button on the “Internet Options” panel to confirm the configuration changes. Then click the OK button 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. Continue to the next task: “Installing the PKI Services CA certificate on a Microsoft Windows Vista system.”

---

**Installing the PKI Services CA certificate on a Microsoft Windows Vista system**

Any system using Microsoft Windows Vista operating systems 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. While 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 Vista 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 Vista 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 Save.
3. Internet Explorer displays a pop-up panel to warn that a Web site wants to open Web content using the browser. 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. Select **Place all certificates in the following store** and click **Browse...**.

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.
Appendix E. Accessibility

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/Library Server versions of z/OS books in the Internet library at: http://www.ibm.com/systems/z/os/zos/bkserv/
Notices

This information was developed for products and services offered in the USA.

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
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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.

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:

IBM Corporation
Mail Station P300
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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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.

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](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 are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows 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.

Other company, product, or service names may be trademarks or service marks of others.
Bibliography

The following lists the titles and numbers of documents referenced in this publication.

- 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
- 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 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
Index

Special characters

CEE_RUNOPTS, updating 372
_PKISERV_CA_DOMAIN environment variable 445
_PKISERV_CONFIG_PATH environment variable 445
_PKISERV_ENABLE_JSP
  description 447
_PKISERV_EXIT
  adding to environment variables file 236
  adding to Web server's environment variables file 240
  description 445
_PKISERV_MSG_LEVEL
  description 446
  message levels 401
  subcomponents 401
_PKISERV_MSG_LOGGING
  STDERR_LOGGING 446
  STDOUT_LOGGING 446
-AdditionalHeadIE 104
-ChallengePassphrase 104
-ChallengePassphrase2 104
-preregok 105
-RecoverEmail 104
-RecoverEmail2 104
-renewkeysetIE 105
-renewkeysetNS 105
-renewrevokead 105
-renewrevokeok 105
-requestbad 104
-requestok 104
-requestor 104
-requestor2 105
-returnp12cert 105
/etc/pkiserv 9
/usr/lpp/pkiserv 9
description 9
subdirectories 437
/var/pkiserv 9
description 9
[
] for substitution variables 102
% in named fields 103
%-renewrevokead% 111
%-renewrevokeok% 111
%-requestor% 119
%-AltOther_1_2_3_4_5% 147
%-AltOther_1_2_3_4_6% 147
%-cadomain% variable 228
%-dn% variable 228
%-notafter% variable 228
%-pendreqlist% variable 228
%-printcert% variable 228
%-quicklink% variable 229
%-recoverylink% variable 229
%-recoverylist% variable 229
%-requestor% variable 229
%-returnp12cert% 111
%-SelectCADomain% 112

Numerials

n-year PKI extensions demonstration certificate
  fields 122

Numerics

1YBSM 114
1YBSSL 114
2YBZOS 114
2YIACS 114
5YSAC 114
5YSCEPP 114
5YSIPS 114
5YSSSL 114

A

abends, recording 391
access
  READ, authorizing 342
  required for administrator 450, 451
  to administration pages 299
  changing 155
  to end-user Web pages 274
  to RACF group 27
  to VSAM data sets 27
access control list (ACL), LDAP 74
access control, setting up 27, 449
accessibility 509
ACL (access control list), LDAP 74
actions
  on certificate requests 303
  on certificates 313
active (status of certificate) 313
active, AutoRenewDisabled (status of certificate) 313
active, NotRenewable (status of certificate) 313
active, AutoRenew (status of certificate) 313
addtype directive 79
admactcert.rexx 153
admacttid.rexx 152
admacttid2.rexx 152
admicl.rexx 152
admicall.rexx 153
admicicert.rexx 152
ADMINAPPROVE subsection (in TEMPLATE section of pkiserv.tmpl) 118
adminDN keyword 73, 78
ADMINFOOTER subsection (in APPLICATION section of pkiserv.tmpl) 112, 154
ADMINHEADER subsection (in APPLICATION section of pkiserv.tmpl) 112, 153
administration
  changing log options 401
  changing requestor e-mail 315
deleting certificate requests 308
deleting certificates 315
administration (continued)
deleting preregistered requests 308
disable automatic renewal of certificates 315
displaying log options settings 402
enable automatic renewal of certificates 315
log options
  changing 401
  displaying 402
modifying certificate requests 306
processing
  certificate requests using searches 309
  certificates using searches 316
  multiple certificate requests 310
  multiple certificates 316
  selected certificate requests 310
  selected certificates 318
  single certificate 314
  single certificate request 304
  single preregistered request 304
RACF
  ongoing administration 341
  running IKYSETUP 27
rejecting certificate requests 308
resuming certificates 315
revoking certificates 315
searching
  certificate requests 308
  certificates 316
  selected certificate requests 310
  selected certificates 318
starting PKI Services 95
stopping PKI Services daemon 97
suspending certificates 315
administration group
  setting up 27
administration group PKIGRP 39
administration home page
  accessing 299
  using 303
administration tasks
  PKI Services 299
    processing certificate requests 302
    processing certificates 312
  RACF
    ongoing administration 341
    running IKYSETUP 27
administration Web application
  PKI Services component description 4
administration Web pages
  changing access to 155
  customizing 151
  fields 302
  steps for customizing 154
  using 299
administration, RACF
  classroom courses xviii
administrative functions
  protecting 27, 451
administrator
  access required 450
administrator notification
  copying files for 48
AdminNotifyForm (parameter in pkiserv.conf) 65
AdminNotifyNew (parameter in pkiserv.conf) 56
AdminNotifyReminder (parameter in pkiserv.conf) 56
adminPW
  LDAP server configuration file 73, 78
ADMINSCOPE subsection (in APPLICATION section of pkiserv.tmpl) 112, 154
admmain.rexx 152
admmoid.rexx 152
admpend.rexx 152
admpendall.rexx 153
admpendtid.rexx 152
advanced customization 187
AIEALNKE 437
alias
  file (for sendmail) 22
  for certificate template
    changing 139
    description 114
    list 114
    for database entry 253
    for IKYSPROC 484
    PKISERVD 484
AltDomain (named field in pkiserv.tmpl) 105
AltEmail (named field in pkiserv.tmpl) 105
alternate name
  domain name 278
  e-mail address 278
  IP address 278
  other name 278
  uniform resource identifier (URI) 278
AltIPAddr (named field in pkiserv.tmpl) 106
AltOther (named field in pkiserv.tmpl) 106
AltURI (named field in pkiserv.tmpl) 106
APPL subsection (in TEMPLATE section of pkiserv.tmpl) 116
application domain
  adding 200
  steps for adding, when you use JSPs 203
  steps for adding, when you use REXX CGIs 202
APPLICATION section of pkiserv.tmpl
  ADMINFOOTER subsection 154
  ADMINHEADER subsection 153
  ADMINSCOPE subsection 112, 154
  APPLICATION sections of pkiserv.tmpl
    ADMINFOOTER subsection 112
    ADMINHEADER subsection 112
    CONTENT subsection 111
    FAILRECONTENT subsection 111
    FINDRECOVERCONTENT subsection 111
    RECONTENT subsection 111
    RECONTENT2 subsection 111
    REFAILRECONTENT subsection 111
    RESUCCESSCONTENT subsection 111
    RETRIEVECONTENT2 subsection 111
    RETURNCONTENT2 subsection 111
    subsections 110
approve (action on certificate request) 303
approve with modifications (action on certificate request) 303
approved (status of certificate request) 302
APROCLIB 437
ARL (authority revocation list) distribution point 199
ARLDist (parameter in pkiserv.conf) 56, 197
ASAMPLIB 437
associating user ID with PKI Services started procedure 449
attributes
HIGHTRUST 343
LDAP, that PKI Services requires 487
NOPASSWORD 449
OU 84, 86
PROTECTED 450
RDN 487
RESTRICTED 33, 450
AuthName1 (parameter in pkiserv.conf) 83
AuthorityInformationAccess certificate extension 370
AuthorityKeyIdIdentifier certificate extension 370
CRL extension 371
authorization checking, using exit 239
AuthPwd1 (parameter in pkiserv.conf) 84
auto-approval
access required 451
indicator in pkitmpl.xml 159
of certificates 112, 113, 114
automatic certificate renewal
exit processing for 235
indicator in pkitmpl.xml 159
setting up 230
automatic deletion
from ICL 55
from ObjectStore 54, 55
automatic renewal of certificates 115
AUTORENEW tag on certificate templates 115

B
backing up
VSAM data sets
sample JCL 481
backup_dsn (variable in IKYSETUP) 37
base64-encoded
#10 certificate request 109
certificate 102, 135, 136
response 6
base64-encoded PKCS (field in end-user Web pages) 279
base64cert substitution variable 102, 105, 135
basic constraints extension, including in certificate request 108
BasicConstraints (certificate extension) 370
bibliography 515
bin subdirectory 437
bind passwords for LDAP
encrypted 361
in the clear 361
binding
distinguished name for LDAP 83
passwords for LDAP servers 84
BindProfile1 (parameter in pkiserv.conf) 85
bp_x_userid. (variable in IKYSETUP) 34
brackets (in substitution variables) 102
browser certificates
n-year PKI certificate for extensions demonstration 114
aliases 114
IdenTrust compliant template 499
installing 286, 287
n-year PKI certificate for extensions demonstration 114
one-year PKI S/MIME browser certificate 113
one-year PKI SSL browser certificate 113
one-year SAF browser certificate 112
requesting 280
retrieving 286, 287
supported types 7
two-year PKI browser certificate for authenticating to z/OS 113
browsertype substitution variable 102

C
CA (certificate authority)
overview 3
CA certificate
backing up 27
creating 451
using IKYSETUP 27
exporting 27
installing 275, 300
installing on Windows Vista 507
key rollover 354
locating 344
rekeying 354
renewing 349
CA certificate and private key
backing up 452
CA certificate profile, recovering 351
CA domains
adding 204
CA functions
authorizing PKI Services daemon user ID for 27
CA revocation list (ARL) 198
c_1_dn (variable in IKYSETUP) 29
c_1_domain (variable in IKYSETUP) 37
c_1_expires (variable in IKYSETUP) 38
c_1_keysiz (variable in IKYSETUP) 34
c_1_label (variable in IKYSETUP) 30
c_1_ring (variable in IKYSETUP) 38
CA, intermediate
establishing PKI Services as 347
cacert_dsn (variable in IKYSETUP) 38
cadisplay.rexx 137
cadomain substitution variable 102

