

z/OS



Security Server PKI Services Guide and Reference

z/OS



Security Server PKI Services Guide and Reference

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 371.

Third Edition, September 2002

This edition applies to Version 1 Release 4 of z/OS (5694-A01), Version 1 Release 4 of z/OS.e (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order documents through your IBM® representative or the IBM branch office serving your locality. Documents are not stocked at the address below.

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
Department 55JA, Mail Station P384
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: mhvrcfs@us.ibm.com

World Wide Web: <http://www.ibm.com/servers/eserver/zseries/zos/webqs.html>

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.

© Copyright International Business Machines Corporation 2001, 2002. All rights reserved.

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

Contents

Tables	ix
Figures	xi
About this document	xiii
Who should use this document	xiii
How to use this document	xiv
How to read syntax conventions	xvi
Where to find more information	xvii
Softcopy publications	xvii
Using LookAt to look up message explanations.	xvii
Accessing z/OS™ licensed documents on the Internet	xviii
Other sources of information	xviii
IBM discussion areas	xviii
Internet sources	xix
To request copies of IBM publications	xx
Summary of changes	xxi

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	7
Chapter 2. Planning your implementation	9
Installing PKI Services	9
Requirements for sysplex support	9
Determining prerequisite products	10
z/OS HTTP Server	10
LDAP directory server	10
OCSF and OCEP	10
ICSF (optional)	10
sendmail (optional)	11
Identifying skill requirements	11
Team members	11
Skills for setting up prerequisite products	11
Skills for setting up PKI Services	12
Creating an implementation plan	14
Task roadmap for implementing PKI Services.	14
Chapter 3. Migration considerations	17
Migration overview	17
Release summary	17
Parallel sysplex support	17
What this change affects	17
Dependencies	18

	Coexistence considerations	18
	Migration Tasks.	18
	References to other documents.	19
	E-mail notification for completed certificate requests and expiration warnings	19
	What this change affects	19
	Dependencies	20
	Coexistence considerations	20
	Migration Tasks.	20
	References to other documents.	20
	Support for MAIL, STREET, and POSTALCODE qualifiers for distinguished	
	names	20
	What this change affects	21
	Dependencies	21
	Coexistence considerations	21
	Migration Tasks.	21
	References to other documents.	21
	Using encrypted passwords for your LDAP servers	21
	What this change affects	22
	Dependencies	22
	Coexistence considerations	22
	Migration Tasks.	22
	References to other documents.	22
	Storing serial number and event files in the VSAM object store	23
	What this change affects	23
	Dependencies	23
	Coexistence considerations	23
	Migration Tasks.	23
	References to other documents.	23
	Summary of interface changes	23
	Code Samples	24
	Messages.	25
	SYS1.SAMPLIB members	25
	Utilities	25
	Chapter 4. Installing and configuring prerequisite products	27
	Tasks to perform before setting up PKI Services	27
	Installing and configuring the z/OS HTTP Server	27
	Steps for installing and configuring the z/OS HTTP Server to work with PKI	
	Services	27
	Installing and configuring OCSF and OCEP	29
	Steps for installing and configuring OCSF and OCEP to work with PKI	
	Services	29
	Installing and configuring LDAP	30
	Steps for installing and configuring LDAP	30
	Installing and configuring ICSF (optional)	32
	Configuring sendmail (optional)	33

Part 2. Configuring your system for PKI Services 35

	Chapter 5. Running IKYSETUP to perform RACF administration	37
	Overview of IKYSETUP.	37
	Before you begin	38
	Variables whose values must change.	39
	Variables whose values may change depending on setup	41
	Variables you can optionally change	46
	Steps for performing RACF tasks using IKYSETUP	47

Chapter 6. Configuring the UNIX runtime environment	53
Steps for copying files	54
Optionally updating PKI Services environment variables	55
(Optional) Steps for updating PKI Services environment variables	56
Optionally updating the pkiserv.conf configuration file	57
(Optional) Steps for updating the configuration file	58
Steps for setting up the /var/pkiserv directory	67
Chapter 7. Tailoring LDAP configuration for PKI Services	69
Steps for updating schema.user.ldif	69
Chapter 8. Updating z/OS HTTP Server configuration and starting the server	71
Steps for updating the z/OS HTTP Server's configuration files	71
Steps for starting the z/OS HTTP Server	74
Chapter 9. Tailoring the PKI Services configuration file for LDAP	75
Excerpt of LDAP section	75
Storing information for encrypted passwords for your LDAP servers	75
Steps for tailoring the LDAP section of the configuration file	76
Chapter 10. Creating VSAM data sets	81
Space considerations for creating VSAM data sets	81
Determining storage needs for ICL	81
Determining storage needs for the object store	82
(Optional) preliminary steps for establishing VSAM RLS	82
Steps for creating the VSAM object store and ICL data sets and indexes	83
(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS	84
Chapter 11. Starting and stopping PKI Services	85
Steps for starting the PKI Services daemon	85
Stopping the PKI Services daemon	86

Part 3. Customizing PKI Services 89

Chapter 12. Customizing the end-user Web application	91
Contents of the pkiserv.tmpl certificates templates file	91
What are substitution variables?	91
What are named fields?	92
INSERT sections	93
The APPLICATION section	97
Templates that PKI Services provides	99
TEMPLATE sections	100
Summary of fields in certificate templates	106
Examining the pkiserv.tmpl file	107
Relationship between CGIs and the pkiserv.tmpl file	114
Steps for performing minimal customization	116
Steps for additional first-time customization	117
Steps for retrofitting release changes into the PKI Services certificate templates	121
Locating code for customizing end-user Web pages	125
Steps for adding a new certificate template	127
Changing the runtime user ID	127
Steps for changing the runtime user ID for requesting certificates	128
Steps for changing the runtime user ID for retrieving certificates	129

	Customizing e-mail notifications sent to users	129
	Steps for customizing e-mail notification forms	132
	Chapter 13. Customizing the administration Web pages	133
	CGIs for administration Web pages	133
	Customizing the administration Web pages	134
	Steps for customizing the administration Web pages	135
	Changing the runtime behavior for accessing administration pages	135
	Steps for changing control of access to administration pages	136
	(Optional) Steps for removing the administration page link from the PKI Services home page	136
	Chapter 14. Advanced customization	139
	Using certificate policies	139
	Steps for creating the CertificatePolicies extension	139
	Updating the signature algorithm	141
	Steps for changing the signature algorithm	141
	Using the PKI exit	141
	Steps for updating the exit code sample	142
	Using the exit for pre- and post-processing	142
	Scenarios for using the PKI exit	152
<hr/>		
	Part 4. Using PKI Services	155
	Chapter 15. Using the end-user Web pages	157
	Steps for accessing the end-user Web pages	157
	Summary of fields	160
	Steps for requesting a new certificate	162
	Retrieving your certificate	167
	Steps for retrieving your certificate from the bookmarked Web page	167
	Steps for retrieving your certificate from the PKI Services home page	169
	Steps for renewing a certificate	169
	Steps for revoking a certificate.	172
	Chapter 16. Using the administration Web pages.	173
	Steps for accessing the administration home page	173
	Fields in the administration Web pages	177
	Processing certificate requests	177
	Status of certificate requests	177
	Actions on certificate requests	178
	Using the PKI Services administration home page	178
	Processing certificates.	188
	Status of certificates	188
	Actions for certificates	189
	Steps for processing a single certificate	189
	Steps for processing certificates by performing searches	190
	Relationship between certificate requests and matching certificates	195
<hr/>		
	Part 5. Administering RACF for PKI Services	197
	Chapter 17. RACF administration for PKI Services	199
	Creating a CA signing key pair using hardware	199
	Steps for creating a CA certificate using the PCICC	199
	Authorizing users for the PKI Services administration group	200
	Connecting members to the group	200

Deleting members from groups	200
Authorizing users for inquiry access	200
Steps for authorizing users for inquiry access	200
Administering HostIdMappings extensions	201
Steps for administering HostIdMappings extensions	201
Locating your PKI Services certificate and key ring	203
Steps for locating the PKI Services certificate and key ring	203
Establishing PKI Services as an intermediate certificate authority	204
Steps for establishing PKI Services as an intermediate CA	205
Renewing your PKI Services certificate authority certificate	206
Steps for renewing your PKI Services certificate authority certificate	206
Recovering a CA certificate profile	207
Steps for recovering a CA certificate profile	207
Controlling applications that call R_PKIServ	211
R_PKIServ end-user functions	211
R_PKIServ administrative functions	213
Using encrypted passwords for LDAP servers	214
Steps for using encrypted passwords	214

Part 6. Troubleshooting 217

Chapter 18. Using information from SYS1.LOGREC	219
Sample LOGREC data	222
Chapter 19. Using information from the PKI Services logs	225
Viewing SYSOUT information	225
_PKISERV_MSG_LEVEL subcomponents and message levels	229
Changing logging options	229
Displaying log options settings	230
Chapter 20. Using PKI Services utilities	231
vosview	232
iclview	235

Part 7. Reference information 237

Chapter 21. Messages	239
Chapter 22. File directory structure	257
Product libraries	257
HFS directory and subdirectories	257
Chapter 23. The pkiserv.conf configuration file	261
Chapter 24. The pkiserv.tmpl certificate templates file	263
Chapter 25. Environment variables	305
Environment variables in the environment variables file	305
The pkiserv.envars environment variables file	307
Chapter 26. The IKYSETUP REXX exec.	309
Actions IKYSETUP performs by issuing RACF commands	309
Setting up the PKI Services daemon user ID	309
Setting up access control to protect PKI Services	309
Creating the CA certificate, private key, and key ring	311

Configuring the z/OS HTTP Server for SSL mode	312
Using RACF to obtain a certificate for the Web server	313
Enabling the z/OS HTTP Server for surrogate operation	313
Enabling the PKI Services daemon to call OCSF functions	313
Code Sample: IKYSETUP	314
Chapter 27. Other code samples	327
z/OS HTTP Server configuration directives	327
IKYCVSAM.	330
IKYRVSAM.	332
PKISERVD sample procedure to start PKI Services daemon	335
Chapter 28. The certificate validation service	337
Overview	337
Certificate policies	339
Certificate extensions	339
CRL extensions and CRL entry extensions	340
Files for PKITP	340
Configuring and getting started with PKITP	341
Steps for configuring PKITP.	341
Trust Policy API	341
CSSM_TP_PassThrough.	343
Providing the certificate validation service	347

Part 8. Appendixes 363

Appendix A. LDAP directory server requirements	365
Appendix B. Using a gskkyman key database for your certificate store	367
Steps for using a gskkyman key database for your certificate store	367
Appendix C. Accessibility	369
Using assistive technologies	369
Keyboard navigation of the user interface.	369
Notices	371
Programming interface information	372
Trademarks.	372
Bibliography	375
Index	377

Tables

1.	Syntax conventions	xvi
2.	Basic components of PKI Services and related products	4
3.	Types of certificates you can request	7
4.	HFS directory variables	9
5.	Tasks and skills needed for installing prerequisite products	12
6.	Roles, tasks, and skills for setting up PKI Services	13
7.	Task roadmap for implementing PKI Services	14
8.	z/OS HTTP Server information you need to record	29
9.	OCSF information you need to record	30
10.	LDAP information you need to record	31
11.	IKYSETUP — Structure and divisions	38
12.	IKYSETUP variables whose values must change	39
13.	Deciding the value of restrict_surrog.	42
14.	Deciding the value of use_icsf	42
15.	Deciding the value of key_backup	42
16.	Deciding the value of unix_sec.	43
17.	IKYSETUP variables you might want to change depending on setup	44
18.	IKYSETUP variables you can optionally change	46
19.	Deciding which files to copy and change	53
20.	Information needed for updating the configuration file	58
21.	LDAP information you need for tailoring LDAP configuration	69
22.	Summary of configuration and usage of each Web server instance	71
23.	LDAP information you need for tailoring z/OS HTTP Server configuration	72
24.	Information needed for updating the LDAP section of the configuration file.	76
25.	VSAM RLS information you need to record	83
26.	pkiserv.tmpl — Structure and main divisions	91
27.	Substitution variables	92
28.	Sample INSERTs.	94
29.	Named fields in INSERT sections.	94
30.	Subsections of the APPLICATION section.	99
31.	Certificate templates PKI Services provides	99
32.	Names of certificate templates	101
33.	Summary of subsections in certificate templates	105
34.	Summary of fields in certificate templates that PKI Services provides	106
35.	CGI actions for end-user Web pages	114
36.	Cross-reference of changes made for new features used in various templates	121
37.	Changed lines in templates	122
38.	Location of code for various Web pages	125
39.	Descriptions of variables for forms	131
40.	Summary of substitution variables in forms	132
41.	CGI actions for administrative Web pages	133
42.	Summary of information about important files for the exit routine	141
43.	Values of arguments for pre- and post-processing	143
44.	Types of certificates you can request	157
45.	Summary of fields in end-user Web pages	160
46.	Summary of fields in the administration pages	177
47.	Statuses of certificate requests	177
48.	Summary of actions to perform on requests and required status	178
49.	Searches to display certificate requests	184
50.	Status of certificates	188
51.	Summary of actions to perform and required status to do so	189
52.	Searches to display certificates	191
53.	Information you need for locating your PKI Services certificate and key ring.	203

54.	Information you need for establishing PKI Services as an intermediate CA	205
55.	Information you need for renewing your PKI Services certificate authority certificate	206
56.	Information you need for recovering a CA certificate profile	207
57.	Summary of accesses required for PKI Services request.	212
58.	LOGREC data for PKI Services	219
59.	Nicknames of certificate templates for appldata	233
60.	Summary of information about important files	237
61.	Meaning of fourth character in message number.	239
62.	Meaning of eighth character in message number	239
63.	Files contained in subdirectories.	257
64.	Subcomponents for message level	306
65.	Message levels	306
66.	Access required if you plan to use an administrator.	310
67.	Access required if you plan to use auto-approval	310
68.	FACILITY class access needed for protecting administrative functions	311
69.	Access PKISERVD needs to use RACF's certificate services	312
70.	Summary of information about important files for PKITP	340
71.	PKI Services OCSF Trust Policy (PKITP) error codes	345
72.	Table of LDAP objectclasses and attributes that PKI Services sets	365
73.	Relationship of named fields to LDAP attributes and object identifiers	365

Figures

1.	Component diagram of a typical PKI Services system.	6
2.	Flowchart of the process of updating IKYSETUP	48
3.	Sample log data set.	51
4.	readymsg.form	130
5.	rejectmsg.form	131
6.	expiringmsg.form	131
7.	PKISERV certificate generation application Web page.	158
8.	The Certificate popup window for installing the CA certificate	159
9.	One-year SSL browser certificate request form	164
10.	Successful request displays transaction ID	165
11.	Web page to retrieve your certificate	166
12.	Browser certificate installation Web page	167
13.	Server certificate installation Web page	168
14.	Popup window listing certificates	170
15.	Renew or revoke a certificate Web page.	171
16.	PKI Services home page	174
17.	The Certificate popup window for installing the CA certificate	175
18.	Entering your user ID and password	176
19.	PKI Services administration home page	179
20.	Single request approval Web page.	180
21.	Processing successful Web page	181
22.	Modifying the request Web page	182
23.	Processing requests after searching	185
24.	Request processing was successful Web page	187
25.	Request processing was not successful Web page	187
26.	Request processing was partially successful Web page	188
27.	Processing a certificate from the single certificate Web page	190
28.	Processing certificates using searches	192
29.	Processing of certificate was successful Web page.	194
30.	Request processing was not successful Web page.	194
31.	Request processing was partially successful Web page	195
32.	Sample JCL data set for restoring the certificate serial number incrementer value	210
33.	Sample LOGREC data	223
34.	Separating the job files	226
35.	Selecting a file to view	227
36.	Messages contained in the file	228
37.	Settings that IKYP025I displays	255
38.	Examples of organizations, certificates, and chains	338

About this document

This document supports z/OS (5694-A01) and z/OS.e (5655-G52). This document contains information about PKI Services, which is part of the z/OS Security Server. The Security Server includes the following components:

- DCE Security Server
- Resource Access Control Facility (RACF)
- Lightweight Directory Access Protocol (LDAP) Server, which includes client and server functions
- Network Authentication Service
- Open Cryptographic Enhanced Plug-ins (OCEP)
- PKI Services
- z/OS Firewall Technologies

This document provides the information for planning, customizing, administering, and using the PKI Services component of the Security Server. For information about other components of the Security Server, see the publications related to those components.

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:

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

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

Preface

- 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 chapters:
 - 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, “Migration considerations” on page 17 provides information about migrating from an earlier release.
 - Chapter 4, “Installing and configuring prerequisite products” on page 27 describes installing and configuring related products: the z/OS HTTP Server, OCSF and OCEP, LDAP, and optionally ICSF.
- Part 2, “Configuring your system for PKI Services” on page 35 describes the tasks your team members need to perform to configure PKI Services.
 - Chapter 5, “Running IKYSETUP to perform RACF administration” on page 37 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 6, “Configuring the UNIX runtime environment” on page 53 explains UNIX programmer tasks including how to copy files, update environment variables, update the PKI Services configuration file, and set up the /var/pkiserv HFS directory.
 - Chapter 7, “Tailoring LDAP configuration for PKI Services” on page 69 explains how the LDAP programmer updates LDAP configuration for PKI Services.
 - Chapter 8, “Updating z/OS HTTP Server configuration and starting the server” on page 71 explains how the Web server programmer updates the z/OS HTTP Server configuration files and starts the z/OS HTTP Server.
 - Chapter 9, “Tailoring the PKI Services configuration file for LDAP” on page 75 explains how the UNIX programmer updates the LDAP section of the PKI Services configuration file.
 - Chapter 10, “Creating VSAM data sets” on page 81 explains how the MVS programmer creates VSAM data sets.
 - Chapter 11, “Starting and stopping PKI Services” on page 85 explains how the MVS programmer starts and stops the PKI Services daemon.
- Part 3, “Customizing PKI Services” on page 89 explains how to customize end-user and administration Web pages and advanced customization using an exit.
 - Chapter 12, “Customizing the end-user Web application” on page 91 provides an overview of the pkiserv.tmpl file, which contains the certificate templates, and explains how to customize the end-user Web pages.

- Chapter 13, “Customizing the administration Web pages” on page 133 provides an overview of the CGI scripts and explains how to customize the administration Web pages.
- Chapter 14, “Advanced customization” on page 139 explains how to use certificate policies, the signature algorithm, and the PKI exit.
- Part 4, “Using PKI Services” on page 155 explains using the end-user and administration Web pages.
 - Chapter 15, “Using the end-user Web pages” on page 157 shows the end-user Web pages and explains how to request a certificate, obtain the certificate, and renew or revoke a certificate.
 - Chapter 16, “Using the administration Web pages” on page 173 shows the administration Web pages and explains how to process certificate requests and certificates.
- Part 5, “Administering RACF for PKI Services” on page 197 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, “Troubleshooting” on page 217 explains using logs and utilities:
 - Chapter 18, “Using information from SYS1.LOGREC” on page 219 describes ‘SYS1.LOGREC’ — which is used to record unusual runtime events, such as an exception.
 - Chapter 19, “Using information from the PKI Services logs” on page 225 discusses using the PKI Services logs to debug problems and explains how to change logging options and display log options settings.
 - Chapter 20, “Using PKI Services utilities” on page 231 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 7, “Reference information” on page 237 provides reference information including messages and important code samples.
 - Chapter 21, “Messages” on page 239 explains PKI Services messages.
 - Chapter 22, “File directory structure” on page 257 describes product and HFS directories for PKI Services and files contained in them.
 - Chapter 23, “The `pkiserv.conf` configuration file” on page 261 provides a code sample of the `pkiserv.conf` configuration file.
 - Chapter 24, “The `pkiserv.tmpl` certificate templates file” on page 263 provides a code sample of the `pkiserv.tmpl` file.
 - Chapter 25, “Environment variables” on page 305 explains the `pkiserv.envars` environment variables file and provides a code sample.
 - Chapter 26, “The IKYSETUP REXX exec” on page 309 explains the contents of the IKYSETUP REXX exec that performs RACF administration and provides a code sample.
 - Chapter 27, “Other code samples” on page 327 provides additional code samples.
 - Chapter 28, “The certificate validation service” on page 337 describes the certificate validation service.
- There are several appendixes, including the following:
 - Appendix A, “LDAP directory server requirements” on page 365 explains using a non-z/OS LDAP server.
 - Appendix B, “Using a `gskkyman` key database for your certificate store” on page 367 explains an alternative method for setting up your key database.

Preface

How to read syntax conventions

This section describes how to read syntax conventions. It defines syntax notations and provides syntax examples that contain these items.

Table 1. Syntax conventions

Notation	Meaning	Example	
		Book Syntax	Sample Entry
Apostrophes	Apostrophes indicate a parameter string and must be entered as shown.	SEND 'message',NOW	SEND 'listings ready',NOW
Comma	Commas must be entered as shown.	DISPLAY C,K	DISPLAY C,K
Ellipsis ...	Ellipsis indicates that the preceding item or group of items can be repeated one or more times. Do not enter the ellipsis.	VARY (devspec[,devspec]...),ONLINE	VARY (282,283,287),ONLINE
Parentheses and special characters	Parentheses and special characters must be entered as shown.	DUMP COMM=(text)	DUMP COMM=(PAYROLL)
Underline	Underline indicates a default option. If you select an underlined alternative, you do not have to specify it when you enter the command.	<u>K T</u> [<u>,REF</u>] [<u>,UTME=nnn</u>]	<u>K T</u>
Lowercase parameter	Lowercase indicates a variable term. Substitute your own value for the item.	MOUNT devnum	MOUNT A30 or mount a30
Uppercase parameter	Uppercase indicates the item must be entered using the characters shown. Enter the item in either upper or lowercase.	DISPLAY SMF	DISPLAY SMF or display smf
Single brackets	Single brackets represent single or group-related items that are optional. Enter one or none of these items.	DISPLAY DMN[=domainum]	DISPLAY DMN=5
Stacked brackets	Stacked brackets represent group-related items that are optional. Enter one or none of these items.	[TERMINAL] [NOTERMINAL]	NOTERMINAL
Single braces	Single braces represent group-related items that are alternatives. You must enter one of the items. You cannot enter more than one.	{COMCHECK COMK}	COMK

Table 1. Syntax conventions (continued)

Notation	Meaning	Example	
		Book Syntax	Sample Entry
Stacked braces	Stacked braces represent group related items that are alternatives. You must enter one of the items. You cannot enter more than one.	MN {DSNAME} {SPACE } {STATUS}	MN SPACE
Or-bar ()	An or-bar indicates a mutually exclusive choice. When used with brackets, enter one or none of the items. When used with braces, you must enter one of the items.	ACTIVATE RECOVER=SOURCE	RECOVER=SOURCE
Stacked items with or-bars () and brackets	Stacked items with or-bars indicates a mutually-exclusive choice. Enter one or none of these items.	CD RESET [,SDUMP] ,SYSABEND ,SYSUDUMP ,SYSDUMP ,ALL	CD RESET,SYSUDUMP

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 Security Server library is available on the following CD-ROMs. The CD-ROM online library collections include the IBM Library Reader, which is a program that enables you to view the softcopy documents.

SK3T-4269 *z/OS Version 1 Release 3 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 the Adobe Acrobat reader.

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

This softcopy collection kit contains the Security Server library for z/OS in both BookManager and Portable Document Format (PDF) files. You can view or print the PDF files with the Adobe Acrobat reader.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

<http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>

Preface

or from anywhere in z/OS where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site that will allow you to access LookAt from a handheld computer (Palm Pilot VIIx suggested).

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the **News** section on the LookAt Web site.

Some messages have information in more than one document. For those messages, LookAt displays a list of documents in which the message appears.

Accessing z/OS™ licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

<http://www.ibm.com/servers/resourceLink>

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code. ¹

To obtain your IBM Resource Link user ID and password, log on to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

Other sources of information

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

IBM discussion areas

IBM provides the following discussion areas for PKI Services, RACF and security-related topics.

- **MVSRACF**

MVSRACF is available to customers through IBM's TalkLink offering. To access MVSRACF from TalkLink:

1. z/OS.e™ customers received a Memo to Licensees, (GI10-0684) that includes this key code.

1. Select S390 (the S/390 Developers' Association).
 2. Use the fastpath keyword: MVSRACF.
- **SECURITY**
SECURITY is available to customers through IBM's DialIBM offering, which may be known by other names in various countries. To access SECURITY:
 1. Use the CONFER fastpath option.
 2. Select the SECURITY CFORUM.

Contact your IBM representative for information on TalkLink, DialIBM, or equivalent offerings for your country and for more information on the availability of the MVSRACF and SECURITY discussions.

Internet sources

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

- **Online library**

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

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

- **Redbooks**

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

<http://www.ibm.com/redbooks/>

- **Enterprise systems security**

For more information about security on the S/390 platform, OS/390, and z/OS, including the elements that comprise the Security Server, use this address:

<http://www.ibm.com/servers/eserver/zseries/zos/security/>

- **PKI Services home page**

You can visit the PKI Services home page on the World Wide Web using this address:

<http://www.ibm.com/servers/eserver/zseries/zos/pki/>

- **RACF home page**

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

<http://www.ibm.com/servers/eserver/zseries/zos/racf/>

- **RACF-L discussion list**

Customers and IBM participants may 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**

Preface

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

To access this code from a Web browser, go to the RACF home page and select the “Downloads” topic from the navigation bar, or go to <ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/>.

The code is also available from [ftp.software.ibm.com](ftp://ftp.software.ibm.com) through anonymous FTP. To get access:

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

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

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

Note: Some Web browsers and some FTP clients (especially those using a graphical interface) might have problems using [ftp.software.ibm.com](ftp://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.

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 6:30 a.m. through 5:00 p.m. Mountain Time. You can use this number to:

- Order or inquire about IBM publications
- Resolve any software manufacturing or delivery concerns
- Activate the program reorder form to provide faster and more convenient ordering of software updates

Summary of changes

Summary of changes
for SA22-7693-01
z/OS Version 1 Release 4

This document contains information previously presented in SA22-7693-00, which supports z/OS Version 1 Release 3.

New information

- Support for enabling a sysplex
- Support for sending e-mail notifications to users when a certificate is ready for retrieval or expiring or when a certificate request has been rejected
- Support for using MAIL, STREET, and POSTALCODE distinguished name qualifiers
- Support for using an encrypted LDAP password
- Support for using the 4758 coprocessor (called PCICC) to generate private keys
- Support for storing serial number and event files that PKI Services uses (previously maintained as separate files) in the VSAM object store
- The vosview utility has added new **-c** and **-r** parameters.
- The iclview utility has added new **-c** and **-r** parameters.
- New messages are included:
 - IKYC030I
 - IKYC031I
 - IKYC032I
 - IKYC033I
 - IKYC034I
 - IKYC035I
 - IKYL003I
 - IKYL004I
- Message IKYP029I changed.
- New code samples include:
 - IKYRVSAM
 - expiringmsg.form
 - readymsg.form
 - rejectmsg.form
- Information is added to indicate that this document supports z/OS.e (5655-G52).

This document includes 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.

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, “Migration considerations” on page 17 summarizes the enhancements for the latest release and describes considerations for migrating from an earlier release.
- Chapter 4, “Installing and configuring prerequisite products” on page 27 describes installing and configuring related products: the z/OS HTTP Server, OCSF and OCEP, LDAP, and optionally ICSF.

|
|
|

Chapter 1. Introducing PKI Services

This chapter provides an overview of PKI Services.

It covers the following topics:

- “What is PKI Services?”
- “What is a certificate authority?”
- “What is PKI?” on page 4
- “Basic components of PKI Services and related products” on page 4
- “Component diagram” on page 5
- “Supported standards” on page 6
- “Supported certificate types” on page 7
- “Supported certificate fields and extensions” on page 7

What is PKI Services?

PKI Services allows you 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.

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.

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.

Introducing PKI Services

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 2. Basic components of PKI Services and related products

Administration Web application	Assists authorized administrators to review requests for certificates, approve or reject requests, renew certificates, or revoke certificates through their own Web browsers. The application consists of sample screens that you can easily customize to display your organization's logo. It also supports the following tasks: <ul style="list-style-type: none">• Reviewing pending certificate requests• Querying pending requests to process those that meet certain criteria• Displaying detailed information about a certificate or request• Monitoring certificate information, such as validity period• Annotating the reason for an administrative action
End-user Web application	Guides your users to request, obtain, and renew certificates through their Web browsers. The application consists of sample screens that you can easily customize to meet your organization's needs for certificate content and standards for appearance. It offers several certificate templates that you can use to create requests for a variety of certificate types, based on the certificate's intended purpose and validity period, and supports certificate requests that are automatically approved.

Table 2. Basic components of PKI Services and related products (continued)

Exit	Provides advanced customization for additional authorization checking, validating, and changing parameters on calls to the R_PKIServ callable service (IRRSPX00), and capturing certificates for further processing. You can call this exit from the PKIServ CGIs and use its IRRSPX00 pre-processing and post-processing functions. A code sample in C language code is included.
ICSF (optional)	Securely stores the PKI Services certificate authority's private signing key.
LDAP	The directory that maintains information about the valid and revoked certificates that PKI Services issues in an LDAP-compliant format. You can use an LDAP server such as z/OS Security Server LDAP.
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: <ul style="list-style-type: none"> • 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.
z/OS HTTP Server	PKI Services uses the Web server to encrypt messages, authenticate requests, and transfer certificates to intended recipients.

Component diagram

Figure 1 on page 6 shows a typical PKI Services system.

Introducing PKI Services

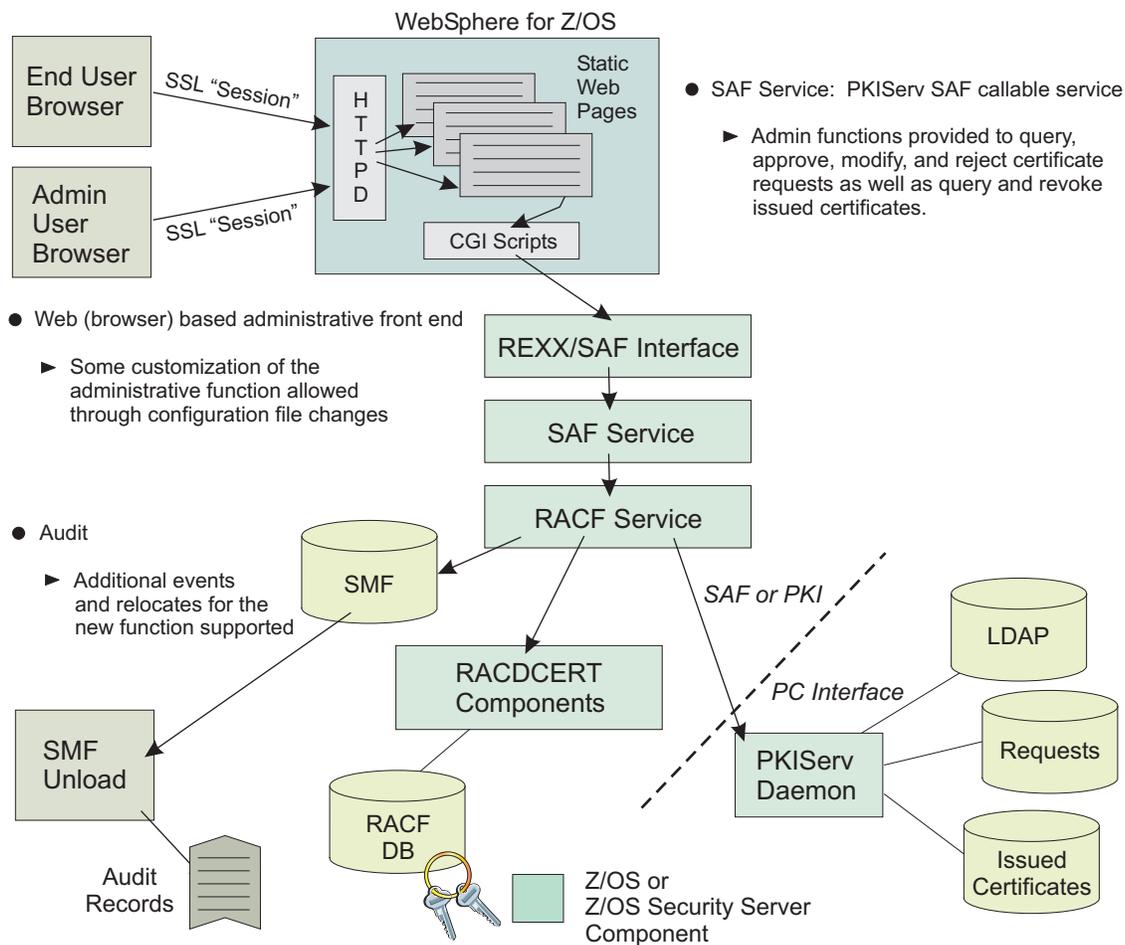


Figure 1. Component diagram of a typical PKI Services system

Supported standards

PKI Services supports the following standards for public key cryptography:

- Secure Sockets Layer (SSL) version 2 and version 3, with client authentication
- PKCS#10 browser and server certificate format, with a base64-encoded response
- IPSEC certificate format
- S/MIME certificate format
- Browser certificates for:
 - Microsoft Internet Explorer version 5.x
 - Netscape Navigator and Netscape Communicator version 4.x
- Server certificates
- LDAP standard for communications with the Directory
- X.509v3 certificates
- Certificate revocation lists (CRLv2)
- Key lengths up to 1024 bits for the CA signing private keys
- RSA algorithms for encryption and signing
- MD5 and SHA-1 hash algorithms

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 3 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 3. Types of certificates you can request

Type of certificate	Use
One-year PKI SSL browser certificate	End-user client authentication using SSL
One-year PKI S/MIME browser certificate	Browser-based e-mail encryption
Two-year PKI browser certificate for authenticating to z/OS	End-user client authorization using SSL when logging onto z/OS
Five-year PKI SSL server certificate	SSL Web server certification
Five-year PKI IPSEC server (firewall) certificate	Firewall server identification and key exchange
Five-year PKI intermediate CA certificate	Subordinate (non-self-signed) Certificate Authority certification
One-year SAF browser certificate	End-user client authentication where RACF (not PKI Services) is the certificate provider
One-year SAF server certificate	Web server SSL certification where RACF (not PKI Services) is the certificate provider

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 certificates can include the following types of extensions:

Standard extensions

The standard X.509v3 certificate extensions:

- authority key identifier
- basic constraints
- certificate policies
- 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.

Introducing PKI Services

To support your organization's policies, PKI Services also provides the means for you to customize and define 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 chapter 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 chapter covers the following topics:

- “Installing PKI Services”
- “Determining prerequisite products” on page 10
- “Identifying skill requirements” on page 11
- “Creating an implementation plan” on page 14

Installing PKI Services

Your MVS programmer uses SMP/E to install PKI Services into an HFS 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 HFS directories were used so you can customize the install process.

Table 4 shows each HFS-related 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 4. HFS directory variables

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

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 1.4 or higher.
- 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 82 for information about creating VSAM data sets suitable for VSAM RLS. For information on establishing a parallel sysplex environment with a coupling facility, see *z/OS MVS Programming: Sysplex Services Guide*. For more information about

Planning your implementation

establishing data sharing for VSAM RLS, see *z/OS DFSMS Introduction* and *z/OS DFSMSdfp Storage Administration Reference*.

Determining prerequisite products

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

- z/OS HTTP Server
- LDAP directory server
- OCSF and OCEP
- ICSF (optional)
- sendmail (optional)

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

z/OS HTTP Server

In a PKI Services system, 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 may 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 27.

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 is recommended, 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), such as IBM z/OS LDAP. If you intend to use the z/OS LDAP server, you must configure it to use the TDBM backend.

Through the integration of IBM z/OS LDAP 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 30.

OCSF and OCEP

PKI Services requires OCSF and OCEP to be installed and configured so that the user ID under which the PKI Services daemon runs can use required services. For more information, see “Installing and configuring OCSF and OCEP” on page 29.

ICSF (optional)

ICSF is recommended but not required. You can begin using PKI Services without installing ICSF and install it later without reinstalling PKI Services. ICSF is strongly recommended to store and protect your certificate authority’s private key. For more information, see “Installing and configuring ICSF (optional)” on page 32.

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 33.

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 may 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 may 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
- LDAP programmer
- MVS programmer
- OCSF and OCEP programmer
- RACF administrator
- UNIX programmer
- Web server programmer

You may wish to include a Web page designer to customize your PKI Services Web applications. This task is listed in the chapter 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 may be advisable to appoint a PKI administrator early, and involve this person in your planning.

Attention: 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. IBM recommends that you give this authority to only those individuals whom you trust with the RACF SPECIAL attribute. For more information on the RACF SPECIAL attribute, see the *z/OS Security Server RACF Security Administrator’s Guide*.

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:

Planning your implementation

Table 5. Tasks and skills needed for installing prerequisite products

Role	Tasks	Required Skills	Documented in:
ICSF programmer	(Optionally) installing and configuring ICSF (if not already done)	ICSF installation and configuration skills	<ul style="list-style-type: none"> • <i>z/OS ICSF Administrator's Guide</i> • <i>z/OS ICSF Application Programmer's Guide</i> • <i>z/OS ICSF System Programmer's Guide</i>
LDAP programmer	Installing and configuring LDAP (if not already done) and recording information	LDAP installation and configuration skills	<ul style="list-style-type: none"> • <i>z/OS Security Server LDAP Server Administration and Use</i>
OCSF and OCEP programmer	Installing and configuring OCSF and OCEP (if not already done) and recording information	OCSF and OCEP installation and configuration skills	<ul style="list-style-type: none"> • <i>z/OS Open Cryptographic Services Facility Application Programming</i> • <i>z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming</i>
UNIX programmer	Configuring sendmail (if your installation is planning to send e-mail notifications to users about certificates)	<ul style="list-style-type: none"> • Basic UNIX commands such as the cp (copy) command and mkdir (make directory) command • sendmail configuration skills 	<ul style="list-style-type: none"> • <i>z/OS Communications Server: IP Configuration Guide</i>
Web server programmer	Installing and configuring the z/OS HTTP Server (if not already configured for at least non-SSL pages) and recording information	z/OS HTTP Server installation and configuration skills	<ul style="list-style-type: none"> • <i>z/OS HTTP Server Planning, Installing, and Using</i>

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.
2. The OCSF and OCEP programmer installs and configures the OCSF and OCEP.
3. The LDAP programmer installs and configures LDAP.
4. Optionally, the ICSF programmer install and configures ICSF.

See Chapter 4, "Installing and configuring prerequisite products" on page 27 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 6. Roles, tasks, and skills for setting up PKI Services

Role	Tasks	Required Skills	Documented in:
LDAP programmer	<ul style="list-style-type: none"> • Customizes LDAP configuration for PKI Services 	<ul style="list-style-type: none"> • LDAP customization skills 	<ul style="list-style-type: none"> • <i>z/OS Security Server LDAP Server Administration and Use</i>
MVS programmer	<ul style="list-style-type: none"> • Creates VSAM object store and ICL data sets and indexes • Optionally setting up VSAM RLS • Starts the PKI Services daemon 	<ul style="list-style-type: none"> • Basic MVS skills <ul style="list-style-type: none"> – Editing a data set – ISPF COPY command – MVS console START command • JCL knowledge to change job card • Basic Web and browser skills 	<ul style="list-style-type: none"> • <i>z/OS MVS System Commands</i>
RACF administrator	<ul style="list-style-type: none"> • Adds groups and user IDs • Sets up access control • Creates certificates • Sets up daemon security 	<ul style="list-style-type: none"> • RACF administration • REXX skills (for working with IKYSETUP REXX exec) • RACF commands such as the following: <ul style="list-style-type: none"> – ADDGROUP – ADDSD – ADDUSER – RACDCERT – RDEFINE – PERMIT – SETROPTS • TSO skills 	<ul style="list-style-type: none"> • <i>z/OS TSO/E REXX Reference</i> • <i>z/OS UNIX System Services Planning</i> • <i>z/OS Security Server RACF Security Administrator's Guide</i>
UNIX programmer	<ul style="list-style-type: none"> • Copies files • (Optionally) customizes environment variables • (Optionally) customizes (non-LDAP sections of) pkiserv.conf configuration file • Sets up /var/pkiserv directory • Updates the LDAP section of the pkiserv.conf configuration file 	<ul style="list-style-type: none"> • Basic UNIX commands, such as the cp (copy) command • Getting superuser authority 	<ul style="list-style-type: none"> • <i>z/OS UNIX System Services Command Reference</i> • <i>z/OS UNIX System Services Planning</i>
Web server programmer	<ul style="list-style-type: none"> • Helps set up PKI Services <ul style="list-style-type: none"> – Updates the z/OS HTTP Server's configuration files – Starts the z/OS HTTP Server • Customizes the PKI Services Web pages 	<ul style="list-style-type: none"> • z/OS HTTP Server customization skills • Editing configuration files • Customizing the PKI Services Web pages 	<ul style="list-style-type: none"> • <i>z/OS HTTP Server Planning, Installing, and Using</i>

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

Table 7. Task roadmap for implementing PKI Services

Subtask	Associated procedure (See ...)
Installing and configuring prerequisite products: <ul style="list-style-type: none">• z/OS HTTP Server• OCSF and OCEP• LDAP directory server• ICSF (optional)• sendmail (optional)	Chapter 4, “Installing and configuring prerequisite products” on page 27 <ul style="list-style-type: none">• “Steps for installing and configuring the z/OS HTTP Server to work with PKI Services” on page 27• “Steps for installing and configuring OCSF and OCEP to work with PKI Services” on page 29• “Steps for installing and configuring LDAP” on page 30• “Installing and configuring ICSF (optional)” on page 32• “Configuring sendmail (optional)” on page 33
Configuring your system for PKI Services: <ul style="list-style-type: none">• RACF• z/OS UNIX• LDAP configuration• z/OS HTTP Server• LDAP• VSAM	Part 2, “Configuring your system for PKI Services” on page 35 <ul style="list-style-type: none">• Chapter 5, “Running IKYSETUP to perform RACF administration” on page 37• Chapter 6, “Configuring the UNIX runtime environment” on page 53• Chapter 7, “Tailoring LDAP configuration for PKI Services” on page 69• Chapter 8, “Updating z/OS HTTP Server configuration and starting the server” on page 71• Chapter 9, “Tailoring the PKI Services configuration file for LDAP” on page 75• Chapter 10, “Creating VSAM data sets” on page 81• Chapter 11, “Starting and stopping PKI Services” on page 85
Customizing PKI Services: <ul style="list-style-type: none">• Customizing end-user Web pages• Customizing administration Web pages	Part 3, “Customizing PKI Services” on page 89 <ul style="list-style-type: none">• Chapter 12, “Customizing the end-user Web application” on page 91• Chapter 13, “Customizing the administration Web pages” on page 133

Planning your implementation

Table 7. Task roadmap for implementing PKI Services (continued)

Subtask	Associated procedure (See ...)
<ul style="list-style-type: none">• Advanced customizing	<ul style="list-style-type: none">• Chapter 14, “Advanced customization” on page 139
Testing PKI Services:	Part 4, “Using PKI Services” on page 155
<ul style="list-style-type: none">• Using end-user Web pages• Using administration Web pages	<ul style="list-style-type: none">• Chapter 15, “Using the end-user Web pages” on page 157• Chapter 16, “Using the administration Web pages” on page 173
Administering PKI Services:	Part 5, “Administering RACF for PKI Services” on page 197
<ul style="list-style-type: none">• RACF	<ul style="list-style-type: none">• Chapter 17, “RACF administration for PKI Services” on page 199

Chapter 3. Migration considerations

Migration overview

The following sections describe the new and changed information for PKI Services introduced in z/OS V1R4. For each change, this section provides the following information:

- Description of change
- Summary of the tasks or interfaces affected by each change
- Coexistence considerations, if any
- Migration procedures, if any
- References to other documents

Release summary

For information about:	Refer to:
Changes introduced in z/OS V1R4	
PKI sysplex support	page 17
E-mail notification about certificates	page 19
Support for MAIL, STREET, and POSTALCODE distinguished name qualifiers.	page 20
Support for encrypted passwords for your LDAP servers	page 21
Storing serial number and event files that PKI Services uses (previously maintained as separate files) in the VSAM object store	page 23

Parallel sysplex support

This support lets you take advantage of a parallel SYSPLEX environment to start multiple independent instances of the PKI Services daemon on different images in the sysplex and to configure PKI Services to run in parallel, acting on one common data store.