Index 519
certificate authority (CA) certificate backing up 27 certificate creating 27 certificate exporting 27 certificate installing 275, 300 certificate renewing 349 certificate overview 3 certificate data set, editing 348 certificate extensions customizing 8 certificate host identity mapping 8 certificate in PKI Services 8 certificate standard 8 certificate supported by PKITP 370 certificate generation application Web page 111 certificate policies CertificatePolicies extension 188 PKITP supports 369 certificate preregistration certificate enabling in pkiserv.conf 58 certificate profile, recovering 351 certificate renewal, automatic certificate exit processing for 235 certificate requests certificate actions on 303 certificate changing 306 certificate deleting 308 certificate modifying 306 certificate processing 299 certificate multiple 310 certificate selected 310 certificate single 304 certificate using searches 308 certificate processing selected 310 certificate rejecting 308 certificate relationship with certificates 320 certificate searching 308 certificate states 302 certificate statuses 302 certificate updating 306 certificate revocation list (CRL) (continued) certificate suspended certificates on 313 certificate time interval between issuances 62 certificate validity period 58 certificate revocation list (CRL), allowing access to 74 certificate serial number incremeneter, restoring 352 certificate store, using gskkyman for 491 certificate suspension grace period 59 certificate templates certificate adding 144 certificate alias 114 certificate AUTORENEW tag 115 certificate file certificate customization, additional first-time 138 certificate customization, minimal 137, 138 certificate customizing for IdenTrust compliance 493 certificate customizing the OtherName field for REXX CGI execs 148 certificate for JSPs 157 certificate retrofitting release changes 142 certificate IdenTrust compliant certificate browser certificate 499 certificate server certificate 502 certificate name 114 certificate nickname 114 certificate one-year PKI generated key 113 certificate pkiserv.tmpl 112 certificate subsections, summary 121 certificate true name 114 certificate validation service 367 CertificateIssuer (CRL entry extension) 371 CertificatePolicies extension certificate defining 188 certificate in certificate 60 certificate organization name for 61 certificate PolicyCritical (parameter in pkiserv.conf) 60 certificate supported by PKITP 370 certificate using 188 certificates certificates actions on 313 certificate auto-approval 112, 113, 114 certificate automatic renewal 230 certificate capturing 239 certificate changing requestor e-mail 315 certificate deleting 315 certificate disabling automatic renewal certificate by administrator 315 certificate enabling automatic renewal certificate by administrator 315 certificate extensions 8 certificate generating keys for 64, 232 certificate locating 344 certificate preregistering a SCEP client 296 certificate preregistration 114 certificate processing 299, 312 certificate processing selected 318 certificate recovering if PKI Services generated the keys 293 certificate relationship with certificate requests 320 certificate renewing certificate via Web page, steps for 289 certificate renewing automatically 230
certificates (continued)
requesting 280
resuming 315
retrieving
from bookmarked page 286
from home page 287
revoking
by administrator 315
by user 292
searching 316
single 314
standard extensions 8
states 313
statuses 313
supported types 7
suspending
by administrator 315
by user 292
uses 7
X.509v3 support 8
certificates, authorization for creation 74
Certification Practice Statement
Uniform Resource Identifier 56
CertPlist class 267
CertPolicy section (of pkiserv.conf)
default values 56
description 51
excerpt 51
information needed 56
CF lock structure
defining in SMS base configuration 89
defining to MVS 89
CGI debugging, flag in pkiserv.tmpl 101
CGIs
admactcert.rexx 153
admaccttid.rexx 152
admaccttid2.rexx 152
admicl.rexx 152
admiclall.rexx 153
admiclcert.rexx 152
admmain.rexx 152
admmodlid.rexx 152
admpend.rexx 152
admpendall.rexx 153
admpendlid.rexx 152
cadisplay.rexx 137
cagetcert.rexx 136
cagorcerv.rexx 137
camain.rexx 136
camodify.rexx 137
careq.rexx 136
carettrieve.rexx 136
catmpl.rexx 136
summary 151
chains 367
challenge passphrase (field in end-user Web pages) 279
ChallengePassPhrase (named field in pkiserv.tmpl) 106
change requestor e-mail (action for certificate) 314
changes
summary xxiii
check boxes 318
CISIZE statements 90
classroom courses, RACF xviii
clear, LDAP bind passwords in 361
client user ID 145
ClientName (named field in pkiserv.tmpl) 106
code samples
configuration directives 471, 472
configuration file 441
environment variables file 448
expiringmsg.form 227
httpd.conf 471
httpd2.conf 472
IKYCVSAM 473
IKYRVSAM 477
IKYSETUP 455
IKYVBKUP 481
IKYVREST 483
JCL to back up VSAM data sets 481
JCL to create VSAM data sets
not using RLS 473
using RLS 477
JCL to restore VSAM data sets 483
pendingmsg.form 227
pkiserv.conf 441
pkiserv.envars 448
pkiserv.tmpl
APPLICATION section 126
INSERT section 133
TEMPLATE section 130
PKISERVD 484
pkitpsamp.c 377
procedure to start PKI Services daemon 484
readymsg.form 226
recoverymsg.form 228
rejectmsg.form 227
renewcertmsg.form 227
code signing server certificate
fields 123
Common Data Security Architecture (CDSA) 3, 4
common name (field in end-user Web pages) 277
CommonName (named field in pkiserv.tmpl) 106
completed (status of certificate request) 303
compliance with IdenTrust 493
components
diagram 5
in message numbers 405
components of PKI Services 4
configurable section of IKYSETUP 27
configuration directives
example 471, 472
configuration file
example 441
for SSL traffic 77
pathname 445
updating 66
for IdenTrust compliance 499
overview 50
steps 52
Index 521
configuration, PKI Services
   testing  95
configuring
   system for PKI Services  25
CONSTANT subsection (in TEMPLATE section of pkiserv.tmpl)  116
CONTENT subsection
   in APPLICATION section of pkiserv.tmpl  111
   in TEMPLATE section of pkiserv.tmpl  116
CONTROL access for IRR.DIGTCERT.GENCERT  451, 453
core function  447
CORE subcomponent  447
country (field in end-user Web pages)  277
Country (named field in pkiserv.tmpl)  107
courses about RACF  xviii
cryptographic service provider (field in end-user Web
   pages)  280
cryptography standards supported  6
CRYPTOZ class
   profiles to allow PKI Services to generate key
   pairs  454
cryptoz_grp (variable in IKYSETUP)  34
CSECTs
   IKY8B  392
   IKYP0N  391
   IKYP81  391
   IKYP8A  391, 392
   IKYP8B  391
   IKYSCHDR  393
   IKYSTART  393
   IKYTIMER  394
CSF:SCSFMOD0  36
CSF:SCSFMOD1  36
csfkeys_profile (variable in IKYSETUP)  35
csferv_profile (variable in IKYSETUP)  35
csfusers_grp (variable in IKYSETUP)  35
CSSM_TP_PassThrough
   DBList  374
   evidence  374
   format  373
   functions
      CertGroupVerify  367
      FreeEvidence  367
   initial policy  374
   parameters  373
   performing certificate validation  376
   purpose  373
   return codes  375
CUSTOMERS
   application name  128
customization
   advanced  187
customizing
   certificate templates file
      minimal  137
certificate ready message form
      minimal  137
certificate reject message form
      minimal  137
end-user Web pages
   minimal  137
pkiserv.tmpl
   minimal  137
D
   debug flag  101
decision tables
      key_backup in IKYSETUP  32
      key_type in IKYSETUP  32
      restrict_surrog in IKYSETUP  33
   DB subcomponent  447
   DB2  20
decision tables
   key_backup in IKYSETUP  32
   key_type in IKYSETUP  32
   restrict_surrog in IKYSETUP  33
   unix_sec in IKYSETUP  33
522  z/OS V1R11.0 Cryptographic Services PKI Services Guide and Reference
default
/etc/pkiserv 9
/usr/lpp/pkiserv 9
/var/pkiserv 9
binding information 361
CA private key size 34
CA-ring (SAF key ring) 38
certificate, data set for backup copy of 37
configuration file for z/OS HTTP Server 78
daemon user ID
  PKI Services 38
  Web server 40
data set
  for copy of PKI Services certificate and private key 37
  for copy of PKI Services certificate to assist backup process 38
  for copy of PKI Services RA certificate and private key 39
  for Web server’s root CA certificate for copying to file system 38
data set for backup copy of PKI Services certificate, private key 37
data set for backup copy of PKI Services RA certificate, private key 39
environment variables file 48
file locations 70
high-level qualifier 52
ICSF
  profile to protect ICSF services 35
  profile to protect PKI Services key 35
installation directory 9
key, data set for backup copy of 37
key, data set for backup copy of RA certificate 39
message level 401, 446
OMVS KERN (z/OS UNIX user ID) 34
PKI Services
  administration group 39
  configuration file 95
  daemon user ID 38, 66
  surrogate user ID 39
PKISERV 39
pkiserv.conf file values 52
PKISRVD (PKI Services daemon user ID) 38
primary and secondary extent allocations (in JCL) 87
RA certificate, data set for backup copy of 39
runtime directory 9
SAF key ring 66
sendmail location 49
sha-1WithRSAEncryption 62
STDOUT LOGGING 446
surrogate user ID for PKI Services 39
time zone 96
UNIX user ID 34
variables directory 9
VSAM data set name
  ICL base cluster 54
  ICL requestor alternate index 54
  ICL status alternate index 54
default (continued)
VSAM data set name (continued)
  ObjectStore alternate index 53
  ObjectStore base cluster 52
  status 53
  Web server’s daemon user ID 40
  WEBSRV 40
delete (action for certificate) 314
delete (action on certificate request) 303
diagnostic messages, logging 447
diagram, PKI Services system 5
digital certificates
  exporting certificates using R_PKIServ callable service 357
  generating certificates using R_PKIServ callable service 357
  retrieving certificates using R_PKIServ callable service 357
DIR parameter 49
directives
  addtype 79
  example 471, 472
  exec 79
  FastCGI 79
  keyfile 79
  log 80
  normalmode 78, 79
  pass 79
  protect 79
  redirect 79
  sslclientauth 79
  sslmode 78, 79
  sslport 78, 79
  sslroot 79
  sslrootca 79
  sslx500caroots 79
  sslx500certs 79
  sslx500host 79
  sslx500password 79
  sslx500port 79
  sslx500user 79
  userid 78, 79
directories
/etc/pkiserv 9
/usr/lpp/pkiserv 9, 437
/var/pkiserv
  description 9
  for installation 9
  for runtime 9
  for variables 9
  runtime 95
  structure 437
  var
    setting up 70
directory server, LDAP
  planning for 10
  requirements 487
disability 509
disable (action for certificate) 314
distinguished name
e-mail address 277
for LDAP binding 83
LDAP administrator’s 73, 78
distinguished name (continued)
qualifier 277
qualifiers
  Email (deprecated) 107
  EmailAddr 107
  Mail 108
  PostalCode 109
  Street 110
distinguished name qualifier
  field in end-user Web pages 277
distribution point ARL 199
distribution point CRLs
customizing 192
DN fields
  mapping to LDAP attributes 488
DNQualifier (named field in pkiserv.tmpl) 107
documents
  bibliography 515
  configuring UNIX runtime environment 47
  installing prerequisite products
    ICSF 21
    LDAP 20
    OCSF 23
    sendmail 22
    z/OS HTTP Server 17
  RACF administration 28
  UNIX programmer 47
Domain component
  field in end-user Web pages 277
domain name
  fully qualified, for LDAP 73, 78, 83
domain name, alternate
  field in end-user Web pages 278
DomainName (named field in pkiserv.tmpl) 107
DP CRLs
customizing 192
DSA
  signature algorithm
    SigAlg1 parameter 62
      updating 191
  standard supported 7
E
  e-mail
    applications 3
      secure 3
    e-mail address
      for alternate name (field in end-user Web pages) 278
      for distinguished name (field in end-user Web pages) 277
      for notifications 280
e-mail notifications (continued)
    environment variables, updating for 49
    ExpireWarningTime
      description 59
      updating 67
    ExpiringMessageForm
      description 65
      updating 68
    expiringmsg.form
      copying 48
      description 45
    forms for
      expiringmsg.form 227
      pendingmsg.form 227
      readymsg.form 226
      recoverymsg.form 228
      rejectmsg.form 227
      renewcertmsg.form 227
    NotifyEmail 108
    pendingmsg.form
      copying 48
      description 45
    NotifyEmail 108
    pkiserv.conf
      updating, overview 50
      updating, steps 66
    ReadyMessageForm
      description 64
      updating 68
    readymsg.form
      copying 48
      description 47
    RecoverForm
      updating 68
    recovermsg.form
      copying 48
      description 47
    RejectMessageForm
      description 64
      updating 68
    rejectmsg.form
      copying 48
      description 47
    RenewCertForm
      updating 68
    renewcertmsg.form
      description 47
    retrieving your certificate 286
    updating
      environment variables 49
      ExpireWarningTime 67
      expiringmsg.form 230
      readymsg.form 230
      recoverymsg.form 230
      rejectmsg.form 230
    variables 228
      %cadomain% 228
      %dn% 228
      %notafter% 228
      %pendreqlist% 228
      %printcert% 228
      %quicklink% 229
e-mail notifications (continued)
   variables (continued)
      %%%recoverylink%%  229
      %%%recoverylist%%  229
      %%%requestor%%  229
      %%%transactionid%%  229

   EAR file for JSPs
      updating  176

   editing
      IKYSETUP  41
   Email (deprecated field in pkiserv.tmpl)  107
   EmailAddr (named field in pkiserv.tmpl)  107
   enable (action for certificate)  314
   EnableSCEP (parameter in pkiserv.conf)  58

   encrypted passwords for LDAP servers  85
      BindProfile1
         description  85
         RACF administration  361
         storing information for  81
         updating LDAP section of pkiserv.conf  82
   encryption  191
   end-user function  357
   end-user functions
      protecting  27, 449
   end-user Web application
      PKI Services component
         description  5
   end-user Web pages
      accessing  274
      code locations  143
      customizing  101
      additional first-time  138
      minimal  137
      OtherName field for REXX CGI execs  148
   fields  276
      using  273
   environment variables
      _PKISERV_CA_DOMAIN  445
      _PKISERV_CONFIG_PATH  445
      _PKISERV_ENABLE_JSP  447
      _PKISERV_EXIT  236, 240, 445
      _PKISERV_MSG_LEVEL  446
      _PKISERV_MSG_LOGGING  446
   code sample file  448
      description  446
      file name
         DIR parameter  49
         FN parameter  49
      in PKISERVD  484
      TZ  49
      updating
         overview  49
         steps  50
   error messages
      list  405
   error messages, logging  447
   errorinfo substitution variable  102
   EVENT  96
      event code, SMF  485
   examples
      _PKISERV_CA_DOMAIN  445

   examples (continued)
      _PKISERV_MSG_LEVEL  446
      configuration directives  471, 472
      configuration file  441
      environment variables file  448
      expiringmsg.form  227
      httpd.conf  471
      httpd2.conf  472
      IKYCVSAM  473
      IKYRVSAM  477
      IKYSETUP  455
      IKYVBKUP  481
      IKYVREST  483
   JCL
      certificate serial number incremeneter,
         restoring  352
      IKYCVSAM  474
      IKYRVSAM  478
      IKYVBKUP  481
      IKYVREST  483
      PKISERVD  484
   log options settings  402
   LOGREC data  394
   named field  103
   output from displaying log options settings  402
   pendingmsg.form  227
   pkiserv.conf  441
   pkiserv.envars  448
   pkiserv.tmpl
      APPLICATION section  126
      INSERT section  133
      TEMPLATE section  130
   PKISERVD  484
   pktsamp.c  377
   procedure to start PKI Services daemon  484
   readymsg.form  226
   recoverymsg.form  228
   rejectmsg.form  227
   renewcertmsg.form  227
   substitution variable  102
   excerpt
      pkiserv.conf
         CertPolicy section  51
         General section  51
         LDAP section  81
         ObjectStore section  51
         OIDs section  51
         SAF section  51
      pkiserv.tmpl  126
   exec directive  79
   exit
      _PKISERV_EXIT environment variable  445
      arguments  240
      description of  5
      environment variable  445
      pathname  445
      post-processing  243, 245
         EXPORT  247
         GENRENEW  245
         REQRENEW  245
         REVOKE  249, 251
exit (continued)

postprocessing
  automatic renewal  238
preprocessing  244
  automatic renewal  237
EXPRT  246
GENCERT  242
GENRENEW  242
QRECOVER  250
REGRENEW  244
REVOKE  248

scenarios
  allowing only selected users to request
    certificates  252
  automatic renewal of certificates  239
  maintaining customized certificate repository  253
  recover lost passphrase  254
    renewal only within 30 days of expiration  253
  updating sample code  236
  updating sample code for CGIs  240
  using  235

exit methods for JSPs  254
ExitTimeout (parameter in pkiserv.conf)  63, 236
expired (status of certificate)  313
ExpireWarningTime (parameter in pkiserv.conf)  59
expiring message form for certificate  65
ExpiringMessageForm (parameter in pkiserv.conf)  65

expiringmsg.form
  code sample  227
  copying  48
  customizing  230
  in samples directory  439
  purpose  45

EXPORT
  accesses required  359
  description  253
  parameters
    post-processing  247
    preprocessing  246
  R_PKIServ function  450
  return codes
    post-processing  247
    preprocessing  246
export_dsn (variable in IKYSETUP)  38
ExportCert class  266
extended key usage (field in end-user Web
  pages)  278
extensions
  CertificatePolicies  60
  supported by PKITP  370
  X.509 version 3 standard  8
extensions demonstration
  certificate
    fields  122
extent allocations in IKYCVSAM  87
ExtKeyUsage (named field in pkiserv.tmpl)  107

F

FACILITY class profile
  IRR.RPKISERV.PKIADMIN  360

FAILURECONTENT subsection
  in APPLICATION section of pkiserv.tmpl  111

FAILURECONTENT subsection (in TEMPLATE section
  of pkiserv.tmpl)  119
FastCGI directive  79
field in end-user Web pages  278
fields
  administration Web pages  302
  end-user Web pages  276
  in IKYSETUP REXX exec
    change based on setup  32
    change optionally  37
    change required  29
    modifiable by administrator  307
    X.509 version 3 standard  8

file directory structure  437
file system
  installation directory  9
  runtime directory  9
  subdirectories  437
files
  CGIs
    administrator Web pages  152
    end-user Web pages  135
    for configuring PKI Services  47
    default locations  403
    exit  235
  expiringmsg.form
    code sample  227
    copying  48
    customizing  230
  httpd.conf
    code sample  471
    updating  78
  httpd1443.conf
    source for  472
    updating  79
  httpd2.conf
    code sample  472
    copying from  79
  IKYCVSAM
    code sample  474
    copying  89
    extent allocations  87
    updating  89
  IKYRVSAM
    code sample  478
    copying  90
    updating  90
  IKYSETUP
    code sample  455
  IKYVBKUP
    code sample  483
  IKYVREST
    code sample  483
  JSPs  182
  Makefile.pkiexit  235
files (continued)
  pendingmsg.form
    code sample 227
copying 48
  pkixexit.c 235
  pkiserv.conf
    code sample 441
    copying 48
    updating 50, 50, 82, 189, 190
  pkiserv.envars
    code sample 448
    copying 48
    updating 50
  pkiserv.tmpl
    additional customization 138
    contents 101
    copying 48
    minimal customization 137
    retrofitting changes 142
  PKIServ.xsd
    copying 48
    description 157
  PKISERVVD
    code sample 484
    updating 50
  pkitmpl.xml
    copying 48
    description 157
    updating 176
  PKITP 371
  pktp.h 437
  pktp.so 438
  pkixgen.tmpl
    description 157
  readymsg.form
    code sample 226
    copying 48
    customizing 230
  recoverymsg.form
    code sample 228
    copying 48
    customizing 230
  rejectmsg.form
    code sample 227
    copying 48
    customizing 230
  renewcertmsg.form
    code sample 227
  FINDRECOVERCONTENT subsection
  in APPLICATION section of pkiserv.tmpl 111
  firewall certificate
    description 113
    fields 123
  five-year PKI intermediate CA certificate
    description 114
    fields 123
  five-year PKI IPSEC server (firewall) certificate
    description 113
    fields 123
  five-year PKI SSL server certificate
    description 113

G
GENCERT
  accesses required 359
  exit scenario use 252, 253
  parameters
    post-processing 243
    preprocessing 242
  R_PKIServ function 450
  return codes
    post-processing 243
    preprocessing 242
General section (of pkiserv.conf)
  default values 63
  description 51
  excerpt 51
  information needed 63
  generated key pairs for certificates
  setting up ICSF 21
  TokenName keyword in configuration file 66
GENRENEW
  accesses required 359
  exit scenario use 253
  parameters
    post-processing 243
    preprocessing 242
  R_PKIServ function 450
  return codes
    post-processing 243
    preprocessing 242
getCertificate method 266
getPassphrase method 266
getTransactionid method 266
GLD.SGLDLNK 36
global ARL (CA revocation list) 198
grace period, certificate suspension 59
groups
  authorizing 342
  deleting 342
GSK.SGSKLOAD 36
gskkyman 491