Using a sysplex requires setting up VSAM record-level sharing (RLS). The vosview and iclview utilities are updated with new parameters to support VSAM RLS. A new SharedVSAM keyword is added to the pkiserv.conf configuration file in the ObjectStore section. Using a sysplex also entails using the sample JCL IKYRVSAM rather than IKYCVSAM.

What this change affects

Area	Considerations
LDAP programming	None.

Area	Considerations
MVS programming	<p>If you want to use a sysplex to run multiple instances of the PKI Services daemon, you need to:</p> <ol style="list-style-type: none"> 1. Perform VSAM data set creation <ul style="list-style-type: none"> • Perform preliminary steps for establishing VSAM Record-Level Sharing (RLS) • Update the VOL statements when creating the VSAM object store and ICL data sets and indexes • Enable existing PKI Services VSAM data sets for VSAM RLS 2. Stop and restart the PKI Services daemons.
RACF administration	None.
Web server programming	None.
UNIX programming	If you want to use a sysplex for PKI Services, you need to specify SharedVSAM=T when updating the pkiserv.conf configuration file.
PKI Services administration	None.

Dependencies

To use a sysplex for PKI Services, you must copy data sets to the appropriate storage class.

Coexistence considerations

There are no unique considerations in the sysplex environment.

Migration Tasks

Review the following high-level tasks to better understand the impacts to your environment. A task is required if you must perform this task to use a sysplex and you are migrating from z/OS V1R3. For detailed procedures, see the reference listed.

Task	Condition	Reference
Update the pkiserv.conf configuration file.	Required.	page 65
Perform preliminary steps to establish VSAM RLS.	Required.	page 82
Enable existing PKI Services VSAM data sets for VSAM RLS.	Required.	page 84
Stop and restart the PKI Services daemons.	Required.	page 86 and page 85

References to other documents

- *z/OS MVS Programming: Sysplex Services Guide*
- *z/OS MVS Programming: Sysplex Services Reference*

E-mail notification for completed certificate requests and expiration warnings

This support is added to enhance user convenience by providing a mechanism to notify your clients through e-mail when their certificates are ready for retrieval or are about to expire. This support includes the following changes:

- **NotifyEmail**: a new optional user input field for the non-SAF certificate request web pages and the R_PKIServ callable service (IRRSPX00).
- Changes in the `pkiserv.conf` configuration file for this support include the following:
 - `ExpireWarningTime` in the `CertPolicy` section
 - In the `General` section:

ReadyMessageForm	notification sent when certificate is ready for retrieval
RejectMessageForm	notification sent when certificate request is rejected
ExpiringMessageForm	notification sent when certificate is expiring.

What this change affects

Area	Considerations
LDAP programming	None.
RACF administration	None.
Web server programming	None.
UNIX programming	If you are sending e-mail notifications: <ol style="list-style-type: none"> 1. Configure <code>sendmail</code>. 2. Copy additional files: <ul style="list-style-type: none"> • <code>rejectmsg.form</code> • <code>readymsg.form</code> • <code>expiringmsg.form</code> 3. Optionally update environment variables (if you are not using the default PKI Services environment variables file <code>(/usr/lpp/pkiserv/samples/pkiserv.envars)</code>) 4. Update the <code>pkiserv.conf</code> configuration file.
MVS programming	If you are sending e-mail notifications you need to stop and restart the PKI Services daemons because of changes to <code>pkiserv.conf</code>
PKI Services administration	If you are sending e-mail notifications: <ol style="list-style-type: none"> 1. Retrofit z/OS Version 1 Release 4 release changes to the PKI Services certificate templates file <code>pkiserv.tmpl</code> for the templates you use. 2. Customize e-mail notifications sent to users.

Dependencies

You must enable sendmail. For more information, see *z/OS Communications Server: IP Configuration Guide*.

Coexistence considerations

There are no coexistence considerations associated with this change.

Migration Tasks

Review the following high-level tasks to better understand the impacts to your environment. A task is required if you must perform this task to send e-mail notifications and you are migrating from z/OS V1R3. For detailed procedures, see the references listed.

Task	Condition	Reference
Configure sendmail.	Required.	page 33
Copy additional files, such as messages notifications forms.	Required.	page 54
Update environment variables	Optional. You need to update environment variables only if you are not using the default PKI Services environment variables file /usr/lpp/pkiserv/samples/pkiserv.envars.	page 55
Update the pkiserv.conf configuration file.	Required.	page 58
Stop and restart the PKI Services daemons.	Required.	page 86 and page 85
Retrofit z/OS Version 1 Release 4 release changes to the PKI Services certificate templates file pkiserv.tmpl for the templates you use. (For sending e-mail notifications, include NotifyEmail on non-SAF certificate request templates.)	Required.	page 121
Customize e-mail notifications sent to users.	Required.	page 129

References to other documents

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

Support for MAIL, STREET, and POSTALCODE qualifiers for distinguished names

In z/OS V1R4 PKI Services has added MAIL, STREET, and POSTALCODE qualifiers for distinguished names. This helps to differentiate distinguished names.

Corresponding with the addition of MAIL, STREET, and POSTALCODE qualifiers, PKI Services has added Email, Street, and PostalCode named fields in various TEMPLATES sections of the pkiserv.tmpl certificate templates file.

What this change affects

Area	Considerations
LDAP programming	None.
MVS programming	None.
RACF administration	None.
Web server programming	None.
UNIX programming	None.
PKI Services administration	Retrofit release changes to the PKI Services certificates template file, pkiserv.tmpl, for changes to support Email, Street, and PostalCode in the templates you use.

Dependencies

There are no dependencies associated with this change.

Coexistence considerations

There are no coexistence considerations associated with this change.

Migration Tasks

Review the following high-level tasks to better understand the impacts to your environment. A task is required if you must perform this task to use MAIL, STREET, and POSTALCODE qualifiers for distinguished names and you are migrating from z/OS V1R3. For detailed procedures, see the reference listed.

Task	Condition	Reference
Retrofit z/OS Version 1 Release 4 release changes to the PKI Services certificate templates file pkiserv.tmpl for the templates you use.	Required.	page 121

References to other documents

None.

Using encrypted passwords for your LDAP servers

PKI Services stores certificates, CRLs, and so forth in an LDAP directory. The LDAP interface requires the caller to authenticate (bind) to the directory either anonymously or using a distinguished name and passwords. Before z/OS V1R4, those passwords (for multiple directories) are stored in clear text in the pkiserv.conf configuration file. Starting in V1R4, the customer can optionally encrypt and store the passwords in the PROXY segment of general resource profiles.

You can either store binding information in the IRR.PROXY.DEFAULTS profile in the FACILITY class or in one or more profiles defined in the LDAPBIND class. The BindProfile1 parameter has been added to the pkiserv.conf file to specify the name of the LDAP bind profile containing the bind information.

What this change affects

Area	Considerations
LDAP programming	None.
MVS programming	Stop and restart the PKI Services daemons because of changes to pkiserv.conf
RACF administration	You need to perform an additional series of steps including defining and activating a RACF KEYSMSTR class profile and creating an LDAPBIND profile.
Web server programming	None.
UNIX programming	You need to update the LDAP section of the pkiserv.conf configuration file to add a bind profile directive.
PKI Services administration	None.

Dependencies

The RACF administrator must set up a RACF KEYSMSTR class and bind profile and tell the UNIX programmer the profile name. The UNIX programmer uses this name when adding the BindProfile1 parameter to the pkiserv.conf configuration file. The UNIX programmer also removes the Server1, AuthName1, and AuthPwd1 parameters from the pkiserv.conf configuration file.

Coexistence considerations

There are no coexistence considerations associated with this change.

Migration Tasks

Review the following high-level tasks to better understand the impacts to your environment. A task is required if you must perform this task to use encrypted passwords for your LDAP servers and you are migrating from z/OS V1R3. For detailed procedures, see the reference listed.

Task	Condition	Reference
Update the LDAP section of the pkiserv.conf configuration file.	Required.	page 76
Perform RACF administration for using encrypted passwords.	Required.	page 214
Stop and restart the PKI Services daemons.	Required.	page 86 and page 85

References to other documents

- *z/OS Security Server LDAP Server Administration and Use*

Storing serial number and event files in the VSAM object store

To simplify setup, PKI Services has been enhanced to store serial number and event files in the VSAM object store. Before z/OS V1R4, PKI Services stored these files as separate files, and the ObjectStore section of the pkiserv.conf configuration file specified the path and file name stem by including the Name and Path keywords.

If you are installing PKI Services for the first time in z/OS V1R4, you do not need to do anything. If you used PKI Services in z/OS V1R3 and are migrating to a later release, leave the Name and Path keywords in the pkiserv.conf configuration file until after you complete migration; after migration, you remove these keywords.

What this change affects

Area	Considerations
LDAP programming	None.
MVS programming	The first time you start the PKI Services daemon, the first system you start must be the one that has access to the old work files (by default, /etc/pkiserv/pkiserv.j*). (These are on the same system you were using for PKI Services before adding sysplex support.)
RACF administration	None.
Web server programming	None.
UNIX programming	You can remove the Name and Path keywords from the pkiserv.conf configuration file after migration. (See Step 1a on page 64.)
PKI Services administration	None.

Dependencies

There are no dependencies associated with this change.

Coexistence considerations

There are no coexistence considerations associated with this change.

Migration Tasks

There are no high-level tasks to perform for this support.

References to other documents

None.

Summary of interface changes

This section summarizes new and changed interface components of PKI Services:

For information about:	Refer to:
Code samples	page 24
Messages	page 25

For information about:	Refer to:
SYS1.SAMPLIB	page 25
Utilities	page 25

Code Samples

The following table lists the changes made to PKI Services code samples for z/OS Version 1 Release 4:

File name	Release	Description	Related support
pkiserv.conf	V1R4	<p>Updated: The following parameters were added:</p> <ul style="list-style-type: none"> • SharedVSAM (in ObjectStore section) • ExpireWarningTime (in CertPolicy section) • ReadyMessageForm (in General section) • RejectMessageForm (in General section) • ExpiringMessageForm (in General section) • BindProfile1 (in LDAP section) <p>Note: This parameter is commented out; you must uncomment to use it.</p> <p>Removed parameters (in ObjectStore section):</p> <ul style="list-style-type: none"> • Name • Path <p>See “(Optional) Steps for updating the configuration file” on page 58 and “Steps for tailoring the LDAP section of the configuration file” on page 76 for information about updating the pkiserv.conf configuration file. See Chapter 23, “The pkiserv.conf configuration file” on page 261 for a code sample.</p>	None.

File name	Release	Description	Related support
pkiserv.tpl	V1R4	Updated: The pkiserv.tpl certificate templates file was updated, and this affects the Web pages. See “Steps for retrofitting release changes into the PKI Services certificate templates” on page 121 for more information about changes and Chapter 24, “The pkiserv.tpl certificate templates file” on page 263 for a code sample.	None.
expiringmsg.form	V1R4	New: A new e-mail form was added to send when a certificate is expiring.	None.
readymsg.form	V1R4	New: A new e-mail form was added to send when a certificate is ready for retrieval.	None.
rejectmsg.form	V1R4	New: A new e-mail form was added to send when a certificate request is rejected.	None.

Messages

For detailed information about the new and changed PKI Services messages, see Chapter 21, “Messages” on page 239. For information about other message changes that may affect your installation, refer to *z/OS Summary of Message Changes*.

SYS1.SAMPLIB members

The following table lists changes to PKI Services members of SYS1.SAMPLIB:

Member name	Release	Description	Related support
IKYRVSAM	V1R4	New: Contains sample IDCAMS JCL to create VSAM data sets that you use if you intend to use RLS and parallel sysplex support.	None.

Utilities

The following table lists the changes made to PKI Services utilities for z/OS Version 1 Release 4:

Utility name	Release	Description	Related support
vosview	V1R4	Updated: Support for parameters -r and -c was added.	None.
iclview	V1R4	Updated: Support for parameters -r and -c was added.	None.

Chapter 4. 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 5, “Running IKYSETUP to perform RACF administration” on page 37 through Chapter 10, “Creating VSAM data sets” on page 81), your team needs to set up prerequisite products:

- z/OS HTTP Server
- OCSF and OCEP
- LDAP
- ICSF (optional)
- sendmail (optional)

You need to install and configure the z/OS HTTP Server, OCSF and OCEP, and LDAP only if you are setting up prerequisite products for PKI Services for the first time. Installing ICSF is optional. You need to configure sendmail only if you are sending e-mail notifications to users (about rejected certificate requests or certificates that are ready for retrieval or expiring).

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 OCSF and OCEP” on page 29
3. “Installing and configuring LDAP” on page 30
4. “Installing and configuring ICSF (optional)” on page 32
5. “Configuring sendmail (optional)” on page 33

This chapter explains these tasks in more detail.

Installing and configuring the z/OS HTTP Server

You need 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 5, “Running IKYSETUP to perform RACF administration” on page 37 and Chapter 8, “Updating z/OS HTTP Server configuration and starting the server” on page 71 — 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 will need Web server programming skills to complete this procedure.
2. You may 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:

If ...	Then ...	Notes
The z/OS HTTP Server is not installed and configured...	Install and configure z/OS HTTP Server by following the instructions in the installation section of <i>z/OS HTTP Server Planning, Installing, and Using</i> .	Recommendation: For PKI Services, when you install the z/OS HTTP Server, do not use a password file.
The z/OS HTTP Server is installed but not configured for SSL...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 5, "Running IKYSETUP to perform RACF administration" on page 37.)	
The z/OS HTTP Server is installed and configured for SSL using a RACF key ring...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 5, "Running IKYSETUP to perform RACF administration" on page 37.)	
The z/OS HTTP Server is installed and configured for SSL using gskkyman...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 5, "Running IKYSETUP to perform RACF administration" on page 37. The RACF programmer also needs to add your CA certificate to an existing keyfile; see Appendix B, "Using a gskkyman key database for your certificate store" on page 367 for information about gskkyman steps.)	

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

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

Table 8. z/OS HTTP Server information you need to record

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

Installing and configuring OCSF and OCEP

PKI Services requires OCSF and OCEP to be installed and configured so that the user ID under which the PKI Services daemon runs can use required services. The OCSF and OCEP programmer also needs to record some information.

Steps for installing and configuring OCSF and OCEP to work with PKI Services

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

Before you begin:

1. Although the base feature of z/OS includes OCSF and ICSF, if you are in the United States or Canada, make sure you have ordered and installed the additional OCSF Security Level 3 feature. (There is no charge for this feature.)
2. You will need OCSF and OCEP programming skills to complete this procedure.
3. You may need to refer to the configuration information in the following documents:
 - *z/OS Open Cryptographic Services Facility Application Programming*
 - *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming*

These documents contain:

- Instructions on how to set up the necessary security authorizations using RACF
- Information on the RACF program control definitions necessary for OCSF
- Instructions on how to run the installation scripts necessary to use OCSF and OCEP.

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

1. If OCSF and OCEP are not already installed and configured, follow the instructions for how to do so in the previously listed documents.
2. If the value set for the registry directory differs from the default of `'/var/ocsf'`, record the new value in the following table. (If it differs from the default, the UNIX programmer will need to update the `OCSFREGDIR` environment variable in the PKI Services environment variables file, `pkiserv.envars`.)

Installing and configuring prerequisites

Table 9. OCSF information you need to record

OCSF information	Explanation	Default value or customized value
Value set for the registry directory	This is the location of the OCSF registry. The default is <code>'/var/ocsf'</code> .	<code>'/var/ocsf'</code>

A later chapter, Chapter 28, “The certificate validation service” on page 337, provides information about the PKI Services OCSF Trust Policy, PKITP. For information about configuring this, see “Configuring and getting started with PKITP” on page 341.

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

Although it may 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 is recommended but not required. 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 365. The remainder of this chapter assumes you will use the z/OS LDAP Server.

Before you begin:

1. You will need LDAP programming skills to complete this procedure.
2. You will need to refer to the following document:

z/OS Security Server LDAP Server Administration and Use

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

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

If ...	Then...	Notes
You do not have LDAP installed and configured...	Follow the instructions in the Administration section of <i>z/OS Security Server LDAP Server Administration and Use</i> .	Note: It is not necessary to set up the LDAP server for SSL because PKI Services does not use SSL to communicate with the LDAP server.
You have LDAP installed and configured but not for the TDBM backend...	You need to migrate to the TDBM backend. See <i>z/OS Security Server LDAP Server Administration and Use</i> for details about how to do this.	

If ...	Then...	Notes
You have LDAP installed and configured for the TDBM backend...	Go to the next step.	

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

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

Table 10. LDAP information you need to record

LDAP information	Explanation	Value
Administrator's distinguished name	<p>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:</p> <pre>CN=Tim Jones,O=IBM,C=US</pre> <p>CAs typically have distinguished names in the following form:</p> <pre>OU=your-CA's-friendly-name,O=your-organization, C=your-country-abbreviation</pre> <p>The LDAP administrator defines the administrator's distinguished name with the adminDN keyword in the <code>/etc/ldap/slapd.conf</code> configuration file. For example, the value is "cn=Admin" in the following:</p> <pre>adminDN="cn=Admin"</pre>	
Administrator password	<p>This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example:</p> <ul style="list-style-type: none"> • By specifying the password as a TDBM entry by using the userPassword attribute in the ldif2tdbm load utility • (Not recommended) by using the adminPW keyword in the <code>slapd.conf</code> configuration file. 	
LDAP fully qualified domain name and port	<p>This is the domain name on which the LDAP server is listening. For example, for <code>ldap.widgets.com:389</code>, the fully qualified domain name is <code>ldap.widgets.com</code> and the port is 389. See Table 8 on page 29 for a definition of fully qualified domain name.</p>	

Installing and configuring prerequisites

Table 10. LDAP information you need to record (continued)

LDAP information	Explanation	Value
Suffix	<p>A suffix in LDAP is the top-level name of the subtree. For example, for the following distinguished name:</p> <p><code>OU=your-CA's-friendly-name,0=your-organization,C=your-country-abbreviation</code></p> <p>the suffix could be either <code>"0=your-company,C=your-country-abbreviation"</code> or <code>"C=your-country-abbreviation"</code>.</p> <p>The suffix value is specified after the suffix keyword in the <code>slapd.conf</code> file:</p> <p><code>suffix "0=your-company,C=your-country-abbreviation"</code></p> <p>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.</p>	

-
- The chapters that follow require the LDAP server to be running. Follow the instructions in the chapter about running the LDAP server in *z/OS Security Server LDAP Server Administration and Use*.
-

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 recommended 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 5, "Running IKYSETUP to perform RACF administration" on page 37.)

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 will need ICSF programming skills to complete this procedure.
- You may need to refer to the following document:

z/OS 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.

If ICSF is not already installed and configured for PKA, do this by following the instructions in *z/OS ICSF Administrator's Guide*.

Configuring sendmail (optional)

If your installation plans to send e-mail notifications to users whose certificate request is rejected or whose certificate is ready for retrieval or expiring, the UNIX programmer needs to configure sendmail.

Before you begin: You need the following document:

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

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

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

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

3. Load this configuration file into sendmail.

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. Enter the following command:

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

Installing and configuring prerequisites

Part 2. Configuring your system for PKI Services

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

- Chapter 5, “Running IKYSETUP to perform RACF administration” on page 37 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 6, “Configuring the UNIX runtime environment” on page 53 explains:
 - Copying files, such as the PKI Services configuration file
 - Updating environment variables
 - Updating the PKI Services configuration file
 - Setting up the /var/pkiserv HFS directory.
- Chapter 7, “Tailoring LDAP configuration for PKI Services” on page 69 explains how to update your LDAP configuration (performed earlier — see “Installing and configuring LDAP” on page 30) for PKI Services.
- Chapter 8, “Updating z/OS HTTP Server configuration and starting the server” on page 71 describes updating the z/OS HTTP Server configuration files and starting the z/OS HTTP Server.
- Chapter 9, “Tailoring the PKI Services configuration file for LDAP” on page 75 explains how to update the LDAP section of the PKI Services configuration file.
- Chapter 10, “Creating VSAM data sets” on page 81 explains how to create VSAM data sets.
- Chapter 11, “Starting and stopping PKI Services” on page 85 explains how to start and stop the PKI Services daemon.

|
|
|
|

Chapter 5. Running IKYSETUP to perform RACF administration

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

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. For simplicity IBM recommends that you use surrogate user IDs for internal clients as well, rather than allowing them to access PKI Services under their own identities.)
 - Associating the 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 CA 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
 - Creating a SAF key ring and associating it with the certificate
 - Exporting the CA certificate to an MVS data set and HFS 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.

Overview of IKYSETUP

IKYSETUP consists of several parts:

- Configurable section — This section assigns values to variables.
- A section that issues RACF commands to perform RACF administration tasks (see “Actions IKYSETUP performs by issuing RACF commands” on page 309 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 Figure 3 on page 51.)

Running IKYSETUP

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 11. IKYSETUP — Structure and divisions

Configurable section — assigns values to variables
<ul style="list-style-type: none">• Values you must change to customize (see Table 12 on page 39)• Values you might change that are related to setup (see Table 17 on page 44)• Values you can optionally change (see Table 18 on page 46)
Issues RACF commands
Records information in the log data set

Before you begin

- Remember: You 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 parameters).
- If you are configuring PKI Services for the first time and you want to generate the CA signing key pair using the using the 4758 coprocessor (called PCICC), you need to create your CA certificate manually using RACF before running IKYSETUP; see “Steps for creating a CA certificate using the PCICC” on page 199 for directions.
- 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.

Note: There are three tables because there are three categories of variables:

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

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

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

- Table 12
- Table 16 on page 43
- The rows of Table 17 on page 44 concerning z/OS UNIX level security:
 - unix_sec
 - (If z/OS level security is already set up:) bpx_userid. and pgmcntl_dsn.
- Review the default values in **all** the tables.

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 12. IKYSETUP variables whose values must change

Variable name	Description	Referenced elsewhere	Default value and your company's information
ca_dn	<p>The CA's distinguished name. (For a definition of distinguished name, see Table 10 on page 31.)</p> <p>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 18 on page 46) 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.</p>	<p>The suffix of the PKI Services CA's distinguished name must match the LDAP suffix. (The LDAP suffix is in the LDAP configuration file, slapd.conf. See Table 10 on page 31 for a definition of suffix.)</p> <p>Note: However, do not specify a C('value') if it is not present in your LDAP suffix.</p>	<p>OU('Human Resources Certificate Authority')</p> <p>O('Your Company')</p> <p>C('Your Country 2 Letter Abbreviation')</p> <p>_____</p>
ca_label	<p>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.</p>	No	<p>Local PKI CA</p> <p>(Replace this default if you already have your CA certificate and private key set up in RACF.)</p> <p>_____</p>
daemon_uid	<p>The z/OS UNIX user identifier (UID) associated with the PKI Services daemon user ID.</p>	No	<p>554</p> <p>_____</p>

Running IKYSETUP

Table 12. IKYSETUP variables whose values must change (continued)

Variable name	Description	Referenced elsewhere	Default value and your company's information
pkgid	The z/OS UNIX group identifier (GID) for the PKI Services administration group.	No	655 _____
pkigroup_mem.	Members of the PKI Services administration group are responsible for administering PKI Services functions. Note: IBM recommends that you restrict PKI Services administration tasks to those with the RACF SPECIAL attribute. See page 11 for more information. 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:) _____ _____ _____ _____ _____
surrog_uid	The UID associated with the surrogate user ID.	No	555 _____

Table 12. IKYSETUP variables whose values must change (continued)

Variable name	Description	Referenced elsewhere	Default value and your company's information
web_dn	<p>Your Web server's distinguished name. (For a definition of distinguished name, see Table 10 on page 31.)</p> <p>Notes:</p> <ol style="list-style-type: none"> The RACF administrator copies the fully qualified domain name from an earlier table: Table 8 on page 29. If you already have your Web server configured for SSL: <ul style="list-style-type: none"> Set web_dn="" Update the web_ring row <p>(You need to connect your PKI Services CA certificate to your key ring. See the web_ring row for directions.)</p>	The value of the Web server's common name (CN), which is your server's symbol IP address, for example, www.yourcompany.com), must match your Web server's fully qualified domain name.	<p>CN('www.YourCompany.com')</p> <p>O('Your Company')</p> <p>L('Your City')</p> <p>SP('Your Full State or Province Name')</p> <p>C('Your Country 2 Letter Abbreviation')</p> <hr/>
web_ring	<p>The name of the Web server's SAF key ring.</p> <p>If your Web server is configured for SSL and you are using a RACF key ring, set web_ring to the value of the RACF key ring. If your Web server is configured for SSL and you are using gskkyman, set web_ring="" and see Appendix B, "Using a gskkyman key database for your certificate store" on page 367 for additional directions.</p>	httpd*.conf - KeyFile directive	<p>SSLring</p> <hr/>

Variables whose values may change depending on setup

To help in completing the next table of variables (see Table 17 on page 44) fill out the following four decision tables:

Decision table for restrict_surrog

Use the following decision table to determine the value of restrict_surrog in Table 17 on page 44. 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.

Running IKYSETUP

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

Table 13. Deciding the value of restrict_surrog

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

Decision table for use_icsf

Use the following decision table to determine the value of use_icsf in Table 17 on page 44. The use_icsf variable determines whether you are using ICSF for private key protection.

Recommendation: By default, IKYSETUP does not use ICSF. IBM recommends that you do not change the default the first time you run IKYSETUP but that you change it before going into a production environment. (For information about installing and configuring ICSF, see “Installing and configuring ICSF (optional)” on page 32.)

Table 14. Deciding the value of use_icsf

If ...	Then ...
You want to use ICSF for private key protection	Set use_icsf=1 You also need to review and possibly change the following additional variables in Table 17 on page 44: <ul style="list-style-type: none">• csfkeys_profile• csfserv_profile• csfusers_grp
You do not want to use ICSF	Do not change the default use_icsf=0

Decision table for key_backup

Use the following decision table to determine the value of key_backup in Table 17 on page 44. The key_backup variable determines whether the PKI Services CA certificate and private key should be backed up to an encrypted data set.

Table 15. Deciding the value of key_backup

If ...	Then ...	Notes
You want to back up your CA's certificate and private key to a passphrase encrypted data set	Do not change the default key_backup=1	Note: When you use IKYSETUP, you need to enter a passphrase whose display is not inhibited — it appears on the screen in the clear.
You do not want to back up your CA's certificate and private key to a passphrase encrypted data set	Set key_backup=0	

Decision table for unix_sec

Use the following decision table to determine the value of `unix_sec` in Table 17 on page 44. The `unix_sec` variable determines whether you want to use z/OS UNIX security, which is a higher level of security. z/OS UNIX provides two levels of security:

UNIX level security

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

z/OS UNIX level security

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

Table 16. Deciding the value of `unix_sec`

If...	Then ...	Notes
You already have z/OS UNIX security set up	Set <code>unix_sec=1</code>	–
You do not have z/OS UNIX security set up and you do not want to set it up	Do not change the default of <code>unix_sec=0</code>	–
You do not have z/OS UNIX security set up and you want to set it up for the first time	Set <code>unix_sec=2</code>	<p>Notes:</p> <ol style="list-style-type: none"> For information about additional manual configuration, see the section about establishing UNIX security in the <i>z/OS UNIX System Services Planning</i>. If you are setting <code>unix_sec=2</code>, you also need to review and possibly update the following variables: <ul style="list-style-type: none"> <code>bpx_userid</code>. <code>pgmcntl_dsn</code>.

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:

Running IKYSETUP

Table 17. IKYSETUP variables you might want to change depending on setup

Variable name	Description	Referenced elsewhere	Default value or your company's information
bpx_userid.	A list of user IDs with daemon and server authority. The bpx_userid.0 is the number of items in the list and the rest of the entries are the bpx user IDs. (This is non-applicable if unix_sec =2.)	No	1(default for number of items) OMVSKERN
csfkeys_profile	A profile to protect the PKI Services key in ICSF. (This is non-applicable if use_icsf= 0.) If you do not want IKYSETUP to create the profile, set csfkeys_profile="". Note: When RACF stores the private key in the PKDS, it generates the label as: 'IRR.DIGTCERT.CERTIFAUTH. unique-time-stamp'	No	IRR.DIGTCERT.CERTIFAUTH.*
csfserv_profile	A profile to protect ICSF services. (This is non-applicable if use_icsf=0.)	No	CSF*
csfusers_grp	A group of authorized ICSF service users. (This is non-applicable if use_icsf=0.)	No	
key_backup	Specifies whether the PKI Services CA certificate and private key should be backed up to an encrypted data set. The value can be: <ul style="list-style-type: none"> • 1 (yes — the default) • 0 (no). Note: When you use IKYSETUP, you need to enter a passphrase whose display is not inhibited — it appears on the screen in the clear.	No	1 (yes)

Table 17. IKYSETUP variables you might want to change depending on setup (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
pgmcntl_dsn.	A list in which pgmcntl_dsn.0 is the number of items in the list and the rest of the entries are a list of load libraries to be program controlled. If you set unix_sec=2, you probably need to update the list of data sets. (This is non-applicable if unix_sec≠2.)	No	9 (default for number of items) <ul style="list-style-type: none"> • 'CEE.SCEERUN' • 'CBC.SCLBDLL' • 'GLD.SGLDLNK' • 'GSK.SGSKLOAD' • 'SYS1.CSSLIB' • 'TCPIP.SEZALINK' • 'SYS1.LINKLIB' • 'CSF.SCSFMOD0' • 'CSF.SCSFMOD1'
restrict_surrog	Specifies whether the surrogate user ID should be marked restricted. The value can be: <ul style="list-style-type: none"> • 0 (no — the default) • 1 (yes) <p>Recommendation: Do not change the default the first time you run IKYSETUP, but change it before going into a production environment.</p>	No	0 (no)
unix_sec	Specifies whether to set up z/OS UNIX level security. (See page 43 for a definition of z/OS UNIX level security.) The value can be: <ul style="list-style-type: none"> • 0 (do not set up — the default) • 1 (is already set up) • 2 (add this level of security) <p>If you are changing unix_sec to 1 or 2, you also need to review and possibly update the bpx_userid. and pgmcntl_dsn. rows.</p> <p>Recommendation: Do not set unix_sec=2 the first time you are running IKYSETUP.</p>	For unix_sec=2, the names of the load libraries need to change.	0 (no)

Running IKYSETUP

Table 17. IKYSETUP variables you might want to change depending on setup (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
use_icsf	<p>Specifies whether PKI Services should use ICSF for private key operations. The value can be:</p> <ul style="list-style-type: none"> • 0 (no — the default) • 1 (yes). <p>If you are changing use_icsf to 1, see also the csfkeys_profile, csfserv_profile, and csfusers_grp rows.</p> <p>Recommendation: Do not change the default the first time you run IKYSETUP, but change it before going into a production environment.</p>	For this to be successful, ICSF must be configured for RSA (PKA) operations and running.	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 18. IKYSETUP variables you can optionally change

Variable name	Description	Referenced elsewhere	Default value or your company's information
backup_dsn	The data set that will contain a backup copy of the PKI Services certificate and private key.	No	'daemon.PRIVATE.KEY.BACKUP.P12BIN' Note: The <i>daemon</i> refers to the daemon variable in this table.
ca_expires	The date the PKI Services CA certificate expires.	No	2020/01/01 Note: You should update this value to the expiration date of your CA certificate.
ca_ring	The name of the PKI Services SAF key ring.	pkiserv.conf - [SAF] KeyRing value	CAring
daemon	The PKI Services daemon user ID.	pkiserv.conf - [SAF] KeyRing value	PKISRVD
export_dsn	The data set that will contain the PKI Services certificate for copying to HFS.	No	'daemon.PRIVATE.CACERT.DERBIN' Note: The <i>daemon</i> refers to the daemon variable in this table.

Table 18. IKYSETUP variables you can optionally change (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
log_dsn	The log data set name.	No	'your-id.PRIVATE.IKYSETUP.LOG' Notes: 1. The <i>your-id</i> refers to the RACF ID of the person running IKYSETUP. (You do not need to add this; MVS adds this for you.) 2. Changing the default is not recommended.
pkigroup	The PKI Services administration group. This is a RACF group containing the list of user IDs that are authorized to use PKI Services administration functions.	No	PKIGRP
surrog	The surrogate user ID for PKI Services. Note: This cannot be an existing user ID (because IKYSETUP creates the user ID with the NOPASSWORD attribute).	httpd*.conf - Surrogate user ID	PKISERV
vsamhlq	The high-level qualifier of the VSAM data sets for PKI Services. Note: The RACF administrator gets this information from the MVS programmer	<ul style="list-style-type: none"> • pkiserv.conf - [ObjectStore] *DSN values • IKYCVSAM - Data sets names 	Same as the daemon variable earlier in this table.
web_expires	The date the Web server certificate expires.	No	2020/01/01 Note: You should update this value to the expiration date of your CA certificate.
web_label	The label for the Web server's certificate.	No	SSL Cert
webserver	The Web server's daemon user ID.	See Web server documentation.	WEBSRV

Steps for performing RACF tasks using IKYSETUP

|
|

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

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

Recommendation: If this is your first attempt to use IKYSETUP, you are recommended to 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.

Running IKYSETUP

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

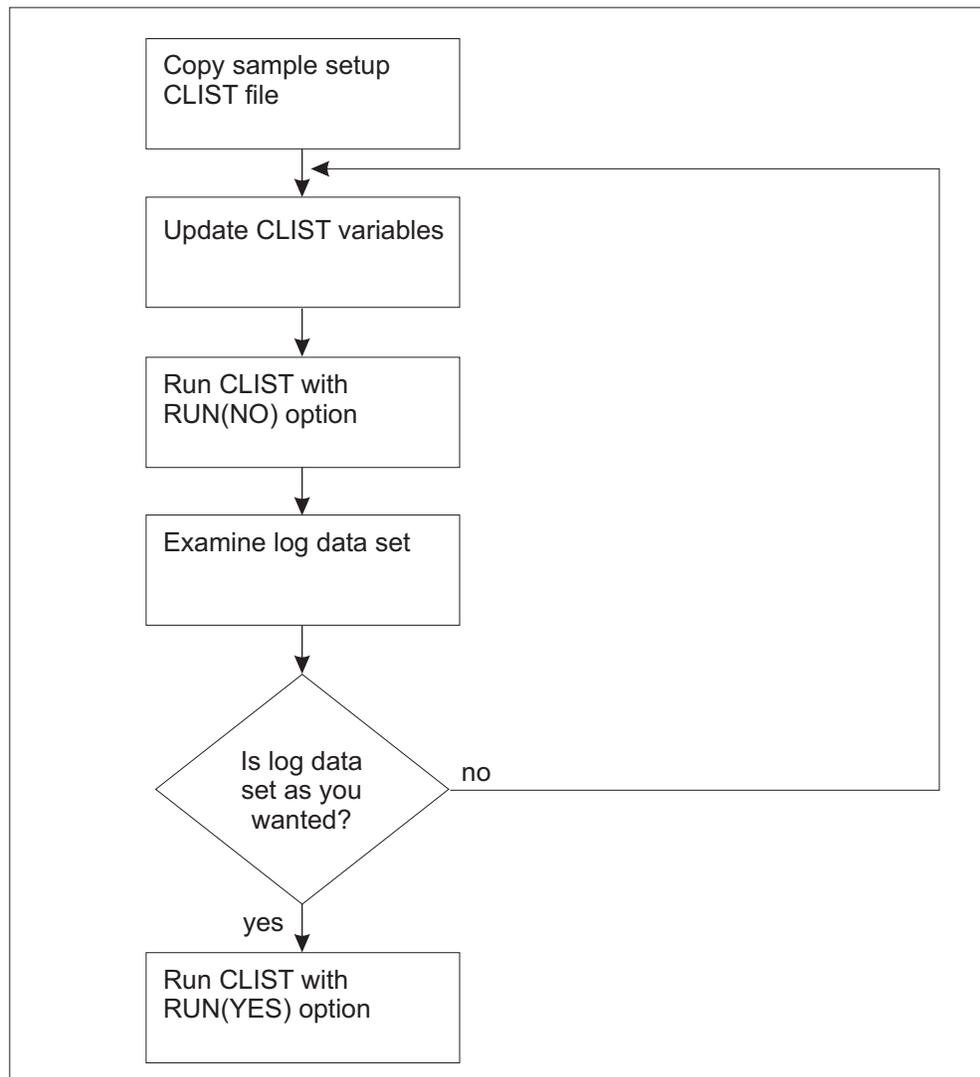


Figure 2. Flowchart of the process of updating IKYSETUP

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

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

2. Edit the IKYSETUP code to update the values of variables you changed in Table 12 on page 39.

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

Example:

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

-
3. If necessary, update the values of variables you changed in Table 17 on page 44.

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

Example:

```
use_icsf=1
```

4. Optionally update any variables you changed in Table 18 on page 46.

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. **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 Figure 3 on page 51 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 309.) The bottom part provides a record of important information that you will need for later steps, such the name of your daemon user ID. Review this information to ensure that the values are the ones you want.

If you want to change any of the commands or information in the log data set, you need to change additional values in IKYSETUP. Remember to record any additional changes in Table 12 on page 39, Table 17 on page 44, and Table 18 on page 46. 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.

Running IKYSETUP

Investigate informational messages and make any necessary corrections. (Informational messages usually indicate a set-up problem that may 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. For the PKI Services proc to start, the PKI Services user ID (by default, PKISRVD) needs read access to the OCSF services. Provide this access by entering the following RACF commands:
 - | PERMIT CDS.CSSM.CLASS(FACILITY) ID(PKISRVD) ACC(READ)
 - | PERMIT CDS.CSSM.CRYPTO CLASS(FACILITY) ID(PKISRVD) ACC(READ)
 - | PERMIT CDS.CSSM.DATALIB CLASS(FACILITY) ID(PKISRVD) ACC(READ)
 - | SETROPTS RACLIST(FACILITY)
- | 10. 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 214.