H
HIghTRUST attribute 343
HoldInstructionCode (CRL entry extension) 371
host identity mapping 8
HostIdMap (named field in pkiserv.tmpl) 107
HostIdMappings
  field (on end-user Web pages) 278
HostIdMappings extension
  administering 342, 343
  PKITP support 370
HTTP server
  installing and configuring 17
  requirement for 10
  starting 80
  updating configuration 77
httpd.conf 471
  application domains
    adding 202
    editing
      to add application domains 202
httpd.envvars 240
httpd1443.conf 77
  editing log directives in 80
httpd2.conf 472

I
ICL
  base cluster
    VSAM data set name for 54
    certificates maintained in 313
    requestor
      VSAM data set name for 54
      space considerations 87
      status alternate index
        VSAM data set name for 54
        time period before automatic deletion expired certificates 55
ICL data sets and indexes
  creating 89
ICLD5N (parameter in pkiserv.conf) 54
ICLRequestorDSN (parameter in pkiserv.conf) 54
ICLStatusDSN (parameter in pkiserv.conf) 54
iclview
  examples 323
  format 322
  parameters 322
  purpose 322

ICSF
  authorizing PKI Services 27
  description of PKI Services component 5
  installing and configuring 21
  migrating CA certificate and private key to 452
  migrating RA certificate and private key to 452
  PKI Services component
    description 5
    public key data set (PKDS) 21
    requirement for 11
    token data set (TKDS) 21
ICSF programmer
  installing and configuring ICSF 21
  skills 12
  tasks 12
  team member 11
IDCAMS 87
IdenTrust, configuring PKI Services for compliance
  adding an intermediate CA 494
  adjusting general settings 494
  defining templates 495
  introduction 493
  roadmap 495
  samples for IdenTrust compliance
    browser certificate template 499
    configuration file 499
    server certificate template 502
  steps
    adjusting general settings 497
    creating templates 497
    modifying pkiserv.conf 496
    system prerequisites 494
  task overview 494
  task roadmap 495
iecert substitution variable 102
IKY8B CSECT 392
IKYALLOC 437
IKYCVSAM
  copying 89
  creating VSAM data sets 87
  extent allocations 87
  in SAMPLIB 437
  sample 473
  updating 89
IKYDDDEF 437
IKYISMKD 437
IKYMKD1R 437
IKYP0N CSECT 391
IKYP81 CSECT 391
IKYP8A CSECT 391, 392
IKYP8B CSECT 391
IKYPKID 437
IKYPRMT 437
IKYVSAM
  copying 90
  in SAMPLIB 437
  sample 477
  updating 90
IKYSCHDR CSECT 393
IKYSETUP REXX exec
  actions 449
PUBLICKEYIE 109
PUBLICKEYNS 109
RecoverEmail 109
Requestor 109
Requestor2 109
returnbrowsercertIE 105
returnbrowsercertNS 105
Security 109
SerialNumber 110
SignWith 110
StateProv 110
Street 110
Title 110
TransactionId 110
Uid 110
UnstructAddr 110
UserId 110
install_pkitp 371, 372
install-dir 9
installation directory 9
installation exit 235
installing
PKI Services 9
  skills 13
prerequisite products
  skills 12
intermediate CA
certificate
  description 114
  description of template 114
  fields 123
  template description 114
establishing PKI Services as 347
intermediate certificate authority
  establishing PKI Services as 346
Internet Explorer
  configuring to trust PKI services 506
  key protection field on end-user Web page 280
  requesting a certificate 281
  selecting a key size 282
  setting up to run end-user Web application on
  Windows Vista 505
  supported standard 6
  verifying certificate installed correctly 287, 288
Internet Protocol Security standard (IPSEC) 3
  interval
  before certificate expiration 59
  between certificate revocation lists 58, 62
  certificate suspension grace period 59
  scanning database for approved requests 57
  warning message about certificate expiration 59
introduction to PKI Services 3
InvalidityDate (CRL entry extension) 371
IP address
  AltIPAddr field 106
  format 106
  LDAP fully qualified domain name 73
IP address, alternate
  field in end-user Web pages 278
IPSEC
certificate format 6
certificates 7
  supported standard 3
IRR.DIGTCERT.ADD 359, 451
IRR.DIGTCERT.CERTIFAUTH.* 35
IRR.DIGTCERT.EXPORT 359
IRR.DIGTCERT.GENCERT 359, 451, 453
IRR.DIGTCERT.GENERENEW 359, 451
IRR.DIGTCERT.LIST 453
IRR.DIGTCERT.LISTRING 453
IRR.DIGTCERT.RECOVERY 359
IRR.DIGTCERT.REQCERT 359, 451
IRR.DIGTCERT.REQRENEW 359, 451
IRR.DIGTCERT.RESPOND 359
IRR.DIGTCERT.REVOKE 360
IRR.DIGTCERT.SCEPREQ 360
IRR.DIGTCERT.SCEPREQ 360
IRR.DIGTCERT.SCEPREQ 360
IRR.DIGTCERT.SCEPREQ 360
IRR.DIGTCERT.SCEPREQ 360
IRR.PROXY.DEFAULTS 82, 361, 362
IRR.PROXY.DEFAULTS profile in FACILITY class 361
IRR.RPKISERV 358
IRR.RPKISERV.PKIADMIN 360, 451
IRRSPX00 callable service
  controlling the use 357
IRRSPX00 SAF callable service
  exits 240
  PKI Services component description 5
  issued certificate list (ICL) 313
IssuerAltName
  certificate extension 370
IssuerAltName (CRL extension) 371
IssuingDistributionPoint (CRL extension) 371
J
Java server pages (JSPs)
  deploying to a Websphere application server 178
  description 157
  directories for 182
  environment variable for enabling 447
  exit processing for 254
  implementing the Web application using 157
  updating 176
  XML template for 157, 176
JCL
  backing up VSAM data sets 481
  creating VSAM data sets
    not using RLS 473
    using RLS 477
  example
    certificate serial number incremeneter, 352
    restoring 352
    IKYCVSAM 474
    IKYRVSAM 478
    IKYVBKUP 481
    IKYVREST 483
    PKISERVD 484
EXEC card, PARM= operand limitation 50
  restoring VSAM data sets 483
  VSAM data sets 481, 483
  not using RLS 473
JCL (continued)
VSAM data sets (continued) using RLS 477
JOB card 89
JSRs (Java server pages)
deploying to a Websphere application server 178
description 157
directories for 182
exit processing for 254
implementing the Web application using 157
updating 176
XML template for 157, 176

K
key certificate
fields 125
key generation for certificates 232
key pairs for certificates
allowing PKI Services to generate 454
key pairs for certificates, generating
setting up ICSF 21
TokenName keyword in configuration file 66
key protection (field in end-user Web pages) 280
key ring
associating Web server and CA certificates with 27
creating 27
locating 344
key size (field in end-user Web pages) 279, 280
key usage (field in end-user Web pages) 279
key_backup (variable in IKYSETUP)
decision table 32
default value 35
description 35
key_gen (variable in IKYSETUP) 35
key_type (variable in IKYSETUP)
decision table 32
default value 36
description 36
keyboard 509
keyfile directive 79
keyid substitution variable 102
KeyProt (named field in pkiserv.tmpl) 107
KeyRing (parameter in pkiserv.conf) 66
keys for certificate requests
setting up PKI Services to generate 64, 232
KEYSMSTR class
activating 362
profile, defining 361
KeyUsage (certificate extension) 370
KeyUsage (named field in pkiserv.tmpl) 108

L
label (field in end-user Web pages) 279
Label (named field in pkiserv.tmpl) 108
LDAP 74
administrator’s distinguished name
description 73, 78
administrator’s password
description 73, 78
LDAP (continued)
allowing access to CRLs 74
attributes
mapped to DN fields 488
mapped to object identifiers 488
PKI Services requires 487
authorization to create certificates and CRLs 74
backend 20
bind passwords
encrypted 361
in the clear 361
configuring 19
description of PKI Services component 5
directory server requirements 487
distinguished name
administrator’s 73, 78
for binding 83
domain name
description 73, 78
Server1 parameter 83
encrypted passwords
BindProfile1 description 85
LDAPBIND class profile 81
RACF administration tasks for 361
storing information for 81
updating LDAP section of pkiserv.conf 82
establishing secure connection with 74
fully qualified domain name
description 73, 78
for LDAP server 83
Server1 parameter 83
installing 19
IP address
for LDAP server 83
IP address and port 83
objectclasses required by PKI Services 487
OU attribute 84, 86
password
administrator’s 73, 78
encrypted 361
for binding 84
in the clear 361
PKI Services component
description 5
PKI Services objectclasses and attributes
requirements 487
port
description 73, 78
for LDAP server 83
profile name 85
retrieving post requests 84
servers available (number of) 82
subcomponent for message logging 447
suffix, description 73
tailoring configuration for PKI Services 73
tailoring pkiserv.conf 81
TDBM DB2 backend 20
time interval for scanning for items to post 82
version supported 7
LDAP programmer
skills 12, 13
LDAP programmer (continued)
tasks
configuration, tailoring LDAP 73
configuring LDAP 19
CRLs, allowing access to 74
installing LDAP 19
LDAP configuration, tailoring 73
schema.user.ldif, updating 74
setting up LDAP ACL for accessing CRLs 74
setting up LDAP ACL for creating certificates 74
summary 12, 13
tailoring LDAP configuration 73
updating schema.user.ldif 74
team member 11
LDAP section (of pkiserv.conf)
default value 82
description 52
excerpt 81
information needed 82
tailoring 82
LDAP server
requirement for 10
LDAP server configuration file
adminPW 73, 78
LDAPBIND class
displaying information about 362, 363
profile
creating 362
specifying name when configuring 82
LDBM backend 10
ldif2tdbm load utility 73, 78
legal statement about certificate issuance and use 63
lib 438
libraries
AIEALNKE 437
APROCLIB 437
ASAMPLIB 437
PROCLIB 437
SAMPLIB 437
SIEALNKE 437
load libraries 36
load utility (ldif2tdbm) 73, 78
local PKI certificate authority 30
Local PKI RA (default RA certificate label) 66
local PKI registration authority 31
locality (field in end-user Web pages) 277
Locality (named field in pkiserv.tmpl) 108
log data set
from running IKYSETUP 43
log directives in httpd1443.conf, editing 80
log options
changing 401
displaying 402
log_dsn (variable in IKYSETUP) 38
logging message level 446
LOGREC
description 391
sample data 394
logs
changing options for 401
IKYSETUP data set sample 43
logs (continued)
using information from 397

M
mail (field in end-user Web pages) 277
Mail (named field in pkiserv.tmpl) 108
mainframe
education xx
Makefile.pkiext 235, 236, 240
Makefile.pkitpsamp 371
mapping
dn fields to LDAP attributes 488
host identity 8
MaxSuspendDuration (parameter in pkiserv.conf) 59
MD2 191
MD5 7, 191
members
connecting
to new group 342
to RACF group 341
deleting from group 342
deleting from RACF group 341
message form
certificate expiring 65
certificate ready 64
certificate rejected 64
message levels
_PKISERV_MSG_LEVEL 401
for logging 446
logging
diagnostic 447
error 447
informational 447
severe 447
verbose diagnostic 447
warning 447
message logging
CORE subcomponent 447
DB subcomponent 447
LDAP subcomponent 447
PKID subcomponent 447
POLICY subcomponent 447
SAFE subcomponent 447
TPOLICY subcomponent 447
message numbers
components identified 405
message types 405
methods, exit, for JSPs 254
Microsoft Internet Explorer
key protection field on end-user Web page 280
requesting a certificate 281
selecting a key size 282
supported standard 6
verifying certificate installed correctly 287, 288
Microsoft Windows Vista
installing CAPICOM 505
installing PKI Services CA certificate on 507
setting up to run end-user Web application 505
MODIFY command
change log options 401
MODIFY command (continued)
display logging options 402
stop PKI Services daemon 97
MVS programmer
installation of PKI Services 9
skills 14
tasks
creating VSAM data sets 87
enabling VSAM data sets for RLS 90
establishing RLS, preliminary steps 88
RLS, enabling VSAM data sets for 90
RLS, preliminary steps for establishing 88
starting PKI Services daemon 95
stopping PKI Services daemon 97
VSAM data sets, creating 87
team member 11
MyPolicy (parameter in pkiserv.conf) 52, 189

N
n-year PKI certificate for extensions demonstration
description 114
name
certificate templates
alias names 114
nicknames 114
short names 114
table summarizing 114
true names 114
field (in end-user Web pages) 280
named fields (in pkiserv.tmpl) 103
Netscape
key size field on end-user Web page 280
requesting a certificate 281
selecting a key size 282
supported standard 6
verifying certificate installed correctly 287, 288
nickname
certificate template 114
NOPASSWORD attribute 449
normal operating mode of z/OS HTTP Server 453
normalmode directive 78, 79
not after date (field in end-user Web pages) 278
not before date (field in end-user Web pages) 278
NotAfter (named field in pkiserv.tmpl) 108
NotBefore (named field in pkiserv.tmpl) 108
notice
legal 63
number 61
Notices 511
notification e-mail address (field in end-user Web pages) 280
notification forms
copying 48
customizing 225
notifications
customizing 230
retrieving your certificate 286
NotifyEmail
deleting 138
must match MAIL 291
NotifyEmail (named field in pkiserv.tmpl)
description 108
notifying users
forms for
pendingmsg.form 227
renewcertmsg.form 227
NumServers (parameter in pkiserv.conf) 82
O
Object ID
for policy 61
signing algorithm 62
object identifiers
mapping to LDAP attributes 488
object store
space considerations 88
objectclasses
LDAP, that PKI Services requires 487
ObjectDSN (parameter in pkiserv.conf) 52
ObjectRequestorDSN (parameter in pkiserv.conf) 53
ObjectStatusDSN (parameter in pkiserv.conf) 53
ObjectStore
DB subcomponent for message logging 447
description 320
enabled for sysplex 55
section of pkiserv.conf
default value 52
description 51
excerpt 51
information needed 52
sysplex enabled 55
time period before automatic deletion
completed requests 54
inactive requests 55
incomplete requests 55
unsuccessful requests 55
VSAM data set names
requestor alternate index 53
status alternate index 53
TID alternate index 53
ObjectTidDSN (parameter in pkiserv.conf) 53
OCEP
configuring for PKITP 371
data library (DL) 374
installing and configuring 23
optional installation 11
programmer
skills 12
team member 11
Trust Policy 367, 369, 374
OCSF
configuring 23
installing 23
programmer 11
installing and configuring OCSF 23
skills 12
requirement for 11
Trust Policy
module 372
overview 367
OCSF (continued)
Trust Policy (continued)
plug-in 367
OCSF Trust Policy
API
CSSM_TP_PassThrough 372
OCSPType (parameter in pkiserv.conf) 60
OIDs section (of pkiserv.conf)
default value 52
description 51
eexcerpt 51
information needed 52
OMVSKERN 34
one-year PKI generated key certificate
fields 125
one-year PKI generated key r certificate
description 113
one-year PKI S/MIME browser certificate
description 113
fields 122
one-year PKI SSL browser certificate
description 113
fields 122
one-year SAF browser certificate
description 112
fields 125
one-year SAF server certificate
description 112
fields 125
optfield substitution variable 102
Org (named field in pkiserv.tmpl) 108
organization (field in end-user Web pages) 277
organization name
for CertificatePolicies extension 61
organizational unit (field in end-user Web pages) 277
organizationalUnit objectclass 84
OrgUnit (named field in pkiserv.tmpl) 109
OrgUnit2 (named field in pkiserv.tmpl) 109
other name, alternate
field in end-user Web pages 278
OtherName (field in end-user Web pages)
customizing for REXX CGI execs 147
OU attribute 84, 86
overview of PKI Services 3

p
p12cert substitution variable 102
Parallel Sysplex support
prerequisites 9
requirements 9
parameters
changing 239
validating 239
pass directive 79
passphrase
recovering 239
passphrase (field in end-user Web pages) 280
PassPhrase (named field in pkiserv.tmpl) 109
passwords
binding 361
passwords (continued)
encrypted, for LDAP servers
LDAPBIND class profile 81
RACF administration for 361
for LDAP binding 84
for LDAP servers, encrypted
storing information for 81
updating LDAP section of pkiserv.conf 82
LDAP
encrypted 361
in the clear 361
LDAP administrator's 73, 78
RACF administration for 361
pathname
certificate expiring message form 65
certificate ready message form 64
certificate reject message form 64
configuration file 445
exit program 445
renewed certificate message form 65
request(s) pending for approval message form 65
requests meeting criteria for recovery message form 65
PDS 437
pending approval (status of certificate request) 302
pendingmsg.form
code sample 227
copying 48
in samples directory 439
purpose 45
pgmctnl_dsn. (variable in IKYSETUP) 36
PKCS #10
browser certificate format 6
certificate request 279
SCEP certificate request 219
server certificate format 6
PKCS #11 TKDS (token data set) 21
PKDS (public key data set) 21
PKI (public key infrastructure)
defined 4
PKI Authenticode–code signing server certificate
fields 123
PKI browser certificate for authenticating to z/OS
description 113
fields 122
PKI certificate for extensions demonstration
description 114
PKI exit
description 5
PKI extensions demonstration certificate
fields 122
PKI generated key certificate
description 113
PKI intermediate CA certificate
description 114
fields 123
PKI IPSEC server (firewall) certificate
description 113
fields 123
PKI S/MIME browser certificate
description 113
PKI S/MIME browser certificate (continued) 122
PKI Services
administration Web application 4
components
administration Web application 4
end-user Web application 5
exit 5
ICSF 5
IRRSPX00 5
LDAP 5
PKI Services daemon 5
R_PKIServ callable service 5
RACF 5
Websphere Application Server 5
z/OS HTTP Server 5
daemon
component 5
daemon user ID
PKISRVD 38
end-user Web application 5
exit 5
ICSF
component 5
IRRSPX00 5
LDAP
component 5
R_PKIServ callable service (IRRSPX00)
component 5
RACF
component 5
starting 95
stopping 95, 97
updating
certificate templates file 138
using
administration Web pages 299
Web pages
using 299
Websphere Application Server
component 5
z/OS HTTP Server
component 5
PKI Services administration
processing certificate requests
selected 310
selected certificate requests 310
PKI Services daemon
starting 95, 484
PKI Services daemon user ID
creating 449
PKI SSL browser certificate
description 113
fields 122
PKI SSL server certificate
description 113
fields 123
PKI Windows logon certificateields 122
pki_gid (variable in IKYSETUP) 30
PkiCertificate class 268
PKID
subcomponent for message logging 447
pkiexit.c
description 235
scenarios
allowing only selected users to request certificates 252
automatic renewal of certificates 239
maintaining customized certificate repository 253
recover lost passphrase 254
renewal only within 30 days of expiration 253
updating sample code 236
updating sample code for CGIs 240
pkigp (variable in IKYSETUP) 39
pkigroup_mem. (variable in IKYSETUP) 30
PKIGRP 39
pkijsp subdirectory 438
pkiprereg
examples 334
format 331
input data file 332
parameters 331
purpose 331
PKISERV
application name 110, 126
runtime user ID 145
surrogate user ID 39, 450
z/OS HTTP Server operating modes required 453
PKIServ subdirectory 438
pkserv.conf
CertPolicy section
default values 56
description 51
excerpt 51
information needed 56
changing
signature algorithm 192
code sample 441
copying 47
distribution point CRLs, customizing 195
editing
for configuring PKI Services 52
to change signature algorithm 192
to create CertificatePolicies extension 189, 190
to customize distribution point CRLs 195
to test configuration 95
General section
default values 63
description 51
excerpt 51
information needed 63
LDAP section
default value 82
description 52
excerpt 81
information needed 82
ObjectStore section
default value 52
description 51
excerpt 51
Index 535
ObjectStore section (continued)

OIDs section
default value 52
description 51
excerpt 51
information needed 52

parameters
AdminNotifyForm 65
AdminNotifyNew 56
AdminNotifyReminder 56
ARLDist 56, 197
AuthName1 83
AuthPwd1 84
BindProfile1 85
CPS1 56, 190
CreateTimeInterval 57
CreateOUValue 84
CRLDistDirPath 57, 197
CRLDistName 57, 196
CRLDistSize 57, 196
CRLDistURI 58, 197
CRLDuration 58
EnableSCEP 58
ExitTimeout 63
ExpireWarningTime 59
ExpiringMessageForm 65
ICLDSN 54
ICLRequestorDSN 54
ICLStatusDSN 54
InitialThreadCount 63
KeyRing 66
MaxSuspendDuration 59
MyPolicy 52, 189
NumServers 82
ObjectDSN 52
ObjectRequestorDSN 53
ObjectStatusDSN 53
ObjectTidDSN 53
OCSPType 60
Policy1Notice1 61, 189
Policy1Notice2 61
Policy1Org 61, 189
PolicyCritical 60, 189
PolicyName1 61, 189
PolicyRequired 60, 189, 190
PostInterval 82
RA_label 66
ReadyMessageForm 64
RecoverForm 65
RejectMessageForm 64
RemoveCompletedReqs 54
RemoveExpiredCerts 55
RemoveExpiredCertsAndKeys= 55
RemoveInactiveReqs 55
RenewCertForm 65
RetryMissingSuffix 84
Server1 83
SharedVSAM 55
SigAlg1 62, 192

parameters (continued)