The following figure shows an example of the data that appears when you run IKYSETUP.

```

Creating users and groups ...
ADDUSER PKISRVD name('PKI Srvs Daemon') nopassword omvs(uid(554) assize(256000000) threads(512))
ADDUSER PKISERV nopassword omvs(uid(555)) name('PKI Srvs Surrogate')
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'))
WITHLABEL('Local PKI CA') NOTAFTER(2020/01/01))
Backing up the CA certificate ...
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA')) DSN('PKISRVD.PRIVATE.KEY.BACKUP.P12BIN')
FORMAT(PKCS12DER) PASSWORD('*****')
Marking CA certificate as HIGHTRUST ...
RACDCERT CERTAUTH ALTER(LABEL('Local PKI CA')) HIGHTRUST
Saving the CA certificate to a data set for OPUT ...
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA')) DSN('PKISRVD.PRIVATE.CACERT.DERBIN') FORMAT(CERTDER)
Creating the PKI Services keyring ...
RACDCERT ADDRING(CAring) ID(PKISRVD)
RACDCERT ID(PKISRVD) CONNECT(CERTAUTH LABEL('Local PKI CA') RING(CAring) USAGE
(PERSONAL) DEFAULT)
Creating the Webserver SSL certificate and keyring ...
RACDCERT GENCERT ID(WEBSRV) SIGNWITH(CERTAUTH LABEL('Local PKI CA')) WITHLABEL
('SSL Cert') SUBJECTSDN(CN('www.YourCompany.com') O('Your Company') L('Your City')
SP('Your Full State or Province Name') C('Your Country 2 Letter Abbreviation'))
NOTAFTER(2020/01/01))
RACDCERT ADDRING(SSLring) ID(WEBSRV)
RACDCERT ID(WEBSRV) CONNECT(CERTAUTH LABEL('Local PKI CA') RING(SSLring))
RACDCERT ID(WEBSRV) CONNECT(ID(WEBSRV) LABEL('SSL Cert') RING(SSLring)
USAGE(PERSONAL) DEFAULT)
Giving PKISRVD access to BPX.SERVER ...
RDEFINE FACILITY BPX.SERVER
PERMIT BPX.SERVER CLASS(FACILITY) ID(PKISRVD) ACCESS(READ)
Allowing the PKI Services daemon to act as a CA ...
RDEFINE FACILITY IRR.DIGTCERT.GENCERT
RDEFINE FACILITY IRR.DIGTCERT.LISTRING
RDEFINE FACILITY IRR.DIGTCERT.LIST
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY) ID(PKISRVD) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(PKISRVD) ACCESS(READ)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(PKISRVD) ACCESS(READ)
Allowing the Webserver to access its keyring ...
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(WEBSRV) ACCESS(READ)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(WEBSRV) ACCESS(READ)
Allowing the Webserver to switch identity to PKISERV ...
SETROPTS CLASSACT(SURROGAT)
RDEFINE SURROGAT BPX.SRV.PKISERV
PERMIT BPX.SRV.PKISERV CLASS(SURROGAT) ID(WEBSRV) ACCESS(READ)
SETROPTS RACLIST(SURROGAT) REFRESH

```

Figure 3. Sample log data set (Part 1 of 2)

```

Creating the STARTED class profile for the daemon ...
RDEFINE STARTED PKISRVD.* STDATA(USER(PKISRVD))
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
SETROPTS RACLIST(STARTED) REFRESH
Allowing PKISERV to request certificate functions ...
SETR GENERIC(FACILITY)
RDEFINE FACILITY IRR.RPKISERV.**
PERMIT IRR.RPKISERV.** CLASS(FACILITY) ID(PKISERV) ACCESS(CONTROL)
Creating the profile to protect PKI Admin functions ...
RDEFINE FACILITY IRR.RPKISERV.PKIADMIN
PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY) ID(PKIGRP) ACCESS(UPDATE)
PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY) ID(PKISERV) ACCESS(NONE)
SETROPTS RACLIST(FACILITY) REFRESH

```

Information needed for PKI Services UNIX set up:

The daemon user ID is:
PKISRVD

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

The PKI Services' DER encoded certificate is in data set:
'PKISRVD.PRIVATE.CACERT.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

Figure 3. Sample log data set (Part 2 of 2)

Chapter 6. Configuring the UNIX runtime environment

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

- Using a sysplex for PKI Services daemons
- Sending e-mail notification for certificates ready for retrieval or expiration or rejected certificate requests

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, set up the /var/pkiserv directory.

The following table summarizes information about copying and updating files:

Table 19. Deciding which files to copy and change

File	Purpose	Need to copy?	Need to change?
pkiserv.conf	Configuration file. Contains various settings and values PKI Services needs.	Only if you are configuring PKI Services for the first time.	<p>The UNIX programmer may need to change the LDAP section of this file, but IBM recommends doing this later (see Chapter 9, “Tailoring the PKI Services configuration file for LDAP” on page 75).</p> <p>The UNIX programmer needs to update the non-LDAP pkiserv.conf configuration file if any of the following is true:</p> <ul style="list-style-type: none"> • You intend to run multiple instances of PKI Services in a sysplex • You are migrating from Release 3 and you intend to send e-mail notifications for certificate-related events • You are configuring PKI Services for the first time and do not intend to send e-mail notifications for certificate-related events
pkiserv.tmpl	Certificate templates file. Contains HTML-style code that builds the Web pages underlying certificate requests.	Only if you are configuring PKI Services for the first time.	Recommendation: Make no changes to this file until later. See Chapter 13, “Customizing the administration Web pages” on page 133 for details about making changes.
pkiserv.envars	The environment variables file.	Only if you are configuring PKI Services for the first time and the file needs changes	UNIX programmer may have to update this file. See “Optionally updating PKI Services environment variables” on page 55.

Configuring the UNIX runtime environment

Table 19. Deciding which files to copy and change (continued)

File	Purpose	Need to copy?	Need to change?
rejectmsg.form	The form for an e-mail sent to a user when the PKI Services administrator has rejected a certificate request.	Only if your company sends an e-mail notification to a user after the PKI Services administrator has rejected a certificate request	Recommendation: Make no changes to this file until later. See Chapter 13, “Customizing the administration Web pages” on page 133 for details about making changes.
readymsg.form	The form for an e-mail sent to a user when the PKI Services administrator has approved a certificate request and the certificate is ready for retrieval.	Only if your company sends an e-mail notification to a user after the PKI Services administrator has approved a certificate request and the certificate is ready for retrieval.	Recommendation: Make no changes to this file until later. See Chapter 13, “Customizing the administration Web pages” on page 133 for details about making changes.
expiringmsg.form	The form for an e-mail sent to a user when a certificate is going to expire.	Only if your company sends an e-mail notification to a user about a certificate that is going to expire	Recommendation: Make no changes to this file until later. See Chapter 13, “Customizing the administration Web pages” on page 133 for details about making changes.

(To view the contents of any of these files, see Chapter 27, “Other code samples” on page 327.)

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 HFS directory where the MVS programmer installed PKI Services and the runtime directory, *HFS-install-dir* and *runtime-dir* in the commands that follow. The defaults are */usr/lpp/pkiserv/* and */etc/pkiserv* respectively. The MVS programmer was asked to record any changes to these defaults; see Table 4 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 and template files by entering the following commands from the UNIX command line.

Note: To use these commands, your user ID must have super user authority.

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

2. If your company is sending e-mail notifications to users (when certificate requests are rejected or when certificates are ready for retrieval or expiring),

Configuring the UNIX runtime environment

copy the appropriate notification files from the samples directory to the runtime directory by entering commands such as the following:

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

-
3. If you are configuring PKI Services for the first time, examine the values in the environment variables file (by default, `pkiserv.envars`). If any values need to change (such as the `OCSFREGDIR`, the environment variable for the OCSF registry directory — see step 2 on page 29), copy this file by entering the following command:

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

Optionally updating PKI Services environment variables

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

- You are configuring PKI Services for the first time
- You 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 are migrating from Release 3 where you are running with customized environment variables and you want to send e-mail notifications.

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 of the OCSF Registry directory (`/var/ocsf`)
- The location for `sendmail` (`/usr/sbin/sendmail`)

Recommendation: 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 335.) `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

Configuring the UNIX runtime environment

pkiserv.envars as recommended, 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
- stdout
- stderr
- time zone

See “Steps for starting the PKI Services daemon” on page 85.

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

You must specify any environment variables that PKI Services requires either in the PKISERVD procedure or in the environment variables file (pkiserv.envars). IBM recommends making additions and changes to the environment variables file.

(Optional) Steps for updating PKI Services environment variables

Before you begin: See page 55 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.envars) and update the file as necessary. (See “Environment variables in the environment variables file” on page 305 for a description of the environment variables and “The pkiserv.envars environment variables file” on page 307 for a code sample of the environment variables file.)

Notes:

- a. If the value set for the OCSF registry directory differs from the default value of '/var/ocsf', you need to update the OCSFREGDIR environment variable.
- b. If you did not install sendmail in its default location (/usr/sbin), you need to update the PATH environment variable.

Note:

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 335 for a code sample of the PKISERVD proc.)
3. If you are migrating from Release 3 where you are already running with a customized environment variables file and you want to send e-mail notifications, you need to add a PATH statement to your environment variables file. If you

Configuring the UNIX runtime environment

installed sendmail in its default location (/usr/sbin), then you can copy the PATH statement from the sample file shipped with PKI Services (/usr/lpp/samples/pkiserv.envars). Otherwise add a PATH statement such as the following:

```
PATH=/directory-where-sendmail-resides
```

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

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. **Everything in the pkiserv.conf file — including section names, keys, and values — is case-sensitive.** Each section of the pkiserv.conf configuration file has a title enclosed in square brackets. The configuration file includes the following sections:

[OIDs]

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

```
<name>=<dotted-decimal>
```

The following excerpt is from the OIDs section:

```
[OIDs]
:
MyPolicy=1.2.3.4
```

[ObjectStore]

The ObjectStore section specifies operational information for 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]
SigAlg1=sha-1WithRSAEncryption
:
```

[General]

The General section is for general information.

The following excerpt is from the General section:

Configuring the UNIX runtime environment

[General]

```
InitialThreadCount=10
⋮
```

[SAF]

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

The following excerpt is from the SAF section:

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

[LDAP]

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

The following excerpt is from the LDAP section:

```
[LDAP]
NumServers=1
⋮
```

The UNIX programmer needs to update the LDAP section of this file, but IBM recommends doing this later (see Chapter 9, “Tailoring the PKI Services configuration file for LDAP” on page 75).

(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 24 on page 76.) Read the parameter descriptions, and examine the default values in the rightmost column to ensure that the values meet your company’s requirements. As necessary, cross out the defaults and enter the information appropriate to your own company’s needs and policies.

Table 20. Information needed for updating the configuration file

Parameter	Information needed	Where to get this information	Default value or customized value
OIDs section			
MyPolicy=	A registered Object ID identifying your organization’s usage policy, for example: 1.2.3.4	Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	1.2.3.4 If you need to use the CertificatePolicies extension, replace 1.2.3.4 with the value of your Object ID: _____
ObjectStore section			
ObjectDSN=	VSAM data set name for ObjectStore data. This is the request database. Each VSAM request record consists of a fixed header followed by a variable-length section.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 18 on page 46. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrvd.vsam.ost' Note that this begins with the VSAM high-level qualifier.

Configuring the UNIX runtime environment

Table 20. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value or customized value
ObjectTidDSN=	VSAM data set name for the ObjectStore alternate index.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 18 on page 46. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrvd.vsam.ost.path' Note that this begins with the VSAM high-level qualifier.
ICLDSN=	VSAM data set name for ICL data. 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.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 18 on page 46. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrvd.vsam.icl' Note that this begins with the VSAM high-level qualifier.
RemoveCompletedReqs=	Time period that completed certificate requests remain in the ObjectStore before automatic deletion. This is a number followed by d (days) or w (weeks).	UNIX programmer decides this value.	1w
RemoveInactiveReqs=	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).	UNIX programmer decides this value.	4w
SharedVSAM=	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).	UNIX programmer decides this value.	F
CertPolicy section			

Configuring the UNIX runtime environment

Table 20. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value or customized value
SigAlg1=	<p>The Object ID for the signature algorithm. This must be an RSA signature algorithm:</p> <ul style="list-style-type: none"> • sha-1WithRSA Encryption (the default) • md-5WithRSAEncryption • md-2WithRSAEncryption <p>Note: Changing the default also requires adding a line in the OIDs section. See “Updating the signature algorithm” on page 141.</p>	Do not change this information until you are performing advanced customization. See “Updating the signature algorithm” on page 141 for more information.	sha-1WithRSA Encryption
CreateInterval=	How often the certificate creation thread scans the database for approved requests. This is a number followed by w (weeks), d (days), h (hours), m (minutes), or s (seconds).	UNIX programmer decides this value.	3m
ExpireWarningTime	<p>Note: You need a value for this parameter only if you are sending e-mail notifications to users when certificates are expiring.</p> <p>This parameter indicates how soon before certificate expiration to send a warning message (that is, the number of days or weeks before the day and time the certificate expires).</p> <p>This name-value pair is optional. Its absence indicates no expiration checking is performed. 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 is done.</p>	UNIX programmer decides this value.	4w

Configuring the UNIX runtime environment

Table 20. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value or customized value
TimeBetweenCRLs=	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). Note: 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.	UNIX programmer decides this value.	1d
CRLDuration=	The amount of time that a certificate revocation list is valid. This is a number followed by w (weeks), d (days), h (hours), m (minutes), or s (seconds).	UNIX programmer decides this value.	2d
PolicyRequired=	Whether the CertificatePolicies extension is included in the certificate. This is T (True) or F (False). Unless you change this to T, the following fields in the CertPolicy section are ignored.	UNIX programmer decides this value. It should be T if you are using the CertificatePolicies extension or F otherwise. Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	F
PolicyCritical=	Whether the CertificatePolicies extension is created with the critical flag turned on. This is T (True) or F (False).	UNIX programmer decides this value. It should be T if you are using the CertificatePolicies extension or F otherwise. Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	F

Configuring the UNIX runtime environment

Table 20. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value or customized value
PolicyName1=	The Object ID for the policy. (This is the same value that is in the MyPolicy parameter of the OIDs section.)	Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	<i>MyPolicy</i> If you changed PolicyRequired=F to PolicyRequired=T, replace the variable <i>MyPolicy</i> with the same value that is in the MyPolicy parameter of the OIDs section. _____
Policy1Org=	This is the organization name for the CertificatePolicies extension, for example, International Business Machines, Inc.	Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	My Company, Inc. If you are changing PolicyRequired=F to PolicyRequired=T, you need to specify your own value for this: _____
Policy1Notice1=	The first company notice number.	Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	1 If you are changing PolicyRequired=F to PolicyRequired=T, you need to specify your own value for this: _____
Policy1Notice2=	The second company notice number.	Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	17 If you are changing PolicyRequired=F to PolicyRequired=T, you need to specify your own value for this: _____
UserNoticeText1=	A legal statement about certificate issuance and use, for example: Certificate for IBM internal use only	Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	<i>statement</i> If you are changing PolicyRequired=F to PolicyRequired=T, you need to replace the variable <i>statement</i> with your own value for this: _____
CPS1=	The Uniform Resource Identifier (URI) where your organization’s Certification Practice Statement (CPS) is located. This is in the form: http://www.mycompany.com/cps.html	Do not change this information until you are performing advanced customization. See “Steps for creating the CertificatePolicies extension” on page 139 for more information.	http://www.mycompany.com/cps.html If you are changing PolicyRequired=F to PolicyRequired=T, you need to replace the variable <i>mycompany</i> with your own value for this: http://www._____.com/cps.html
General section			

Configuring the UNIX runtime environment

Table 20. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value or customized value
InitialThreadCount=	Number of threads (at least 2 and no more than 100) the PKI Services daemon should create at program initialization.	UNIX programmer decides this value.	10
ReadyMessageForm=	The full pathname or data set name containing the 'Your certificate is ready' message form. Using this name-value pair is optional. If you do not specify this name-value pair, no message is sent.	UNIX programmer decides this value.	/etc/pkiserv/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
ExpiringMessageForm=	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.	UNIX programmer decides this value.	/etc/pkiserv/expiringmsg.form
SAF section			
KeyRing=	The fully qualified name of the SAF key ring for PKI Services to use. (This must consist of an uppercase user ID "/" case-sensitive ring name.)	See Table 18 on page 46.	PKISRVD/Caring
LDAP section — For information about the LDAP section, see Table 24 on page 76.			

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

Configuring the UNIX runtime environment

- Sending e-mail notifications to users if the PKI Services administrator rejects certificate requests or certificates are ready for retrieval or expiring

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

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 are configuring PKI Services for the first time, you can omit the following change. (The pkiserv.conf configuration file that is shipped with the product starting in z/OS Version 1 Release 4 does not contain the Name= line. This line was included in z/OS Version 1 Release 3.)

If you are migrating from z/OS Version 1 Release 3 for PKI Services, you may have updated the value `pkica` on the following line. Leave this line in the pkiserv.conf file until migration is completed. You can tell when migration has completed because the PKI Services daemon renames the HFS files, appending `.MIGRATED` to the file names. After migration is completed, you can delete the Name= line if you wish:

```
Name=pkica
```

If are configuring PKI Services for the first time, you can omit the following change. (The pkiserv.conf configuration file that is shipped with the product starting in z/OS Version 1 Release 4 does not contain the Path= line. This line was included in z/OS Version 1 Release 3.)

If you are migrating from Release 3, you may have updated the value `/var/pkiserv` on the following line. Leave this line in the pkiserv.conf file until migration is completed. After this, you can delete the Path= line if you wish:

```
Path=/var/pkiserv
```

- b. If necessary, change `pkisrvd` in the following lines to the value of the VSAM high-level qualifier in the ObjectDSN=, ObjectTidDSN=, and ICLDSN= rows in Table 20 on page 58; if you changed the file names after the period, replace these values also:

```
ObjectDSN=pkisrvd.vsam.ost'  
ObjectTidDSN=pkisrvd.vsam.ost.path'  
ICLDSN=pkisrvd.vsam.icl'
```

Note: 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 83), you must make equivalent changes in pkiserv.conf.

If you are migrating from Release 3 and you want to use a sysplex, you need to set up RLS. This involves running IKYRVSAM to reallocate your VSAM data sets. It may also involve changing the names of the source and destination data sets in IKYRVSAM. If you change the names of the destination data sets in IKYRVSAM to

Configuring the UNIX runtime environment

different names than you currently have in the pkiserv.conf configuration file, then you need to update pkiserv.conf as well.

- c. If necessary, change **1w** in the following line to the value in the RemoveCompletedReqs= row in Table 20 on page 58:
RemoveCompletedReqs=**1w**
- d. If necessary, change **4w** in the following line to the value in the RemoveInactiveReqs= row in Table 20 on page 58:
RemoveInactiveReqs=**4w**
- e. If necessary, update the SharedVSAM lines:
 - If you intend to use sysplex and you are migrating from z/OS Version 1 Release 3 for PKI Services, locate the following lines in the sample configuration file (/usr/lpp/pkiserv/samples/pkiserv.conf) and copy them to the bottom of the ObjectStore section; then change F in the last line to T.
Are the VSAM data sets shared in a sysplex with other instances
of PKI Services. True (T) or False (F)
SharedVSAM=F
 - 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 migrating from Release 3 or 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 20 on page 58:
CreateInterval=**3m**
- b. If necessary, update the ExpireWarningTime line(s):
 - If you are sending e-mail notifications (about rejected certificate requests or certificates ready for retrieval or expiring) and you are migrating from Release 3, locate the following lines in the sample configuration file (/usr/lpp/pkiserv/samples/pkiserv.conf) and copy them into the CertPolicy section (after the CreateInterval parameter). If necessary, change the value 4w to the value in the ExpireWarningTime row of Table 20 on page 58.
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
 - 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 20 on page 58. :
ExpireWarningTime=4w
 - If you are not using e-mail notifications and you are configuring PKI Services for the first time, remove the ExpireWarningTime=4 line from the pkiserv.conf file.
 - If you are not using e-mail notifications and you are migrating from Release 3, you do not need to do anything.
- c. If necessary, change **1d** in the following line to the value in the TimeBetweenCRLs= row in Table 20 on page 58:
TimeBetweenCRLs=**1d**

Configuring the UNIX runtime environment

- d. If necessary, change **2d** in the following line to the value in the CRLDuration= row in Table 20 on page 58:

```
CRLDuration=2d
```

- e. If necessary, change **F** in the following line to the value in the PolicyRequired= row in Table 20 on page 58:

```
PolicyRequired=F
```

-
3. If necessary, update the General section:

- a. If necessary, change **10** in the following line to the value in the InitialThreadCount row in in Table 20 on page 58:

```
InitialThreadCount=10
```

- b. If necessary update the ReadyMessageForm, RejectMessageForm, and ExpiringMessageForm lines:

- If you are sending e-mail notifications (about rejected certificate requests or certificates that are ready for retrieval or expiring) and you are migrating from Release 3, copy the following lines from the sample configuration file (/usr/lpp/pkiserv/samples/pkiserv.conf) into your pkiserv.conf configuration file (at the bottom of the General section). If necessary, change the values of the pathname in the uncommented lines to the corresponding values in Table 20 on page 58.

```
# 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
```

- 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 following three lines to the corresponding values in Table 20 on page 58:

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

```
RejectMessageForm=/etc/pkiserv/rejectmsg.form
```

```
ExpiringMessageForm=/etc/pkiserv/expiringmsg.form
```

- If you are not sending e-mail notifications and you are configuring PKI Services for the first time, delete all of the following lines in the pkiserv.conf configuration file:

```
# 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
```

Configuring the UNIX runtime environment

- If you are not sending e-mail notification and you are migrating from Release 3, you do not need to do anything.

-
4. If necessary, in the SAF section, change **PKISRVD/CAring** in the following line to the value in the KeyRing= row in Table 20 on page 58:

```
KeyRing=PKISRVD/CAring
```

Steps for setting up the /var/pkiserv directory

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

PKI Services needs to set up HFS files in a directory. (The default location is /var/pkiserv.) You need to set up this location and make the PKI Services daemon (by default, PKISRVD) the owner.

Then you copy the CA certificate from its MVS data set to the cacert.der in the directory (the default location is /var/pkiserv) and change its permission settings. (The data set was created earlier. See “Before you begin” on page 38; the default name of the data set is 'pkisrvd.private.cacert.derbin'.)

Perform the following steps to set up the /var/pkiserv directory (if you are configuring PKI Services for the first time):

1. Change the ownership of the directory to PKISRVD by entering the following command from the UNIX command line:

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

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

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

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

```
chmod 755 /var/pkiserv/cacert.der
```

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

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

Configuring the UNIX runtime environment

Chapter 7. Tailoring LDAP configuration for PKI Services

You need to tailoring LDAP configuration for PKI Services only if you are configuring PKI Services for the first time.

The directions in this section are for using the z/OS Security Server LDAP for PKI Services. If you intend to use a different LDAP product, you need to refer to the documentation for this product. See Appendix A, “LDAP directory server requirements” on page 365 for information about installing a non-z/OS LDAP.

The LDAP programmer needs to update the schema.user.ldif file so that the LDAP server understands the format of entries that will be stored in the directory.

Steps for updating schema.user.ldif

Before you begin:

- Remember: You need to perform this task only if you are configuring PKI Services for the first time.
- 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 30.
- You need to know the following information from LDAP installation. Copy the information into the following table from (completed) Table 10 on page 31:

Table 21. LDAP information you need for tailoring LDAP configuration

LDAP information	Explanation	Value
Administrator's distinguished name	This is the distinguished name to use for LDAP binding. (For a definition of distinguished name, see Table 10 on page 31. The LDAP administrator defines the administrator's distinguished name with the adminDN keyword in the /etc/ldap/slapd.conf configuration file. For example, the value is "cn=Admin" in the following: adminDN="cn=Admin"	
Administrator password	This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example: <ul style="list-style-type: none"> – By specifying the password as a TDBM entry by using the userPassword attribute in the ldif2tdbm load utility – (Not recommended) by using the adminPW keyword in the slapd.conf configuration file. 	
LDAP fully qualified domain name and port	This is the IP address and port on which the LDAP server is listening. For example, for ldap.widgets.com:389, the fully qualified domain name is ldap.widgets.com and the port is 389. See Table 8 on page 29 for a definition of fully qualified domain name.	
Suffix	(For a definition of suffix, see Table 10 on page 31.) The suffix value is specified after the suffix keyword in the slapd.conf file. suffix "o=your-company,c=your-country-abbreviation"	

You need to update the schema.user.ldif file only if you are configuring PKI Services for the first time.

If you have already configured your LDAP schema using a schema.user.ldif file that is from before z/OS V1R2 (or if you are using schema files other than

Tailoring LDAP configuration for PKI Services

| schema.user.ldif), see *z/OS Security Server LDAP Server Administration and Use*
| for instructions on how to change the schema. Refer to the sections about the
| LDAP directory schema for TDBM and the minimum schema for TDBM. PKI
| Services requires everything in the minimum schema, plus RFC2587.ldif and its
| prerequisites.

| If you are configuring your LDAP schema for the first time, perform the following
| steps:

1. Copy the /usr/lpp/ldap/etc/schema.user.ldif file to the directory from which you are working by entering the following z/OS UNIX shell command:

```
cp /usr/lpp/ldap/etc/schema.user.ldif .
```

-
2. Edit the schema.user.ldif file in the current directory, ensuring that the "dn:" line (the first line in the file) has the following form and replacing *Your Company Suffix* with the suffix from Table 21 on page 69:

```
dn: cn=schema, Your Company Suffix
```

-
3. Load the schema defined in the schema.user.ldif file into the directory by entering the following command. Replace *adminDN* and *passwd* with the adminDN and adminPW values from Table 21 on page 69.

```
ldapmodify -D adminDN -w passwd -V 3 -f schema.user.ldif
```

Chapter 8. Updating z/OS HTTP Server configuration and starting the server

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

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

- Updating the z/OS HTTP Server's 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's 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 27). 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 22. Summary of configuration and usage of each Web server instance

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

Before you begin:

- Remember: You need to perform these steps only if you are configuring PKI Services for the first time.
- You need to know the HFS install directory (the HFS directory where the MVS programmer installed PKI Services), called *HFS-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 4 on page 9.
- You need to know the following LDAP information. Record the information in the rightmost row of the following table:

Updating z/OS HTTP Server configuration and starting the server

Table 23. LDAP information you need for tailoring z/OS HTTP Server configuration

LDAP information	Explanation	Value
Administrator's distinguished name	This is the distinguished name to use for LDAP binding. (For a definition of distinguished name, see Table 10 on page 31.) The LDAP administrator defines the administrator's distinguished name with the adminDN keyword in the <code>/etc/ldap/slapd.conf</code> configuration file. For example, the value is "cn=Admin" in the following: adminDN="cn=Admin"	
Administrator password	This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example: <ul style="list-style-type: none"> – By specifying the password as a TDBM entry by using the userPassword attribute in the ldif2tdbm load utility – (Not recommended) by using the adminPW keyword in the <code>slapd.conf</code> configuration file. 	
LDAP fully qualified domain name	This is the IP address on which the LDAP server is listening, for example, for <code>ldap.widgets.com</code> . See Table 8 on page 29 for a definition of fully qualified domain name.	
LDAP port	This is the port for LDAP, for example, <code>389</code> in <code>ldap.widgets.com:389</code>	

Perform the following steps to update the z/OS HTTP Server's 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, `/HFS-install-dir/samples/httpd.conf` file) into the default configuration file, `/etc/httpd.conf`.

Note: The `HFS-install-dir`, your HFS installation directory, by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default (see Table 4 on page 9).

- a. Copy the `keyfile`, `sslmode`, `sslport`, and `normalmode` directives as is, replacing any existing values.
- b. If your organization customized the value of `web_ring` (see Table 12 on page 39), 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.

Recommendation:

You are recommended to 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.

Updating z/OS HTTP Server configuration and starting the server

- 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 8 on page 29.)
- i. Change all instances of *application-root* to your HFS installation directory, which is `usr/lpp/pkiserv` by default.

Note: Your HFS installation directory by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default (see Table 4 on page 9).

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

Note: The *HFS-install-dir*, your HFS installation directory, by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default (see Table 4 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 12 on page 39), 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 23 on page 72.
- e. Copy the protection and protect directives after any protection and protect directives you already have. Do not change the order in which these directives appear.
- f. Copy the redirect directives after any redirect directives you already have. Do not change the order in which these directives appear.
- g. Copy the exec directives before any pass and exec directives you already have.
- h. Change all instances of *server-domain-name* to your Web server's fully qualified domain name, for example, www.ibm.com. (For information about your Web server's fully qualified domain name, see Table 8 on page 29.)
- i. Change all instances of *application-root* to your HFS installation directory.

Note: Your HFS installation directory by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default (see Table 4 on page 9).

Updating z/OS HTTP Server configuration and starting the server

- j. If you created `httpd1443.conf` by copying `httpd.conf`, optionally change the directories in `httpd1443.conf` for the report, log, and pid files. (IBM recommends that you do this to ensure the two servers are not using the same files at the same time.) To do this:

- 1) Create a new directory for the `httpd1443` files by using the following command:
- 2) Assign ownership to `WEBSRV` with the following command:
- 3) Edit the `*Log` directives in the new `httpd1443.conf` file to provide unique path names.

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 30.)

2. Enter 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 9. 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 6, “Configuring the UNIX runtime environment” on page 53 describes tasks the UNIX programmer performs. The other team members perform additional tasks before the UNIX programmer updates the LDAP section of the `pkiserv.conf` configuration file (described in this chapter) and starts the PKI Services daemon (described in Chapter 11, “Starting and stopping PKI Services” on page 85).

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=myldapservers.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 214 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 n` line specifies the fully qualified domain and port of your LDAP server. If your file contains a `Server n` line, PKI Services looks for the matching `AuthName n` and `AuthPwd n` lines and uses these values.

Tailoring the PKI Services configuration file for LDAP

2. The BindProfilen parameter specifies the name of the LDAPBIND class profile. If your file does not contain a Servern line but does contain a BindProfilen line, PKI Services looks for the bind information in the LDAPBIND class profile. (If Servern is present, PKI Services does not look for bind information in BindProfilen, even if the value in Servern 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:

- Remember: 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 24 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 24. Information needed for updating the LDAP section of the configuration file

Parameter	Information needed	Where to get this information	Default value and your company's information
NumServers=	The number of available LDAP servers. These are replicas that can post certificates and CRLs.	From LDAP programmer	1
PostInterval=	How often the posting thread should scan the database for items to post in weeks (w), days (d), hours (h), minutes (m), or seconds (s).	UNIX programmer decides this. Specify a number followed by h (hours), m (minutes) or s (seconds). Example: 6m	5m
Server1=	You use this parameter only if you are storing LDAP passwords in the clear. This parameter's value is the fully qualified domain name (domain name or IP address and port) for the first LDAP server.	Copy this information from the earlier (completed) table, Table 21 on page 69.	<i>myldapserver.mycompany.com:389</i> Note: If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for each server.

Tailoring the PKI Services configuration file for LDAP

Table 24. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
AuthName1=	<p>You use this parameter only if you are storing LDAP passwords in the clear.</p> <p>This parameter's value is the distinguished name to use for LDAP binding. Note: You must specify the OID qualifiers in uppercase and without any spaces surrounding the equal signs or commas that separate the attribute value assertions (AVAs).</p> <p>(See Table 10 on page 31 for a definition of distinguished name.) The LDAP administrator defines the administrator's distinguished name with the adminDN keyword in the <code>/etc/ldap/slapd.conf</code> configuration file. For example, the value is "cn=Admin" in the following: <code>adminDN="cn=Admin"</code></p>	Copy this information from the earlier (completed) table, Table 21 on page 69.	<p><i>CN=root</i></p> <p>Note: If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for each server.</p>

Tailoring the PKI Services configuration file for LDAP

Table 24. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
AuthPwd1=	<p>You use this parameter only if you are storing LDAP passwords in the clear.</p> <p>This parameter's value is the password to use for LDAP binding. The LDAP programmer sets this.</p> <p>Note: Include this parameter, Server1, and AuthName1 only if you are storing the LDAP password in the clear. Alternately, if you are 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 FACILITY class profile. (For more information, see "Using encrypted passwords for LDAP servers" on page 214.)</p>	Copy this information from the earlier (completed) table, Table 21 on page 69.	<p><i>root</i></p> <p>Note: If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for each server.</p>
CreateOUValue=	Value to use for the OU attribute when creating LDAP entries under the objectclass organizationalUnit (see Table 72 on page 365). This is used only when no OU value is specified in the relative distinguished name.	UNIX programmer decides this (after consulting with LDAP programmer)	Created by PKI Services
RetryMissingSuffix=	True (T) or False (F) setting that indicates whether LDAP post requests should be retried later if the distinguished name suffix does not exist. When set to F, LDAP post requests that fail because of a missing suffix are discarded.	UNIX programmer decides this (after consulting with LDAP programmer)	T

Tailoring the PKI Services configuration file for LDAP

Table 24. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
BindProfile1=	<p>You use this parameter only if you intend to use an encrypted password for your LDAP server.</p> <p>This parameter's value is the name of the LDAPBIND class profile containing the bind information for the LDAP server. (For more information, see "Using encrypted passwords for LDAP servers" on page 214.)</p>	<p>Get the profile name from the RACF administrator who creates the profile. See "Using encrypted passwords for LDAP servers" on page 214 for more information.</p>	<p>LOCALPKI.BINDINFO.LDAP1</p> <p>Note: If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for each server.</p>

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 24 on page 76:

```
NumServers=1
```

2. Optionally change 5m in the following line to the posting interval in Table 24 on page 76:

```
PostInterval=5m
```

3. If necessary, update the BindProfile1 line or the Server1, AuthName1, and AuthPwd1 lines:

- If you intend to use encrypted passwords for your LDAP servers and you are migrating from Release 3:

- If you intend to use an LDAPBIND class profile, perform the following steps:

- a. Copy the following lines in the sample configuration file (/usr/lpp/pkiserv/samples/pkiserv.conf) into your pkiserv.conf configuration file (at the bottom of the LDAP section):

```
# 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
```

- b. Remove the comment delimiter (#) from the start of the fourth line and change LOCALPKI.BINDINFO.LDAP1 to the name of the LDAPBIND class profile. (See Step 3 on page 214).

- c. Delete the following three lines in the LDAP section:

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

Tailoring the PKI Services configuration file for LDAP

- If you intend to use the FACILITY CLASS profile IRR.PROXY.DEFAULTS, delete the following three lines in the LDAP section:

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

- 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 214).

```
# 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 24 on page 76:

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

- b. Change *CN=root* in the following line to the value of the administrator distinguished name in Table 24 on page 76:

```
AuthName1=CN=root
```

- c. Change *root* in the following line to the value of the administrator password in Table 24 on page 76:

```
AuthPwd1=root
```

- If you are not using encrypted passwords for your LDAP servers and you are migrating from Release 3, you do not need to do anything.

-
4. If the value of NumServers= is greater than 1, repeat Step 3 on page 79 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 24 on page 76:

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

-
6. If necessary, change 'T' in the following line to the RetryMissingSuffix value in Table 24 on page 76:

```
RetryMissingSuffix=T
```

Chapter 10. Creating VSAM data sets

You need to perform the tasks this chapter describes only if you meet one of the following conditions:

- You are configuring PKI Services for the first time
- Your company is using a sysplex for PKI Services daemons.

The MVS programmer performs the following tasks:

- If using a sysplex, perform the preliminary steps for establishing VSAM RLS
- If configuring PKI Services for the first time, create the VSAM object store and ICL data sets and indexes
- If using a sysplex and migrating from z/OS Version 1 Release 3, enable existing PKI Services VSAM data sets for VSAM RLS.

Note: If you are migrating from z/OS Version 1 Release 3 and do not want sysplex support, you do not need to perform any of the tasks in this chapter.

Space considerations for creating VSAM data sets

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 IKYCVSAM sample JCL contains default values for the primary and secondary extent allocations. Both are 50 (RECORDS(50 50)). You need to update these values based on your anticipated future needs. Use the following guidelines to update the RECORDS parameter for the DEFINE CLUSTER statements. Keep in mind that IDCAMS allocates extents on a track basis. (For more information about IDCAMS, see *z/OS DFSMS Access Method Services for Catalogs*.) After determining the size of the extent desired, IDCAMS rounds up to the next whole track. This may increase the actual size of the extents allocated.

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. The ICL grows over time as more certificates are issued. Assuming average size certificates, one ICL record will occupy 1024 bytes of storage. However, IDCAMS allocates based on the record's maximum size, 32756. Therefore, one allocation record will hold 31 certificates ($32756 / 1024 = 31$). If you use the default RECORDS(50 50), each extent will hold 50 allocation records or 1599 certificates ($32756 \times 50 / 1024 = 1599$). Because VSAM data sets can have up to 128 extents, the total number of certificates that can be stored using RECORDS(50 50) is 204672.

Summary of storage considerations for ICL

1 RECORD = 31 certificates

1 50 RECORD EXTEND = 1599 certificates

Entire data set using RECORDS(50 50) = 204672 certificates

If your anticipated needs differ greatly from the above values, you need to adjust the RECORDS parameter on the DEFINE CLUSTER statement for the ICL. (This is the second DEFINE CLUSTER statement in IKYCVSAM. See "IKYCVSAM" on page 330 for a code sample of this file.)

Creating VSAM data sets

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 spike in certificate request activity. Assuming average size certificate requests, one object store record and its companion posting record will occupy a total of 2560 bytes of storage. IDCAMS allocates based on the maximum size, 32756. Therefore, one allocation record will hold 12 concurrent certificate requests ($32756 / 2560 = 12$). If you use the default RECORDS(50 50), each extent will hold 50 allocation records or 639 concurrent certificate requests ($32756 \times 50 / 2560 = 639$). Because VSAM data sets can have up to 128 extents, the total number of concurrent certificate requests that can be stored using RECORDS(50 50) is 81792.

Summary of storage considerations for the object store

1 RECORD = 12 concurrent certificate requests

1 50 RECORD EXTEND = 639 concurrent certificate requests

Entire data set using RECORDS(50 50) = 81792 certificates

If your anticipated needs differ greatly from the preceding values, you need to adjust the RECORDS parameter on the DEFINE CLUSTER statement for the object store. (This is the first DEFINE CLUSTER statement in IKYCVSAM. See “IKYCVSAM” on page 330 for a code sample of this file.)

(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 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*.

Perform the following steps to establish VSAM RLS. For specific information on how to perform these steps, see the chapter about administering VSAM Record-Level Sharing in *z/OS DFSMSdfp Storage Administration Reference*.

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 RLS.

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

Table 25. VSAM RLS information you need to record

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

See “(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS” on page 84 for additional information about setting up VSAM datasets to run PKI Services in a sysplex.

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.

Before you begin: If you also want run multiple instances of PKI Services in a parallel sysplex (one per image), you need to have performed the steps described in “(Optional) preliminary steps for establishing VSAM RLS” on page 82.

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

1. Copy the sample JCL in 'SYS1.SAMPLIB(IKYCVSAM)' to your JCL data set. (See “IKYCVSAM” on page 330 for a code sample of this file.)
2. Update your data set as directed in the instructions in the prolog of the sample JCL:
 - a. Change the JOB card.
 - b. Change the VOL statements.
 - If you are running multiple instances of PKI Services in a parallel sysplex, replace the VOL statements with STORCLAS statements that specify the storage class recorded in Table 25, for example:
 STORCLAS(VSAMRLS)
 - If you are running without a parallel sysplex, replace the vvvvvv in the VOL statements with a VOL=SER suitable for your VSAM data sets.
 - c. If you are running multiple instances of PKI Services in a parallel sysplex, remove the SPANNED and CISIZE statements in the file. These lines follow:
 SPANNED -
 CISZ(512)-
 - 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 1b on page 64.)
 - e. Update the primary and secondary extent allocations (both are 50 by default) based on your anticipated future needs. (See “Space considerations for creating VSAM data sets” on page 81 for guidelines on determining the space you will need.) The following line shows these allocations:
 RECORDS(50 50)

Creating VSAM data sets

- f. Do **not** change any numeric values, other than the primary and secondary values specified for RECORDS.

-
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 20 on page 58). The MVS programmer enables the sysplex to access the VSAM datasets.

Before you begin: You need to have performed the steps described in “(Optional) preliminary steps for establishing VSAM RLS” on page 82.

Perform the following steps to enable your existing PKI Services data sets for VSAM RLS:

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

Attention: Do **not** copy IKYCVSAM by mistake as 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.
 - 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 58.

- e. Update the primary and secondary extent allocations (both are 50 by default) based on your anticipated future needs. (See “Space considerations for creating VSAM data sets” on page 81 for guidelines on determining the space you need.)

The following line shows these allocations:

```
RECORDS(50 50) -
```

- f. Do **not** change any numeric values, other than the primary and secondary values specified for RECORDS.

3. Submit the job when your changes are complete.

Chapter 11. 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 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

Before you begin:

- Your z/OS HTTP Server should be SSL-enabled (see Chapter 8, “Updating z/OS HTTP Server configuration and starting the server” on page 71) 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 that follows. The default is `/etc/pkiserv/`. The MVS programmer was asked to record any changes to the default; see Table 4 on page 9.

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 (recommended), 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 2 on page 87) and then restart PKI Services.

3. Start the PKI Services daemon from the MVS console by entering the following command:

```
S PKISERVD
```

Notes:

- a. If you are migrating from Release 3 and adding parallel sysplex support (you want to start multiple instances of the PKI Services daemon on different images in the sysplex), the first time you start the daemon, you must ensure that the system you start first is the one that has access to the

Starting and stopping PKI Services

old work files (by default, /var/pkiserv). (These are on the same system you were using for PKI Services before adding sysplex support).

- b. 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).
- c. 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 3 on page 55), 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,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 12 on page 39.

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"), undo the change in Step 2 on page 85 (see Step 2 on page 87 for steps on undoing the change), and then restart PKI Services.

Stopping the PKI Services daemon

Perform the following steps to stop the PKI Services daemon:

1. To stop the PKI Services daemon, enter one of the following two commands
You can use either the following MODIFY (or F) console command:

```
F PKISERVD,STOP
```

or the STOP (P) command:

```
P PKISERVD
```

- | 2. If you changed the PKI Services configuration file (as recommended previously
| — see Step 2 on page 85), have the UNIX programmer undo that change now
| by performing the following steps:
 - | a. Edit the PKI Services configuration file (by default /etc/pkiserv/pkiserv.conf).
 - | b. Set NumServers= in the LDAP section of the file to the correct number of
| LDAP servers.
 - | c. Exit to save your changes.

Starting and stopping PKI Services

Part 3. Customizing PKI Services

This part includes the following:

- Chapter 12, “Customizing the end-user Web application” on page 91 provides an overview of the pkiserv.tpl file, which contains the certificate templates, and explains customizing the end-user Web pages.
- Chapter 13, “Customizing the administration Web pages” on page 133 provides an overview of the CGI scripts and explains how to customize the administration Web pages.
- Chapter 14, “Advanced customization” on page 139 explains:
 - Using certificate policies
 - Updating the signature algorithm
 - Using the PKI exit.