TimeBetweenCRLs 62
TokenName 66
UserNoticeText1 63, 190

SAF section
default value 66
description 51
excerpt 51
information needed 66

sections
CertPolicy section 51
General section 51
LDAP section 52
ObjectStore section 51
OID section 51
SAF section 51
signature algorithm
changing 192
steps for updating LDAP section 85
updating 66
overview 50
steps 52

pkiserv.envars
code sample 448
purpose 46
updating 49

pkiserv.tmpl
APPLICATION section 126
ADMINHEADER subsection 112
ADMINSCOPE subsection 112
ADMINFOOTER subsection 112
CONTENT subsection 111
FAILURECONTENT subsection 111
FINDRECOVERCONTENT subsection 111
RECONTENT subsection 111
RECONTENT2 subsection 111
REFAILEDCERT subsection 111
RESUCCESSCONTENT subsection 111
RETRIEVECONTENT2 subsection 111
RETURNCERT subsection 111
subsections 110
certificate templates 112
changing
to add application sections 201
copying 47
customizing
customization, additional first-time 138
customizing the OtherName field 148
minimally 137
retrofitting release changes 142
debug flag 101
description 101
editing
administration Web pages 155
pkiserv.tmpl (continued)
editing (continued)
end-user Web pages 137
to add application sections 201
INSERT sections 103, 133
INSERTs
-AdditionalHeadIE 104
-ChallengePassphrase 104
-ChallengePassphrase2 104
-preregok 105
-RecoverEmail 104
-RecoverEmail2 104
-renewkeysetIE 105
-renewkeysetNS 105
-renewrevokebad 105
-renewrevokeok 105
-requestbad 104
-requestok 104
-requestor 104
-requestor2 105
-return12cert 105
AltDomain 105
AltEmail 105
AltIPAddr 106
AltOther 106
AltURI 106
ChallengePassPhrase 106
ClientName 106
CommonName 106
Country 107
DNQualifier 107
DomainName 107
Email (deprecated) 107
EmailAddr 107
ExtKeyUsage 107
HostIdMap 107
KeyProt 107
KeyUsage 108
Label 108
Locality 108
Mail 108
NotAfter 108
NotBefore 108
NotifyEmail 108
Org 108
OrgUnit 109
OrgUnit2 109
PassPhrase 109
PostalCode 109
PublicKey 109
PublicKeyIE 109
PublicKeyNS 109
RecoverEmail 109
Requestor 109
Requestor2 109
returnbrowsercertIE 105
returnbrowsercertNS 105
returnp12cert 105
Security 109
SerialNumber 110
SignWith 110

pkiserv.tmpl (continued)
INSERTs (continued)
StateProv 110
Street 110
Title 110
TransactionId 110
Uid 110
UnstructAddr 110
Userld 110
named fields 103
%%%renewrevokebad%%% 111
%%%renewrevokeok%%% 111
%%%requestok%%% 119
%%%returnp12cert%%% 111
%%%SelectCADomain%%% 112
purpose 47
sections 101
substitution variables 102
TEMPLATE section 130
ADMINAPPROVE subsection 118
APPL subsection 116
CONSTANT subsection 116
CONTENT subsection 116
FAILURECONTENT subsection 119
PREREGISTER subsection 120
RETRIEVECONTENT subsection 120
RETURNCERT subsection 120
subsections 115
SUCCESSCONTENT subsection 119
updating
customization, additional first-time 138
customizing the OtherName field 148
minimally 137
retrofitting release changes 142
PKIServ.xsd
in samples directory 439
PKISERVD
code sample 484
in PROCLIB 437
updating environment variables 49
PKISRVD
PKI Services daemon user ID 38, 66
pkitmpl.xml
description 157
in samples directory 439
purpose 47
pkitmpl.xml file
description 157
updating 176
PKITP
API called CSSM_TP_PassThrough 372
certificate extensions supported 370
certificate policies supported 369
configuring 371
files 371
overview 367
PKI Services Trust Policy plug-in for OCSF 367
pkitp_ipv 371, 372
pkitp.h 371
pkitp.so 371
pktpsamp.c
   description and directory 371
   editing 376
   sample code 377
PKIX 3, 4
pkixgen.tmpl
description 157
planning
   for PKI Services 9
policies
   notice number 61
   Object ID for 61
   usage 52
POLICY
   subcomponent for message logging 447
   Policy1Notice1 (parameter in pkiserv.conf) 61, 189
   Policy1Notice2 (parameter in pkiserv.conf) 61
   Policy1Org (parameter in pkiserv.conf) 61, 189
   PolicyCritical (parameter in pkiserv.conf) 60, 189, 369
   PolicyName1 (parameter in pkiserv.conf) 61, 189
   PolicyRequired (parameter in pkiserv.conf) 60, 189, 190, 369
ports
   1443, port 77
   443, port 77
   80, port 77
   for HTTP traffic 77
   for SSL traffic 77
   LDAP 73, 78
post requests
   retrying for LDAP 84
post-processing
   exit 240
   EXPORT 247
   GENCERT 243
   GENRENEW 243
   REQCERT 245
   REVOKE 249, 251
postal code
   DN field supported 488
   field in end-user Web pages 277
   PostalCode (named field in pkiserv.tmpl) 109
   postExport method 261
   postGenReqCert method 257
   postGenReqRenew method 259
   posting to LDAP
   frequency 82
   PostInterval (parameter in pkiserv.conf) 82
   postprocessing
      automatic renewal 238
   postQRRecover method 265
   preExport method 260
   preGenReqCert method 256
   preGenReqRenew method 258
   preprocessing
      automatic renewal 237
      exit 240
      EXPORT 246
      GENCERT 242
      GENRENEW 242
preprocessing (continued)
   QRECOVER 250
   REQCERT 244
   REVOKE 248
   preQRRecover method 264
   PREREREGISTER subsection (in TEMPLATE section of pkiserv.tmpl) 120
   preregistered (status of certificate request) 303
   preregistered requests
      deleting 308
      processing
      single 304
   preregistration
      certificate for SCEP 114
      enabling in pkiserv.conf 58
   preregistration certificate
      certificate
      fields 125
   prerequisites
      for sysplex support 9
   preRevoke method 262
   printablecert substitution variable 102
private key
   backing up 27
   creating 27, 451
   migrating to ICSF 27
   storing
   in ICSF 21
   private key of PKI Services
      rekeying 354
      replacing 354
      retiring 354
      rollover 354
problems, diagnosing 391
proCLIB 437
product libraries 437
products related to PKI Services 4
products, prerequisite
   installing and configuring 17
profile
   CA certificate, recovering 351
   FACILITY class
   IRR.PROXY.DEFINITIONS 361, 362
   IRR.SIGITCERT.ADD 359
   IRR.SIGITCERT.EXPORT 359
   IRR.SIGITCERT.GENCERT 359
   IRR.SIGITCERT.REQCERT 359
   IRR.SIGITCERT.REQRENEW 359
   IRR.SIGITCERT.REQRENEW 359
profile (continued)
IRR.DIGTCERT:RESPOND 359
IRR.DIGTCERT:REVOKE 360
IRR.DIGTCERT:SCEPREQ 360
IRR.DIGTCERT:VERIFY 360
IRR.RPKISERV:PKIADMIN 360
KEYSMSTR class
  LDAP:BINDPW.KEY 361
  LDAPBIND class profile
    defining 362
  protect directive 79
  PROTECTED attribute 450
  protection directive 79
protocols
  supported in PKI Services 6
province (field in end-user Web pages) 277
public key cryptography
  standards supported 6
public key data set (PKDS) 21
Public Key Infrastructure for X.509 3
publications
  on CD-ROM xviii
  on CD-ROM and DVD xviii
  softcopy xviii
PublicKey (named field in pkiserv.tmpl) 109
PublicKeyIE (named field in pkiserv.tmpl) 109
PublicKeyNS (named field in pkiserv.tmpl) 109

Q
QRECOVER
  accesses required 359
  parameters
    preprocessing 250
    R_PKIServ function 450
  return codes
    post-processing 251
    preprocessing 250
QRecover class 266
QrecoverResultsList class 269

R
R_PKIServ callable service
  administrative functions 360
  controlling the use 357
  description of PKI Services component 5
  end-user functions 357
  FACILITY class resources that protect 358, 360
  PKI Services component description 5
  protected by FACILITY class resources 450
R_PKIServ SAF callable service
  exits 240
RA certificate
  creating 451
  using IKYSETUP 27
  locating 344
RA certificate and private key
  backing up 452
ra_backup_dsn (variable in IKYSETUP) 39
ra_dn (variable in IKYSETUP) 30
RA_label (parameter in pkiserv.conf) 66
ra_label (variable in IKYSETUP) 31
RACF
  administering PKI Services 341
  authorizing
    READ access 342
    users for inquiry access 341
  connecting members
    to group 341
    to new group 342
  deleting groups 342
  deleting members from group 341, 342
  description of PKI Services component 5
  documents
    on CD-ROM xviii
    softcopy xviii
  PKI Services component
    description 5
    setting up PKI Services 27
RACF administration
  for PKI Services, ongoing 341
  for setting up PKI Services using IKYSETUP 27
  steps for 40
  using IKYSETUP 449
RACF administrator
  ongoing administration for PKI Services 341
  running IKYSETUP
    overview 27
    steps 42
  skills 14
  tasks 14
    IKYSETUP, running 27
    ongoing administration for PKI Services 341
    performed by IKYSETUP 449
    running IKYSETUP 27
    setting up PKI Services using IKYSETUP 40
  team member 11
RACF group
  providing access 27
RDB (request database) 320
RDN attribute 487
READ access
  authorizing 342
    IRR.DIGTCERT:GENCERT 453
    IRR.DIGTCERT:GENRENEW 451
    IRR.DIGTCERT:LIST 453
    IRR.DIGTCERT:LISTSTRING 453
    IRR.DIGTCERT:REQCERT 451
    IRR.DIGTCERT:REQRENEW 451
    IRR.RPKISERV:PKIADMIN 451
readonly substitution variable 103
ready message form for certificate 64
ReadyMessageForm (parameter in pkiserv.conf) 64
readymsg.form
  code sample 226
  copying 48
  customizing 230
  in samples directory 439
  purpose 47
recent activity (field in administration Web pages) 302
return codes  (continued)
REQCERT
  post-processing  245
  preprocessing  244
REQRENEW
  post-processing  245
  preprocessing  244
REVOKE
  post-processing  249
  preprocessing  248
returnbrowsercertIE  105
returnbrowsercertNS  105
RETURNCERT subsection (in APPLICATION sections of pkiserv.tmpl)  111
RETURNCERT subsection (in TEMPLATE section of pkiserv.tmpl)  120
returnpkcs10  105
revocation list, certificate authority (ARL)  198, 199
REVOKE
  accesses required  360
  parameters
    post-processing  249, 251
    preprocessing  248
  R_PKIServ function  450
return codes
  post-processing  249
  preprocessing  248
revoke (action for certificate)  314
RevokeCert class  266
revoked (status of certificate)  313
revoked, expired (status of certificate)  313
RLS
  enabling VSAM data sets for  90
  MVS programmer task  14
  preliminary steps for establishing  89
  setting up, preliminary steps  88
  RLS, VSAM record-level sharing  477
roadmaps
  configuring PKI Services for IdenTrust compliance  495
  for implementing PKI Services  15
roles of team members  11
rollover, PKI Services private key  354
RpkiservException class  269
RSA
  signature algorithm
    SigAlg1 parameter  62
    updating  191
  standard supported  7
runtime directory  95
runtime environment
  configuring  45
runtime user ID
  changing  145
  for requesting certificates  146
  for retrieving certificates  146
runtime-dir  9

S
S/MIME
  certificate format  6
  description of certificate  113
  fields of certificate  122
  supported standard  3
  use of certificate  7
SAF
  browser certificate
    description  112
    fields  125
  key ring
    creating  27, 451
    KeyRing parameter  66
  section (of pkiserv.conf)
    default value  66
    description  51
    excerpt  51
    parameter description  66
  server certificate
    description  112
    fields  125
subcomponent for message logging  447
SAMPLB  114
samples
  _PKISERV_CA_DOMAIN  445
  _PKISERV_MSG_LEVEL  446
  browser certificate template  499
  configuration directives  471, 472
  configuration file  441
  configuration file for IdenTrust compliance  499
  directives  471, 472
  environment variables file  448
  expiringmsg.form  227
  httpd.conf  471
  httpd2.conf  472
  IKYCVSAM  473
  IKYRVSAM  477
  IKYSETUP  455
  IKYVBKUP  481
  IKYVREST  483
  JCL
    certificate serial number incrementer,
      restoring  352
    IKYCVSAM  474
    IKYRVSAM  478
    IKYVBKUP  481
    IKYVREST  483
    PKISERVD  484
  log data set from IKYSETUP  43
  LOGREC data  394
  pendingmsg.form  227
  pkiserv.conf  441
  pkiserv.environ  448
  pkiserv.tmpl
    APPLICATION section  126
    INSERT section  133
    TEMPLATE section  130
  PKISERVD sample procedure  484
  pkitpsamp.c  377
  readymsg.form  226
samples (continued)
  recoverymsg.form 228
  rejectmsg.form 227
  renewcertmsg.form 227
  server certificate template for IdenTrust compliance 502
samples subdirectory 439
SAMPLIB 437
scenarios
  PKI exit
    allowing only selected users to request certificates 252
    maintaining customized certificate repository 253
    providing customized TITLE 252
    recover lost passphrase 254
    renewal only within 30 days of expiration 253
SCEP certificates
  requesting 296
SCEP preregistration certificate
  fields 125
SCEPREQ
  accesses required 360
sections of pkiserv.conf
  CertPolicy section 51
  General section 51
  LDAP section 52
  ObjectStore section 51
  OIDs section 51
  SAF section 51
secure
  e-mail 3
secure connection with LDAP, establishing 74
Secure Multipurpose Internet Mail Extensions (S./MIME) 3
Secure Sockets Layer (SSL) 3, 6, 74
security topics for RACF
  classroom courses xviii
Securityyn (named field in pkiservtmpl) 109
sendmail
  configuring 22
  not using default 49
  requirement for 11
serial number
  field in administration Web pages 302
  field in end-user Web pages 277
  incremener, restoring 352
serialno substitution variable 103
SerialNumber (named field in pkiservtmpl) 110
SERVAUTH class 343
server certificates
  aliases 114
  five-year PKI intermediate CA certificate 114
  five-year PKI IPSEC server (firewall) certificate 113
  five-year PKI SSL server certificate 113
  generating 27
  IdenTrust compliant template 502
  installing 286, 287
  one-year SAF server certificate 112
  retrieving 286, 287
  supported types 7
Server1 (parameter in pkiserv.conf) 83
setPassphrase method 266
setting up
  PKI Services 25
  var directory 70
setting up an ACL 74
settings
  contained in pkiserv.conf 46
  displaying log options 402
  IKYP025I displays 426
  log options, displaying 402
setTransactionid method 266
severe messages, logging 447
SHA1 7, 191
SHA256 191
SharedVSAM (parameter in pkiserv.conf) 55
sharing control data sets (SHCDS) 89
SHCDS 89
shortcut keys 509
SIEALNKE 437
SigAlg1 (parameter in pkiserv.conf) 62, 192
signature algorithm
  Object ID 62
  updating 191
signing key 21
signing_ca_label (variable in IKYSETUP) 39
SignWith (named field in pkiservtmpl) 110
Simple Certificate Enrollment Protocol (SCEP)
certificate
  description 114
  description of template 114
  template description 114
certificate fields 125
enabling 219
enabling in pkiserv.conf 58
fingerprint checking 222
overview 219
preregistration 220, 221
overview 220
RA certificate
  creating using IKYSETUP 27
  record 220
  request processing, overview 221
  request states 221
  steps for enabling 223
  variables 220, 221
single certificate, processing 314
single request, processing 304
skill requirements 11
skills
  ICSF programmer 12
  installing PKI Services 13
  installing prerequisite products 12
  LDAP programmer 12, 13
  MVS programmer 14
  OCEP programmer 12
  OCSF programmer 12
  RACF administrator 14
  UNIX programmer 12, 14
  Web server programmer 13, 15
smart cards 3
SMP/E 9
steps (continued)
creating (continued)
  ICL data sets  89
  VSAM object store  89
customizing
  administration Web pages  154
  distribution point CRLs  195
e-mail notifications  230
  end-user Web pages  137, 138, 142
  pkserv.tmpl  137, 138, 142
deleting
  multiple certificates  316
  selected certificates  318
  single certificate  314
  single request  304
e-mail notifications, customizing  230
enabling Simple Certificate Enrollment Protocol (SCEP)  223
encrypted passwords for LDAP servers  361
environment variables, updating  50
establishing your CA and RA certificates  451
gskkymam for certificate store  491
HostldMappings extensions, administering  343
ICL data sets, creating  89
IdenTrust compliance
  adjusting general settings  497
  creating templates  497
  modifying pkserv.conf  496
IKYSETUP, using  40
inquiry access, authorizing users for  342
installing
  ICSF  21
  LDAP  19
  z/OS HTTP Server  17
intermediate certificate authority, changing PKI Services to  347
invoking the certificate validation service  376
key ring, locating  344
LDAP
  schema.user.ldif, updating  74
  section of PKI Services configuration file, tailoring  82
  updating schema.user.ldif  74
LDAP bind passwords, encrypted
  IRR.PROXY.DEFAUTLS profile  362
  LDAPBIND class  362
LDAP servers, encrypted passwords for  361
locating
  CA certificate  344
  key ring  344
  PKI Services certificates  344
  RA certificate  344
modifying single request  304
passwords, encrypted LDAP binding
  IRR.PROXY.DEFAUTLS profile  362
  LDAPBIND class  362
performing RACF administration using
  IKYSETUP  40
PKI Services certificate authority certificate, renewing  349
PKI Services certificates, locating  344
steps (continued)
PKI Services daemon
  starting  95
  stopping  97
PKI Services RA certificate, renewing  350
pkserv.conf
  copying  47
  updating  52, 66
pkserv.tmpl
  copying  48
  customization, additional first-time  138
  customization, minimal  137
  customizing the OtherName field  148
  retrofitting release changes  142
PKITP, configuring  372
pkitpsamp.c  376
preregistering a SCEP client  296
processing
  multiple certificates  316
  multiple requests through searches  309
  selected certificates  318
  selected requests  310
  single certificate  314
  single request  304
RA certificate
  renewing  350
RA certificate, locating  344
RACF administration using IKYSETUP  40
recovering a CA certificate profile  351
rejecting single request  304
renewing
  certificate  289
  PKI Services certificate authority certificate  349
  PKI Services RA certificate  350
requesting a certificate  280
retrieving a certificate
  from bookmarked Web page  286
  from PKI Services home page  287
retrofitting changes into certificate templates  142
revoking
  certificate (by user)  292
  multiple certificates  316
  selected certificates  318
  single certificate  314
RLS
  enabling VSAM data sets for  90
  preliminary steps for establishing  89
running IKYSETUP  40
searching for requests  309
sendmail configuration, testing  22
setting up the var directory  70
Simple Certificate Enrollment Protocol (SCEP)
  enabling  223
starting
  PKI Services  95
  PKI Services daemon  95
z/OS HTTP Server  80
stopping PKI Services daemon  97
suspending
  certificate (by user)  292
steps (continued)
tailoring LDAP section of PKI Services configuration file 82
testing sendmail configuration 22
updating
  configuration file 52, 66
  environment variables 50
  exit code sample 236
  exit code sample for CGIs 240
  LDAP section of pkiserv.conf 85
  pkixexit.c 236
  pkixexit.c for CGIs 240
  pkiserv.conf 52, 66
  signature algorithm 192
single request 304
z/OS HTTP Server configuration files 77, 78
user ID for requesting certificates, changing 146
user ID for retrieving certificates, changing 146
using
gskkyman 491
IKYSETUP 40
using encrypted passwords for LDAP servers 361
  creating IRR.PROXY.DEAFULTS profile 362
  creating LDAPBIND class profile 362
var directory, setting up 70
viewing Web pages 95
VSAM object store, creating 89
z/OS HTTP Server configuration files, updating 77
STOP command 97
stopping
  PKI Services 95, 97
storage needs
  for ICL 87
  for object store 88
STORCLAS statements 89
store
  creating 89
  determining size requirements 88
storing
  certificate requests 320
  certificate revocation lists 5
  certificates 5
  private key
    in ICSF 5
street (field in end-user Web pages) 277
Street (named field in pkiserv.tmpl) 110
subcomponent level
  for logging 446
subdirectory
  bin 437
  include 437
  lib 438
  pkiijsp 438
  PKIServ 438
  samples 439
SubjectAltName (certificate extension) 370
SubjectKeyIdentifier (certificate extension) 370
subordinate certificate authority
  using PKI Services as 346
subsections in certificate templates, summary 121
substitution variables
  base64cert 102, 105
  browsertype 102
  cadomain 102
  errorinfo 102
  iecert 102
  keyid 102
  optfield 102
  p12cert 102
  pkiserv.tmpl 102
  printablecert 102
  readonly 103
  requestor 103
  serialno 103
  tmplname 103
  transactionid 103
SUCCESSCONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 119
suffix
  LDAP 73
summary of changes xxiii
surrog (variable in IKYSETUP) 39
surrog_uid (variable in IKYSETUP) 31
surrogate operation
  setting up 27, 454
surrogate user ID 145
  creating 27
  PKISERV 39, 450
suspend (action for certificate) 314
suspended (status of certificate) 313
SYS1.CSSLIB 36
SYS1.LINKLIB 36
SYS1.LOGREC 391
SYS1.SAMPLIB(IKYSETUP) 27
SYSS0UT
  records, contents 400
  viewing information 397
sysplex enabled
  in ObjectStore 55
sysplex support
  daemon, PKI Services, starting 95
  PKI Services daemon, starting 95
  pkiserv.conf
    updating, overview 50
    updating, steps for 66
  prerequisites 9
  requirements 9
  RLS
    enabling VSAM data sets for 90
    preliminary steps for establishing 88
SharedVSAM
  description 55
  updating 67
  starting PKI Services daemon 95
  updating pkiserv.conf 50
  updating SharedVSAM 67
system architecture diagram 5
task roadmaps
  configuring PKI Services for IdenTrust
    compliance 495
  for implementing PKI Services 15
  tasks
    (noun, gerund phrase)
      steps 163
  allowing WebSphere users to renew and revoke
  browser certificates
    steps 167
  application domain, creating
    steps 203
  automatic certificate renewal, setting up
    steps 231
  backing up the VSAM data sets
    steps 92
  CAPICOM, installing on Windows Vista system
    steps 506
  configuring Internet Explorer to trust PKI Services
    steps 506
  EAR file, updating
    steps 177
  installing the PKI Services CA certificate on a
  Windows Vista system
    steps 507
  JSP files, deploying to a WebSphere application
  server
    steps 178
  PKI Services Web application, implementing using the
  JSPs
    roadmap 162
  preparing to implement the PKI Services Web
  application using the JSPs
    steps 162
  preparing Windows Vista and Internet Explorer for use with end-user Web application
    roadmap 505
  recovering a certificate whose keys were generated by PKI Services
    steps 293
  restoring the VSAM data sets
    steps 93
  retrieving a PKI generated key certificate
    steps 288
  setting up key generation for certificate requests
    steps 232
  TCP/IP.SEZALOAD 36
  TDBM 20
    specifying password as entry 73, 78
  TDBM backend 10
  team members 11
  TEMPLATE section of pkiserv.tmpl (continued)
    RETURNCERT subsection 120
    subsections 115
    SUCCESSCONTENT subsection 119
  templates
    adding 144
    customizing
      additional first-time changes 138
      minimal 137
      OtherName field 148
      retrofitting release changes 142
    description 157
  for JSPs 157
  updating 176
  XML 157, 176
  TemplateTool utility 336
  threads
    created at initialization 63
  TID
    alternate index
      VSAM data set name for 53
  time interval
    before certificate expiration 59
    between certificate revocation lists 58, 62
    certificate suspension grace period 59
    for scanning for items to post 82
    scanning database for approved requests 57
    warning message about certificate expiration 59
  time period
    in ICL before automatic deletion 55
    in ObjectStore before automatic deletion 54, 55
  time zone
    default 96
  TimeBetweenCRLs (parameter in pkiserv.conf) 62
  timeout value
    for PKI Services exit 63
  Title 110
  title (field in end-user Web pages) 277
  TKDS
    time period before automatic deletion expired certificates 55
  TKDS (token data set) 21
    specifying a token in for PKI Services 66
  TLS (Transport Layer Security) 74
  tmplname substitution variable 103
  token data set
    time period before automatic deletion expired certificates 55
  token data set (TKDS) 21
    specifying a token in for PKI Services 66
  TokenName (parameter in pkiserv.conf) 66
  TPOLICY
    subcomponent for message logging 447
  transaction ID
    field in administration Web pages 302
    field in end-user Web pages 280
  TransactionId (named field in pkiserv.tmpl) 110
  transactionid substitution variable 103
  Transport Layer Security (TLS) 74
  true name of certificate templates 114
  trust level for PKI Services 343
Trust Policy
API called CSSM_TP_PassThrough 372
overview 367
two-year PKI Authenticode–code signing server certificate
description 113
fields 122
two-year PKI browser certificate for authenticating to
z/OS
description 113
fields 122
two-year PKI Windows logon certificate
description 113
fields 122
type 80 SMF record
table of event codes and qualifiers 485
table of relocate section variable data 485
TZ environment variable 49