Chapter 12. Customizing the end-user Web application

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 chapter describes the `pkiserv.tmpl` certificate templates file and explains how to use it to customize the end-user Web pages. (See Chapter 24, “The `pkiserv.tmpl` certificate templates file” on page 263 for a code sample of the templates file.) This chapter also explains the relationship between CGIs and the certificate templates file. Finally, this chapter 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 26:

Table 26. `pkiserv.tmpl` — Structure and main divisions

A prolog section of comments explaining main sections, subsections, named fields, and substitution variables. (To examine these comments, see Chapter 24, “The `pkiserv.tmpl` certificate templates file” on page 263.)

APPLICATION section

The APPLICATION section contains subsections, which produce certain Web pages, such as the PKI Services Home page (see Figure 7 on page 158). For details, see “The APPLICATION section” on page 97.

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 100.

INSERT sections

These contain HTML for certain Web pages (for example, the “Request submitted successfully” Web page) and certificate field dialogs (for example, text entry boxes (the common name INSERT produces a text box where the user enters this information) and drop-downs). For details, see Figure 10 on page 165.

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 section 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.

Customizing the end-user Web pages

You use square brackets to delineate a substitution variable.

Example:

```
[base64cert]
```

Notes:

1. Substitution variables are case-sensitive.
2. Depending on the section where a substitution variable is present, it may not have a valid meaning. For example, the base64cert substitution variable is meaningless before the certificate is retrieved. Therefore, in this case, the value of [base64cert] would be the null string (an empty string).

The following table summarizes valid substitution variables:

Table 27. Substitution variables

Substitution variable	Description
base64cert	The requested certificate, base64-encoded.
browsertype	A special substitution variable to qualify named fields only. It enables the different browsers, Netscape and Internet Explorer, to perform browser-specific operations, such as generating a public and private key pair. To do this, Netscape uses a KEYGEN HTML tag while Internet Explorer uses ActiveX controls. For example, suppose you specify %%PublicKey[browsertype]%% in a TEMPLATE CONTENT section. If the user referencing this section uses the Netscape Navigator browser, then INSERT PublicKeyNS is included. If the user's browser is Microsoft Internet Explorer, INSERT PublicKeyIE is included.
iecert	The requested certificate in a form that Microsoft Internet Explorer accepts.
optfield	A special substitution variable that should be placed in any certificate field name INSERT where the end user can supply the value. It makes the input field optional.
printablecert	This contains the certificate details so that the end user can confirm that the certificate is the correct one to renew or revoke. The displayed data is extracted from the ICL entry.
tmplname	A certificate template name. This is primed from the HTML tag <SELECT NAME="Template"> in the <APPLICATION NAME=PKISERV> section. The end user selects it on the first Web page.
transactionid	A unique value returned from a certificate request.

What are named fields?

Named fields insert common HTML code, such as a common input field or a page header or footer, in a Web page. (Each named field refers to a corresponding INSERT section.) A named field is delineated with %%.

Examples:

```
%%Country%%  
%%-pagefooter%%
```

Note: Named fields are case-sensitive.

Customizing the end-user Web pages

A named field can include or not include a dash. A named field without a dash, such as `%%Label%%` may 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 114 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 100) 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. As previously indicated, any named field used in the `pkiserv.tmpl` file must be defined in a corresponding `INSERT` section.

Unlike the `APPLICATION` section and `TEMPLATE` sections, `INSERT` sections can have no subsections. The following is the format of an `INSERT` section:

```
<INSERT NAME=insert-name>...</INSERT >
```

An `INSERT` contains HTML that either:

- defines a certificate field
- defines other common HTML that can be referenced in other sections.

The following example of an `INSERT` defines a certificate field.

Example:

```
<INSERT NAME=Country>
<p> Country [optfield] <BR>
<INPUT NAME="Country" TYPE="text" SIZE=2 maxLength="2">
</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:

- Sample `INSERT`s, which are includable code inserts (This is common HTML for Web page content as listed in Table 28 on page 94)
- Certificate fields that are defined in `INSERT` sections. (See Table 29 on page 94.) These include:
 - X.509 fields (for example, `OrgUnit`)
 - non-X.509 fields (for example `Userld`).

Customizing the end-user Web pages

Table 28. Sample INSERTs

INSERT NAME	Contents
-AdditionalHeadIE	ActiveX controls to enable Internet Explorer to generate a key pair
-requestok	HTML for the Web page "Request submitted successfully" after a successful certificate request (for both original requests and renewals). (For a sample of this Web page, see Figure 10 on page 165.)
-requestbad	HTML for the Web page that says, "Request was not successful"
-renewrevokeok	HTML for the Web page that says, "Request submitted successfully" after a successful attempt to revoke a certificate (see Figure 15 on page 171 for a sample of the Web page to renew or revoke a certificate).
-renewrevokebad	HTML for the Web page that says, "Request was not successful" after an unsuccessful attempt to renew or revoke a certificate (see Figure 15 on page 171 for a sample of the Web page to renew or revoke a certificate).
-return10cert	This returns a #10 certificate.
returnbrowsercertNS	This contains [base64cert], which is the base64 substitution variable.
returnbrowsercertIE	This contains a script for producing a popup window installing your certificate (if you are using the Microsoft Internet Explorer browser). See Figure 12 on page 167 for a sample of this Web page.

Named fields in INSERT sections

Most of the following fields are X.509 fields. The following table summarizes the named fields in INSERT sections:

Table 29. Named fields in INSERT sections

Field	Description
AltDomain	The host name of the machine where a certificate will be installed. This is a text field of up to 100 characters. Note: The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
AltEmail	The user's e-mail address, including the @ character and any periods (.). This is a text field of up to 100 characters. Note: The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
AltIPAddr	The unique IP version 4 address that specifies the location of the server or device on the Internet, for example, 9.67.97.103. (PKI Services supports only IP version 4 addresses.) The IP address is in dotted decimal format and is a text field of up to 15 characters. Note: The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
AltURI	A name or address referring to an Internet resource; a URL is one kind of uniform resource identifier. This is a text field of up to 100 characters. Note: The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
ChallengePassPhrase	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.

Table 29. Named fields in INSERT sections (continued)

Field	Description
CommonName	<p>For browser certificates, this is your name, such as John Smith. (You can use your first and last name, in that order.) For server certificates, this is name by which the server's administrator wants it to be known. For SSL servers, the SSL protocol requires the CommonName to be the fully qualified domain name of the server, for example, www.ibm.com. CommonName is a text field of up to 64 characters. See the Note 1 on page 107 for more information about this field.</p> <p>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>
Country	<p>The country where your organization is located. This is a 2-character text field.</p> <p>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>
Email	<p>The e-mail address for the distinguished name. This is a text field of up to 64 characters.</p> <p>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>
HostIdMap	<p>This is the user ID for authorization purposes, in an e-mail type of format: subject-id@host-name</p> <p>For example, this could be dsmit@ibm.com. This is a text field of up to 100 characters.</p> <p>There are three ways to use %%HostIdMap%%:</p> <ul style="list-style-type: none"> • If you place it in the CONTENT section, the end user can specify the value (or values since it may be repeated). • You can also place it in the APPL section that the application provides. If you do so, it should have the following form: %%HostIdMap=@host-name% <p>The host-name is the hardcoded system name for the current system.</p> <p>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.</p> <ul style="list-style-type: none"> • A third way to specify HostIdMap is to place %%HostIdMap%% in the ADMINAPPROVE section. This allows the administrator to fill in the value when approving the certificate request. See "Administering HostIdMappings extensions" on page 201 for more information.
KeyProt	<p>(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.</p>
KeyUsage	<p>The intended purpose of the certificate. Possible values are:</p> <ul style="list-style-type: none"> • handshake — Protocol handshaking (for example, SSL) • dataencrypt — Data encryption • certs sign — Certificate signing • docsign — Document signing
Label	<p>The label assigned to the requested certificate. This is a text field of up to 32 characters.</p>
Locality	<p>The city or municipality where your organization is located, such as Pittsburgh or Paris. This is a text field of up to 64 characters.</p> <p>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>

Customizing the end-user Web pages

Table 29. Named fields in INSERT sections (continued)

Field	Description
NotBefore	Number of days (0 or 30) before the certificate becomes valid.
NotAfter	Length of time that the certificate is current. This is 365 days (1 year) or 720 days (2 years).
NotifyEmail	The e-mail address for notification purposes. This is a text field of up to 64 characters. Note: When a certificate is created and posted to LDAP, the NotifyEmail value, if specified, is posted as the MAIL attribute. If the MAIL attribute already exists in that directory entry, its value is replaced by the new value. If both NotifyEmail and Email appear on one request, they must have the same value.
Org	Organization. The legally registered name (or trademark name, for example, IBM) of your organization. This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
OrgUnit	The name of your division or department. This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
OrgUnit2	The name of your division or department. (There can be more than one organizational unit field on a request form. For example, one could be for your department and another for your division.) This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
PassPhrase	The user decides this and enters and then reenters it when requesting a certificate (and must later supply this value when retrieving the certificate). This is a case-sensitive text field of up to 32 characters. There is no minimum number of characters, and the user can use any characters, but alphanumeric characters (A–Z, a–z, and 0–9) are recommended.
PostalCode	The zipcode or postal code. This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
PublicKey	The base64-encoded #10 certificate request. (This is for server or device enrollment only.) You create a certificate request on behalf of another server (which could be a z/OS server or other type of server) or device for which you are requesting a certificate. You use software specific to that server to generate the #10 request before going to the PKI Services Web site. Save the request in a file. Then open the file in a text editor such as Windows Notepad and copy the and paste the contents into the text box on the enrollment form. A text area of 70 columns and 12 rows is allocated for this certificate request. Here is an example of the certificate request: <pre> MIIBiDCB8gIBADAZMRcwFQYDVQQDEw5Kb2huIFEuIFB1YmxpYzCBnzANBghkqhkiG 9w0BAQEFAAOBjQAwgYkCgYEAAsCT1cJHAGPqi60jAyl+xNbt8z5ngmvq02V003oYu /mEnQtRM96e+2jbmDCRo5tWVklG40Yf9ZVB5biURMJFLztfa4AVdEVtun8DH2pwc wiNIZZcC1Zym5adurUmyDk64Pgi iIPMQS/t0ttG4c5U8uWSK0b1J4V4f7ps+t1aG t+cCAwEAAaAwMC4GCSqGS1b3DQEDjEhMB8wHQYDVR00BBYEFAlKTovBBvnFqDA0 1oIhtRinwRC9MA0GCSqGS1b3DQEBBQUAA4GBAIBCVpwYvppIX3HHmpKZPNY8Snsz AJrDsgAEH51W0IRGywhqKcLLxa9htoQai6cdc8RpFVTwk6UfdCOGxMn4aFb34Tk3 5WYdz0iHXg8MhHiB3EruwdWs+S7Fv3JhU3FLwU6lFLFAjbi+35iEWQymOR6mE5W CathprmGfKRrSDE5E </pre>
PublicKeyIE	(This is for the Internet Explorer browser only.) This is the cryptographic service provider. The user selects a value from a drop-down list (Microsoft Base Cryptographic Provider or Microsoft Enhanced Cryptographic Provider).
PublicKeyNS	(This is for the Netscape browser only.) This is the key size for your public/private key pair. The user selects a value from the drop-down list. Larger keys are more secure, but they also increase the time needed for connecting to a secure session.

Table 29. Named fields in INSERT sections (continued)

Field	Description
Requestor	The user's name, used for tracking the request. This can be in any format, for example, John Smith or John. J. Smith. (This can differ from the common name, especially if the request is for a server certificate.) The value is saved with the request and issued certificate, but it is not a field in the created certificate. The default value is taken from the leftmost RDN in the subject's distinguished name, truncated to 32 characters.
SignWith	<p>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.</p> <p>Examples:</p> <p>"SAF:CERTAUTH/Local CA Cert"</p> <p>"PKI:"</p> <p>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'.</p> <p>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.</p>
StateProv	<p>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.</p> <p>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>
Street	<p>The street address. This is a text field of up to 64 characters.</p> <p>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>
Title	<p>Job title. This is a text field of up to 64 characters.</p> <p>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>
TransactionId	PKISERV Web pages assign this after the user requests a certificate. When it is displayed, the user needs to record this number. This is a text field of up to 56 characters.
UserId	The owning SAF user ID. This is a text field of up to 8 characters.

The APPLICATION section

The APPLICATION section identifies the applications that will use PKI Services. The following is the format of the APPLICATION section:

<APPLICATION NAME=appl-name>...</APPLICATION>

The product ships with one application, PKISERV. Therefore, the pkiserv.tmpl certificate templates file that ships with PKI Services contains the following line:

Example:

```
<APPLICATION NAME="PKISERV">
```

The APPLICATION section can contain the following subsections:

- CONTENT
- RECONTENT

Customizing the end-user Web pages

- RESUCCESSCONTENT
- REFAILURECONTENT
- ADMINHEADER
- ADMINFOOTER

<CONTENT> ...</CONTENT>

This subsection contains the HTML to display the PKI Services home Web page to the end user who is requesting and retrieving certificates. (See Figure 7 on page 158 for a sample Web page.) This subsection should contain one or more named fields (see “What are named fields?” on page 92) 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.

<RECONTENT> ...</RECONTENT>

This subsection contains the HTML to display information about the certificate so that the end user can confirm that this is the correct certificate to renew or revoke. (See Figure 15 on page 171 for a sample Web page.) This subsection uses the substitution variable [printablecert], which contains the data extracted from the ICL entry. (See “What are substitution variables?” on page 91.)

<RESUCCESSCONTENT> ...</RESUCCESSCONTENT>

This subsection contains the HTML to display a Web page to the end user when the revocation request is successful. Any named fields in this subsection are interpreted as HTML content inserts (for example, a page footer) that INSERT sections define. For PKISERV, the INSERT sections are included as part of the HTML for the Web page displayed to the end user.

<REFAILURECONTENT> ...</REFAILURECONTENT>

This subsection contains the HTML to display a Web page to the end user when renewal or revocation request is unsuccessful. Any named fields in this subsection are interpreted as content inserts (for example, a page footer) that INSERT sections define. For PKISERV, the INSERT sections are included as part of the HTML for the Web page displayed to the end user.

<ADMINHEADER>...</ADMINHEADER>

This subsection contains the general installation-specific HTML content for the header of all administration Web pages. See “Steps for customizing the administration Web pages” on page 135 for more information.

<ADMINFOOTER>...</ADMINFOOTER>

This subsection contains the general installation-specific HTML content for the footer of all administration Web pages. See “Steps for customizing the administration Web pages” on page 135 for more information.

The following table summarizes the contents (Web pages) that the subsections of the APPLICATION section generate.

Table 30. Subsections of the APPLICATION section

Section or subsection	Contents
CONTENT	HTML for the Web page "PKISERV certificate generation application." (For a sample of this Web page, see Figure 7 on page 158.)
RECONTENT	HTML for the Web page "Renew or revoke a browser certificate." (For a sample of this Web page, see Figure 15 on page 171.)
RESUCCESSCONTENT	Contains only the named field %%-renewrevokeok%% (whose associated INSERT contains HTML for the Web page "Request submitted successfully").
REFAILURECONTENT	Contains only the named field %%-renewrevokebad%% (whose associated INSERT contains HTML for the Web page "Request was not successful").
ADMINHEADER	This is for an administration page; see "Customizing the administration Web pages" on page 134 for more information.
ADMINFOOTER	This is for an administration page; see "Customizing the administration Web pages" on page 134 for more information.

Templates that PKI Services provides

PKI Services provides the templates to request the following certificates:

- One-year SAF server certificate
- One-year SAF browser certificate
- One-year PKI SSL browser certificate (See Figure 9 on page 164 to see a sample of this Web page.)
- One-year PKI SSL S/MIME browser certificate
- Two-year PKI browser certificate for authenticating to z/OS
- Five-year PKI SSL server certificate
- Five-year PKI IPSEC server (firewall) certificate
- Five-year PKI intermediate CA certificate

The following table describes the certificate templates that PKI Services provides:

Table 31. Certificate templates PKI Services provides

Certificate template	Description
One-year SAF server certificate	The template allows end users to request certificates for servers, using native SAF certificate generation facilities (rather than PKI Services certificate generation facilities). The certificate is used for handshaking only (for example, SSL). This certificate is auto-approved.
One-year SAF browser certificate	This template is for requesting 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 29 on page 94 for descriptions of fields) because the certificate is stored in a RACF database. This certificate is auto-approved.

Customizing the end-user Web pages

Table 31. Certificate templates PKI Services provides (continued)

Certificate template	Description
One-year PKI SSL browser certificate	A template for requesting a browser certificate that PKI Services generates. The end user enters the common name. (See Table 29 on page 94 for descriptions of fields.) This template contains an ADMINAPPROVE section. Therefore, certificates requested using this template require administrator approval before being issued. The user ID and password are not required but the passphrase is required.
One-year PKI S/MIME browser certificate	A template for requesting a browser certificate that PKI Services generates. This is similar to the one-year PKI SSL browser certificate except the end user selects AltEmail.
Two-year PKI browser certificate for authenticating to z/OS	<p>A template for requesting 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.</p> <p>%%HostIdMap%% is intended as a replacement for adding (and mapping) the certificate to a RACF user ID.</p> <p>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 29 on page 94.)</p>
Five-year PKI SSL server certificate	A template for requesting 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	A template for requesting a server certificate that PKI Services generates. This is similar to the five-year PKI SSL server certificate except that keyusages of handshake and dataencrypt are hardcoded. Also, the end user selects AltEmail, AltIPAddr, AltURI, and AltDomain.
Five-year PKI intermediate CA certificate	A template for requesting 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.

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 eight TEMPLATE sections, for the eight certificates the preceding section describes.

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:

Customizing the end-user Web pages

```
<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 32. Names of certificate templates

True name	Alias	Nickname
1 Year PKI SSL Browser Certificate	PKI Browser Certificate	1YBSSL
1 Year PKI S/MIME Browser Certificate	PKI Browser Certificate	1YBSM
2 Year PKI Browser Certificate For Authenticating To z/OS	PKI Browser Certificate	2YBZOS
5 Year PKI SSL Server Certificate	PKI Server Certificate	5YSSSL
5 Year PKI IPSEC Server (Firewall) Certificate	PKI Server Certificate	5YSIPS
5 Year PKI Intermediate CA Certificate	PKI Server Certificate	5YSCA
1 Year SAF Server Certificate	SAF Server Certificate	none
1 Year SAF Browser Certificate	SAF Browser Certificate	none

TEMPLATE sections can have the following subsections:

- CONTENT
- APPL
- CONSTANT
- ADMINAPPROVE
- SUCCESSCONTENT
- FAILURECONTENT
- RETRIEVECONTENT
- RETURNCERT

<CONTENT>...</CONTENT>

This subsection contains the HTML to display a Web page to the end user requesting a certificate of a specific type. (See Figure 9 on page 164 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

Customizing the end-user Web pages

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>...</APPL>

This subsection identifies certificate fields for which the application itself should provide values. This subsection should contain only named fields, one per line. The only supported named fields allowed in this section are:

- UserId
- HostIdMap

Example:

```
<APPL>
%%UserId%%
%%HostIdMap=@www.ibm.com%%
<APPL>
```

<CONSTANT>...</CONSTANT>

This subsection identifies certificate fields that have a constant (hardcoded) value for everyone. This subsection should contain only named fields, one per line. The syntax for specifying the values is `%%field-name=field-value%%`:

Example:

```
%%KeyUsage=handshake%%
```

<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 may 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

Customizing the end-user Web pages

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 certificate

You can put the following fields in the ADMINAPPROVE section:

- AltDomain
- AltEmail
- AltIPAddr
- AltURI
- CommonName
- Country
- EndDate
- HostIdMap (can repeat)
- KeyUsage
- Locality
- Org
- OrgUnit (can repeat)
- StartDate
- StateProv
- Title

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 29 on page 94.)

Example:

```
<ADMINAPPROVE>
%%KeyUsage%%
%%CommonName%%
%%OrgUnit%%
%%Org%%
%%Country%%
```

Customizing the end-user Web pages

```
%%HostIdMap%%  
%%HostIdMap%%  
%%HostIdMap%%  
%%HostIdMap%%  
<ADMINAPPROVE>
```

<SUCCESSCONTENT>...</SUCCESSCONTENT>

This subsection contains the HTML to display to the end user a Web page saying that the certificate request was submitted successfully. Any named fields in this subsection are interpreted as content inserts defined by INSERT sections. For PKISERV, the INSERT sections are included as part of the HTML to display a Web page to the end user.

In all of the templates included with PKI Services, <SUCCESSCONTENT> contains only the named field %%-requestok%%. (See “What are named fields?” on page 92 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 10 on page 165.)

<FAILURECONTENT>...</FAILURECONTENT>

This subsection contains the HTML to display to the end user a Web page saying the certificate request was not submitted successfully. Any named fields in this subsection are interpreted as content inserts defined by INSERT sections. For PKISERV, the INSERT sections are included as part of the HTML to display a Web page to the end user.

In all of the templates included with PKI Services, <FAILURECONTENT> contains only the named field %%-requestbad%%. (See “What are named fields?” on page 92 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 11 on page 166. You may 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 92 for an explanation of named fields.)
- The title of the Web page (This appears in the banner of your browser. Figure 11 on page 166 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

Customizing the end-user Web pages

- A heading that says "Retrieve Your (name of certificate)." This uses the substitution variable [tplname]. (See "What are substitution variables?" on page 91 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 to the end user a Web page upon successful certificate retrieval. For PKISERV, if the certificate being retrieved is a browser certificate, then this section must contain a single line containing a browser qualified INSERT name.

Example:

```
%%returnbrowsercert[browsertype]%%
```

Additionally, INSERTs for Netscape (returnbrowsercertNS) and Internet Explorer (returnbrowsercertIE) containing browser-specific HTML for returning certificates must be defined elsewhere in the pkiserv.tpl certificates template file. If the certificate being retrieved is a server certificate, this section should contain the HTML necessary to present the certificate to the user as text.

Summary of subsections contained in certificate templates

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

Table 33. Summary of subsections in certificate templates

Subsection (in TEMPLATE section)	One-year SAF browser	One-year SAF server	One-year PKI SSL browser	One-year PKI SSL S/MIME browser	Two-year PKI browser cert. for authen. to z/OS	Five-year PKI SSL server	Five-year PKI IPSEC server (firewall)	Five-year PKI int. CA
CONTENT	X	X	X	X	X	X	X	X
APPL	X	X			X			X
CONSTANT	X	X	X	X	X	X	X	X
ADMINAPPROVE			X	X		X	X	
SUCCESSCONTENT	X	X	X	X	X	X	X	X
FAILURECONTENT	X	X	X	X	X	X	X	X
RETRIEVECONTENT	X	X	X	X	X	X	X	X
RETURNCERT	X	X	X	X	X	X	X	X

Summary of fields in certificate templates

The following table summarizes fields in the various certificate templates. Fields can be:

- Required
- Optional
- Provided by the application
- Constant (values specified)

Table 34. Summary of fields in certificate templates that PKI Services provides

Field name	Template							
	One-year PKI SSL browser cert.	One-year PKI S/MIME browser cert.	One-year SAF server cert.	One-year SAF browser cert.	Two-year PKI browser cert. for auth to z/OS	Five-year PKI SSL server cert.	Five-year PKI IPSEC server (firewall) cert.	Five-year PKI int. CA cert.
AltDomain			Optional			Optional		
AltEmail		Required	Optional			Optional		
AltIPAddr			Optional			Optional		
AltURI			Optional			Optional		
ChallengePassPhrase	Optional				Optional			
CommonName	Required		Optional	Constant ¹		Optional		
Country			Required	Constant - US		Optional		
Email	Optional					Optional		
HostIdMap ²					App. provides			
KeyUsage	Constant - handshake					Constant ³	Constant - certsign	
Label			Required					
Locality			Optional			Optional		
NotBefore	Constant - 0				Constant - 0			
NotAfter	Constant - 365				Constant - 730	Constant - 1825		
NotifyEmail	Optional				Optional	Optional		
Org	Constant - The Firm		Required	Constant - The Firm		Optional		
OrgUnit	Constant - Class 1 Internet Certificate CA		Required	Constant ⁴	Constant - Class 1 Internet Certificate CA	Optional		
OrgUnit2			Optional			Optional		
PassPhrase	Required				Required			
PostalCode						Optional		
PublicKey ⁵	Required							
Requestor	Optional				Optional			

Table 34. Summary of fields in certificate templates that PKI Services provides (continued)

	Template			
SignWith	Constant - PKI:	Constant - SAF:CERAUTH/ taca	Constant - PKI:	
StateProv		Optional		Optional
Street				Optional
Title	(Not included in any template)			
TransactionId	Required			
UserId		Application provides		App. provides

Notes:

- 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.
 - HostIdMap is formed by concatenating UserId with @host-name.
 - You can have more than one KeyUsage. The template contains two:
KeyUsage=handshake
KeyUsage=dataencrypt
 - You can have more than one OrgUnit. The template contains two:
OrgUnit=SAF template certificate
OrgUnit=Nuts and Bolts Division
 - The browser provides the PublicKey.
For the one-year SAF browser certificate, PublicKey is coded with the substitution variable browsertype. For Internet Explorer, this generates two fields:
 - CSP — the cryptographic service provider. (Defaults to Microsoft Enhanced Cryptographic Provider)
 - KeyProt — Enable strong private key protection. Defaults to No.
 For Netscape, this generates one field:
 - PublicKeyNS — key size. (Defaults to high grade.)
- In any of the server certificate templates, the PublicKey is the #10 request.

Examining the pkiserv.tmpl file

After the initial section of comments, the next section of the pkiserv.tmpl file is the APPLICATION section. The following example shows the APPLICATION section. (The vertical ellipses indicate omitted sections.)

```

<APPLICATION NAME=PKISERV> 1
<CONTENT> 2
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Application </TITLE>
%%-copyright%%
</HEAD>
<BODY>
<H1>PKISERV Certificate Generation Application</H1>
<p>
<A HREF="/PKIServ/cacerts/cacert.der">Install 3
  our CA certificate into your browser </A>
<H2>Choose one of the following:</H2>
<ul>
<li><h3>Request a new certificate using a model</h3>
<FORM name=mainform METHOD=GET ACTION="/PKIServ/ssl-cgi/catmpl.rexx"> 4
<p> Select the certificate template to use as a model

```

Customizing the end-user Web pages

```
<SELECT NAME="Template"> 5
  %%1 Year PKI SSL Browser Certificate%%
  <OPTION>1 Year PKI SSL Browser Certificate
  %%1 Year PKI S/MIME Browser Certificate%%
  <OPTION>1 Year PKI S/MIME Browser Certificate
  %%2 Year PKI Browser Certificate For Authenticating To z/OS%%
  .
  .
  .
</HTML>
</CONTENT>
<RECONTENT> 6
<HTML><HEAD>
<TITLE> PKISERV Renew or Revoke a Browser Certificate </TITLE>
%%-copyright%%
</HEAD>
<BODY>
<H1>Renew or Revoke a Browser Certificate</H1>
.
.
.
</BODY>
</HTML>
</RECONTENT>
<RESUCCESSCONTENT> 7
  %%-renewrevokeok%%
</RESUCCESSCONTENT>
<REFAILURECONTENT> 8
  %%-renewrevokebad%%
</REFAILURECONTENT>
<ADMINHEADER> 9
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Administration </TITLE> 10
%%-copyright%%
</HEAD>
<BODY>
</ADMINHEADER>
<ADMINFOOTER>
<p> %%-pagefooter%% 11
</BODY>
</HTML>
</ADMINFOOTER>
</APPLICATION>
```

The numbers in the following list refer to the highlighted items in the preceding example:

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 end user requests or retrieves a certificate. The <H1> indicates the main heading of that Web page, "Web Based Certificate Generation Application." (See Figure 7 on page 158 for a sample of that Web page.)
3. The HREF tag is the link to install the 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

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 15 on page 171 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 10 on page 165 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 ADMINHEADER subsection references the %%-copyright%% named field, which is defined in the INSERT section. This should contain the copyright statement for your company.
10. The title appears in the banner across the very top of the browser window.
11. 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.

The TEMPLATE sections follow the APPLICATION section. The following example shows a TEMPLATE section. (The vertical ellipses indicate omitted sections.)

```
# =====
#
# Template Name - 2 Year PKI Browser Certificate For Authenticating
#                 to z/OS 1
#
# Function - Creates a 2 year certificate good for authenticating to
#           z/OS....
#
#
# User input fields:
# Requestor - optional
# PassPhrase - required
# PublicKey - required (Provided by the browser itself)
# NotifyEmail - optional
#
# =====
#
<TEMPLATE NAME=2 Year PKI Browser Certificate For Authenticating To z/OS> 2
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=2YBZOS>
<CONTENT> 3
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE> 4
%%-copyright%% 5
%%-AdditionalHead[browsertype]%%
<SCRIPT LANGUAGE="JavaScript"> 6
<!--
:
:
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1>2 Year Browser Certificate For Authenticating To z/OS</H1> 7
<p>
```

Customizing the end-user Web pages

```
<H2>Choose one of the following:</H2>
:
:
#<FORM NAME="CertReq" METHOD=POST ACTION=
#      "/PKIServ/ssl-cgi-bin/careq.rexx" onSubmit= 8
#      "if(ValidateEntry()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<p> Enter values for the following field(s) 9
  %%Requestor (optional)%%
  %%NotifyEmail (optional)%%
  %%PassPhrase%%
  %%PublicKey2[browsertype]%%
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%% 10
</BODY>
</HTML>
</CONTENT>
<APPL> 11
  %%UserId%%
  %%HostIdMap=@host-name%%
</APPL>
<CONSTANT> 12
  %%NotBefore=0%%
  %%NotAfter=730%%
  %%KeyUsage=handshake%%
  %%OrgUnit=Class 1 Internet Certificate CA%%
  %%Org=The Firm%%
  %%SignWith=PKI:%%
  %%CommonName=%%
</CONSTANT>
<SUCCESSCONTENT> 13
  %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT> 14
  %%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT> 15
<HTML><HEAD>
  %%-copyright%%
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
:
:
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tplname]</H1> 16
<H3>Please bookmark this page</h3>
:
:
#<FORM NAME=retrieveform METHOD=POST ACTION=
#      "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit= 17
:
:
</FORM>
:
:
<p>%%-pagefooter%%
```

```

</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT> 18
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>

```

The numbers in the following list refer to the highlighted items in the preceding example:

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.
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. This JavaScript script provides the underlying logic for the text entry that the user must perform.
7. The heading is the main heading on the Web page for requesting the selected certificate.
8. The ACTION tag indicates that the CGI script that gets control when the user clicks the **Submit certificate request** button is careq.rexx.
9. Fields for which the user can supply input include %%Requestor%%, %%PassPhrase%%, 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.
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 later. See Item 1 on page 113.)
14. The FAILURECONTENT subsection contains the HTML to display when the certificate request is unsuccessful. This subsection contains the %%-requestbad%% named field. (This named field is defined in the INSERT section, shown later.)
15. The requestok INSERT (mentioned in Item 13) includes an ACTION that calls caretrieve.rexx, which displays the HTML in the RETRIEVECONTENT

Customizing the end-user Web pages

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 Item 15 on page 111 indicates.
18. The RETURNCERT subsection contains the `%%return10cert%%` named field, which is defined in an INSERT. (See Item 4 on page 114.)

The final section of the `pkiserv.tmpl` certificate templates file includes sample INSERTS. The following example shows sample INSERTS. (The vertical ellipses indicate omitted sections.)

```
# =====
#
# Sample INSERTS
#
# =====
#
<INSERT NAME=-AdditionalHeadIE>
<OBJECT
  classid="clsid:43F8F289-7A20-11D0-8F06-00C04FC295E1"
  CODEBASE="xenroll.cab"
  id="certmgr"
>
</OBJECT>
</INSERT>

<INSERT NAME=-requestok> 1
<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 METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx"> 2
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT NAME="TransactionId" TYPE="hidden" VALUE="[transactionid]">
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>

<INSERT NAME=-requestbad> 3
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Failure</TITLE>
</HEAD>
<BODY>
<H1> Request was not successful</H1>
<p> Please correct the problem or report the error to your Web admin
person<br>
<PRE>
[errorinfo]
</PRE>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
```

```

:
<INSERT NAME=-return10cert> 4
<HTML><HEAD>
<TITLE> Web Based SAF Certificate Generation Application Pg 4</TITLE>
</HEAD>
<BODY>
<H1> Here's Your Certificate. Cut and Paste it to a File</H1>
<TABLE BORDER><TR><TD>
<PRE>
[base64cert] 5
</PRE>
</TD></TR></TABLE>
<p>%-pagefooter%
</BODY>
</HTML>
</INSERT>
:
</BODY>
</HTML>
</INSERT>
#
# =====
#
# X.509 fields (INSERTs) valid for certificate requests
#
# =====
#
:
:
<INSERT NAME=PublicKeyIE> 6
<SCRIPT LANGUAGE="VBScript">
<!--
:
:
// -->
<
:
:
// -->
</SCRIPT>

# =====
:
:
<INSERT NAME=PassPhrase> 7
<p> Pass phrase for securing this request. You will need to supply
this value when retrieving your certificate [optfield] <BR>
<INPUT NAME="PassPhrase" TYPE="password" SIZE=32 maxlength="32"> <BR>
<p> Reenter your pass phrase to confirm <BR>
<INPUT NAME="ConfirmPassPhrase" TYPE="password" SIZE=32
maxlength="32">
</INSERT>
:
:
<INSERT NAME=-pagefooter>
<p>email: webmaster@your_company.com
</INSERT>

```

The numbers in the following list refer to the highlighted items in the preceding example:

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.

Customizing the end-user Web pages

4. (The caretrieve.rexx CGI displays the RETRIEVECONTENT subsection (see Item 15 on page 111) 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 Item 18 on page 112).
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 Item 9 on page 111.)

Relationship between CGIs and the pkiserv.tmpl file

CGIs are REXX 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/auth/careq.rexx
- /usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/caretrieve.rexx
- /usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx
- /usr/lpp/pkiserv/PKIServ/clientauth-cgi-bin/cadisplay.rexx
- /usr/lpp/pkiserv/PKIServ/clientauth-cgi-bin/camodify.rexx

The following table summarizes the actions the CGIs perform:

Table 35. CGI actions for end-user Web pages

REXX exec	Action	Sample associated Web page
camain.rexx	<ul style="list-style-type: none">• When user clicks the Request certificate button, this calls catmpl.rexx, passing it a parameter identifying the selected template.• The user can click the Pick up certificate button to go directly to caretrieve.rexx (if the certificate is already requested).• The user can click the Renew or revoke certificate button to go to cadisplay.rexx.• An administrator can click the Go to administration page button to go to admmain.rexx. (See Table 41 on page 133 for more information about admmain.rexx.)	See Figure 7 on page 158.

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

REXX exec	Action	Sample associated Web page
catmpl.rexx	<ul style="list-style-type: none"> • Displays Web page coded in the HTML under the CONTENT subsection (of a TEMPLATE section). • When the user clicks the Submit certificate request button, this passes template and field name parameters to careq.rexx. • When the user clicks the Retrieve your certificate button, this passes control to caretrieve.rexx. 	See Figure 9 on page 164.
careq.rexx	<ul style="list-style-type: none"> • Processes field names under the APPL subsection (of a TEMPLATE section). Note: Depending on the template, this can be: <ul style="list-style-type: none"> – UserId only – UserId and HostIdMap. • Processes hardcoded field names under the CONSTANT subsection (of a TEMPLATE section). • Depending on the results, displays Web page coded in the HTML under the SUCCESSCONTENT or FAILURECONTENT subsection (of a TEMPLATE section): <ul style="list-style-type: none"> – The SUCCESSCONTENT subsection includes a Continue button the user can click to continue to caretrieve.rexx. 	See Figure 10 on page 165.
caretrieve.rexx	<ul style="list-style-type: none"> • Displays Web page coded in the HTML under the RETRIEVECONTENT subsection (of a TEMPLATE section). This HTML prompts the user to enter the transaction ID and a password if the user entered one when requesting the certificate. • When the user clicks the Retrieve and install certificate button, this passes the transaction ID parameter to cagetcert.rexx. 	See Figure 11 on page 166.
cagetcert.rexx	<ul style="list-style-type: none"> • Displays Web page coded in the HTML under RETURNCERT subsection (of a TEMPLATE section). This HTML determines which of the following forms to use when returning the certificate: <ul style="list-style-type: none"> – as a base64-encoded certificate (for server certificates) – as an ActiveX object (for Microsoft Internet Explorer browser certificates) – as an application/x-x509-user-certificate MIME type (for Netscape browser certificates). 	See Figure 12 on page 167.
cadisplay.rexx	<ul style="list-style-type: none"> • Displays Web page coded in the HTML under the RECONTENT subsection (of the APPLICATION section). • For renewing a certificate, the user fills in the passphrase and clicks the Renew button. For revoking a certificate, the user clicks the Revoke button. Both actions call camodify.rexx. 	See Figure 15 on page 171.
camodify.rexx	<ul style="list-style-type: none"> • Displays Web page coded in the HTML under the SUCCESSCONTENT subsection (of a TEMPLATE section) for a successful renewal. The SUCCESSCONTENT subsection includes a Continue button the user can click to call caretrieve.rexx. • Displays the Web page coded in HTML under the RESUCCESSCONTENT subsection (of the APPLICATION section) for a successful revocation. 	See Figure 10 on page 165.

Steps for performing minimal customization

You need to perform these steps only if you are customizing certificate templates for the first time. If your company used an earlier release of PKI Services, you do **not** need to do so again.

Before you begin: Review the certificate templates and decide if there are any that you want to remove from the pkiserv.tpl certificates template file. If so, do this first. (To remove a certificate template, you can simply remove its name from the APPLICATION section.)

Perform the following steps to do the minimal updates on the remaining certificate templates:

Note: Fields such as `%%Org%%`, `%%Country%%`, and so forth 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 `slapd.conf` file).

1. For the SAF templates, update the following fields as needed:
 - a. If present, replace the OrgUnit values in the following lines with values more appropriate to your organization:

```
%%OrgUnit=Nuts and Bolts Division%%  
%%OrgUnit=SAF template certificate%%
```

- b. Replace `taca` in the following line with the correct label of the CERTAUTH signing certificate:

```
%%SignWith=SAF:CERTAUTH/taca%%
```

-
2. For the PKI templates, replace the OrgUnit value in the following line with a value more appropriate for your organization:

```
%%OrgUnit=Class 1 Internet Certificate CA%%
```

-
3. If present, replace `The Firm` with the name of your company in the following `%%Org` line:

```
%%Org=The Firm%%
```

-
4. If your company location is not the United States, update the following line by specifying the correct two-letter country abbreviation:

```
%%Country=US%%
```

-
5. If present, replace `host-name` with the domain name of this system in the following `%%HostIdMap` line:

```
%%HostIdMap=@host-name%%
```

You also need to follow the instructions in “Administering HostIdMappings extensions” on page 201.

-
6. For non-SAF certificates, you can notify users when certificate requests are rejected or when certificates are ready for retrieval or are expiring.

Customizing the end-user Web pages

- |
- |
- |
- a. If you do not want to have NotifyEmail appear as an input field for any non-SAF certificates, delete the NotifyEmail lines in the following locations in the TEMPLATE section for this certificate:

- In the header:
NotifyEmail - optional
- In the list of fields:
%%NotifyEmail (optional)%%

- b. If you do not want to have NotifyEmail appear as an input field for renewal of any non-SAF certificates, delete the following NotifyEmail line in the APPLICATION section and in the list of fields:

%%NotifyEmail (optional)%%

-
7. Insert the copyright statement for your company in the -copyright named field in the INSERT section.

-
8. Insert the e-mail address of your company's PKI Services administrator in the -pagefooter named field in the INSERT section.
-

Steps for additional first-time customization

|

|

|

You need to perform these steps only if you are customizing certificate templates for the first time. If your company used an earlier release of PKI Services, you do **not** need to perform these steps.

Perform the following steps if you want to perform additional customization of the end-user Web pages:

1. Review the templates and decide which one(s) you need to update.

-
2. If necessary, change the true name, alias, or nickname, as in the following lines.

```
<TEMPLATE NAME=true_name>  
<TEMPLATE NAME=alias>  
<NICKNAME=nickname>
```

true_name

Is the whole and complete name of the certificate template.

alias

Differentiates browser from server certificates. An alias is not required. You can have more than one alias.

nickname

Is an 8-character name. SAF certificates do not have nicknames. If a nickname is not present, the certificate is not renewable.

Example:

```
<TEMPLATENAME=1 Year PKI SSL Browser Certificate>  
<TEMPLATENAME= PKI Browser Certificate>  
<NICKNAME=1YBSSL>
```

Customizing the end-user Web pages

3. If necessary, in the CONTENT subsection, change the certificate fields listed. The following example is from the one-year PKI SSL browser certificate template.