U
Uid (named field in pkiserv.tmpl) 110
unencrypted, LDAP bind passwords 361
Uniform Resource Identifier (URI)
Certificate Practice Statement 56
uniform resource identifier (URI), alternate
field in end-user Web pages 278
UNIX programmer
skill planning 12, 14
task planning 12, 14
tasks
configuring sendmail 22
configuring UNIX runtime environment 45
LDAP section of pkiserv.conf, updating 81
runtime environment, configuring 45
sendmail, configuring 22
UNIX runtime environment, configuring 45
updating LDAP section of pkiserv.conf 81
team member 11
UNIX runtime environment
configuring 45
unix_sec (variable in IKYSETUP)
decision table 33
default value 37
description 37
UnstructAddr (named field in pkiserv.tmpl) 110
unstructured address
field in end-user Web pages 277
unstructured name
field in end-user Web pages 277
UPDATE access
IRR.DIGTCERT.ADD 451
IRR.RPKISERV.PKIADMIN 451
updating
certificate templates file
minimal 137
retrofitting changes 142
pkiserv.tmpl
minimal 137
URI
containing CPS 56
URI, alternate 278
usage policy 52
user ID
associating with PKI Services started procedure 27, 449
changing
for retrieving certificates 146
requesting certificates 146
field in end-user Web pages 277
PKI Services daemon 66
runtime
changing 145
user ID, surrogate
creating 27
user notifications
copying files 48
customizing forms 230
UserExit class 255
UserExitException class 266
Userld (named field in pkiserv.tmpl) 110
userld directive 78, 79
UserNoticeText1 (parameter in pkiserv.conf) 63, 190
userPassword attribute 73, 78
uses of PKI Services 3
utilities
executables 437
iclview 322
PKI Services 321
pkiprereg 331
TemplateTool 336
vosview 326

V
var directory
setting up 70
variables
in IKYSETUP REXX exec
change based on setup 32
change optionally 37
change required 29
customizable section 28
in notification forms
%cadomain%% 228
%dn% 228
%notafter% 228
%pendrequest% 228
%printcert% 228
%quicklink% 229
%recoverylink% 229
%recoverylist% 229
%requestor% 229
%transactionid% 229
variables-dir 9
verbose diagnostic messages, logging 447
VERIFY
accesses required 360
RPKI^PKIServ function 450
viewing
VSAM ICL data set records 323
VSAM ObjectStore data set records 327

Index 547
virtual private network (VPN) devices 3
Vista
installing CAPICOM 505
installing PKI Services CA certificate on 507
setting up to run end-user Web application 505
VOL statements 89
vosview
example 327
output 327
parameters 326
purpose 326
VPN devices 3, 8
VSAM
data set name for ICL data 54
data set name for ICL requestor alternate index 54
data set name for ICL status alternate index 54
data set name for ObjectStore base cluster 52
data set name for ObjectStore requestor alternate index 53
data set name for ObjectStore status alternate index 53
data set name for ObjectStore TID alternate index 53
RLS
enabling data sets for 90
preliminary steps for establishing 89
VSAM data sets
backing up 92
sample JCL 481
creating 87
not using RLS 473
using RLS 89, 90, 477
giving administrators access to 27
restoring from backup 93
sample JCL 483
RLS, enabling for 90
VSAM object store, creating 89
cmsmlq (variable in IKYSETUP) 40

Web server programmer (continued)
tasks (continued)
configuring z/OS HTTP Server 17
installing and configuring 17
starting 80
updating configuration 77
team member 11
uninstalling the z/OS HTTP Server configuration files 77
Web server programmer
installing and configuring z/OS HTTP Server 17
web_dn (variable in IKYSETUP) 31
web_expires (variable in IKYSETUP) 40
web_label (variable in IKYSETUP) 40
web_ring (variable in IKYSETUP) 31
webservlet (variable in IKYSETUP) 40
Websphere application server
deploying JSP file to 178
Websphere Application Server 5
installing and configuring 19
requirement for 10
Websphere users
authorizing to renew and revoke browser certificates 166
authorizing to use PKI Services functions 163
WEBSRV 40
Windows logon certificate
description 113
Windows Vista
installing CAPICOM 505
installing PKI Services CA certificate on 507
setting up to run end-user Web application 505

X
X.509v3 certificates 7, 8
XML template file
description 157, 176

Z
z/OS
PKI browser certificate for authenticating to
description 113
fields 122
PKI Windows logon certificate
description 113
fields 122
z/OS Basic Skills information center xx
z/OS HTTP Server
configuration files
updating 77
description of PKI Services component 5
installing and configuring 17
operating modes PKISSERT requires 453
PKI Services component
description 5
requirement for 10
setting up for surrogate operation 27, 454
starting 80
z/OS HTTP Server (continued)
  updating configuration  77
z/OS product libraries
  AIEALNKE  437
  APROCLIB  437
  ASAMPLIB  437
  PROCLIB  437
  SAMPLIB  437
  SIEALNKE  437
z/OS UNIX level security  34
zip code (field in end-user Web pages)  277
Readers’ Comments — We’d Like to Hear from You

z/OS
Cryptographic Services PKI Services
Guide and Reference

Publication No. SA22-7693-11

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

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 state on this form.

Comments:

Thank you for your support.
Submit your comments using one of these channels:
• Send your comments to the address on the reverse side of this form.
• Send your comments via e-mail to: mhvrdfs@us.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address