Example:

```
<p> Enter values for the following field(s)
%%CommonName%%
%%Requestor (optional)%%
%%PassPhrase%%
%%PublicKey2[browsertype]%%
```

4. If you add required fields in the preceding step, update the JavaScript code that is part of the embedded HTML to check for required fields that are missing.
-

5. If necessary, in the APPL subsection, change the list of certificate fields that the application provides. (Currently, the only supported fields are UserId and HostIdMap.) The following example is from the two-year PKI browser certificate for authenticating to z/OS:

Example:

```
<APPL>
%%UserId%%
%%HostIdMap=@host-name%%
<APPL>
```

6. If necessary, in the CONSTANT subsection, update the list of certificate fields whose values are hardcoded. The following example is from the one-year PKI SSL browser certificate template:

Example:

```
<CONSTANT>
%%NotBefore=0%%
%%NotAfter=365%%
%%KeyUsage=handshake%%
%%OrgUnit=Class 1 Internet Certificate CA%%
%%Org=The Firm%%
%%SignWith=PKI:%%
<CONSTANT>
```

Note: If you update the CONSTANT subsection to create subject distinguished names, make sure that the names match the LDAP suffix defined for your LDAP server. Otherwise the certificates are not posted to LDAP. PKI Services constructs the subject distinguished name from the fields specified in the following order:

- CommonName
 - Title
 - OrgUnits (in the order that they appear in the template file)
 - Org
 - Locality
 - StateProv
 - Country
-

Customizing the end-user Web pages

7. If necessary, edit the ADMINAPPROVE subsection. (Certificates requiring an administrator's approval have an ADMINAPPROVE subsection. The absence of the ADMINAPPROVE subsection indicates auto-approval for requests.) Make sure the ADMINAPPROVE subsection, if present, correctly lists the minimum set of certificate fields that the administrator can change.

Notes:

- a. There may be more fields in the ADMINAPPROVE subsection than fields that the user can complete in the certificate request (because the users do not necessarily see all fields).
- b. Do not include the Requestor, Label, UserId, PublicKey, or SignWith fields in the ADMINAPPROVE subsection; these fields cannot be changed and are ignored if present. (See page 103 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 METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT NAME="TransactionId" TYPE="hidden" VALUE="[transactionid]">
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
```

Customizing the end-user Web pages

```
<p>%%-pagefooter%%  
</BODY>  
</HTML>  
</INSERT>
```

- The FAILURECONTENT subsection contains only the %%-requestbad%% named field, which contains the HTML for the Web page whose main heading is "Request was not successful." To make changes to this Web page, update the requestbad INSERT:

```
<INSERT NAME=-requestbad>  
<HTML><HEAD>  
<TITLE> Web Based Certificate Generation Failure</TITLE>  
</HEAD>  
<BODY>  
<H1> Request was not successful</H1>  
<p> Please correct the problem or report the error to your Web admin  
person<br>  
<PRE>  
[errorinfo]  
</PRE>  
<p>%%-pagefooter%%  
</BODY>  
</HTML>  
</INSERT>
```

-
9. If necessary, update the RETRIEVECONTENT subsection.

Note: See "Steps for changing the runtime user ID for retrieving certificates" on page 129 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 116.)

```
<INSERT NAME=-copyright>  
<!--  
/*****  
/*  
/* LICENSED MATERIALS - PROPERTY OF IBM  
/* THIS SCRIPT IS "RESTRICTED MATERIALS OF IBM"  
/* 5647-A01 (C) COPYRIGHT IBM CORP. 2000,2001  
/*  
/*****  
-->  
</INSERT>
```

- b. If necessary, update any desired Web page content (such as headers, footers, titles, background colors, frames, links, and so on) for the Web page whose main heading is "Retrieve Your (certificate template name)."

-
10. If you are updating the template for a server certificate, you can update the HTML in the RETURNCERT subsection to customize the returned Web page. (For a browser template, you cannot change the RETURNCERT subsection. It must contain the %%returnbrowsercert%% named field, which contains the [browserstype] 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, depending on the Release 4 features you are implementing, you may need to retrofit changes in the pkiserv.tmpl certificate templates file. (You would not want to replace the file if you customized it in the previous release.) The features related to changed lines in the pkiserv.tmpl certificate templates file are:

- Sending e-mail notifications (NotifyEmail named field)
- Using MAIL, STREET, and POSTALCODE distinguished name qualifiers (Email, Street, or PostalCode named fields, respectively)

The main differences between the Release 4 and Release 3 versions of the pkiserv.tmpl certificate templates file are:

- In the APPLICATION section: Added NotifyEmail field
- In the TEMPLATE section:
 - Comments in the header sections

Note: Comments are lines that begin with #. Inserting or changing these lines does not change the way the pkiserv.tmpl certificate templates file works, but IBM recommends that you retrofit changes in comment lines for better program documentation.

- Changes in the JavaScript related to new certificate named fields
- New certificate named fields (named fields begin with %%)
- In the INSERTs section: Line changes to update the field descriptions of the Subject Alternate Name extension and new sections for named fields

You can use a file comparison tool to compare the new PKI Services certificates template file (/usr/lpp/pkiserv/samples/pkiserv.tmpl) and your existing PKI Services certificates template file (/etc/pkiserv/samples/pkiserv.tmpl). For a code sample of the current pkiserv.tmpl certificates template file, see Chapter 24, “The pkiserv.tmpl certificate templates file” on page 263; revision bars on this code sample indicate lines changed from z/OS Version 1 Release 3. (Of course, revision bars on this code sample will not indicate changes for any customization you performed.)

The following table lists the templates, summarizing the template sections that changed for each enhancement.

Note: If your installation added any certificate templates, you would need additional rows in the table for these additional templates.

Table 36. Cross-reference of changes made for new features used in various templates

Certificate Template	e-mail notifications	MAIL distinguished name qualifier	STREET or POSTALCODE distinguished name qualifiers
1 Year SAF Server Certificate	Not changed	Not changed	Not changed
1 Year SAF Browser Certificate	Not changed	Not changed	Not changed
1 Year PKI SSL Browser Certificate	Changed	Changed	Not changed

Customizing the end-user Web pages

Table 36. Cross-reference of changes made for new features used in various templates (continued)

Certificate Template	e-mail notifications	MAIL distinguished name qualifier	STREET or POSTALCODE distinguished name qualifers
1 Year PKI S/MIME Browser Certificate	Changed	Not changed	Not changed
2 Year PKI Browser Certificate for Authenticating to z/OS	Changed	Not changed	Not changed
5 Year PKI SSL Server Certificate	Changed	Changed	Changed
5 Year PKI IPSEC (Firewall) Certificate	Changed	Changed	Changed
5 Year PKI Intermediate CA Certificate	Changed	Changed	Changed

Perform the following steps to retrofit changes into the pkiserv.tmpl certificate templates file:

Note: Throughout these directions, highlighted lines or parts of lines indicate new or changed lines, respectively. Non-highlighted lines are included only to provide a context for the new or changed lines.

1. If you are implementing e-mail notifications, insert the NotifyEmail named field in the APPLICATION section.

```

</SCRIPT>
<INPUT NAME="action" TYPE="hidden" VALUE="renew">
%%NotifyEmail (optional)%%
%%PassPhrase%%

```

2. Use the following table to update the template sections, based on the enhancements you have chosen to implement. (The table is ordered in the same order as the pkiserv.tmpl certificate templates file. The Enhancement column specifies the feature related to each changed section.) Go through the table, using the feature column to decide if a change is necessary. For example, if you are implementing e-mail notifications but not the Email, Street, or PostalCode named fields, in the first template ("1 Year PKI SSL Browser Certificate"), you need to make only the changes in rows 1 and 5.

Table 37. Changed lines in templates

Change Number	Enhancement	Changed lines
1 Year PKI SSL Browser Certificate		
1.	E-mail notifications	# PublicKey - required (Provided by the browser itself) # NotifyEmail - optional
2.	MAIL distinguished name qualifier	# PublicKey - required (Provided by the browser itself) # NotifyEmail - optional # Email - optional Note: Whether the NotifyEmail line appears depends on whether you added the line in the preceding row.

Table 37. Changed lines in templates (continued)

Change Number	Enhancement	Changed lines
3.	E-mail notifications and the MAIL distinguished name qualifier Note: Include these lines only if you are using both enhancements.	<pre>var STRING_UnmatchPwdPrompt= "The passwords do not match. Enter again." var STRING_UnmatchEmailPrompt= "The Email addresses for Distinguished name and notification do not match. Enter again."</pre>
4.	E-mail notifications and the MAIL distinguished name qualifier Note: Include these lines only if you are using both enhancements.	<pre>else if(document.CertReq.PassPhrase.value!= document.CertReq.ConfirmPassPhrase.value){ alert(STRING_UnmatchPwdPrompt); document.CertReq.ConfirmPassPhrase.focus(); return true; } else if((document.CertReq.Email.value != "" && document.CertReq.NotifyEmail.value != "") && (document.CertReq.Email.value!= document.CertReq.NotifyEmail.value)){ alert(STRING_UnmatchEmailPrompt); document.CertReq.NotifyEmail.focus(); return true; }</pre>
5.	E-mail notifications	<pre>%%Requestor (optional)%% %%NotifyEmail (optional)%%</pre>
6.	MAIL distinguished name qualifier	<pre><p> Enter values for the following field(s) %%CommonName%% %%Email (optional)%%</pre>
1 Year PKI S/MIME Browser Certificate		
1.	E-mail notifications	<pre># PublicKey - required (Provided by the browser itself) # NotifyEmail- optional</pre>
2.	E-mail notifications	<pre>%%Requestor (optional)%% %%NotifyEmail (optional)%%</pre>
2 Year PKI Browser Certificate For Authenticating to z/OS		
1.	E-mail notifications	<pre># PublicKey - required (Provided by the browser itself) # NotifyEmail - optional</pre>
2.	E-mail notifications	<pre><p> Enter values for the following field(s) %%Requestor (optional)%% %%NotifyEmail (optional)%%</pre>
5 Year PKI SSL Server Certificate:		
1.	MAIL distinguished name qualifier	<pre># User input fields: # Email - optional</pre>
2.	STREET or POSTALCODE distinguished name qualifier (or both)	<pre># Org - optional # Street - optional # Locality - optional # StateProv - optional # PostalCode - optional</pre>

Customizing the end-user Web pages

Table 37. Changed lines in templates (continued)

Change Number	Enhancement	Changed lines
3.	E-mail notifications	# PublicKey - required (This is the #10 request) # NotifyEmail - optional
4.	MAIL distinguished name qualifier	Same lines as row 3 of 1 Year PKI SSL Browser Certificate
5.	MAIL distinguished name qualifier	<pre>else if((document.serverform.Email.value != "" && document.serverform.NotifyEmail.value != "") && (document.serverform.Email.value != document.serverform.NotifyEmail.value)){ alert(String_UnmatchEmailPrompt); document.serverform.NotifyEmail.focus(); return true; }</pre> <p>Note: These lines are similar but not identical to the lines in Step 4 of the 1 Year PKI SSL Browser Certificate. (Differences are because these lines are for a server certificate.)</p>
6.	MAIL distinguished name qualifier	<p> Enter values for the following field(s) %% Email (Optional) %%
7.	STREET or POSTALCODE distinguished name qualifier (or both)	%%Org (Optional)%% %% Street (Optional) %% %%Locality (Optional)%% %%StateProv (Optional)%% %% PostalCode (Optional) %%
8.	E-mail notifications	%%Requestor (Optional)%% %% NotifyEmail (Optional) %%
5 Year PKI IPSEC Server (Firewall) Certificate		
Use list of changes from 5 Year SSL Server Certificate		
5 Year PKI Intermediate CA Certificate:		
Use list of changes from 5 Year SSL Server Certificate		

- If you added templates when you customized the pkiserv.tmpl certificate templates last release, make any needed changes in these templates for the enhancements you want to implement.
- Make the following highlighted changes and additions to the INSERTs section:


```
<INSERT NAME=AltIPAddr>
<p> IP address for alternate name in dotted decimal form [optfield] <BR>
<INPUT NAME="AltIPAddr" TYPE="text" SIZE=15 maxlength="15">
</INSERT>

<INSERT NAME=AltEmail>
<p> Email address for alternate name [optfield] <BR>
<INPUT NAME="AltEmail" TYPE="text" SIZE=100 maxlength="100">
</INSERT>

<INSERT NAME=AltURI>
<p> Uniform Resource Identifier for alternate name [optfield] <BR>
<INPUT NAME="AltURI" TYPE="text" SIZE=100 maxlength="255">
</INSERT>

<INSERT NAME=AltDomain>
<p> Domain name for alternate name [optfield] <BR>
```

```

<INPUT NAME="AltDomain" TYPE="text" SIZE=100 maxlength="100">
</INSERT>
<INPUT NAME="AltDomain" TYPE="text" SIZE=100 maxlength="100">
</INSERT>

<INSERT NAME=Street>
<p> Street address [optfield] <BR>
<INPUT NAME="Street" TYPE="text" MAXLENGTH=64 SIZE=64>
</INSERT>

<INSERT NAME=PostalCode>
<p> Zipcode or postal code [optfield] <BR>
<INPUT NAME="PostalCode" TYPE="text" MAXLENGTH=64 SIZE=64>
</INSERT>
<INSERT NAME=Email>
<p> Email address for distinguished name [optfield] <BR>
<INPUT NAME="Email" TYPE="text" MAXLENGTH=64 SIZE=64>
</INSERT>

<INSERT NAME=SignWith>
:
:
<INSERT NAME=TransactionId>
<p> Enter the assigned transaction ID [optfield] <BR>
<INPUT NAME="TransactionId" TYPE="text" SIZE=56 maxlength="56"
VALUE="[transactionid]">
</INSERT>

<INSERT NAME=NotifyEmail>
<p> Email address for notification purposes [optfield] <BR>
<INPUT NAME="NotifyEmail" TYPE="text" SIZE=64 MAXLENGTH="64">
</INSERT>

```

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 38. Location of code for various Web pages

Main header (and sample Web page if any)	Location of code in pkiserv.tpl certificate templates file
"1 Year S/MIME Browser Certificate"	TEMPLATE section, CONTENT subsection
"1 Year SSL Browser Certificate" (See Figure 9 on page 164.)	TEMPLATE section, CONTENT subsection
"2 Year Browser Certificate For Authenticating To z/OS"	TEMPLATE section, CONTENT subsection
"5 Year PKI Intermediate CA Certificate"	TEMPLATE section, CONTENT subsection
"5 Year PKI IPSEC Server (Firewall) Certificate"	TEMPLATE section, CONTENT subsection
"5 Year PKI SSL Server Certificate"	TEMPLATE section, CONTENT subsection
"Here's Your Certificate. Cut and Paste it to a File"	INSERT section, -return10cert INSERT Note: This is referenced in the RETURNCERT subsection of the TEMPLATE section of each certificate template.
"Internet Explorer Certificate Install" (See Figure 12 on page 167.)	INSERT section, returnbrowsercertIE INSERT

Customizing the end-user Web pages

Table 38. Location of code for various Web pages (continued)

Main header (and sample Web page if any)	Location of code in pkiserv.tpl certificate templates file
"PKISERV Certificate Generation Application" (See Figure 7 on page 158.)	APPLICATION section, CONTENT subsection
"Renew or Revoke a Browser Certificate" (See Figure 15 on page 171.)	APPLICATION section, RECONTENT subsection
"Request submitted successfully" (For submitting a successful certificate request or renewal, see Figure 10 on page 165.)	<ul style="list-style-type: none"> For a successful certificate request or renewal: INSERT section, -requestok INSERT Note: 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. Note: This is referenced in the RESUCCESSCONTENT subsection of the APPLICATION section.
"Request was not successful"	<ul style="list-style-type: none"> For an unsuccessful certificate request: INSERT section, -requestbad INSERT Note: This is referenced in the FAILURECONTENT subsection of the TEMPLATE section of each certificate template. For an unsuccessful certificate revocation request: INSERT section, -renewrevokebad INSERT Note: This is referenced in the REFAILURECONTENT subsection of the APPLICATION section.
"Retrieve Your 1 Year S/MIME Browser Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 1 Year SSL Browser Certificate" (See Figure 11 on page 166.)	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 2 Year Browser Certificate For Authenticating To z/OS"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 5 Year PKI Intermediate CA Certificate"	TEMPLATE section RETRIEVECONTENT subsection
"Retrieve Your 5 Year PKI IPSEC Server (Firewall) Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 5 Year PKI SSL Server Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your SAF Browser Certificate 1 Year"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your SAF Server Certificate 1 Year"	TEMPLATE section, RETRIEVECONTENT subsection
"SAF Browser Certificate 1 Year (Auto Approved)"	TEMPLATE section, CONTENT subsection
"SAF Server Certificate 1 Year (Auto Approved)"	TEMPLATE section, CONTENT subsection

Note: Fields (such as the Key Usage (KeyUsage) drop down or the Organizational Unit (OrgUnit) text field) are defined in the pkiserv.tmpl certificate templates file, in the INSERT section. (See Table 29 on page 94 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 eight certificate templates provided with PKI Services to determine the one that most closely approximates the certificate template you want to add.

2. After you have determined the certificate template to use as a model, copy this section in the certificate templates file.

3. Provide a new name, alias, and, if present, nickname for the certificate template.

4. Follow the remaining steps, starting at Step 3 on page 118 in the preceding section.

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 17, “RACF administration for PKI Services” on page 199 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

Customizing the end-user Web pages

- Under the surrogate user ID without user ID and password authentication.

Protection directives in the z/OS HTTP Server's configuration file (which defaults to /etc/httpd.conf) enforce these three levels of access control. The default configuration for PKI Services maps the three levels of access control to the following CGI directories respectively:

- /PKIServ/ssl-cgi-bin/auth
- /PKIServ/ssl-cgi-bin/surrogateauth
- /PKIServ/ssl-cgi-bin

Each of the request and retrieve CGIs reside in all three directories. Thus, when you run a CGI you get the protection established for the directory from which it is called.

Each certificate template contains several FORM statements (two commented out and one uncommented, which is active) that determines which of these applies. You can change the access control by uncommenting one of the FORM statements that is commented out and commenting out the one that is active.

Steps for changing the runtime user ID for requesting certificates

Perform the following steps to change the runtime user ID for requesting a certificate.

1. In the pkiserv.tmpl file, find the CONTENT subsection of the TEMPLATE section for the template whose user ID you want to change. Locate the lines containing the FORM statements, such as those in the following example:

Example:

```
<h3><li>Request a New Certificate
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#           "/PKIServ/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#           "/PKIServ/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
<FORM NAME="CertReq" METHOD=POST ACTION=
           "/PKIServ/ssl-cgi-bin/careq.rexx" onSubmit=
```

Notice that the preceding lines contain three FORM statements. The first two FORM statements are commented out, so they are not active. They are for:

- Requesting the certificate under the client's ID and using user ID and password authentication
- Requesting the certificate under the surrogate ID and using user ID and password authentication

The third FORM statement is for requesting the certificate under the surrogate user ID without user ID and password authentication. This is active (it is not commented out).

2. To change the runtime user ID, remove the comment delimiter (#) from in front of the lines for the commented-out FORM statement you want to use and insert the comment delimiter in front of the lines for the bottom FORM statement.

Steps for changing the runtime user ID for retrieving certificates

Perform the following steps to change the runtime user ID for retrieving a certificate.

1. In the pkiserv.tmpl file, find the RETRIEVECONTENT subsection of the TEMPLATE section for the template whose user ID you want to change. Locate the lines containing the FORM statements, such as those in the following example:

Example:

```
<H1> Retrieve Your [tmplname]
<H3>Please bookmark this page
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
<FORM NAME=retrieveform METHOD=POST ACTION=
     "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
```

Notice that the preceding lines contain three FORM statements. The first two FORM statements are commented out (they are not active). These are for:

- Retrieving the certificate under the client's ID
- Retrieving it under the surrogate ID, but requiring user ID and password authentication.

The third FORM statement is for retrieving the certificate under the surrogate user ID without user ID and password authentication. This is active (it is not commented out).

2. To change the runtime user ID, remove the comment delimiter (#) from in front of the lines for the commented-out FORM statement you want to use and insert the comment delimiter in front of the lines for the bottom FORM statement.

Customizing 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 ready to expire (unless it has already been renewed or revoked).

Once a day, PKI Services checks the issued certificate list (ICL) for expiring certificates. (The ExpireWarningTime parameter (see the CertPolicy section in Table 20 on page 58) sets the time interval of how long before the certificate expires

Customizing the end-user Web pages

that the message is sent.) When PKI Services finds an expiring certificate, it 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.

If you are not sending e-mail notifications, see Step 6b on page 117 for directions.

If you are sending e-mail notifications, you need to:

- Have copies of the forms in the runtime directory. (For information about copying the message forms to the runtime directory, see Step 2 on page 54.
- Customize the forms. (For details, see “Steps for customizing e-mail notification forms” on page 132.)
- 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 117.)

For more information about the NotifyEmail field, see Table 29 on page 94. For information about fields on request forms, see Table 45 on page 160. For the pkiserv.tmpl certificate template code sample, see Chapter 24, “The pkiserv.tmpl certificate templates file” on page 263.

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 http://www.dimeocert.com/PKIServ/public-cgi/camain.rexx
to retrieve your certificate. You will need the transaction ID
listed below and your passphrase that you entered when
you submitted the request.

%%transactionid%
```

Figure 4. *readymsg.form*

```

From:dime-o-cert PKI
Subject:Certificate Request Rejected

Attention - Please do not reply to this message as it was automatically sent by a service machine.

Dear %%requestor%%,

Thank you for choosing dime-o-cert PKI. We are sorry to inform you that
your certificate request for subject %%dn%% has been rejected.
Please contact the PKI Services administrator at 1-800-xxx-xxx.
You will need the transaction ID listed below.

%%transactionid%%
    
```

Figure 5. rejectmsg.form

```

From:dime-o-cert PKI
Subject:Certificate Expiration

Attention - Please do not reply to this message as it was automatically sent by a service machine.

Dear %%requestor%%,

Thank you for choosing dime-o-cert PKI. The certificate your requested for
subject %%dn%% expires at %%notafter%% local time. If you wish to renew
your certificate, please visit http://www.dimeocert.com/PKIServ/public-cgi/camain.rexx.
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 6. expiringmsg.form

Notes:

1. PKI Services automatically provides the "To:" value in the forms. You can include "From:" or "Subject:" or both at the top of the file.
2. You must have a blank line between the Subject and the body of the form.

The following table summarizes the variables you can use in the forms when you customize them. At runtime, PKI Services replaces these with their actual values.

Table 39. Descriptions of variables for forms

Variable	Description
%%dn%%	The subject's distinguished name. (This is valid in all the forms.)
%%notafter%%	The certificate expiration date and time in local time in the format YYYY/MM/DD HH:MM:SS. (This is valid only in the expiring.form. It is ignored in the other forms.)
%%requestor%%	The requestor of the certificate. PKI Services obtains this information from the Requestor field the user submits on the original certificate request. (This field is valid in all the forms.)
%%transactionid%%	The transaction ID (CertId) returned. (This is valid for ready and reject forms only. It is ignored in the expiring form.)

Customizing the end-user Web pages

The following table summarizes the substitution variables contained in the ready, rejected, and expiring examples:

Table 40. Summary of substitution variables in forms

Referenced substitution variables	readymsg.form	rejectmsg.form	expiring.form
%%dn%%	X	X	X
%%notafter%%	(ignored)	(ignored)	X
%%requestor%%	X	X	X
%%transactionid%%	X	X	(ignored)

Steps for customizing e-mail notification forms

Perform the following steps to customize the ready, rejected, and expiring forms:

1. Make sure the forms you want to use (readymsg.form, rejectmsg.form, and expiringmsg.form) are present in the runtime directory. (By default, the runtime directory is /etc/pkiserv/. For information about copying files, see Step 2 on page 54.)
2. Update the form. At minimum:
 - a. Delete the first four (comment) lines (as shown in the following), so that the first two lines in your file are the "From:" and "Subject:" lines:

```
# Licensed Materials - Property of IBM
# 5694-A01
# (C) Copyright IBM Corp. 2002
# Status = HKY7707
```
 - b. Specify your company (instead of dime-o-cert) in the From: line and in the first line of the main paragraph .
 - c. If appropriate, update the subject.

Note: There must be a blank line between the subject and the body of the note.

- d. If you are updating a ready or expiring form, change the URL in the main paragraph to customize it for your company.
- e. 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.
-

Chapter 13. Customizing the administration Web pages

CGIs for administration Web pages

CGIs are REXX execs that gain control when the user clicks an action button. The administrative CGIs are connector REXX execs that render Web pages dynamically.

All of the administrative CGIs are contained in the `usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth/directory`.

The following table (which lists the REXX execs in logical order) summarizes the actions the CGIs perform:

Table 41. CGI actions for administrative Web pages

REXX exec	Action	Sample Web page
admmain.rexx	This displays the administration home page. The main heading is "PKI Services Administration." This Web page lets the administrator work with a single certificate request or certificate or search for certificate requests or certificates.	See Figure 19 on page 179.
admpend.rexx	On the administration home page, the administrator can search for certificate requests. This displays a Web page whose main heading is one of the following: <ul style="list-style-type: none">• "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 23 on page 185.
admpendtid.rexx	On the administration home page, the administrator can enter a transaction ID to work with a single certificate request. This displays a Web page whose main heading is one of the following: <ul style="list-style-type: none">• "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 20 on page 180.
admmodtid.rexx	This displays the "Modify and Approve Request" Web page that appears when the administrator decides to modify a request before approving it (on the "Single Request" Web page).	See Figure 22 on page 182.
admicl.rexx	On the administration home page, the administrator can search for certificates. This displays a Web page whose main heading is one of the following: <ul style="list-style-type: none">• "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 23 on page 185.
admiclcert.rexx	On the administration home page, the administrator can enter a serial number to work with a single certificate. This displays a Web page whose main heading is one of the following: <ul style="list-style-type: none">• "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 27 on page 190.

Customizing the administration Web pages

Table 41. CGI actions for administrative Web pages (continued)

REXX exec	Action	Sample Web page
admacttid.rexx	Displays a Web page after the administrator processes a single certificate request (approving it with or without modifications, rejecting, or deleting it). This Web page has one of the following as its main heading: <ul style="list-style-type: none">• "Processing successful"• "Processing was not successful"	For a sample of the Web page whose main heading is "Processing successful" see Figure 21 on page 181.
admacttid2.rexx	This displays a Web page after the administrator approves a certificate request with modifications. The Web page has one of the following main headings: <ul style="list-style-type: none">• "Processing successful"• "Processing was not successful"	For a sample of the Web page whose main heading is "Processing successful" see Figure 21 on page 181.
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: <ul style="list-style-type: none">• "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	<ul style="list-style-type: none">• For an example of the "Processing successful" Web page, see Figure 24 on page 187.• For an example of the "Processing was not successful" Web page, see Figure 25 on page 187.• For an example of the "Processing partially successful" Web page, see Figure 26 on page 188.
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: <ul style="list-style-type: none">• "Processing successful"• "Processing was not successful"	None
admiclall.rexx	After the administrator searches for certificates and admicl.rexx displays the results, the administrator clicks a button to revoke or delete selected certificates. This calls admiclall.rexx, which displays a Web page whose main heading is one of the following: <ul style="list-style-type: none">• "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	None

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 two subsections in the APPLICATION section of the pkiserv.tmpl certificate template file:

ADMINHEADER

Contains the general installation-specific HTML content for the header of all the administration Web pages.

ADMINFOOTER

Contains the general installation-specific HTML content for the footer of all the administration pages.

Steps for customizing the administration Web pages

Perform the following steps to customize the administration Web pages:

1. Add any desired Web page header for the administration pages to the ADMINHEADER subsection of the PKISERV APPLICATION section. (The ADMINHEADER subsection is near the end of the APPLICATION section.)

Example:

```
<ADMINHEADER>
<HTML>h<HEAD>
<TITLE>Web-Based Certificate Generation Administration</TITLE></HEAD>
<BODY>
</ADMINHEADER>
```

-
2. Add any desired Web page footer for the administration pages to the ADMINFOOTER subsection of the APPLICATION section. (The ADMINFOOTER subsection is near the end of the APPLICATION section.)

Example:

```
<ADMINFOOTER>
<p> email: webmaster@company.com
</BODY>
</HTML>
</ADMINFOOTER>
```

Changing the runtime behavior for accessing administration pages

When the administrator tries to access the administration pages (by clicking the **Go to administration page** button on the PKI Services home page), access to the administration pages is controlled in one of the following ways:

- A popup window appears, requiring the administrator to enter a user name and password. (See Figure 18 on page 176 for a sample of the authentication popup window.)
- Alternately, the administrator may 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 17, “RACF administration for PKI Services” on page 199.)

Customizing the administration Web pages

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.tpl certificate templates file and find the following lines in the 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).
-

(Optional) Steps for removing the administration page link from the PKI Services home page

Optionally remove the administration page link from the PKI Services home page and provide an alternative way for administrators to access the administration home page.

Recommendation: Do these steps before going into production mode for added security. It will prevent your general end-user population from trying to access the administration pages if they are not authorized to do so.

To remove the administration page link and provide an alternative way to access the administration pages, perform the following steps:

1. If you want to require your administrators to authenticate using a certificate, make sure you have performed the steps in “Steps for changing control of access to administration pages” and that you have tested to ensure your changes work.
-
2. Edit the pkiserv.tpl file and remove the entire HTML <FORM>...</FORM> in “Steps for changing control of access to administration pages”. (These are the very same lines you edited in that section.)
-
3. Make the administration URL available to your administrators through an alternate means. (For example, you can use a link in another internal web page or in an e-mail message, and so on.)

Customizing the administration Web pages

In the following URLs, *webserver-fully-qualified-domain-name* is the common name (CN) portion of the public webserver's distinguished name; see Table 12 on page 39. Web server redirection ensures that an SSL connection is established.

- If you are allowing your administrators to authenticate by specifying a user ID and password, the URL is:

`http://webserver-fully-qualified-domain-name/PKIServ/ssl-cgi/auth/admmain.rexx`

- If you want your administrators to authenticate using a client certificate, the URL is:

`http://webserver-fully-qualified-domain-name/PKIServ/clientauth-cgi/auth/admmain.rexx`

Note: Your administrators still need to visit the PKI Services home page to install the CA certificate into their browsers. See “Steps for accessing the administration home page” on page 173 for more information.

Customizing the administration Web pages

Chapter 14. Advanced customization

This chapter describes advanced customization methods that PKI Services provides, including:

- Using certificate policies
- Updating the signature algorithm
- Using the PKI exit.

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 the Internet Engineering Task Force (IETF) Web site (www.ietf.org) for RFC2459.)

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. 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

Unlike other extensions, which you can define on a per certificate template basis, PKI Services supports the CertificatePolicies extension only on a global basis. Either all the certificates PKI Services creates have the same CertificatePolicies extension or none of them have it.

Steps for creating the CertificatePolicies extension

Perform the following steps to create your own CertificatePolicies extension:

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 recommended), 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 value of MyPolicy is 1.2.3.4. The value of

Advanced customization

MyPolicy should be a customer-specific (registered) Object ID identifying your organization's certificate. Replace the value of MyPolicy in the following line with your Object ID. Make a note of the value (you need it for the next step).

Example:

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

5. Go back to the CertPolicy section and update the PolicyName1 line to change *MyPolicy* to the value of MyPolicy in the OIDs section:

```
[CertPolicy]  
PolicyName1=MyPolicy
```

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

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

```
Policy1Org=My Company, Inc  
Policy1Notice1=1
```

Policy1Org Your organization's name, for example, International Business Machines, Inc.

Policy1Notice1 through Policy1Notice*n*

Your notice numbers. (You may need more than one Policy1Notice*n* line, depending on how many notice numbers you have. Repeat the line as needed, by incrementing the suffix number on the keyword, for example Policy1Notice1, Policy1Notice2, and so forth.)

- b. Change the value of the UserNoticeText1 line shown in the following. The *statement* should be your notice text string, for example, Certificate for IBM internal use only.

```
UserNoticeText1=statement
```

- c. Change the value of the CPS1 line shown in the following. The value should be your CPS URI, for example, <http://www.ibm.com/cps.html>.

```
CPS1=http://www.mycompany.com/cps.html
```

If you do not want to add qualifiers, delete or comment out (by inserting a # character at the start of the line) the preceding lines.

7. If you need multiple qualifiers, repeat the following fields as needed, incrementing the suffix numbers, for example:

```
PolicyName2=MyOtherPolicy  
Policy2Org=International Business Machines, Inc.  
Policy2Notice1=5  
Policy2Notice2=9  
UserNoticeText2=Certificate is intended for testing only  
CPS2=http://www.ibm.com/cps2.html
```

Updating the signature algorithm

By default, PKI Services uses the SHA-1 with RSA encryption signature algorithm for signing certificates and CRLs. If you need to use one of the older RSA algorithms, you can change the `SigAlg1` value in the `CertPolicy` section of the `pkiserv.conf` configuration file. The signature algorithm must be one of the following:

- `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`

Steps for changing the signature algorithm

Perform the following steps to change the signature algorithm:

1. Edit the `pkiserv.conf` configuration file and find the `OIDs` section.

2. If you want to change from SHA-1 encryption to MD-5, add the following line:


```
md-5WithRSAEncryption=1.2.840.113549.1.1.4
```

Otherwise, to change to MD-2, add the following line:

```
md-2WithRSAEncryption=1.2.840.113549.1.1.2
```

3. Find the `CertPolicy` section.

4. If you want to change from SHA-1 encryption to MD-5, change `sha-1WithRSAEncryption` in the following line to `md-5WithRSAEncryption`. If you want to change to MD-2, change `sha-1WithRSAEncryption` to `md-2WithRSAEncryption`.


```
SigAlg1=sha-1WithRSAEncryption
```

Using the PKI exit

Programming Interface information

For the end-user functions except `VERIFY`, the `PKISERV` 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.

PKI Services provides the following files for the exit. Both files are, by default, located in: `/usr/lpp/pkiserv/samples/`.

Table 42. Summary of information about important files for the exit routine

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

Advanced customization

If the exit exists, it must be a UNIX executable residing in the HFS, and it must have appropriate permission assigned. To specify the exit, the UNIX programmer sets the `_PKISERV_EXIT` environment variable (see page 307). On input it receives standard UNIX parameters (that is, `argc` and `argv[]`). It communicates back to PKISERV through the return code and by writing to STDOUT.

Steps for updating the exit 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 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 43. Values of arguments for pre- and post-processing

Time of processing	Argument 1	Argument 2
Preprocessing	0	The function number from the SAF callable service in EBCDIC: 1 GENCERT 2 EXPORT
Post-processing	1	9 REQCERT 11 REVOKE 12 GENRENEW 13 REQRENEW

Return Codes

The sections 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.

Advanced customization

GENCERT and GENRENEW - preprocessing

Purpose: Provide additional authorization checking and parameter validation and modification.

Arguments:

argument 3...argument n

The parameters as input to the CGI plus values resolved by the CGI in *name=value* form, for example, "CommonName=Sam Smith".

Return Codes:

Return Code	Meaning
0	Continue with the request with possible modifications.
4	Continue with the request with possible modifications, but change it to require administrator approval.
>=8 <50	Deny the request and return to the caller immediately.

STDOUT: Zero or more additional *CertPlist* parameters to add to the request in *name=value* form, one per line. For those fields defined as non-repeating (according to the documentation for the IRRSPX00 callable service, for example, *CommonName*), specifying the parameters here in effect replaces the CGI input values.

GENCERT and GENRENEW - post-processing

Purpose: Capture the TransactionId or failing return codes for further processing.

Arguments:

argument 3...argument n-3

The final set of parameters as determined by the preprocessing exit in *name=value* form.

argument n-2

The RACF return code from the callable service.

argument n-1

The RACF reason code from the callable service.

argument n

The *TransactionId*. This is a string of undetermined value if the request was unsuccessful.

Return Codes:

Return Code	Meaning
0	Normal

STDOUT: Optional replacement *TransactionId*.

Advanced customization

REQCERT and REQRENEW - preprocessing

Purpose: Provide additional authorization checking and parameter validation and modification.

Arguments:

argument 3...argument n

The parameters as input to the CGI plus values resolved by the CGI in *name=value* form, for example, "CommonName=Sam Smith".

Return Codes:

Return Code	Meaning
0	Continue with the request with possible modifications.
4	Continue with the request with possible modifications, but change it to not require administrator approval.
>=8 <50	Deny the request and return to the caller immediately.

STDOUT: Zero or more additional *CertPlist* parameters to add to the request in *name=value* form, one per line. For those fields defined as non-repeating (according to the documentation for the IRRSPX00 callable service, for example, *CommonName*), specifying the parameters here in effect replaces the CGI input values.

REQCERT and REQRENEW - post-processing

Purpose: Capture the *TransactionId* or failing return codes for further processing.

Arguments:

argument 3...argument n-3

The final set of parameters as determined by the preprocessing exit in *name=value* form.

argument n-2

The RACF return code from the callable service.

argument n-1

The RACF reason code from the callable service.

argument n

The *TransactionId*. This is a string of undetermined value if the request was unsuccessful.

Return Codes:

Return Code	Meaning
0	Normal

STDOUT: Optional replacement *TransactionId*.

Advanced customization

EXPORT - preprocessing

Purpose: Provide additional authorization checking and parameter validation and modification.

Arguments:

argument 3...argument n

The parameters as input to the CGI in *name=value* form, for example, "TransactionId=12345".

Return Codes:

Return Code	Meaning
0	Continue with the export.
>=8 <50	Deny the request and return to the caller immediately.

STDOUT: Optional replacement *TransactionId* and *ChallengePassPhrase* parameters in *name=value* form, one per line. If these values are provided, they replace the user-provided values on the call to the SAF callable service. If *TransactionId* is specified without *ChallengePassPhrase*, the user-provided *ChallengePassPhrase* is used. If *ChallengePassPhrase* is specified without *TransactionId*, the user-provided *TransactionId* is used.

EXPORT - post-processing

Purpose: Capture the certificate or failing return codes for further processing.

Arguments:

argument 3...argument n-3

The parameters as input to the CGI in *name=value* form, followed by any modified value provided by the preprocessing exit, also in *name=value* form.

argument n-2

The RACF return code from the callable service.

argument n-1

The RACF reason code from the callable service.

argument n

The base64–encoded certificate with header and footer. This is a string of undetermined value if the request was unsuccessful.

Return Codes:

Return Code	Meaning
0	Normal

STDOUT: Non-applicable.

Advanced customization

REVOKE - preprocessing

Purpose: Provide additional authorization checking and parameter validation.

Arguments:

argument 3...argument n

The parameters as input to the CGI in *name=value* form, for example, "reason=1".

Return Codes:

Return Code	Meaning
0	Continue with the request.
>=8 <50	Deny the request and return to the caller immediately.

STDOUT: Non-applicable.

REVOKE - post-processing

Purpose: Capture the certificate or failing return codes or both for further processing.

Arguments:

argument 3...argument n-2

The parameters as input to the CGI in *name=value* form, for example, "reason=1".

argument n-1

The RACF return code from the callable service.

argument n

The RACF reason code from the callable service.

Return Codes:

Return Code	Meaning
-------------	---------

0	Normal
---	--------

STDOUT: Non-applicable.

Advanced customization

Scenarios for using the PKI exit

The sample PKI exit supplied with PKI Services, `pkixit.c`, written in the C language. It is intended to demonstrate the power of the exit 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. The individual subroutines in the program handle the following 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 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 the user's level of access to RACF FACILITY class resources called `PROJ.MEMBER` and `PROJ.PARTNER`. The access values are as follows:

NONE	No access for either resource. The user is not permitted to request this type of certificate. The certificate request is denied.
READ to PROJ.MEMBER	The user is a team member and is permitted to request the certificate. The TITLE value is set to "Team Member." Certificate requests for team members are automatically approved. (No administrator approval is required.)
UPDATE to PROJ.MEMBER	The user is the team's leader and is permitted to request the certificate. The TITLE value is set to "Team Leader." A certificate request by the team leader is automatically approved. (No administrator approval is required.)
READ to PROJ.PARTNER	The user is considered to be a general partner of the team, not an active team member. The user is allowed to request certificates, but the requests require administrator approval before being issued. The TITLE value is set to "Team Partner."
UPDATE to PROJ.PARTNER	The user is considered to be a trusted partner of the team, not an active team member. The user is allowed to request certificates, and unlike requests of the general partner, the certificate request are automatically approved. The TITLE value is set to "Team Trusted Partner."

The preprocessing exit 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 access 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 use 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 `requestPeriod`.

Advanced customization

- Subroutine `determineExpiration` is called to extract the expiration date from the user's certificate. This subroutine calls several lower subroutines to base64 decode the certificate, DER decode the binary certificate, and convert the expiration date to a seconds value.
- Upon return from `determineExpiration`, the variable `timeBeforeExp` is the number of seconds from now that the certificate expires. This is compared against the number of seconds in 30 days ($86400 * 30$) to see if it is greater than 30 days.
 - If it is greater than 30, the request is rejected by setting return code 8.
 - If it is not greater than 30, the new `NotAfter` value is computed as $timeBeforeExp/86400 + requestPeriod$.
- This new `NotAfter` value is set by writing it to STDOUT.

_____ **End of Programming Interface information** _____

Part 4. Using PKI Services

This part explains how to use the PKI Services Web pages.

- Chapter 15, “Using the end-user Web pages” on page 157 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 16, “Using the administration Web pages” on page 173 shows the administration Web pages and explains how to process certificate requests and certificates.

Chapter 15. Using the end-user Web pages

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

Note: The PKI Services Web pages in this chapter may differ slightly from those on the Web. If you need to see the exact content, view the pages on the Web. Additionally, the pages may contain differences depending on the browser you are using. This chapter assumes you are using Internet Explorer.

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

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

Table 44. Types of certificates you can request

Type of certificate	Use
One-year PKI SSL browser certificate	End-user client authentication using SSL
One-year PKI S/MIME browser certificate	Browser-based e-mail encryption
Two-year PKI browser certificate for authenticating to z/OS	End-user client authorization using SSL when logging onto z/OS
Five-year PKI SSL server certificate	SSL Web server certification
Five-year PKI IPSEC server (firewall) certificate	Firewall server identification and key exchange
Five-year PKI intermediate CA certificate	Subordinate (non-self-signed) Certificate Authority certification
One-year SAF browser certificate	End-user client authentication where RACF (not PKI Services) is the certificate provider
One-year SAF server certificate	Web server SSL certification where RACF (not PKI Services) is the certificate provider

Note: If your installation has not customized the certificate templates, the PKI Services Web pages in this chapter may still differ slightly from those on the Web; if your installation customized the templates, the Web pages in this chapter may differ greatly from those you view on the Web.

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 PKISERV certificate generation application Web page (shown in the following figure):

PKISERV Certificate Generation Application

[Install our CA certificate into your browser](#)

Choose one of the following:

- **Request a new certificate using a model**
Select the certificate template to use as a model
- **Pick up a previously requested certificate**
Enter the assigned transaction ID
Select the certificate return type
- **Renew or revoke a previously issued browser certificate**
- **Administrators click here**

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 7. PKISERV certificate generation application Web page

2. If this is the first time you have accessed the forms on these Web pages, to install the CA certificate into your browser. Click the **Install our CA certificate into your browser** 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 our CA certificate into your browser** 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. This displays the following popup window:

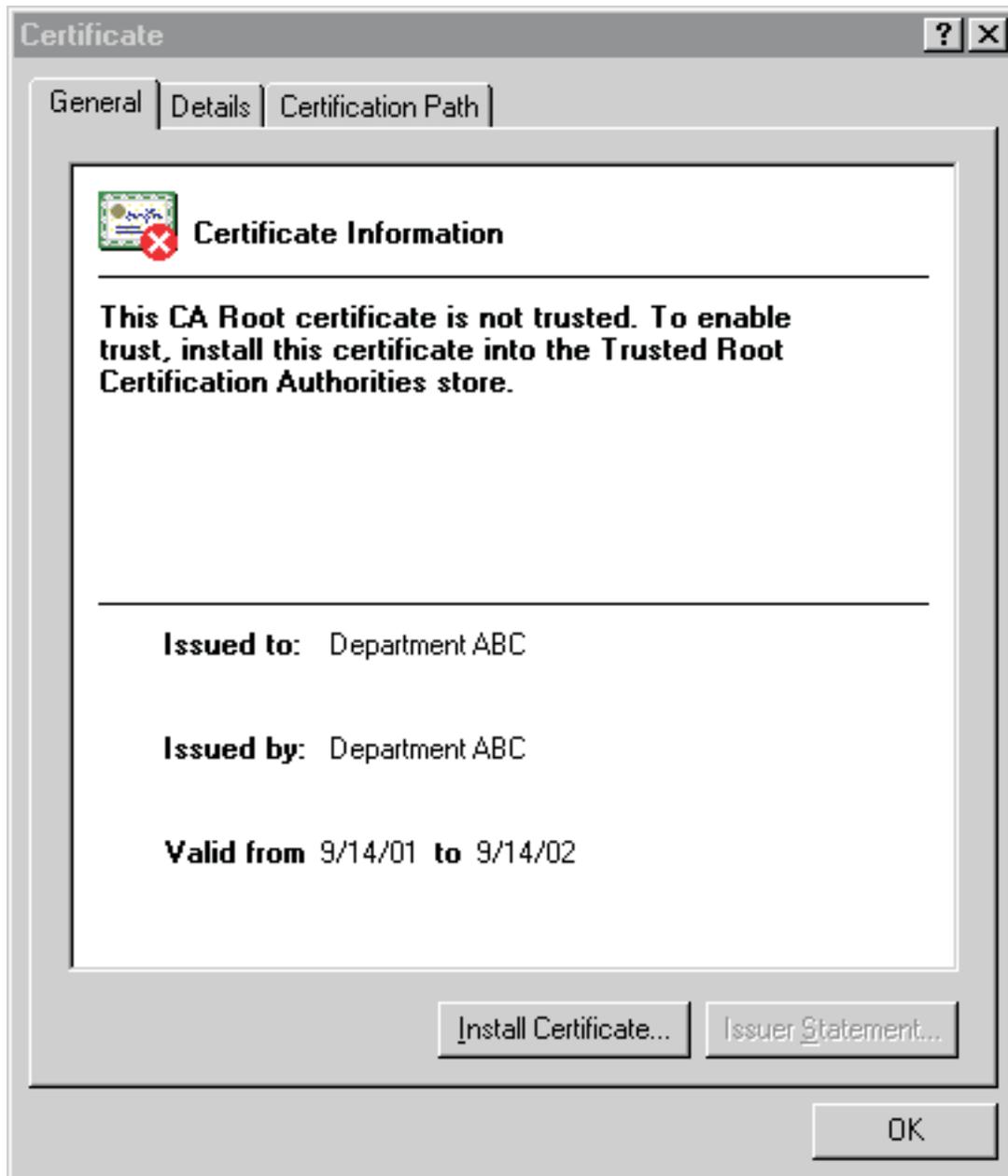


Figure 8. The Certificate popup window for installing the CA certificate

- b. Click the **Install certificate** button. (This initiates a sequence 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.")

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

Using the end-user Web pages

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 45. Summary of fields in end-user Web pages

Field	Description
Base64-encoded PKCS#10 certificate request	<p>(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:</p> <pre>MIIBiDCB8gIBADAZMRcwFQYDVQQDEw5Kb2huIFEuIFB1YmxpYzCBnzANBgkqhkiG 9w0BAQEFAAOBjQAwgYkCgYEAAsCT1cJHAGPqi60jAyL+xNbt8z5ngmvq02V003oYu /mEnQtRM96e+2jbmDCRo5tWVklG40Yf9ZVB5biURMJFLztfa4AVdEVtun8DH2pwc wiNIZZcC1Zym5adurUmyDk64PgiiIPMQS/t0ttG4c5U8uWSK0b1J4V4f7ps+t1aG t+cCAwEAAaAwMC4GCSqGSIB3DQEJJDjEhMB8wHQYDVR00BBYEFAlKTovBBvnFqDAO 1oIhtRinwRC9MA0GCSqGSIB3DQEBBQUAA4GBAIBCVpwYvppIX3HHmpKZPNY8Snsz AJrDsgAEH51W0IRGywhqKcLLxa9htoQai6cdc8RpFVTwk6UfdCOGxMn4aFb34Tk3 5WYdz0iHXg8MhHiB3EruwdWs+S7Fv3JhU3FLwU61FLfAjbVi+35iEWQymOR6mE5W CathprmGfKRsDE5E</pre>
Challenge passphrase	<p>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.</p>
Common name	<p>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.</p> <p>Notes:</p> <ol style="list-style-type: none">1. For SSL servers, the common name is the server's fully qualified domain name, for example, <code>www.ibm.com</code>.2. The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Country	<p>The country where your organization is located. This is a 2-character text field.</p> <p>Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>
Cryptographic service provider	<p>(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. The default selection is "Microsoft Enhanced Cryptographic Provider"; this provides 1024-bit key encryption. The other choice is the "Microsoft Base Cryptographic Provider"; this provides 512-bit key encryption. Larger keys are more secure, but they also increase the time that is needed for connecting to a secure session.</p>
Domain name	<p>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.</p> <p>Note: The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>
E-mail address for distinguished name	<p>E-mail address for the distinguished name. This is a text field of up to 64 characters.</p> <p>Notes:</p> <ol style="list-style-type: none">1. The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.2. If you specify a value for this parameter and for "Notification e-mail address," the two values must be the same.

Table 45. Summary of fields in end-user Web pages (continued)

Field	Description
E-mail address for alternate name	E-mail address for alternate name, including the @ character and any periods (.). This is a text field of up to 100 characters. Note: The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
HostIdMappings extension	This is the user ID for authorization purposes in the following format: subject-id@host-name for example, DSmith@ibm.com. This is a text field of up to 100 characters.
IP address	The IP address for the alternate name. This unique IP version 4 address specifies the location of each device or workstation on the Internet, for example, 9.67.97.103. (PKI Services supports only IP version 4 addresses.) The IP address is in dotted decimal format and is a text field of up to 15 characters. Note: The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
Key protection	(This is for the Internet Explorer browser only.) This asks if you want to enable private key protection. (The dropdown choices are Yes and No.)
Key size	(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.
Key usage	This indicates the intended purpose of the certificate. Possible values are: handshake Protocol handshaking (for example, SSL) dataencrypt Data encryption certsign Certificate signing docsign Document signing
Label	The label assigned to the requested certificate. This is a text field of up to 32 characters.
Locality	The city or municipality where your organization is located, such as Pittsburgh or Paris. This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Not before (date)	A number of days, added to the current date (by default, you can select either 0 or 30), before which the certificate is not valid.
Not after (date)	A number of days, added to the current date, after which the certificate expires. By default, you can select either 1 year or two years for the time at which the certificate expires.
Notification e-mail address	E-mail address for notification purposes. This is a text field of up to 64 characters. Note: If you specify a value for this parameter and for "E-mail address for distinguished name," the two values must be the same.
Organization	The legally registered name (or trademark name, for example, IBM) of your organization. This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Organizational unit	The name of your division or department. (There can be more than one organizational unit field on a request form. For example, one could be for your department and another for your division.) This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Pass phrase	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 recommended.

Using the end-user Web pages

Table 45. Summary of fields in end-user Web pages (continued)

Field	Description
Postal code	Your postal code or zip code. This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
State or Province	The state or province where your organization is located. Your registration policies determine whether you spell out the full name of the state or province or use an abbreviation. This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Street	Your street address. This is a text field of up to 64 characters. Note: The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Title	Your 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.
Transaction ID	PKISERV Web pages assign this 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.
Uniform resource identifier (URI)	Uniform resource identifier for the alternate name. This is a name or address referring to an Internet resource; a URL is one kind of uniform resource identifier. This is a text field of up to 100 characters. Note: The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
Your name	Your name (for tracking purposes). This can be in any format, for example, John Smith or John. J. Smith. This is a text field of up to 32 characters.

Steps for requesting a new certificate

To request a new certificate, first go to the PKI Services home page (see Figure 7 on page 158).

Perform the following steps to request a new certificate:

1. Click the down arrow to the right of the field beside "Select the certificate template to use as a model." This displays a list of certificate templates from which you can select.

Note: The following list shows the certificate templates that PKI Services provides by default. This list may differ from the certificate templates your installation provides because your installation can customize the certificate templates and Web pages.

- 1-year PKI SSL browser certificate
- 1-year PKI S/MIME browser certificate
- 2-year PKI browser certificate for authenticating to z/OS
- 5-year PKI SSL server certificate
- 5-year PKI IPSEC server (Firewall) certificate
- 5-year PKI intermediate CA certificate
- 1-year SAF browser certificate
- 1-year SAF server certificate

Using the end-user Web pages

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 may 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. For example, if you are requesting a one-year SSL browser certificate, the following form appears:

1 Year SSL Browser Certificate

Choose one of the following:

- **Request a New Certificate**

Enter values for the following field(s)

Common Name

Email address for distinguished name (optional)

Your name for tracking this request (optional)

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

Enable strong private key protection?
- **Pick Up a Previously Issued Certificate**

email: webmaster@your-company.com

Figure 9. One-year SSL browser certificate request form

Note: 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.

- In the case of the one-year SSL browser certificate, fill in your common name (see Table 45 on page 160 for descriptions of fields) and passphrase (twice). 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.
- When you are satisfied with the information you have entered, click the **Submit certificate request** button.

5. If the request is successful, you see a page like the following, which tells you your transaction ID.

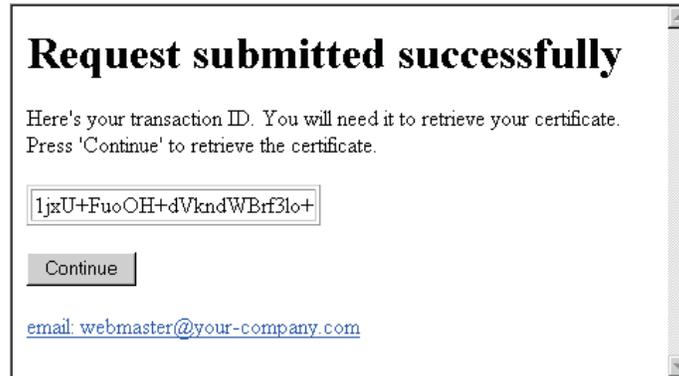


Figure 10. Successful request displays transaction ID

- a. Make a note of the Transaction ID. (You can copy and paste the Transaction ID to a file so that you have it for future reference, or you can write it in the box below. The reason for keeping a record of the Transaction ID is that, depending on how you go to the Web page to retrieve your certificate (see Figure 11 on page 166), you may have to fill in the transaction ID on that Web page.)

Transaction ID:	
------------------------	--

- b. Click the **Continue** button. This displays the following Web page:

Retrieve Your PKI 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 return to this 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

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.

Home page

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 11. 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 may 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 the 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 167) or you can return to the PKI Services home page (by clicking the **Home** button).

Retrieving your certificate

You can retrieve your certificate:

- From Web page you bookmarked in Step 5c on page 166. (This Web page contains your transaction ID, so you do not have to enter it.) The steps that follow are for retrieving your certificate from the bookmarked Web page.
- From the PKI Services home page (see Figure 7 on page 158 and “Steps for retrieving your certificate from the PKI Services home page” on page 169).

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 when your certificate is ready for retrieval (or if your certificate request has been rejected).

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 11 on page 166.)
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 on page 168. If you are using Internet Explorer and the retrieval of a certificate is successful, this displays the Web page shown in Figure 12. (This is for a browser certificate. For a server certificate, Figure 13 on page 168 shows an example of the Web page.)

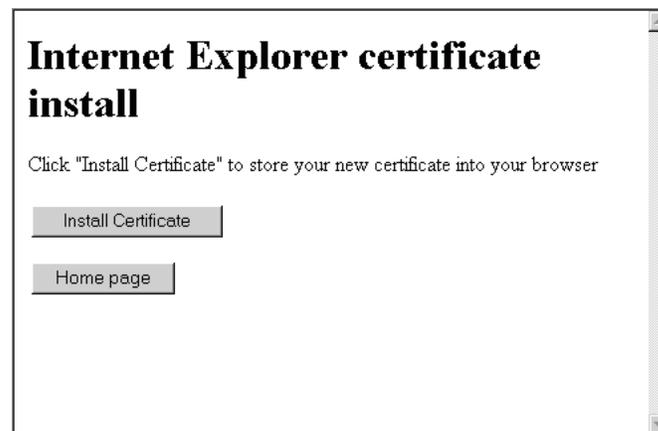


Figure 12. Browser certificate installation Web page



Figure 13. Server 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 10 on page 165.)

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 7 on page 158.) This displays the Web page that Figure 11 on page 166 displays.

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 12 on page 167 shows. (This is for a browser certificate. For a server certificate, Figure 13 on page 168 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."

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 renewing a certificate

Perform the following steps to renew a certificate:

1. On the PKI Services home page (see Figure 7 on page 158), click the **Renew or revoke certificate** button. This displays a popup window with a list of certificates, such as the following figure shows:

Using the end-user Web pages

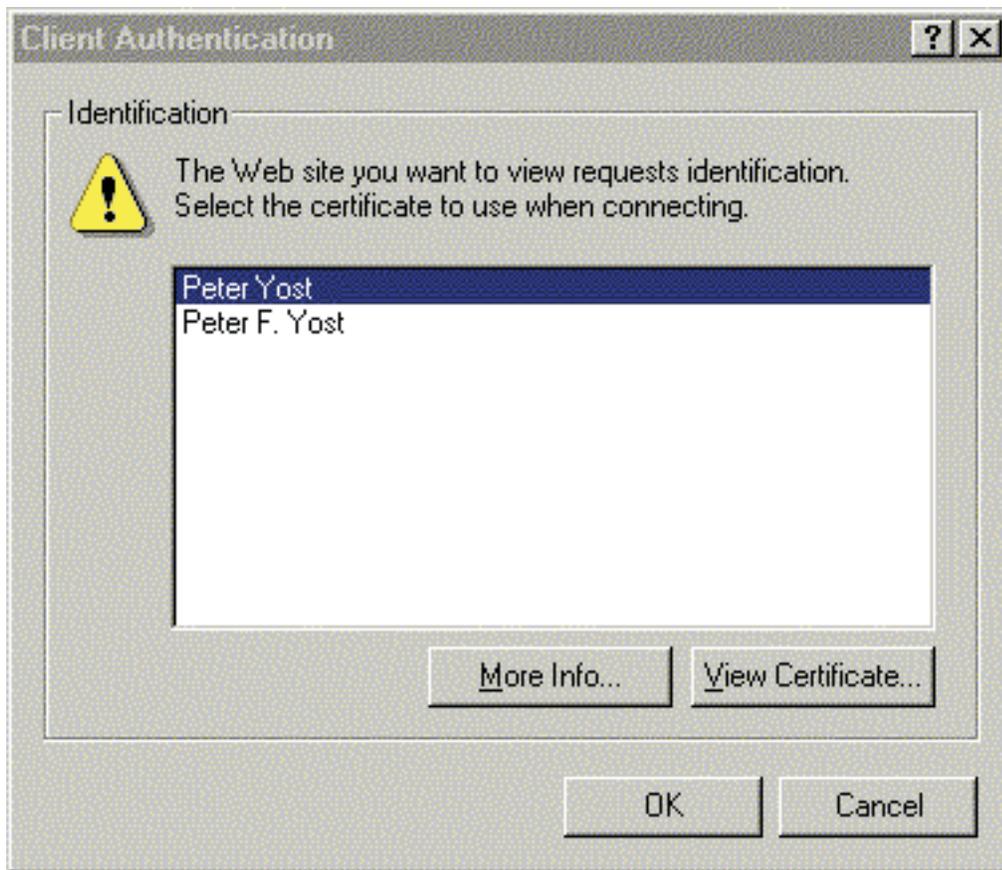


Figure 14. Popup window listing certificates

2. The popup window may list more than one certificate. It lists certificates by nicknames of how they are installed in the browser. Therefore, you may not be able to identify the PKI Services certificate you want to renew. Highlight the entry you think is the right one and click the **OK** button. If the certificate you selected is one that PKI Services issued and it is not expired or revoked, this displays the following Web page:

Renew or Revoke a Browser Certificate

Here is the certificate you selected:

Requestor	Certificate ID / Certificate Names/ Validity	Usage	Status	Dates
pfy@mycomp.com	Serial #:2 Template: 1 Year PKI SSL Browser Certificate	handshake	Active	Created: 2002/05/02
	Subject: MAIL=pfy@mycomp.com, CN=Peter Yost, OU=Class 1 Internet Certificate CA, O=The Firm Issuer: OU=PKI Department, O=IBM, C=us Validity: 2002/05/02 00:00:00 - 2003/05/01 23:59:59			Modified: 2002/05/02

If this is the correct certificate, choose one of the following:
(otherwise you need to restart your browser to pick another certificate)

- 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
- Revoke the above certificate**

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 15. Renew or revoke a certificate Web page

Notes:

- 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 169.
- If the certificate has the MAIL attribute in the subject's distinguished name, the value of NotifyEmail must match it.

Using the end-user Web pages

3. Under the "Renew the above certificate" section, enter your passphrase in the two fields requesting it.

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 the Web page from which you retrieve your certificate (see Figure 11 on page 166 for an example of this Web page and "Steps for retrieving your certificate from the bookmarked Web page" on page 167 for the directions to follow).

Steps for revoking a certificate

Revoking a certificate means that you cannot continue to use the certificate. You might want to revoke your certificate if you suspect your private key has been compromised.

Perform the following steps to revoke a certificate:

1. On the PKI Services home page (see Figure 7 on page 158), click the **Renew or revoke certificate** button. This displays a popup window with a list of certificates, as in Figure 14 on page 170.

2. The popup window may list more than one certificate. The way it lists certificates by nicknames of how they are installed in the browser. You may not be able to identify the PKI Services certificate you want to revoke. Highlight the entry you think is the right one and click the **OK** button. 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 169).

Note: If this is not the PKI Services certificate you want to revoke, you need to close your browser before again clicking the **Renew or revoke certificate** button as in Step 1 on page 169.

3. Make sure the certificate you want to revoke is the one described at the top of the Web page. You can click the drop-down list (of reasons) to select a reason if you wish. Click the **Revoke** button.

4. This displays a Web page that says "Request submitted successfully" You can click the **Home page** button to return to the PKI Services Home page.

Chapter 16. Using the administration Web pages

This chapter 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 certificate
 - Revoke a certificate
 - Delete a certificate
- Perform searches for certificate requests and certificates

Note: The PKI Services Web pages in this chapter may differ slightly from those on the Web. If you need to see the exact content, view the pages on the Web. Additionally, the pages may contain differences depending on the browser you are using. This chapter 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 home page (as shown in the following figure):

PKISERV Certificate Generation Application

[Install our CA certificate into your browser](#)

Choose one of the following:

- **Request a new certificate using a model**
Select the certificate template to use as a model
- **Pick up a previously requested certificate**
Enter the assigned transaction ID
Select the certificate return type
- **Renew or revoke a previously issued browser certificate**
- **Administrators click here**

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 16. PKI Services home page

2. If this is the first time you have accessed the forms on these Web pages, to install the CA certificate into your browser. Click the **Install our CA certificate into your browser** 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 our CA certificate into your browser** 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. This displays the following popup window:

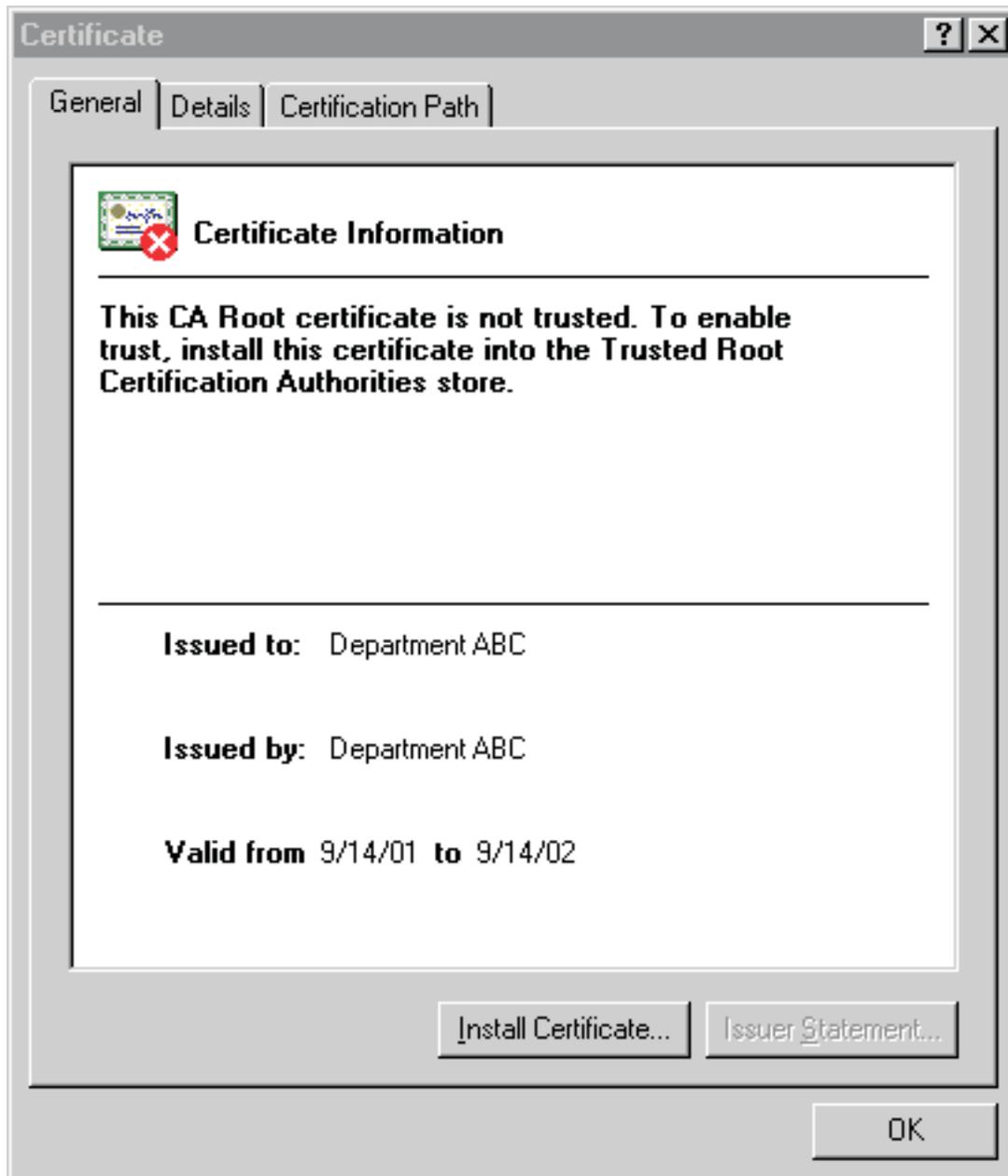


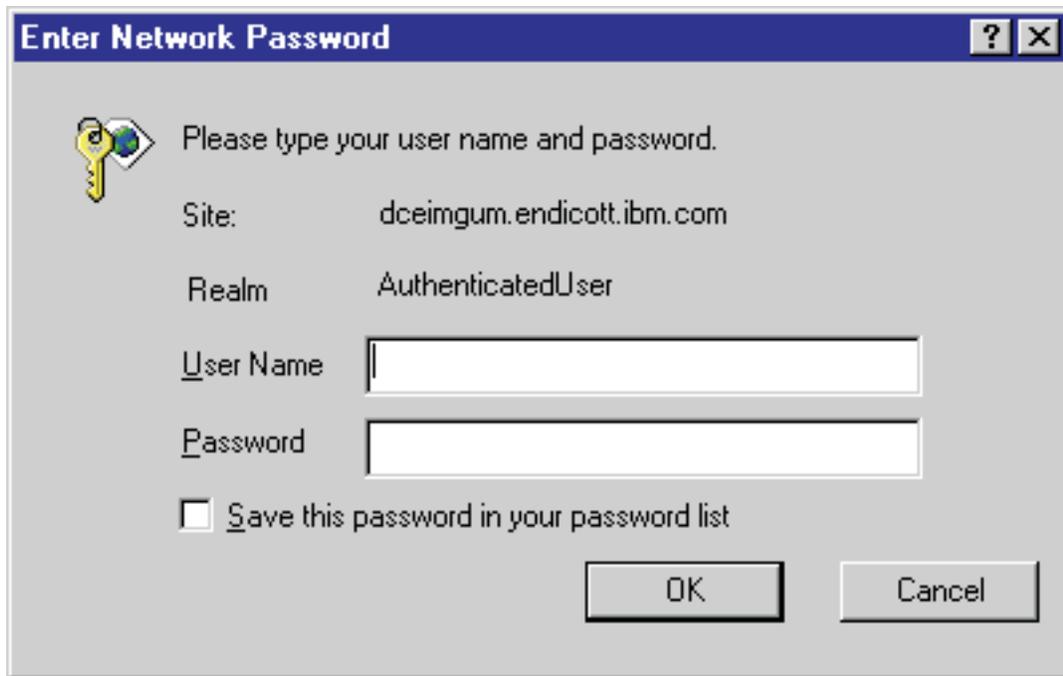
Figure 17. The Certificate popup window for installing the CA certificate

- b. Click the **Install certificate** button. (This initiates a sequence 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.")

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:

Using the administration Web pages



Enter Network Password ? X

 Please type your user name and password.

Site: dceimgum.endicott.ibm.com

Realm AuthenticatedUser

User Name

Password

Save this password in your password list

OK Cancel

Figure 18. Entering your user ID and password

- a. Fill in your z/OS user ID and password
- b. If you want to eliminate having to reenter your user ID and password each time you access the administration pages, check the check box.
- c. Click **OK**.
This calls up the "PKI Services administration" Web page. (See Figure 19 on page 179.)

Notes:

- a. Your Web server programmer may provide you with an alternate URL for accessing the administration home page. You may also have to authenticate using a certificate instead of a user ID and password.
 - b. 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 18 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 46. Summary of fields in the administration pages

Field	Description
Recent activity	This specifies a time range for searches. Possible values include: <ul style="list-style-type: none"> • Not selected • Within the past day • Within the past week • Within the past month • Within the past six months
Requestor name	The name of the person requesting the certificate, as it appears in the common name field of the certificate request form.
Serial number	PKI Services assigns this number to a certificate when you approve it.
Transaction ID	PKI Services assigns this number to a request when a user requests it. This is a text field of up to 56 characters.

Processing certificate requests

Before you can use the Web page to process certificate requests, you need to understand the statuses of certificate requests and the actions you can perform on these certificate requests.

Status of certificate requests

Requests for certificates are kept in a request database while they are active. This is from the moment they are created until an event occurs that causes them to be deleted. The following table summarizes possible statuses. During the time period when a certificate request is active, it can have only one of the following statuses at a time:

Table 47. Statuses of certificate requests

Status	Meaning
Pending Approval	The request requires administrative approval. No action has been taken on the request yet.
Approved	The administrator explicitly approved the request or it was submitted as an auto-approved certificate request. The actual certificate may or may not have been created at this point.
Completed	The certificate has been issued and the requestor has retrieved it. This is a final state.
Rejected	The administrator rejected the request, and the requestor has not been informed of this action (because the user has not tried to retrieve the certificate).
Rejected, User notified	The administrator rejected the request and the requestor has been informed of this action when attempting to retrieve the certificate. This is a final state.

Using the administration Web pages

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

Action	Required status of request
Approve	"Pending Approval"
Approve with modifications	
Reject	
Delete	All statuses ("Pending Approval," "Approved," "Completed," "Rejected," or "Rejected, Notified")

Using the PKI Services administration home page

The following figure shows the PKI Services administration home page:

PKI Services Administration

Choose one of the following:

- **Work with a single certificate request**

Enter the Transaction ID:
- **Work with a single issued certificate**

Enter the Serial Number:
- **Specify search criteria for certificates and certificate requests**

<p>Certificate Requests</p> <p><input type="radio"/> Show all requests</p> <p><input checked="" type="radio"/> Show requests pending approval</p> <p><input type="radio"/> Show approved requests</p> <p><input type="radio"/> Show completed requests</p> <p><input type="radio"/> Show all rejected requests</p> <p><input type="radio"/> Show rejections in which the client has been notified</p>	<p>Issued Certificates</p> <p><input type="radio"/> Show all issued certificates</p> <p><input type="radio"/> Show all revoked certificates</p> <p><input type="radio"/> Show all expired certificates</p> <p><input type="radio"/> Show non-expired non-revoked certificates only</p> <p><input type="radio"/> Show non-expired certificate revocations only</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Additional search criteria (Optional)

Requestor's name

Show recent activity only ▼

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 19. PKI Services administration home page

This Web page allows you to:

- Process a single certificate request (by specifying its transaction ID)
- Process a single certificate (by specifying its serial number)
- Search for groups of certificate requests or certificates by status and additional search criteria so that you can process them

Using the administration Web pages

You can process a single 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 19 on page 179), enter the transaction ID in the field provided for it, and click the **Process request** button. This displays the single request approval Web page as shown in the following figure:

Requestor	Certificate ID / Certificate Names / Validity	Usage	Status	Dates
pfy@mycomp.com	Trans ID: 1jCpqNeYqW0+2SHV+++++ Template: 1 Year PKI SSL Browser Certificate Subject: MAIL=pfy@mycomp.com,CN=Peter Yost,OU=Class 1 Internet Certificate CA,O=The Firm Issuer: OU=PKI Department,O=IBM,C=us Validity: 2002/05/02 00:00:00 - 2003/05/01 23:59:59	handshake	Pending Approval	Created: 2002/05/02 Modified: 2002/05/02

Passphrase

Action to take:

Action Comment (Optional)

email: webmaster@your-company.com

Figure 20. Single request approval Web page

2. Make sure the request is the correct one by reviewing the information in the top part of the Web page.
3. Optionally insert a comment.

4. Click one of the buttons to process the request:
 - **Approve the request as is** button
 - **Approve the request with modifications** button
 - **Reject request** button
 - **Delete request** button

Note: The buttons that appear on the Web page depend on the status of the request. For example, the top three buttons in the preceding list appear only if the status of a request is "Pending Approval." If the administrator has already processed the request, the only button that appears is the **Delete request** button.

- a. If 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:



Figure 21. Processing successful Web page

(Otherwise, the Web page says "Processing is not successful.") From these Web pages, you can then click the **Process more request(s)** button to return to the PKI Services administration home page (Figure 19 on page 179).

- b. If you click the **Approve the request with modifications** button, this displays the following Web page:

Using the administration Web pages

Modify and Approve Request

Requestor	Certificate ID	Dates
pfy@mycomp.com	Trans ID: 1jCpqNeYqW0+2SHV+++++++ Template: 1 Year PKI SSL Browser Certificate	Created: 2002/05/02 Modified: 2002/05/02

You may modify the following fields by providing new values. To remove a field simply blank it out.

Common Name

Email for distinguished name

Organizational Unit

Organizational Unit

Organization

Indicate the intended purpose for the certificate

- Protocol handshaking (e.g., SSL)
- Data encryption
- Certificate signing
- Document signing (nonrepudiation)

Figure 22. Modifying the request Web page (Part 1 of 2)

Date certificate becomes valid Date certificate expires (at end of day)

2002 5 2 2003 5 1

HostIdMappings Extension value(s) in subject-id@host-name form

Action Comment (Optional)

Approve with specified modifications

Reset Modified Fields

Administration Home Page

Home Page

email: webmaster@your-company.com

Figure 22. Modifying the request Web page (Part 2 of 2)

On this Web page, you can change the following fields:

- Common name
- Organizational unit(s) (This can be multiple fields)
- Organization
- E-mail address

Note: If you change the value of the E-mail address field (Email) and if the original request included the notification e-mail address field (NotifyEmail), the value of the latter field is changed to match the changed E-mail address value.

- Street
- Postal code
- Certificate purpose
- Date certificate becomes valid
- Date certificate expires
- HostIdMappings extensions (This can be multiple fields)
- Optional comment about action you perform on the certificate.

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 16 on page 174).

- c. If you click the **Reject request** button, this displays a Web page that informs you that "Processing is successful" or that "Processing is not successful."

Using the administration Web pages

From these Web pages, click the **Process more request(s)** button to return to the PKI Services administration home page (Figure 19 on page 179).

- d. If 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 19 on page 179).

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 49. Searches to display certificate requests

Search criteria	Results
Show all requests	Displays all certificate requests (all statuses: "Pending Approval," "Approved," "Completed," "Rejected," and "Rejected, User Notified").
Show requests pending approval	Displays only certificate requests whose status is "Pending Approval."
Show approved requests	Displays certificate requests whose status is "Approved" or "Completed."
Show completed requests	Displays certificate requests whose status is "Completed."
Show all rejected requests	Displays certificate requests whose status is "Rejected" or "Rejected, User Notified."
Show Rejections in which the client has been notified	Displays certificate requests whose status is "Rejected, User Notified."

To process requests by performing a search for requests of a particular status, perform the following steps:

1. On the PKI Services administration home page (see Figure 19 on page 179), select one of the searches by clicking the appropriate radio button under "Certificate Requests." (The preceding table describes these searches.) You can optionally fill in additional search criteria ("Requestor's name" and "Show recent activity only").
2. Click **Find certificates or certificate requests** button. This displays the following Web page:

Certificate Requests

The following certificate requests matched the search criteria specified:

Select <input checked="" type="checkbox"/>	Requestor	Certificate ID / Certificate Names / Validity	Usage	Status	Date
<input checked="" type="checkbox"/>	Dorothy Smith	Trans ID: 1jxUaJshBbEeVknDWBrf3lo+ Template: 1 Year PKI SSL Browser Certificate Subject: CN=Dorothy Smith,OU=Class 1 Internet Certificate CA,O=The Firm Issuer: OU=Department ABC,O=Company Inc.,L=Anywhere City,C=zz Validity: 2001/09/14 00:00:00 - 2002/09/13 23:59:59	handshake	Pending Approval	Created: 2001/09/14 Modified: 2001/09/14
<input checked="" type="checkbox"/>	Peter Yost	Trans ID: 1jxUaQwaY8scVknDWBrf3lo+ Template: 1 Year PKI SSL Browser Certificate Subject: CN=Peter Yost,OU=Class 1 Internet Certificate CA,O=The Firm Issuer: OU=Department ABC,O=Company Inc.,L=Anywhere City,C=zz Validity: 2001/09/14 00:00:00 - 2002/09/13 23:59:59	handshake	Pending Approval	Created: 2001/09/14 Modified: 2001/09/14

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 without modification all requests selected above that are "Pending Approval"

- Reject all requests selected above that are "Pending Approval"

- Delete all requests selected above

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 23. Processing requests after searching

Note: The table at the top of the Web page shows the certificate requests that match your search criteria. (If multiple certificates requests match the search criteria, up to ten appear on a Web page, and a button at the bottom of the Web 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:

- 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 20 on page 180.
- From the single request Web page, you can perform the steps in the preceding section, starting with Step 2 on page 180).

To perform the same action on all the certificate requests that are listed:

- Optionally enter a comment.

Using the administration Web pages

- 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 19 on page 179) 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 19 on page 179) or the **Home page** button to return to the PKI Services home page (see Figure 16 on page 174).

-
4. After you click an action button, the next Web page is one of the following:
 - Processing successful (see Figure 24 on page 187)
 - Processing was not successful (see Figure 25 on page 187)
 - Processing partially successful (see Figure 26 on page 188)

If "Processing was not successful," you can click on the transaction ID to display the 'Single Request' Web page; see Figure 20 on page 180. Processing can be unsuccessful because requests do not have the status required for the action you selected; see Table 48 on page 178.

If you get "Processing partially successful," you can click on the transaction ID to display the 'Single Request' Web page; see Figure 20 on page 180. This message can occur when your organization has more than one administrator and it involves the following sequence:

- One administrator performs a search

Using the administration Web pages

- Another administrator performs a search before the first administrator has approved requests displayed in the search results
- One of the administrators approves only some of the requests
- The other administrator tries to approve requests including at least one the preceding administrator has already approved and one that the preceding administrator has not already approved.

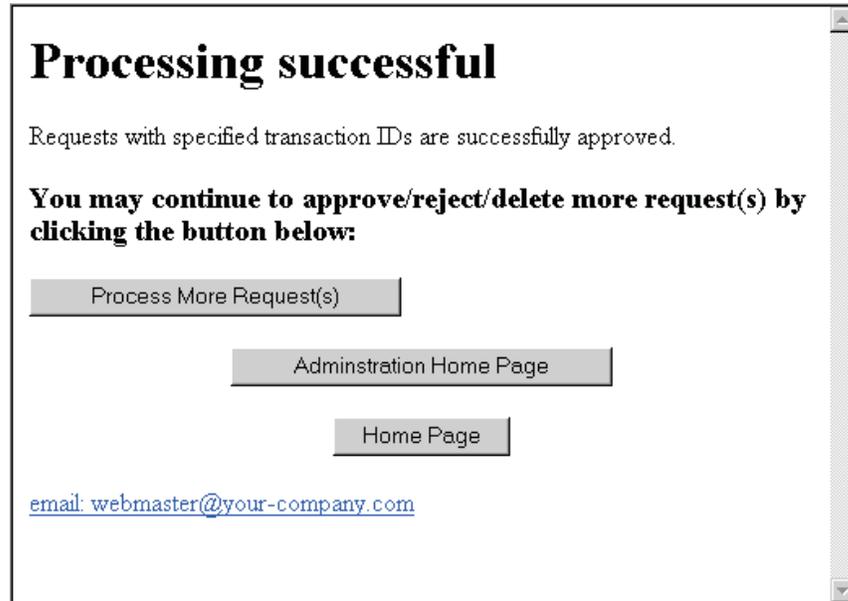


Figure 24. Request processing was successful Web page

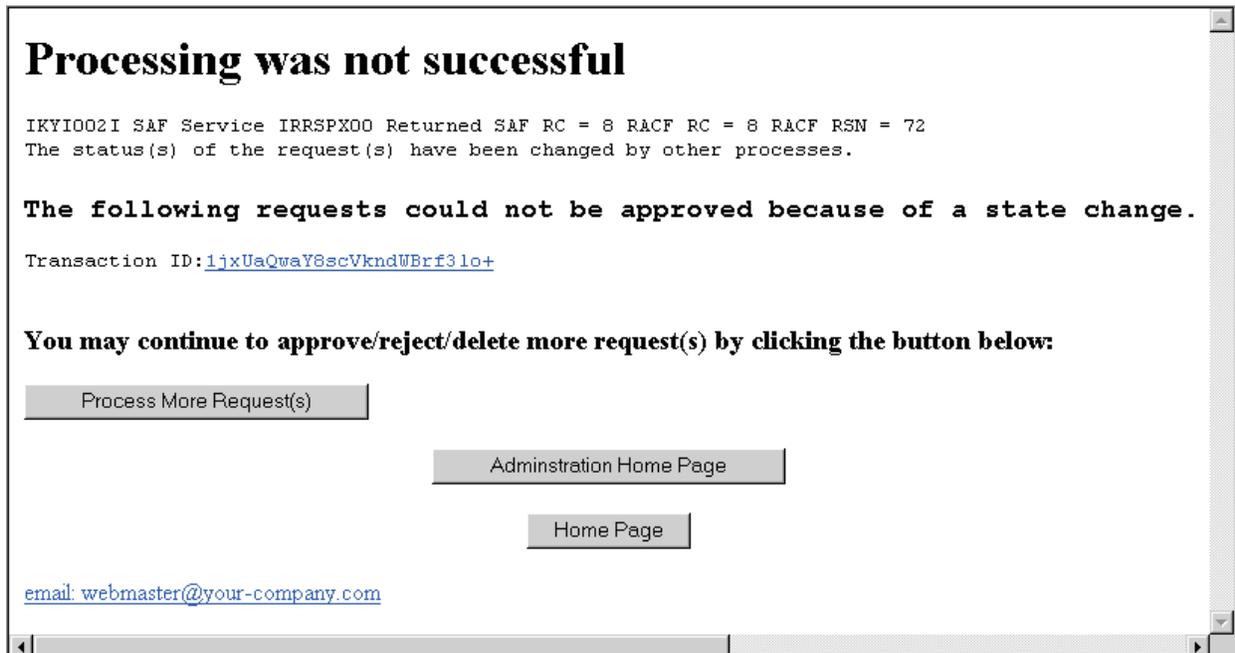


Figure 25. Request processing was not successful Web page

Using the administration Web pages



Figure 26. Request processing was partially successful Web page

5. After approving requests as appropriate, you can:
 - Click **Process more request(s)** to return to Figure 23 on page 185
 - Click **Administration home page** to return to Figure 16 on page 174
 - Click **Home page** to return to Figure 19 on page 179.

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 50. Status of certificates

Active	The certificate has not yet expired and has not been revoked.
Expired	The certificate has not been revoked but has expired.
Revoked	The certificate has not expired but it has been revoked. Such certificates are published on the next certificate revocation list (CRL).

Table 50. Status of certificates (continued)

Revoked, Expired The certificate was revoked and time has elapsed such that it has expired too. Such certificates would not be published on the next CRL.

The administrator must approve a request for the certificate to have a status (as enumerated in the preceding list) or for the administrator to delete the certificate from the ICL. (An administrator can delete a certificate from the ICL, but this would not be a normal situation.) Alternately, the administrator can reject a request or delete the request from the request database (RDB). If the administrator does not approve the request, it is never listed in the ICL.

Actions for certificates

The following table summarizes actions on certificates and the required status to perform these actions:

Table 51. Summary of actions to perform and required status to do so

Action	Required status of certificate	Who performs action
Renew	"Active"	End user
Revoke	"Active"	End user or administrator
Delete	All ("Active," "Expired," "Revoked," or "Revoked, Expired")	Administrator

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 19 on page 179), enter the serial number of the certificate you want to process in the field provided for it. This displays the following Web page:

Using the administration Web pages

Single Issued Certificate

Requestor	Certificate ID / Certificate Names / Validity	Usage	Status	Dates
Peter Yost	Serial #:2 Template: 1 Year PKI SSL Browser Certificate Subject: CN=Peter Yost,OU=Class 1 Internet Certificate CA,O=The Firm Issuer: OU=Department ABC,O=Company Inc.,L=Anywhere City,C=zz Validity: 2001/08/20 00:00:00 - 2002/08/19 23:59:59	handshake	Active	Created: 2001/08/20 Modified: 2001/08/20

Previous Action Comment: Issued certificate

Action to take:

Action Comment (Optional)

Revoke Certificate | No Reason

Delete Certificate

email: webmaster@your-company.com

Figure 27. Processing a certificate from the single certificate Web page

2. Make sure the certificate is the correct one by reviewing the information in the top part of the Web page.
3. If you are going to process a certificate from this Web page, you can optionally insert a comment.
4. Click one of the following buttons to process the certificate:

Revoke certificate	Revokes the certificate.
Delete certificate	Deletes the certificate. (This is for cleanup purposes.)

Note: The **Revoke** button appears only if the status of the certificate is Active.

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 52. Searches to display certificates

Searches	Results
Show all issued certificates	Displays all certificates (can be any status — "Active," "Expired," "Revoked," or "Revoked, Expired").
Show all revoked certificates	Displays certificate requests whose status is "Revoked" or "Revoked, Expired."
Show all expired certificates	Displays certificates whose status is "Expired" or "Revoked, Expired."
Show non-expired, non-revoked certificates only	Displays certificates whose status is "Active."
Show non-expired certificate revocations only	Displays certificate requests whose status is "Revoked."

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 19 on page 179), 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.

Using the administration Web pages

Issued Certificates

The following issued certificates matched the search criteria specified:

Select <input checked="" type="checkbox"/>	Requestor	Certificate Names / Validity	Usage	Status	Date
<input checked="" type="checkbox"/>	Peter Yost	Serial #: 3 Template: 1 Year PKI SSL Browser Certificate Subject: CN=Peter Yost,OU=Class 1 Internet Certificate CA,O=The Firm Issuer: OU=Department ABC,O=Company Inc.,L=Anywhere City,C=zz Validity: 2001/12/06 00:00:00 - 2002/12/05 23:59:59	handshake	Active	Created: 2001/12/06 Modified: 2001/12/06
<input checked="" type="checkbox"/>	Dorothy Smith	Serial #: 4 Template: 1 Year PKI SSL Browser Certificate Subject: CN=Dorothy Smith,OU=Class 1 Internet Certificate CA,O=The Firm Issuer: OU=Department ABC,O=Company Inc.,L=Anywhere City,C=zz Validity: 2001/12/06 00:00:00 - 2002/12/05 23:59:59	handshake	Active	Created: 2001/12/06 Modified: 2001/12/06

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)

[email_webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 28. Processing certificates using searches

Note: The table at the top of the Web page shows the certificates that match your search criteria. (If multiple certificates match the search criteria, up to ten appear on a Web page, and a button at the bottom of the Web page allows you to view the next set.)

3. You can use this Web page:
 - To process a single certificate
 - To perform the same action on all of the certificates that are listed
 - To process selected certificates

To process a single certificates:

- 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 27 on page 190.
- b. From the single certificate Web page, you can perform the steps in the preceding section, starting with Step 2 on page 190).

To perform the same action on all the certificates 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 certificates:

Using the administration Web pages

Revoke all selected active certificates

Revokes the certificates.

Delete all selected certificates

Deletes the certificates.

Note: For the **Revoke** button to appear, your search must match at least one certificate whose status is active.

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 all selected active certificates

Revokes the certificates.

Delete all selected certificates

Deletes the certificates.

Note: For the **Revoke** button to appear, your search must match at least one certificate whose status is active.

Instead of processing one or more certificates, you can click the **Respecify your search criteria Web page** button to return to the PKI Services administration home page (see Figure 19 on page 179) or the **Home page** button to return to the PKI Services home page (see Figure 16 on page 174).

-
4. After you click an action button, the next Web page tells you:
 - "Processing was successful" (see Figure 29 on page 194)
 - "Processing was not successful" (see Figure 30 on page 194)
 - "Processing partially successful" (see Figure 31 on page 195)

If "Processing was not successful," you can click on a serial number to display the "Single Certificate" Web page; see Figure 27 on page 190. Processing can be unsuccessful because certificates do not have the status required for the action you selected; see Table 51 on page 189.

If you get "Processing partially successful," you can click on the serial number to display the "Single Certificate" Web page; see Figure 27 on page 190. The "Processing partially successful" message can occur when your organization has more than one administrator and it involves the following sequence:

- One administrator performs a search
- Another administrator performs a search before the first administrator has revoked or deleted certificates displayed in the search results
- One of the administrators revokes or deletes some of the certificates
- 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.

Using the administration Web pages

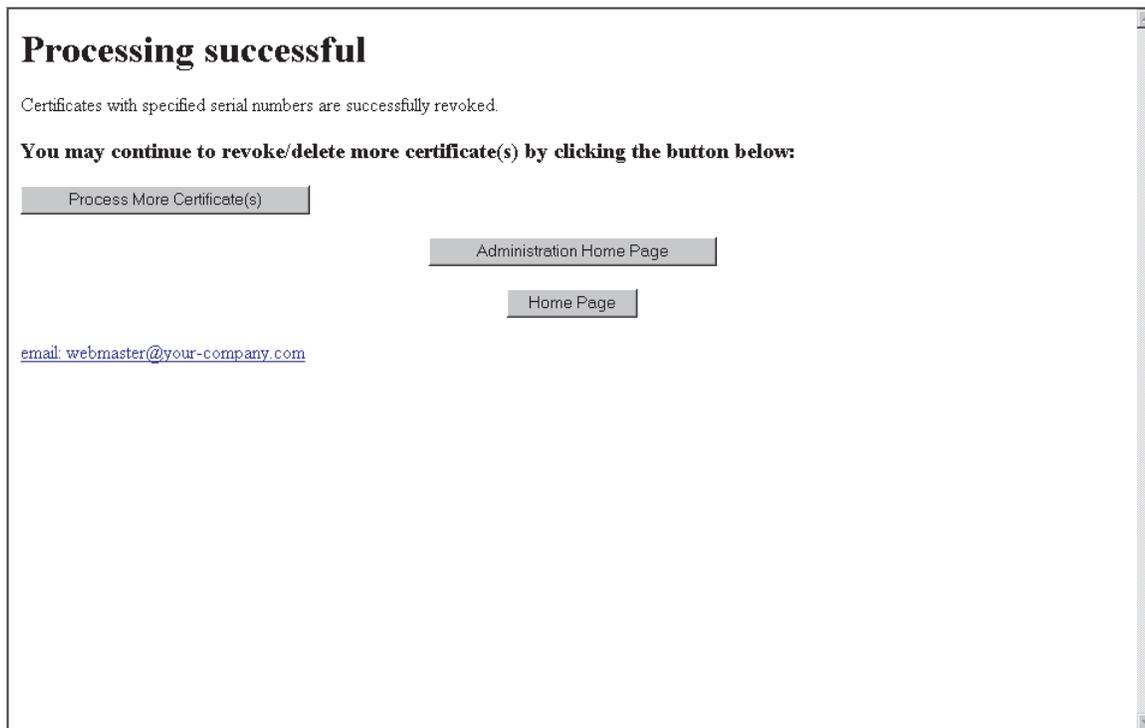


Figure 29. Processing of certificate was successful Web page



Figure 30. Request processing was not successful Web page



Figure 31. Request processing was partially successful Web page

You can click the **Home page** button to return you to the PKI Services home page (see Figure 16 on page 174).

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 may 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 unapprove 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.

Using the administration Web pages

- 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.

Part 5. Administering RACF for PKI Services

This part explains how to perform the following tasks:

- Authorizing users for the PKI Services administration group (connecting and deleting members)
- Authorizing users for inquiry access
- Administering HostIdMappings extensions
- Locating your PKI Services certificate and key ring
- Establishing PKI Services as an intermediate certificate authority
- Renewing your PKI Services certificate authority certificate
- Recovering a CA certificate profile
- Controlling applications that call R_PKIServ
- Using encrypted LDAP passwords.

I

Chapter 17. RACF administration for PKI Services

This chapter describes the tasks that the RACF administrator performs after PKI Services has been set up and customized.

The following topics are covered:

- “Steps for creating a CA certificate using the PCICC”
- “Authorizing users for the PKI Services administration group” on page 200
- “Authorizing users for inquiry access” on page 200
- “Administering HostIdMappings extensions” on page 201
- “Locating your PKI Services certificate and key ring” on page 203
- “Establishing PKI Services as an intermediate certificate authority” on page 204
- “Renewing your PKI Services certificate authority certificate” on page 206
- “Recovering a CA certificate profile” on page 207
- “Controlling applications that call R_PKIServ” on page 211
- “Using encrypted passwords for LDAP servers” on page 214.

For more information about the RACF commands shown in this chapter, see *z/OS Security Server RACF Command Language Reference*.

Creating a CA signing key pair using hardware

You can use the 4758 coprocessor (called the PCICC) to generate your CA's signing public-private key pair. The keys can be up to 1024 bits in length. The generated private key is stored back into the PKDS. The advantages of using the PCICC are that key generation is faster and the key is encrypted and always contained within the bounds of ICSF. However, you cannot back up such keys using RACDCERT. (For information about backing up the entire PKDS, see *z/OS ICSF Administrator's Guide*.)

To generate RSA keys using the PCICC, use the RACDCERT GENCERT command with the PCICC operand. ICSF must be active and configured to use the 4758. If the 4758 is not present or not operational, the command stops and an error message is issued.

Steps for creating a CA certificate using the PCICC

Before you begin: ICSF must be active and configured to use the 4758. (See “Installing and configuring ICSF (optional)” on page 32 for information about installing and configuring ICSF.)

Perform the following steps to create a CA certificate using the PCICC:

1. Enter a RACDCERT command that includes the PCICC parameter, such as the following:

```
RACDCERT GENCERT CERTAUTH PCICC SUBJECTSDN(OU('Human Resources Certificate Authority ')
O('Your Company ')C('Your Country 2 Letter Abbreviation '))
WITHLABEL('Local PKI CA ')NOTAFTER(DATE(2020/01/01))
```

Make sure to fill in your own information for:

- 'Human Resources Certificate Authority' — The name of the organizational unit
- 'Your Company' — The name of your organization

RACF administration for PKI Services

- 'Your Country 2 Letter Abbreviation'
- 'Local PKI CA' — The label you want to use
- 2020/01/01 — The date up to which the certificate will be valid.

2. Run IKYSETUP, making sure that the *ca_dn* parameter in IKYSETUP is set to null and the *ca_label* parameter in IKYSETUP is set to the label you specified in the preceding RACDCERT command. See Table 12 on page 39 and “Steps for performing RACF tasks using IKYSETUP” on page 47.

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, enter 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 18 on page 46 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, enter 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 PKISRV default. (See Table 18 on page 46.)

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))
```

RACF administration for PKI Services

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 on page 200.

Example:

```
DELGROU(HELPDESK)
```

Administering HostIdMappings extensions

You can add a HostIdMappings extension to certificates you create for certain users, allowing you to specify the user IDs that each user will be able to use for login to particular servers (or hosts). Controlling an identity used for login purposes is a very important security objective. Therefore, you must exercise administrative control in the following areas by authorizing:

- PKI Services as a highly trusted certificate authority whose certificates will be honored when they contain HostIdMappings extensions
- Particular servers to accept logins from clients whose certificates contain HostIdMappings extensions

Steps for administering HostIdMappings extensions

Perform the following steps to allow the Web server to accept logins from clients who have been issued PKI Services certificates with HostIdMappings extensions:

1. Determine if PKI Services is defined as a highly trusted certificate authority on your system by listing its certificate authority definition by using the RACDCERT CERTAUTH LIST command.

Example:

```
RACDCERT CERTAUTH LIST(LABEL('Local PKI CA'))
```

RACF administration for PKI Services

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 HostIdMapping is not honored if the target user ID was created after the start of the validity period for the certificate containing the HostIdMappings extension. Therefore, if you are creating user IDs specifically for certificates with HostIdMappings extensions, make sure that you create the user IDs before the certificate requests are submitted. Alternately, when approving the certificate, you can modify the date the certificate becomes valid so that it is not earlier than the date the user ID was created. For renewed certificates, all the original information is replicated in the new certificate, including the date the certificate becomes valid and any HostIdMappings. If you want to change a HostIdMapping 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 certificate and key ring

The IKYSETUP exec sets up the RACF environment for PKI Services. After the set up is complete, you may need to go back and locate the PKI Services certificate or key ring, 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 53. Information you need for locating your PKI Services certificate and key ring

Information needed	Where to find this information	Record your value here
ca_label - Label of your CA certificate in RACF	See Table 12 on page 39.	
ca_ring - PKI Services SAF key ring	See Table 18 on page 46.	
daemon - User ID for the PKI daemon	See Table 18 on page 46.	
log_dsn - Data set name of the IKYSETUP log	See Table 18 on page 46.	
export_dsn - Data set name of your CA certificate as exported from RACF	See Table 18 on page 46.	

Steps for locating the PKI Services certificate and key ring

Perform the following steps to locate the PKI Services certificate and key ring:

1. Locate the certificate by using one of the following two commands. To locate the certificate in RACF, using the export data set containing the certificate as saved by IKYSETUP, enter the following RACF command from a TSO command prompt:

```
RACDCERT CHECKCERT(export_dsn)
```

The output should be something like the following:

```
Digital certificate information for CERTAUTH:
Label: Local PKI CA
Certificate ID: 2QiJmZmDhZmjgdOWg4GTQNFsyUDDwUBA
Status: HIGHTRUST
Start Date: 2001/06/04 23:00:00
End Date: 2020/01/01 22:59:59
Serial Number:
>00<
Issuer's Name:
>OU=Human Resources Certificate Authority.0=IBM.C=US<
Subject's Name:
>OU=Human Resources Certificate Authority.0=IBM.C=US<
Key Usage: CERTSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
```

It is important to note the following if diagnosing errors:

- The first line must indicate that this is a CERTAUTH certificate.
- The Label must match your ca_label value (as in the preceding table).
- The Subject's Name must match the original value recorded for the PKI Services SUBJECTSDN in the IKYSETUP log.

RACF administration for PKI Services

- The Private Key Type and Size must be present.
- If the Issuer's Name differs from the Subject's Name, this indicates that the certificate was issued by another certificate authority.
- If the Serial Number is not equal to 00, this indicates that the certificate has been renewed or was issued by another certificate authority.

Alternately, you can locate the certificate directly by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH LIST(LABEL('ca_label'))
```

This should produce the same information as the preceding. In addition, any ring associations are also displayed:

```
Ring Associations:  
  Ring Owner: PKISRVD  
  Ring:  
    >CAring<
```

In this information, you should ensure that one of the associations listed has the daemon user ID as the owner and that the ring name matches your `ca_ring` value (as listed in the preceding table).

-
2. Examine the CA key ring. To do so, from a TSO command prompt, enter the following RACF command:

```
RACDCERT ID(daemon) LISTRING(ca_ring)
```

This command should produce information such as the following:

Digital ring information for user PKISRVD:

```
Ring:  
  >CAring<  
Certificate Label Name          Cert Owner    USAGE        DEFAULT  
-----  
Local PKI CA                   CERTAUTH     PERSONAL     YES
```

The entry for the PKI Services CA certificate must have USAGE PERSONAL and DEFAULT YES.

Establishing PKI Services as an intermediate certificate authority

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 a intermediate (subordinate) certificate authority. An intermediate certificate authority is one whose certificate is signed by another higher certificate authority. This higher certificate authority may be a root CA or another intermediate CA. If the root CA certificate has previously been trusted, any lower intermediate CA certificate can be verified using the higher certificate.

Steps for establishing PKI Services as an intermediate CA

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 on the blank lines:

Table 54. Information you need for establishing PKI Services as an intermediate CA

Information needed	Where to find this information	Information values
<i>ca_label</i> - This is the label of your CA certificate in RACF	See Table 12 on page 39.	_____
<i>cert_dsn</i> - This is name of the data set to contain your new certificate request and returned certificate.	You decide this based on local data set naming conventions.	_____
<i>export_dsn</i> - This is the data set name of your CA certificate as exported from RACF.	See Table 18 on page 46.	_____

Perform the following steps to establish PKI Services as an intermediate certificate authority:

1. If you have not yet configured your system for PKI Services, then perform all required steps in Chapter 5, “Running IKYSETUP to perform RACF administration” on page 37.

2. 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.)

3. Create a new certificate request from your self-signed CA certificate by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH GENREQ(LABEL('ca_label')) DSN(cert_dsn)
```

4. Send the certificate request to the higher certificate authority, following the procedures that higher authority requires.

5. After the certificate has been issued, receive the certificate back into the certificate data set (*cert_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:
 - a. Delete all existing lines in *cert_dsn*.
 - b. Copy the base64 encoded text.
 - c. Paste this into the ISPF edit window.
 - d. 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.

RACF administration for PKI Services

6. Receive the certificate back into the RACF data base by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH ADD(cert_dsn)
```

7. Export the certificate in DER format to the export data set by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH EXPORT(LABEL('ca_label')) DSN(export_dsn) FORMAT(CERTDER)
```

8. To make your new certificate available to your clients, set up the `/var/pkiserv/directory` by performing step 2 on page 67 through step 4 on page 67 in “Steps for setting up the `/var/pkiserv` directory” on page 67.

Renewing your PKI Services certificate authority certificate

Eventually, your PKI Services CA certificate will expire. To avoid complications related to an expired CA certificate, you should renew the certificate before it actually expires.

Note: You will receive MVS console message IKYP026E as the expiration date approaches.

Steps for renewing your PKI Services certificate authority 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 on the blank lines:

Table 55. Information you need for renewing your PKI Services certificate authority certificate

Information needed	Where to find this information	Information values
<i>ca_label</i> - This is the label of your CA certificate in RACF	See Table 12 on page 39.	_____
<i>cert_dsn</i> - This is name of the data set to contain your new certificate request and returned certificate.	You decide this based on local data set naming conventions.	_____
<i>export_dsn</i> - This is the data set name of your CA certificate as exported from RACF.	See Table 18 on page 46.	_____

Perform the following steps to renew your PKI Services CA certificate:

1. Create a new certificate request from your self-signed CA certificate by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH GENREQ(LABEL('ca_label')) DSN(cert_dsn)
```

2. If your PKI Services certificate authority is a root CA (that is, it has a self-signed certificate, which is the default), then generate the self-signed renewal certificate by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH GENCERT(cert_dsn) 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 certificate authority, following the procedures that higher authority requires.
 - b. After the certificate has been issued, receive the certificate back into the certificate data set (*cert_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 *cert_dsn*.
 - 2) Copy the base64 encoded text.
 - 3) Paste this 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. Receive the certificate back into the RACF data base by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH ADD(cert_dsn)
```

4. Export the certificate in DER format to the export data set by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH EXPORT(LABEL('ca_label')) DSN(export_dsn) FORMAT(CERTDER)
```

5. To make your new certificate available to your clients, set up the */var/pkiserv/directory* by performing steps 2 on page 67 through 4 on page 67 in “Steps for setting up the */var/pkiserv* directory” on page 67.

Recovering a CA certificate profile

Unless you change the IKYSETUP REXX exec to disable the function, IKYSETUP automatically backs up the PKI Services CA certificate and private key to a passphrase-encrypted data set that has PKCS#12 format. If the 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 on the blank lines:

Table 56. Information you need for recovering a CA certificate profile

Information needed	Where to find this information	Information values
<i>backup_dsn</i> - The name of the data set containing the backup copy of your private key.	See Table 18 on page 46.	_____

RACF administration for PKI Services

Table 56. Information you need for recovering a CA certificate profile (continued)

Information needed	Where to find this information	Information values
<i>ca_label</i> - This is the label of your CA certificate in RACF	See Table 12 on page 39.	_____
<i>ca_ring</i> - The PKI Services SAF key ring.	See Table 18 on page 46.	_____
<i>cert_dsn</i> - This is name of the data set to contain your new certificate request and returned certificate.	You decide this based on local data set naming conventions.	_____
<i>daemon</i> - The user ID for the PKI daemon.	See Table 18 on page 46.	_____
<i>export_dsn</i> - This is the data set name of your CA certificate as exported from RACF.	See Table 18 on page 46.	_____
<i>your-passphrase</i> - The pass phrase you used when backing up the private key.	You specified this when running IKYSETUP.	_____

Perform the following steps to recover a CA certificate profile:

1. Enter the following TSO commands:

```
RACDCERT CERTAUTH ADD(backup_dsn) PASSWORD(your-passphrase)
    WITHLABEL('ca_label') ICSF
RACDCERT CERTAUTH ADD(export_dsn)
RACDCERT ID(daemon) CONNECT(CERTAUTH LABEL('ca_label')
    RING(ca_ring) USAGE(PERSONAL) DEFAULT)
```

Note: If you are not using ICSF, omit the ICSF keyword on the ADD.

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 incremter value that is stored in the profile because otherwise, PKI Services resumes issuing certificates starting from serial number 1.)
 - a. Make sure PKI Services is stopped. (See “Stopping the PKI Services daemon” on page 86 for details on how to do this.)
 - b. Enter the following command from the UNIX command line to run the iclview utility:

```
iclview \pkisrzd.vsam.icl\'
```

Record the serial number displayed (in hex) of the last certificate listed:

Serial number (in hex) of last certificate:	
----------------------------------------------------	--

- c. To determine your CA certificate’s profile name, issue the following command to perform an *unsuccessful* ADD:

```
RACDCERT CERTAUTH ADD(export_dsn) WITHLABEL('*** Bad Label ***')
```

The unsuccessful ADD displays an error message including the profile name. Record the profile name:

Profile name:	
----------------------	--

RACF administration for PKI Services

- d. Create the following ICHEINTY ALTER job in your JCL data set, replacing the highlighted values based on the information you recorded in the previous steps:

RACF administration for PKI Services

```

//SAMPIC JOB 'xxxxxxx',NOTIFY=xxxxxx,
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
// REGION=4M
//*****
//ASM EXEC ASMHCL,PARM.C='OBJECT,DECK,TEST',
// PARM.L='MAP,LET,LIST,NCAL,AC(1)'
//C.SYSIN DD *
SAMPICHE CSECT
SAMPICHE AMODE 31
SAMPICHE RMODE ANY
        STM R14,R12,12(R13) Save registers
        BALR R12,0
        USING *,R12 R12 = base register
        B DOIT
*****
* Update the following declares with your certificate info *
*****
ENTRY EQU * Your CA certificate profile
ENTBLEN DC H'54' Length of cert profile name
ENTALEN DC H'54' Length of cert profile name
        DC CL43'00.00=HumanResourcesCertificateAuthority'
        DC CL11'.0=IBM.C=US'
LSER EQU * CERTLSER is an 8 byte field
LSERHIGH DC X'00000000' High word - set to zero
LSERLOW DC X'000000FF' Set to your last serial # (hex)
*****
* Establish standard linkage *
*****
DOIT ST R13,SAVE+4 Save caller's save area address
     LA R15,SAVE Get the next save area address
     ST R15,8(R13) Link the save areas
     LR R13,R15 R13 points to next save area
     ICHEINTY ALTER,TYPE='GEN',ENTRYX=ENTRY,RELEASE=1.9, X
         SEGMENT='CERTDATA',CLASS=DIGTCLAS, X
         ACTIONS=(A_LSER)
CLOSEUP EQU *
        L R13,SAVE+4 Get caller's save area address
        ST R15,16(R13) Save ICHEINTY RC
        LM R14,R12,12(R13) Restore registers except R13
        BR R14 Back to invoker
*****
* CONSTANTS, SAVE AREAS, ETC *
*****
SAVE DS 18F
DIGTCLAS DC CL8'DIGTCERT'
        ORG
A_LSER ICHEACTN FIELD=CERTLSER,FLDATA=(8,LSER),RELEASE=1.9,MF=L
*****
* General Equates *
*****
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8

```

Figure 32. Sample JCL data set for restoring the certificate serial number incrementer value (Part 1 of 2)

```

R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
      END SAMPICHE
//C.SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DSN=SYS1.MODGEN,DISP=SHR
//C.SYSPRINT DD SYSOUT=*
//L.SYSLMOD DD DSN=SYS1.LINKLIB(SAMPICHE),DISP=SHR
//L.SYSPRINT DD SYSOUT=*
//L.SYSIN DD *
      NAME SAMPICHE(R)
/*
//RUNIT EXEC PGM=SAMPICHE
/*

```

Figure 32. Sample JCL data set for restoring the certificate serial number incrementer value (Part 2 of 2)

- e. Submit the job and check its return code.

Controlling applications that call R_PKIServ

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. See *z/OS Security Server RACF Callable Services* for details of invoking IRRSPX00.

You authorize these applications by administering RACF profiles in the FACILITY class, based on whether the application requests end-user functions or administrative functions.

R_PKIServ end-user functions

The end-user functions are:

EXPORT	Retrieves (exports) a previously requested certificate.
GENCERT	Generates an auto-approved certificate.
GENRENEW	Generates an auto-approved renewal certificate.
	Note: The request submitted is automatically approved.
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.
REVOKE	Revokes a certificate that was previously issued.
VERIFY	Confirms that a given user certificate was issued by this CA and, if so, returns the certificate fields.

RACF administration for PKI Services

For end-user functions, FACILITY class profiles protect this interface. The form of the FACILITY class profiles is:

IRR.RPKISERV.<function>

<function>

Is one of the following end-user function names in the preceding list. The user ID for the application (user ID from the ACEE associated with the address space) is used to determine access:

- NONE** Access is denied.
- READ** Access is permitted based on subsequent access checks against the caller's user ID. To determine 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.
- UPDATE** Access is permitted based on subsequent access checks against the application's user ID.
- CONTROL (or user ID is RACF SPECIAL)**
Access is permitted, and no subsequent access checks are made.

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 profiles. 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, REQCERT, REQRENEW, REVOKE, and VERIFY requests in which the application has READ and UPDATE access, subsequent access checks are performed against the IRR.DIGTCERT.<function> FACILITY profiles. The following table summarizes the access requirements for the user ID whose access is checked:

Table 57. Summary of accesses required for PKI Services request

Request	Access
EXPORT	<ul style="list-style-type: none"> • IRR.DIGTCERT.EXPORT <ul style="list-style-type: none"> – UPDATE access if no pass phrase is specified on the call – READ access if a pass phrase is specified.
GENCERT	<ul style="list-style-type: none"> • IRR.DIGTCERT.GENCERT — CONTROL access • IRR.DIGTCERT.ADD <ul style="list-style-type: none"> – UPDATE access if any HostIdMappings information is specified in the certificate request parameter list or the UserId field in the certificate request parameter list indicates the certificate is being requested for another user other than the caller – READ access otherwise
GENRENEW	<ul style="list-style-type: none"> • IRR.DIGTCERT.GENRENEW — READ access • IRR.DIGTCERT.GENCERT — CONTROL access <p>Note: It is assumed that the calling application has already verified the input certificate using the VERIFY function.</p>
REQCERT	<ul style="list-style-type: none"> • IRR.DIGTCERT.REQCERT — READ access

Table 57. Summary of accesses required for PKI Services request (continued)

Request	Access
REQRENEW	<ul style="list-style-type: none"> • IRR.DIGTCERT.REQRENEW — READ access <p>Note: It is assumed that the calling application has already verified the input certificate using the VERIFY function.</p>
REVOKE	<ul style="list-style-type: none"> • IRR.DIGTCERT.REVOKE — READ access <p>Note: It is assumed that the calling application has already verified the target certificate using the VERIFY function.</p>
VERIFY	<ul style="list-style-type: none"> • IRR.DIGTCERT.VERIFY — READ access <p>Note: It is assumed that the calling application has already verified that the end user possesses the private key that correlates to the input certificate.</p>

R_PKIServ 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 detail information about one PKI Services certificate request.

For the administrative functions, a single FACILITY class profile — IRR.RPKISERV.PKIADMIN — protects this interface:

- 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, (MODIFYREQS and MODIFYCERTS).

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 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.

Recommendation: Give UPDATE authority only to those individuals whom you would trust with the RACF SPECIAL attribute. If you do assign PKI Services administrators who do not have the RACF SPECIAL attribute, do not also give these individuals direct access to the end-user functions of the R_PKIServ callable service as described in the previous section.

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 79.)

Before creating either of the preceding profiles, the RACF administrator defines the `LDAP.BINDPW.KEY` profile in the `KEYSMSTR` class. This profile contains a `SSIGNON` segment, which holds either the masked or encrypted value for the key that encrypts passwords stored in the RACF database. Then the RACF administrator creates either of the preceding profiles with a `PROXY` segment that stores the binding information -- the server name, bind Distinguished Name, and password.

Steps for using encrypted passwords

Perform the following steps to use encrypted LDAP bind passwords:

1. Define a RACF `KEYSMSTR` class profile by entering the following command, replacing the highlighted value with your own key:

Example:

```
RDEFINE KEYSMSTR LDAP.BINDPW.KEY SSIGNON(KEYENCRYPTED(0023528875DECFAC))
```

In this example:

- LDAP BIND passwords are masked by using a key saved in the `KEYSMSTR` class, `LDAP.BINDPW.KEY`.
- The key is `0023528875DECFAC`. (Replace this with your own key.)
- `KEYENCRYPTED` is specified (rather than `KEYMASKED`) because ICSF is active. (If ICSF is not active, replace `KEYENCRYPTED` with `KEYMASKED`.)

2. Activate the `KEYSMSTR` class by entering the following command:

```
SETROPTS CLASSACT(KEYSMSTR)
```

3. If you intend to use the `LDAPBIND` class, for each LDAP directory, create a RACF `LDAPBIND` class profile by entering the following command:

```
RDEFINE LDAPBIND MY.LDAP.SERVER1  
  PROXY(LDAPHOST(ldap://some.ldap.host:389)  
  BINDDN('cn=Joe User,ou=Poughkeepsie,o=IBM,c=US') BINDPW(MYPASS1)
```

RACF administration for PKI Services

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.
- c. Replace *cn=Joe User,ou=Poughkeepsie,o=IBM,c=US* with the bind DN.
- d. Replace *MYPASS1* with the bind password.

Note: The bind password is case-sensitive.

-
4. If you intend to use *IRR.PROXY.DEFAULTS* instead of the *LDAPBIND* class for encrypted LDAP bind passwords, enter 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.
- b. Replace *cn=Joe User,ou=Poughkeepsie,o=IBM,c=US* with the bind DN.
- c. Replace *MYPASS1* with the bind password.

Note: The bind password is case-sensitive.

-
5. Optionally, check your work by listing the segment with the *RLIST* command. If you are using the *LDAPBIND* class, enter 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, enter the following command:

```
RLIST FACILITY IRR.PROXY.DEFAULTS PROXY NORACF
```

Results: This command displays information like the following:

```
CLASS      NAME
FACILITY   IRR.PROXY.DEFAULTS

PROXY INFORMATION
LDAPHOST=  LDAP://SOME.LDAP.HOST:389
BINDDN=    cn=LDAP Administrator,ou=Poughkeepsie,o=IBM,c=US
BINDPW=    YES
```

RACF administration for PKI Services

Part 6. Troubleshooting

This part explains using logs and utilities, including the following:

- Chapter 18, “Using information from SYS1.LOGREC” on page 219 discusses ‘SYS1.LOGREC’, which is used to record unusual runtime events, such as an exception.
- Chapter 19, “Using information from the PKI Services logs” on page 225 discusses using the PKI Services logs, which are ongoing, to debug problems and explains how to change logging options and display log options settings.
- Chapter 20, “Using PKI Services utilities” on page 231 explains using PKI Services utilities:

vosview displays the entries contained in the VSAM ObjectStore data set (request database)

iclview displays the entries in the VSAM issued certificate list (ICL) data set.

Chapter 18. Using information from SYS1.LOGREC

'SYS1.LOGREC' records 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 58. LOGREC data for PKI Services

CSECT	Description
IKYAPIMS	Issued when an exception occurs during new_cert_post_rtn() processing (creating an ObjectStore entry for purposes of posting an issued certificate to LDAP).
	<p>Primary Symptom String:</p> <p>Component ID (PIDS): 5752XXPKI</p> <p>Load module: IKYAPI#L</p> <p>CSECT: IKYAPIMS</p> <p>Failing routine: IKYNEWCP</p> <p>Error information: Consists of <i>either</i> an abend code and a reason code <i>or</i> a facility ID and a message number.</p> <p>Abend code: If present, either the character <i>U</i> followed by 4 decimal digits or the character <i>S</i> followed by 3 hexadecimal digits.</p> <p>Reason code: If present, 8 hexadecimal digits.</p> <p>Facility ID: If present, 3 characters.</p> <p>Message number: If present, 8 hexadecimal digits.</p>
	<p>Secondary Symptom String:</p> <p>NEWID An 8-digit hexadecimal string that is the ObjectStore entry ID.</p> <p>UFN A character string that is the user-friendly-name.</p>

Using information from SYS1.LOGREC

Table 58. LOGREC data for PKI Services (continued)

CSECT	Description
IKYSCHDR	<p>Issued from the dispatcher() function when an exception is caught while creating and posting a CRL to LDAP.</p> <p>Primary Symptom String:</p> <p>Component ID (PIDS): 5752XXPKI</p> <p>Load module: IKYAPI#L</p> <p>CSECT: IKYSCHDR</p> <p>Failing routine: IKYDSPER</p> <p>Error information: Consists of <i>either</i> an abend code and a reason code <i>or</i> a facility ID and a message number.</p> <p>Abend code: If present, either the character <i>U</i> followed by 4 decimal digits or the character <i>S</i> followed by 3 hexadecimal digits.</p> <p>Reason code: If present, 8 hexadecimal digits.</p> <p>Facility ID: If present, 3 characters.</p> <p>Message number: If present, 8 hexadecimal digits.</p> <p>Secondary Symptom String:</p> <p>THREAD The string "DISPATCHR".</p>
IKYTIMER	<p>Issued when an exception is caught while processing a timer event in wakeup_rtn().</p> <p>Primary Symptom String:</p> <p>Component ID (PIDS): 5752XXPKI</p> <p>Load module: IKYOSSRV#L</p> <p>CSECT: IKYTIMER</p> <p>Failing routine: IKYWAKUP</p> <p>Error Information: Consists of <i>either</i> an abend code and a reason code <i>or</i> a facility ID and a message number.</p> <p>Abend code: If present, either the character <i>U</i> followed by 4 decimal digits or the character <i>S</i> followed by 3 hexadecimal digits.</p> <p>Reason code: If present, 8 hexadecimal digits.</p> <p>Facility ID: If present, 3 characters.</p> <p>Message number: If present, 8 hexadecimal digits.</p> <p>Secondary Symptom String:</p> <p>EVENTFUNC The name of the event routine being processed (postEvt, createEvt, or removeEvt).</p>

Table 58. LOGREC data for PKI Services (continued)

CSECT	Description
IKYP0N IKYP81 IKYP8A IKYP8B	<p data-bbox="370 254 1448 289">Issued when an ABEND occurs in the one of the CSECTs running on the Monitor Thread.</p> <hr/> <p data-bbox="370 296 667 331">Primary Symptom String:</p> <p data-bbox="370 338 878 373">Component ID (PIDS): 5752XXPKI</p> <p data-bbox="370 380 878 415">Load module: IKYPKID#L</p> <p data-bbox="370 422 1162 457">CSECT: IKYP0N, IKYP81, IKYP8A, or IKYP8B</p> <p data-bbox="370 464 854 499">Recovery routine: ESTEXIT</p> <p data-bbox="370 506 1239 541">Error Information: Consists of an abend code and reason code:</p> <p data-bbox="753 548 1448 653">Abend code: The character <i>S</i> followed by 4 hexadecimal digits or the character <i>U</i> followed by 4 decimal digits.</p> <p data-bbox="753 659 1073 730">Reason code: 8 hexadecimal digits.</p>
IKYP8B	<p data-bbox="370 768 1117 804">Issued when an ABEND occurs in the PC routine (or helper routines).</p> <hr/> <p data-bbox="370 810 667 846">Primary Symptom String:</p> <p data-bbox="370 852 878 888">Component ID (PIDS): 5752XXPKI</p> <p data-bbox="370 894 878 930">Load module: IKYPKID#L</p> <p data-bbox="370 936 837 972">CSECT: IKYP8B</p> <p data-bbox="370 978 854 1014">Recovery routine: ARREXIT</p> <p data-bbox="370 1020 1260 1056">Error information: Consists of an abend code and a reason code.</p> <p data-bbox="753 1062 1448 1167">Abend code: The character <i>S</i> followed by 4 hexadecimal digits or the character <i>U</i> followed by 4 decimal digits.</p> <p data-bbox="753 1173 1073 1245">Reason code: 8 hexadecimal digits.</p>

Using information from SYS1.LOGREC

Table 58. LOGREC data for PKI Services (continued)

CSECT	Description
IKYP8A	<p data-bbox="337 254 1417 317">Issued when an exception is caught in the service thread routine IKYP8A01 or in the services thread request routine IKYP8A02.</p> <p data-bbox="337 323 1417 359">Primary Symptom String:</p> <p data-bbox="337 365 1417 401">Component ID (PIDS): 5752XXPKI</p> <p data-bbox="337 407 1417 443">Load module: IKYPKID#L</p> <p data-bbox="337 449 1417 485">CSECT: IKYP8A</p> <p data-bbox="337 491 1417 527">Failing routine: IKYP8A01 or IKYP8A02</p> <p data-bbox="337 533 1417 611">Error information: Consists of <i>either</i> an abend code and a reason code <i>or</i> a facility ID and a message number.</p> <p data-bbox="721 617 1417 737">Abend code: If present, either the character <i>U</i> followed by 4 decimal digits or the character <i>S</i> followed by 3 hexadecimal digits.</p> <p data-bbox="721 743 1417 821">Reason code: If present, 8 hexadecimal digits.</p> <p data-bbox="721 827 1417 884">Facility ID: If present, 3 characters.</p> <p data-bbox="721 890 1417 968">Message number: If present, 8 hexadecimal digits.</p> <p data-bbox="337 995 1417 1031">Secondary Symptom String:</p> <p data-bbox="337 1037 1417 1073">USER The user ID of the requestor.</p> <p data-bbox="337 1079 1417 1115">FUNC A function code of 8 hexadecimal digits.</p>

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
                                           REPORT DATE: 221 01
SCP:   VS 2 REL 3              MODEL:  9672                               HH MM SS.TH
                                           SERIAL: 048288          TIME: 19:05:16.02

```

```

SEARCH ARGUMENT ABSTRACT:
  PIDS/5752XXPKI RIDS/IKYPKID#L RIDS/IKYP8A RIDS/IKYP8A01 AB/S0C4
  FLDS/RSNCODE VALU/H00000000

```

```

SYSTEM ENVIRONMENT:
  CPU MODEL: 9672          DATE: 221 01
  CPU SERIAL: 048288      TIME: 19:05:16.02
  SYSTEM:    DCEIMGUI      BCP:  MVS
  RELEASE LEVEL OF SERVICE ROUTINE:  HBB7703
  SYSTEM DATA AT ARCHITECTURE LEVEL: 10
  COMPONENT DATA AT ARCHITECTURE LEVEL: 10
  SYSTEM DATA: 00000000 00000000 |.....|

```

```

COMPONENT INFORMATION:
  COMPONENT ID:           5752XXPKI
  COMPONENT RELEASE LEVEL: 7706
  SERVICE RELEASE LEVEL:  HKY7706
  DESCRIPTION OF FUNCTION: PKI SERVICES DAEMON

```

```

PRIMARY SYMPTOM STRING:
  PIDS/5752XXPKI RIDS/IKYPKID#L RIDS/IKYP8A RIDS/IKYP8A01 AB/S0C4
  FLDS/RSNCODE VALU/H00000000

```

SYMPTOM	SYMPTOM DATA	EXPLANATION
PIDS/5752XXPKI	5752XXPKI	COMPONENT IDENTIFIER
RIDS/IKYPKID#L	IKYPKID#L	ROUTINE IDENTIFIER
RIDS/IKYP8A	IKYP8A	ROUTINE IDENTIFIER
RIDS/IKYP8A01	IKYP8A01	ROUTINE IDENTIFIER
AB/S0C4	0C4	ABEND CODE - SYSTEM
FLDS/RSNCODE	RSNCODE	DATA FIELD NAME
VALU/H00000000	00000000	ERROR RELATED HEXADECIMAL VALUE

```

SECONDARY SYMPTOM STRING:
  FLDS/USER VALU/CG422253 FLDS/FUNC VALU/H00000000

```

SYMPTOM	SYMPTOM DATA	EXPLANATION
FLDS/USER	USER	DATA FIELD NAME
VALU/CG422253	G422253	ERROR RELATED CHARACTER VALUE
FLDS/FUNC	FUNC	DATA FIELD NAME
VALU/H00000000	00000000	ERROR RELATED HEXADECIMAL VALUE

```

THE SYMPTOM RECORD DOES NOT CONTAIN FREE FORMAT COMPONENT INFORMATION.
HEX DUMP OF RECORD:

```

```

HEADER
+000  4C831800  00000000  0001221F  19051602  |<C.....|
+010  FF048288  96720000  |..BH0...|

```

Figure 33. Sample LOGREC data (Part 1 of 2)

```

SYMPTOM RECORD
+000 E2D9F9F6 F7F2F0F4 F8F2F8F8 FFFCA5B |SR9672048288...$
+010 B64312D1 0360F103 40404040 40404040 |...J.-1.
+020 4040C4C3 C5C9D4C7 E4C9F5F7 F5F2C8C2 | DCEIMGUI5752HB
+030 C2F7F7F0 F3400080 00000000 00000000 |B7703 .....
+040 F1F00030 00640070 005C0138 003101A0 |10.....*.....
+050 LENGTH(0032) ==> ALL BYTES CONTAIN X'00'.
+070 E2D9F2F1 F1F0F5F7 F5F2E7E7 D7D2C900 |SR21105752XXPKI.
+080 F7F7F0F6 C8D2E8F7 F7F0F640 00000000 |7706HKY7706 ....
+090 00000000 00000000 00000000 D7D2C940 |.....PKI
+0A0 E28599A5 898385A2 40848185 94969540 |SERVICES DAEMON
+0B0 40404040 40404040 40404040 00000000 |.....
+0C0 00000000 00000000 00000000 00000000 |.....
+0D0 00000000 0B41465C 0B414668 0B414699 |.....*.....R
+0E0 0B4146A8 0B4146A8 0B4146A8 01000000 |...Y...Y...Y....
+0F0 0B4144C8 00000000 00000000 F0F1F2F3 |...H.....0123
+100 F4F5F6F7 F8F9C1C2 C3C4C5C6 00680040 |456789ABCDEF...
+110 0000000F 0B414530 00000000 0B414374 |.....
+120 00000000 F0F00000 00000008 00000008 |...00.....
+130 00000000 40E70030 D7C9C4E2 61F5F7F5 |... X..PIDS/575
+140 F2E7E7D7 D2C940D9 C9C4E261 C9D2E8D7 |2XXPKI RIDS/IKYP
+150 D2C9C47B D340D9C9 C4E261C9 D2E8D7F8 |KID#L RIDS/IKYP8
+160 C140D9C9 C4E261C9 D2E8D7F8 C1F0F140 |A RIDS/IKYP8A01
+170 C1C261E2 F0C3F440 C6D3C4E2 61D9E2D5 |AB/S0C4 FLDS/RSN
+180 C3D6C4C5 40E5C1D3 E461C8F0 F0F0F0F0 |CODE VALU/H00000
+190 F0F0F040 0B414780 00000001 00000000 |000 .....
+1A0 C6D3C4E2 61E4E2C5 D940E5C1 D3E461C3 |FLDS/USER VALU/C
+1B0 C7F4F2F2 F2F5F340 C6D3C4E2 61C6E4D5 |G422253 FLDS/FUN
+1C0 C340E5C1 D3E461C8 F0F0F0F0 F0F0F0F0 |C VALU/H00000000
+1D0 40

```

Figure 33. Sample LOGREC data (Part 2 of 2)

Chapter 19. Using information from the PKI Services logs

This chapter 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 proc (see “PKISERVD sample procedure to start PKI Services daemon” on page 335 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 proc, 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 34 on page 226 shows this.

Using information from the PKI Services logs

```

D - gdlvmg15.ws - [43 x 80]
File Edit Transfer Appearance Communication Assist Window Help
SDSF STATUS DISPLAY ALL CLASSES LINE 34-73 (95)
COMMAND INPUT ==> SCROLL ==> CSR
NP JOBNAME JobID Owner Prty Queue C Pos SAff ASys Status
? _ PKISERVD STC00687 PKISRVD 15 EXECUTION EIMG EIMG
BPXAS STC00691 STCUSER 15 EXECUTION EIMG EIMG
BPXAS STC00692 STCUSER 15 EXECUTION EIMG EIMG
BPXAS STC00693 STCUSER 15 EXECUTION EIMG EIMG
BPXAS STC00694 STCUSER 15 EXECUTION EIMG EIMG
$MASCOMM STC00596 15 PRINT A 1
SUIMGUN TSU00581 SUIMGUN 1 PRINT 2
BPXAS STC00592 STCUSER 1 PRINT 3
BPXAS STC00593 STCUSER 1 PRINT 4
BPXAS STC00595 STCUSER 1 PRINT 5
BPXAS STC00591 STCUSER 1 PRINT 6
BPXAS STC00594 STCUSER 1 PRINT 7
SOFV3VS2 STC00541 STCUSER 1 PRINT 8
RACF STC00571 STC1 1 PRINT 9
CSNET STC00545 STC1 1 PRINT 10
TSO STC00544 STC1 1 PRINT 11
SYSLOG STC00549 +MASTER+ 1 PRINT 12
INIT STC00550 STC1 1 PRINT 13
DFSCM STC00547 DFS 1 PRINT 14
INIT STC00551 STC1 1 PRINT 15
INIT STC00552 STC1 1 PRINT 16
INIT STC00553 STC1 1 PRINT 17
INIT STC00554 STC1 1 PRINT 18
INIT STC00556 STC1 1 PRINT 19
INIT STC00555 STC1 1 PRINT 20
INIT STC00557 STC1 1 PRINT 21
INIT STC00558 STC1 1 PRINT 22
INIT STC00559 STC1 1 PRINT 23
INIT STC00560 STC1 1 PRINT 24
INIT STC00561 STC1 1 PRINT 25
INIT STC00562 STC1 1 PRINT 26
INIT STC00563 STC1 1 PRINT 27
INIT STC00564 STC1 1 PRINT 28
ASCHINT STC00565 STC1 1 PRINT 29
ASCHINT STC00566 STC1 1 PRINT 30
ASCHINT STC00567 STC1 1 PRINT 31
ASCHINT STC00568 STC1 1 PRINT 32
BPXAS STC00570 STCUSER 1 PRINT 33
BPXAS STC00569 STCUSER 1 PRINT 34
BPXAS STC00572 STCUSER 1 PRINT 35

```

Figure 34. Separating the job files

After using the question mark line command, you can select the file you want to view by entering an S before this file name. Figure 35 on page 227 shows this:

```

SDSF JOB DATA SET DISPLAY - JOB PKISERV D (STC00687)      LINE 1-5 (5)
COMMAND INPUT ==>
NP  DDNAME  StepName ProcStep DSID Owner   C Dest          Rec-Cnt PAGE
   JESMSG LG JES2      2 PKISRV D A           2
   JESJCL  JES2      3 PKISRV D A          27
   JESYSMSG JES2      4 PKISRV D A           2
s_  STDOUT  PKISERV D 101 PKISRV D A        69,681
   STDERR  PKISERV D 102 PKISRV D A           0
    
```

MA d 07/003

Connected to remote server/host gdlvmg15.endicott.ibm.com using port 23

Figure 35. Selecting a file to view

Figure 36 on page 228 shows the messages contained in the file:

Using information from the PKI Services logs

```
D - gdlvmg15.ws - [43 x 80]
File Edit Transfer Appearance Communication Assist Window Help
SDSF OUTPUT DISPLAY PKISERVD STC00687 DSID 101 LINE 1,105 COLUMNS 02- 81
COMMAND INPUT ==>
Wed Aug 8 15:44:46 2001 (00000001) CORE IKYC026I Deleting inactive object 37. L
Wed Aug 8 15:44:46 2001 (00000001) DB ----- CSR
Vsam::get_flags -
key = 37 flags = 2140030 rlen = 745 RBA = 38912
name = ""
issuedDate = "20010710170839"
lastChangeDate = "20010710170839"
longkey = 1jwBokYQxQ6/VkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::delete_record -
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::release_record - record contents before release
key = 37 flags = 2140030 rlen = 745 RBA = 38912
name = ""
issuedDate = "20010710170839"
lastChangeDate = "20010710170839"
longkey = 1jwBokYQxQ6/VkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::obj fetch - obj key = 38
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::read_record - read OK
key = 38 flags = 2140030 rlen = 745 RBA = 39936
name = ""
issuedDate = "20010710171341"
lastChangeDate = "20010710171341"
longkey = 1jwBoCXyk+2fVkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::get_flags -
key = 38 flags = 2140030 rlen = 745 RBA = 39936
name = ""
issuedDate = "20010710171341"
lastChangeDate = "20010710171341"
longkey = 1jwBoCXyk+2fVkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::getLastTime -
key = 38 flags = 2140030 rlen = 745 RBA = 39936
name = ""
issuedDate = "20010710171341"
lastChangeDate = "20010710171341"
longkey = 1jwBoCXyk+2fVkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) CORE IKYC026I Deleting inactive object 38. L
MA d 02/021
Connected to remote server/host gdlvmg15.endicott.ibm.com using port 23
```

Figure 36. 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 may 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 may span multiple lines.

Informational, warning, error, and severe level messages begin with a message number. (See Chapter 21, "Messages" on page 239.) Verbose and debug level messages do not have message numbers and are not documented.

The following is an example of an informational message:

```
Wed Aug 8 15:44:46 2001 (00000001) CORE IKYC026I Deleting inactive object 37.
Last changed at 2001/07/10 17:08:39
```

_PKISERV_MSG_LEVEL subcomponents and message levels

_PKISERV_MSG_LEVEL is an environment variable that specifies the subcomponent and message level for logging messages.

The subcomponents are:

Subcomponent	Meaning
*	This is the wildcard character, which represents all subcomponents.
CORE	The core functions of PKI Services that are not specific to the other subcomponents.
DB	Activity related to the request or issued certificate VSAM data stores
LDAP	LDAP posting operations
PKID	The PKI Services daemon address setup and infrastructure.
POLICY	Certificate creation and revocation policy processing
SAF	SAF key ring, OCEP, and R_datalib calls

The message levels are:

Debug level (hierarchically listed)	Meaning
S	This indicates logging only Severe messages.
E	This indicates logging Severe and Error messages.
W	This indicates logging Severe, Error, and Warning messages. This is the default message level for all subcomponents if you do not set the environment variable.
I	This indicates logging Severe, Error, Warning, and Informational messages.
D	This indicates logging Severe, Error, Warning, Informational and Diagnostic.
V	This indicates logging ALL messages, including Verbose Diagnostic messages. This is very verbose. Do not use it unless IBM support personnel instruct you to do so.

(For information about updating the environment variables during configuration, see “Optionally updating PKI Services environment variables” on page 55.)

After PKI Services is up and running, if a problem occurs, the MVS programmer can:

- Change the logging options dynamically — by using the MODIFY (minimum abbreviation F) console command
- Display the current settings — by using another MODIFY console command

Changing logging options

To change logging options dynamically, enter the following MODIFY (or F) console command:

```
F PKISERVD,LOG sub-component.level[,sub-component.level...]
```

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, enter the following MODIFY (or F) console command:

```
F PKISERVD,DISPLAY
```

Example (of output):

```
12.55.51 IKYP027I PKI SERVICES SETTINGS:
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
MESSAGE LOGGING SETTING: STDERR_GING
CONFIGURATION FILE IN USE:
/etc/pkiserv/pkiserv.conf
```

Chapter 20. Using PKI Services utilities

This chapter describes the following utility programs, which are shipped with PKI Services. These programs are installed in the /bin subdirectory (/usr/lpp/pkiserv/bin).

- | | |
|----------------|------------------------------------------------------------------------------------|
| vosview | displays the entries contained in the VSAM ObjectStore data set (request database) |
| iclview | displays the entries in the VSAM issued certificate list (ICL) data set. |

vosview

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.

Format

```
vosview {[-r]vsam-dataset-name [data-decode-command-string] |
        -c [data-decode-command-string]}
```

Parameters

-r Indicates opening the VSAM data set in record-level sharing (RLS) mode.

Note: The **-r** and **-c** options are mutually exclusive.

-c Indicates retrieving the data set name and RLS information from the pkiserv.conf file. (This file is located in the directory the `_PKISERV_CONFIG_PATH` environment variable specifies. If the environment variable is not defined, the directory defaults to `/etc/pkiserv`.)

Note: The **-r** and **-c** options are mutually exclusive.

vsam-dataset-name

Is an MVS-style data set name (DSN).

Note: Make sure to include the escape character, which is a backslash (`\`), before the quotation marks enclosing the MVS data set name, for example, `\'pkisrvd.vsam.ost\'`.

data-decode-command-string

Is 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` with the `-o` option, use the following:

```
vosview \'pkisrvd.vsam.ost\' "dumpasn1 -o -"
```

Note: A `dumpasn1` utility is not shipped with PKI Services.

The fixed header data is displayed for each record.

The output for record 1 would look like the following:

```
Object key = 1
Last used key = 726, CRL serial number = 518, ARL serial number = 518,
  Current DP = 11, Number of certs for current DP = 3
name = ""
longkey = ??????????????????????????????
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

```

The output for record 2 would look like the following:

```

| Object key = 2
| name = ""
| longkey = ??????????????????????
| appldata =
| comment =
| data len = 33
| flags = 0 - Type = ??? ObjSt??????
| Creation time is: 2002/04/22 17:29:49
| Last modified time is: 2002/04/29 18:23:51

```

The output for a certificate request record would look like the following:

```

Object key = 105
name = "John Q. Public"
longkey = 1F45AEF2D3729FA35156BC47
appldata = "1YBSSL"
comment = ""
data len = 570
flags = 1020111 - Type = Cert State = RA CertReqActive [State Flag]

```

Object key

Is the index into the VSAM data set name.

name

The requestor's name.

longkey

The transaction ID data.

appldata

An 8-character string identifying to the application the short name or nickname of the certificate template. (PKI Services provides eight certificate templates but it is RACF, or an equivalent security product, rather than PKI Services that handles the SAF templates.) The following table lists the nicknames for the certificate templates. (These are the nicknames that are in the pkiserv.tmpl certificate templates file by default. Your installation may have changed these nicknames or added others during customization. See "TEMPLATE sections" on page 100 for more information.)

Table 59. Nicknames of certificate templates for appldata

Type of certificate	Nickname
One-year PKI SSL browser certificate	1YBSSL
One-year PKI S/MIME browser certificate	1YBSM
Two-year PKI browser certificate for authenticating to z/OS	2YBZOS
Five-year PKI SSL server certificate	5YSSSL
Five-year PKI IPSEC server (firewall) certificate	5YSIPS
Five-year PKI intermediate CA certificate	5YSCA
One-year SAF browser certificate	(No nickname)
One-year SAF server certificate	(No nickname)

comment

A comment the administrator supplied the last time the request was updated.

vosview

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:

- 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.

iclview

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.

Format

```
iclview {[-r] vsam-dataset-name [data-decode-command-string]|
        -c [data-decode-command-string]}
```

Parameters

-r Indicates opening the VSAM data set in record-level sharing (RLS) mode.

Note: The -r and -c options are mutually exclusive.

vsam-dataset-name

Is an MVS-style DSN.

Note: Make sure to include the escape character, which is a backslash (\), before the quotation marks enclosing the MVS data set name, for example, \`'pkisrvd.vsam.icl'`.

data-decode-command-string

Is 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.

-c Indicates retrieving the data set name and RLS information from the pkiserv.conf file. (This file is located in the directory the `_PKISERV_CONFIG_PATH` environment variable specifies. If the environment variable is not defined, the directory defaults to `/etc/pkiserv`.)

Note: The -r and -c options are mutually exclusive.

Examples

To view the records in VSAM ICL data set 'PKISRV.D.VSAM.ICL', passing the certificate to a utility called `dumpsan1` with the `-o` option, use the following:

```
iclview \'pkisrvd.vsam.icl' "dumpsan1 -o -"
```

Note: A `dumpsan1` utility is not shipped with PKI Services.

The fixed header data that is displayed for each record would look like the following:

```
Cert 8: John Q. Public
      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,0=The Firm
      Issuer: OU=PKI Services CA,0=IBM,C=US
      Requester: John Smith
      AppData: 1YBSSL
      Serial Number: CCD
      Email flag: Off
```

An explanation of these lines follows:

iclview

- 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 of one of the following, and comment (if any):
 - ISSUED
 - REVOKED, not posted
 - REVOKED, awaiting CRL post
 - REVOKED, on posted CRL
- Issued at is when the certificate was issued.
- Last changed is when the administrator last changed the certificate.
- Subject: is the name of the person owning the certificate.
- Issuer: is the name of the certificate authority that issued the certificate.
- Requestor: is the requestor's name.
- Appldata: is an 8-character string identifying to the application the short name or nickname of the certificate template. (See Table 59 on page 233 for a list and explanation of nicknames.)
- Serial Number: is the serial number of the certificate as a hexadecimal number.
- Email flag: is the indicator of whether or not to send an expiration warning message. The possible values are "On" or "Off".

Part 7. Reference information

This part provides reference information, including code samples for important files.

Note: The code samples in this chapter might not be identical to the code shipped with the product. If you want to see the most current code, look in the appropriate source directory.

- Chapter 21, “Messages” on page 239 explains PKI Services messages.
- Chapter 22, “File directory structure” on page 257 describes product and HFS directories for PKI Services and files contained in them.
- Chapter 23, “The pkiserv.conf configuration file” on page 261 provides a code sample of the pkiserv.conf configuration file.
- Chapter 24, “The pkiserv.tmpl certificate templates file” on page 263 provides a code sample of the pkiserv.tmpl file. (For detailed explanations about the contents of this file, see Chapter 12, “Customizing the end-user Web application” on page 91.)
- Chapter 25, “Environment variables” on page 305 explains the pkiserv.envars environment variables file and provides a code sample.
- Chapter 26, “The IKYSETUP REXX exec” on page 309 explains the contents of the IKYSETUP REXX exec that performs RACF administration and provides a code sample.
- Chapter 27, “Other code samples” on page 327 provides additional code samples. The following table 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.

Table 60. Summary of information about important files

File	Description	Source location (default)	For code sample...
httpd.conf and httpd2.conf	Contain z/OS HTTP Server directives.	/usr/lpp/pkiserv/samples/	See page 327
IKYCVSAM	Sample IDCAMS JCL to create VSAM data sets (regardless of whether you are using a sysplex or non-sysplex).	SYS1.SAMPLIB	See page 330
IKYRVSAM	Sample IDCAMS JCL to use if you are migrating from z/OS Version 1 Release 3 and want sysplex support. IKYRVSAM reallocates your z/OS Version 1 Release 3 VSAM data sets in preparation for sharing in a sysplex.		
IKYSETUP	REXX exec to set up RACF profiles.	SYS1.SAMPLIB	See page 309
pkiserv.conf	PKI Services configuration file.	/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)	See page 261
PKISERV.D	Sample proc to start PKI Services daemon.	SYS1.PROCLIB	See page 335
pkiserv.envars	PKI Services environment variables file.	/usr/lpp/pkiserv/samples/ (You might need to copy this file to the runtime directory, /etc/pkiserv.)	See page 307

Table 60. Summary of information about important files (continued)

File	Description	Source location (default)	For code sample...
pkiserv.tmpl	PKI Services certificate template file.	/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)	See page 263

- Chapter 28, “The certificate validation service” on page 337 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 21. 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 61. Meaning of fourth character in message number

Character — Meaning	Component producing messages	Where messages appear
C — CORE	Core subcomponent	PKI Services log
D — DB	Database accessing subcomponent	PKI Services log
I — INTERFACE	PKISERV CGIs	In the user's Web browser window
L — LDAP	LDAP bind subcomponent	PKI Services log
O — POLICY	Certificate creation and revocation policy subcomponent	PKI Services log
P — PKID	PKI Services daemon address space controller	<ul style="list-style-type: none">• PKI Services log• (For those with destination and routing codes) operators console
S — SAF	SAF interfacing subcomponent	PKI Services log

Characters five through seven are numeric. The eighth character is the message type:

Table 62. Meaning of eighth character in message number

Character — Meaning	Action required
I — Informational (Status message)	No action required
E — Eventual Action	Possible problem that may require eventual action
A — Action Required	Problem that requires immediate attention

For information about setting messages options using environment variables, see page 305.

Messages

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.

System Programmer 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 are displayed. A description of the error is also displayed, if known.

System Action: The certificate is not created.

System Programmer Response: Report the error to the IBM support center.

IKYC003I Error *nnnn* registering the next CRL
cutting job: *error-code-description*

Explanation: PKI Services has just finished creating the current CRL and is attempting to schedule the next CRL creation thread. An error was encountered. The error code encountered is displayed. A description of the error is also displayed, if known.

System Action: Future CRLs are not created until the problem is corrected and PKI Services is restarted.

System Programmer Response: Look for other error messages that may 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 OCSF 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. Note: If the error code is an OCSF 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 no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. If the error 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 configuration file (slapd.conf) and recycle the LDAP server. For all other LDAP errors, follow the instructions in *z/OS Security Server LDAP Client Programming*. Report non-OCSF errors 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 OCSF 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 no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application*

Programming. If the error is LDAPDL_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 *z/OS Security Server LDAP Client Programming*. Report non-OCSF errors to the IBM support center. If message IKYC009I is also displayed, report that information as well.

IKYC008I **Error *nnnn* creating an {CSSM_DL_DB_PKICA entry for CA Certificate | CSSM_DL_DB_RECORD_CRL entry for {CRL | ARL} | CSSM_DL_DB_PKIUSER entry for User Cert} 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 OCSF 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 may also see message IKYC005I or IKYC007I. If so, follow the instructions for the message displayed. Otherwise, if no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. Follow related instructions in *z/OS Security Server LDAP Client Programming*. Report non-OCSF 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, IKYC006I, and IKYC008I.

System Programmer Response: If reporting message IKYC005I, IKYC006I, 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: In some cases the error code description may be self-explanatory. If not, 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 58.

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 58.

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 58.

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

Messages

"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 58.

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 76.

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_invalidDate: *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_invalidDate

Explanation: PKI Services is processing a revocation request and has encountered an internal error.

System Action: The revocation request is not processed.

System Programmer Response: Report the error to the IBM support center.

IKYC024I Failed to schedule event in nnnn seconds, status = nnnn: error-code-description

Explanation: PKI Services is attempting to schedule a timed event and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

System Action: The event is not scheduled.

System Programmer Response: Report the error to the IBM support center.

IKYC025I Failed to schedule event status = nnnn: error-code-description

Explanation: PKI Services is attempting to schedule a timed event and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

System Action: The event is not scheduled.

System Programmer Response: Report the error to the IBM support center.

IKYC026I Deleting {inactive | completed} object object-id. Last changed at YYYYMMDD HH:MM:SS

Explanation: PKI Services is attempting to purge the request database of inactive and completed requests. A request that has met the criteria for deletion has been found. The request's ID is displayed along with information on when it was last changed. This is an informational message only.

System Action: The request is deleted. PKI Services continues normal processing.

IKYC027I Removing certificate post request after nnnn unsuccessful attempts

Explanation: PKI Services is attempting to purge the request database of unsuccessful LDAP post requests. A request that has met the criteria for deletion has been found. The number of unsuccessful attempts for this request is displayed. This is an informational message only.

System Action: The request is deleted. PKI Services continues normal processing.

IKYC028I Export for CertId certificate-id unsuccessful. Request is still pending approval or yet to be issued

Explanation: A client has requested a certificate and is attempting to retrieve it. The retrieval was unsuccessful because the certificate is not yet available. The request either has yet to be approved by a PKI Services administrator or has been approved, but has not yet been issued by PKI Services. This is an informational message only.

System Action: The state of the request is unchanged. PKI Services continues normal processing.

PKI Services Administrator Response: Use PKI Services administrative functions to query the request to check its state. If the request is still pending approval, determine whether the request should be approved or rejected and take action accordingly. For more information, see "Processing certificate requests" on page 177.

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 OCSF return code, no error description is displayed.

System Programmer Response: If no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. Follow related instructions in *z/OS Security Server LDAP Client Programming*. Report non-OCSF errors to the IBM support center.

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

Messages

| warning message. The warning message is not sent at
| this time but sending will be tried later.

| **IKYC031I** **Error** *nnnn* **invoking sendmail with**
| **email address** *email-address* **retrieved**
| **from LDAP entry** *distinguished-name*

| **Explanation:** PKI Services is trying to call the
| sendmail utility to notify a client that his or her certificate
| is expiring. The call was unsuccessful. This message
| displays the e-mail address and distinguished name
| from which it was retrieved and the error code
| encountered.

| **System Programmer Response:** Diagnose the
| problem by consulting the *z/OS Communications*
| *Server: IP Diagnosis* and related manuals. Report
| non-Communications Server errors to the IBM support
| center.

| **System Action:** The warning message may or may
| 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 either ready or rejected.
| 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 Programmer Response:** Diagnose the
| problem by consulting the *z/OS Communications*
| *Server: IP Diagnosis* and related manuals. Report
| non-Communications Server errors to the IBM support
| center.

| **System Action:** The message may or may not have
| been sent.

| **IKYC033I** **Error** *nnnn* **accessing**
| **{ReadyMessageForm**
| **RejectMessageForm |**
| **ExpiringMessageForm}** *form-value*

| **Explanation:** PKI Services is attempting to notify a
| client that his or her certificate is either ready, rejected,
| or expiring. 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 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 58.

| **System Action:** The message is not sent. If this is the
| expiring warning message, sending will be attempted
| later.

| **IKYC034I** **Error** *nnnn* **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 Programmer Response:** Stop and restart PKI
| Services. If the problem reoccurs, report the error to the
| IBM support center.

| **System Action:** PKI Services processing continues.
| However, further processing of certificate requests may
| fail until PKI Services is stopped and restarted.

| **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 Programmer Response:** Correct the value
| and restart PKI Services if desired. For more
| information, see "(Optional) Steps for updating the
| configuration file" on page 58.

| **System Action:** PKI Services continues, but no
| expiration warning messages will be issued.

| **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 is stopped.

System Programmer Response: This message may 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.

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 is stopped.

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 58.

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 is stopped.

System Programmer Response: This message may 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. The open has failed. The data set name is displayed.

System Action: PKI Services is stopped.

System Programmer Response: Locate the failing "DSN" value in the pkiserv.conf file. Make sure that the value specifies the correct VSAM data set name and that the data set has been created. If 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 more information, see "(Optional) Steps for updating the configuration file" on page 58 and "Steps for creating the VSAM object store and ICL data sets and indexes" on page 83.

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 is not performed.

User Response: Contact your Web administrator.

Web Administrator Response: Determine why the exit program denied the request and correct the program if necessary.

IKYI002I **SAF Service IRRSPX00 Returned SAF RC = *nn* RACF RC = *nn* RACF RSN = *nn* {diagnostic-information}**

Explanation: A user is requesting PKI Services. The PKIServ Web application called the IRRSPX00 SAF

Messages

callable service as requested. The service was unsuccessful. The diagnostic information that follows the message should describe the problem in greater detail:

- 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.
- 6 Certificate generation provider indicated the following error: *<provider-specific-error-msg>*.
- 7 Unexpected Error.

System Action: The request in not performed.

User Response: Correct the problem if applicable. If you cannot correct the problem, contact your Web administrator.

Web Administrator Response: Problems 1, 2, and 3 probably indicate an error with the certificate template. Change the certificate template definition in the *pkiserv.tmpl* file to correct the error.

Problem 4 indicates the user ID assigned to the unit of work calling the IRRSPX00 callable service is not RACF-authorized to perform the request. Determine if the user should have access. If so, use RACF commands to permit the user ID to the required resources.

Problem 5 indicates the PKI Services daemon process has not been started. Start PKI Services; then retry the request.

For problems 6 and 7 or for more information on any of the preceding problems, see earlier chapters in this document and *z/OS Security Server RACF Callable Services*.

IKYI003I **PKI Services CGI error in**
cgi-program-name: *diagnostic-error-*
information

Explanation: A user is requesting PKI Services. The PKIServ Web application CGI program processing the request detected a problem. The name of the CGI program and additional diagnostic information is displayed in the message.

System Action: The request in not performed.

User Response: Contact your Web administrator.

Web Administrator Response: Locate the CGI program mentioned in the message. (Its default installation location is in a subdirectory under */usr/lpp/pkiserv/PKIServ*.) Examine the CGI program's source code to determine the spot where it is failing and why. In most cases, the problem is caused by an error

in the PKI Services template file (usually located in */etc/pkiserv/pkiserv.tmpl*). Correct the problem and retry the request. For more information, see Chapter 12, "Customizing the end-user Web application" on page 91 and Chapter 13, "Customizing the administration Web pages" on page 133.

IKYI004I **Installation exit failed. RC = nn**

Explanation: A user is requesting PKI Services. The PKIServ Web application called an installation-provided exit program. The exit program either terminated abnormally or returned an unsupported return code value. The return code from the invocation of the exit program is displayed in the message.

System Action: The request in not performed.

User Response: Contact your Web administrator.

Web Administrator Response: Determine why the exit program has failed and correct the program as necessary.

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 = LDAPDL_INSUFFICIENT_ACCESS)

System Programmer Response: Locate the incorrect "AuthName" value in the *pkiserv.conf* file and correct it. The value must be specified as an LDAP distinguished name, for example, CN=root,O=IBM. Note: The OID qualifiers must be specified in uppercase and there cannot be any spaces surrounding the equal signs or commas separating the attribute value assertions (AVAs). Make corrections as needed, then stop and restart PKI Services. For more information, see "Steps for tailoring the LDAP section of the configuration file" on page 76.

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

OCSF return code, no error description will be displayed.

System Action: PKI Services attempts to bind to your other LDAP servers, if any. If PKI Services is unable to bind to any LDAP servers, the LDAP posting of certificates and CRLs is temporarily suspended. PKI Services attempts to bind again during the next posting interval. All post requests will remain in the request database to be attempted later, subject to being deleted after one week of unsuccessful attempts.

System Programmer Response: If no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. Diagnose the problem indicated by the return code. For LDAPDL_SERVER_DOWN, ensure that your LDAP server is running. If so, you may 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 *z/OS Security Server LDAP Client Programming*. Report non-OCSF errors to the IBM support center. If message IKYC0091 is also displayed, report that information as well. For more information, see "Steps for tailoring the LDAP section of the configuration file" on page 76.

IKYL0031 Incorrect value specified for LDAPBIND or FACILITY Class profile *profile-name*

Explanation: PKI Services LDAP bind processing is trying to retrieve its LDAP bind information in preparation for communicating with the LDAP server. The bind information is contained in either an LDAPBIND class profile or the IRR.PROXY.DEFAULTS FACILITY class profile. 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 Programmer Response: Locate the profile name in the PKI Services configuration file and correct it if needed. If the host name is specified as a URL, you cannot specify it as an SSL URL (for example, ldaps://). PKI Services does not use SSL to communicate with the LDAP server. If you make corrections, stop and restart PKI Services. If the profile name is already correct, contact your RACF administrator. For more information, see "(Optional) Steps for updating the configuration file" on page 58.

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*.

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.

IKYL0041 Bad LDAP Server value *server-value* in pkiserv.conf file

Explanation: PKI Services LDAP bind processing is trying to retrieve its LDAP bind information in preparation for communicating with the LDAP server. (The *Server1*, *Server2*, and so forth keywords in the LDAP section of the PKI Services configuration file specify the server host name information.) The host name has been specified incorrectly. Its value is displayed.

System Programmer Response: Locate the server name in the PKI Services configuration file, and correct it if needed. If the host name is specified as a URL, you cannot specify it as an SSL URL (for example, ldaps://). PKI Services does not use SSL to communicate with the LDAP server. If you make corrections, stop and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 58.

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.

IKYO0011 Error *nnnn* {setting** | **getting**} *certificate-field* {**in certificate** | **from template**}: *error-code-description***

Explanation: PKI Services is processing a certificate request field and has encountered an internal error. The field name and the error code encountered are displayed. A description of the error is also displayed, if known.

System Action: The certificate request is not processed.

System Programmer Response: Report the error to the IBM support center.

Messages

IKYO002I *nnnn bytes of unconsumed data transferring certificate-field to certificate*

Explanation: PKI Services is processing a certificate request field and has found that the field is larger than it should be. This is an internal error. The field name and the number of extra bytes are displayed.

System Action: The certificate request is not processed.

System Programmer Response: Report the error to the IBM support center.

IKYO003I **The certificate request failed validity checks. Status is *nnnn*;**
status-code-description

Explanation: PKI Services is processing a certificate request field and has encountered an internal error. The field name and the status (error) code encountered are displayed. A description of the error is also displayed, if known.

System Action: The certificate request is not processed.

System Programmer Response: Report the error to the IBM support center.

IKYO004I *action-being-performed returned nnnn;*
error-code-description

Explanation: PKI Services is processing a request and has encountered an internal error. The action being performed and the error code encountered are displayed. A description of the error is also displayed, if known.

System Action: The request is not processed.

System Programmer Response: Report the error to the IBM support center.

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, and it is not available, either because ICSF is inactive or not configured properly or because the pkiserv daemon user ID does not have authority to use the key

Destination: Descriptor code is 6. Routing code is 2.

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 is operational. Follow the documentation pertaining to any issued messages having the "CSF" prefix. If ICH408I messages are issued for insufficient authority to CSFKEYS or CSFSERV class resources, then the pkiserv daemon user ID does not have authority to use the key. Give the user ID the requires access to the specified resource. To determine if the key you are using requires the PCI cryptographic coprocessor, see Chapter 17, "RACF administration for PKI Services" on page 199. For more information, see "Installing and configuring ICSF (optional)" on page 32, *z/OS ICSF System Programmer's Guide*, and *z/OS ICSF Administrator's Guide*.

IKYP002I **PKI SERVICES INITIALIZATION COMPLETE**

Explanation: PKI Services has just been started and has finished initializing.

Destination: Descriptor code is 6. Routing code is 2.

System Action: PKI Services processing continues.

IKYP003I **PKI SERVICES SHUTDOWN REQUESTED**

Explanation: An operator command was issued to stop PKI Services.

Destination: Descriptor code is 5. Routing code is 2.

System Action: PKI Services is stopped.

IKYP004I **LOG OPTION PROCESSED:** *log-option*

Explanation: A MODIFY operator command was issued to alter the current log setting for PKI Services.

Destination: Descriptor code is 5. Routing code is 2.

System Action: The log setting for PKI Services is changed as requested.

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.

Destination: Descriptor code is 5. Routing code is 2.

System Action: The MODIFY command is not processed. The log setting for PKI Services is unchanged.

System Programmer Response: Reenter the MODIFY command, specifying a correct log parameter. For more information, see "Changing logging options" on page 229.

**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.

Destination: Descriptor code is 5. Routing code is 2.

System Action: The MODIFY command is not processed. PKI Services continues processing unchanged.

System Programmer Response: Reenter the MODIFY command, specifying a supported PKI Services command. For more information, see “Stopping the PKI Services daemon” on page 86 and “Changing logging options” on page 229.

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.

Destination: Descriptor code is 5. Routing code is 2.

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 reenter the command.

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 85 and “Stopping the PKI Services daemon” on page 86.

**IKYP008E DIRECTORY POST UNSUCCESSFUL.
LDAP DATA LIBRARY MODULE RC =
nnnn**

Explanation: PKI Services background certificate processing is attempting to post information (certificate, CRL, and so forth) to a directory. The post was unsuccessful. The OCSF Data Library Module (LDAPDL) return code is displayed in the message.

Destination: Descriptor code is 6. Routing code is 2.

System Programmer Response: Determine the cause of the failure from the return code displayed and take appropriate action. These return codes are documented in *z/OS Open Cryptographic Services Facility Application Programming*. If the error is LDAPDL_NO_SUCH_OBJECT, the LDAP entry could not be created because the required suffix does not exist. Check the PKI Services log to determine the entry that could not be created. (Indicated on messages IKYC005I and IKYC008I.) If the entry should be posted to LDAP, you need to define the suffix in the LDAP

configuration file (slapd.conf) and recycle the LDAP server. For more information, see “Steps for installing and configuring LDAP” on page 30 and *z/OS Security Server LDAP Server Administration and Use*.

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 76.

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.

**IKYP009I PKI SERVICES IS STARTING, FMID
product-fmid**

Explanation: The START operator command was issued to start PKI Services. The START command could have been entered directly at the operator’s console or indirectly through a COMMNDxx parmlib member.

Destination: Descriptor code is 6. Routing code is 2.

System Action: PKI Services initialization proceeds.

**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.

Destination: Descriptor code is 6. Routing code is 2.

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.

System Action: PKI Services is stopped.

**IKYP011I PKI SERVICES ADDRESS SPACE
COULD NOT BE MADE
NON-SWAPPABLE: ERROR nnnn**

Explanation: The PKI Services daemon process is starting. Initialization processing is attempting to make the PKI Services address space non-swappable. The attempt was unsuccessful. The SYSEVENT TRANSWAP error code is displayed.

Destination: Descriptor code is 6. Routing code is 2.

System Programmer Response: Look up the error code for SYSEVENT TRANSWAP in *z/OS MVS Programming: Authorized Assembler Services*

Messages

Reference *SET-WTO* to determine what to do. Then, restart PKI Services.

System Action: PKI Services is stopped.

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.

Destination: Descriptor code is 6. Routing code is 2.

System Programmer Response: See documentation related to the service that failed. Make any necessary corrections. Then, restart PKI Services.

System Action: PKI Services is stopped.

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.

Destination: Descriptor code is 6. Routing code is 2.

System Programmer Response: Determine the failing service by examining the error code. The values are as follows

- 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 ETCRE 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 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.

System Action: PKI Services is stopped.

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 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.

System Action: PKI Services termination processing continues.

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 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.

System Action: PKI Services processing continues.

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.

Destination: Descriptor code is 6. Routing code is 2.

System Programmer Response: Look for other PKI Services log messages related to this error. For more information, see Chapter 19, "Using information from the PKI Services logs" on page 225.

System Action: PKI Services is stopped.

IKYP017I **PKI SERVICES IS ALREADY RUNNING**

Explanation: An attempt was made to start more than one instance of the PKI Services daemon.

Destination: Descriptor code is 6. Routing code is 2.

System Action: The first instance of PKI Services continues processing. The second instance is stopped.

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.

Destination: Descriptor code is 6. Routing code is 2.

System Programmer Response: Make the IKYPKID load module in SYS1.LINKLIB APF-authorized. Then, restart PKI Services.

System Action: PKI Services is stopped.

IKYP019I PKI Services dump created.

Explanation: PKI Services encountered a severe error during processing and has dumped the process (using the CEE3DMP callable service).

Operator Response: Contact your system programmer.

System Programmer Response: Examine the dump to determine the error. Contact your 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 85 and “Stopping the PKI Services daemon” on page 86.

System Action: PKI Services processing ends.

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).

Destination: Descriptor code is 6. Routing code is 2.

System Action: PKI Services processing continues.

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.

Destination: Descriptor code is 6. Routing code is 2.

System Action: PKI Services processing continues.

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.

Destination: Descriptor code is 6. Routing code is 2.

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*.

System Action: PKI Services initialization continues without automatic restart capability.

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 Programmer Response: Report the error to the IBM support center.

System Action: The settings are not displayed. PKI Services processing continues.

IKYP024I PKI SERVICES DUMPING FOR ABEND *abend-code RC nnnn*

Explanation: PKI Services has incurred an abend. The abend and reason codes are displayed.

Destination: Descriptor code is 6. Routing code is 2.

System Programmer Response: Use IPCS to examine the dump and diagnose the problem. Contact IBM support if necessary. Restart PKI Services after the error has been corrected.

System Action: PKI Services is stopped.

IKYP025I PKI SERVICES SETTINGS:

Explanation: A MODIFY operator command was issued to display the current settings for PKI Services. See Figure 37 on page 255 for the settings that are displayed.

Destination: Descriptor code is 5. Routing code is 2.

System Action: The settings are displayed. The possible subcomponent message levels 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

Messages

subcomponent message levels with the MODIFY operator command if desired. For more information, see “Changing logging options” on page 229.

IKYP026E PKI SERVICES CA CERTIFICATE EXPIRES ON *yyyy/mm/dd*

Explanation: The certificate that contains the PKI Services CA public key expires on the date shown.

Destination: Descriptor code is 6. Routing code is 2.

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 certificate authority certificate” on page 206 and *z/OS Security Server RACF Security Administrator’s Guide*.

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.

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_data1ib SAF callable service. The attempt failed. Message IKYS015I should also appear in the PKI Services log.

Destination: Descriptor code is 6. Routing code is 2.

System Programmer Response: You need to determine why the access failed. Look up the R_data1ib 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 17, “RACF administration for PKI Services” on page 199 and *z/OS Security Server RACF Security Administrator’s Guide*.

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.

IKYP028E PKI SERVICES DISTINGUISHED NAME OR KEY CHANGE ERROR

Explanation: PKI Services is starting. Initialization processing has retrieved the PKI Services signing certificate from the key ring assigned to PKI Services. The certificate is incompatible with certificate processing that has previously transpired. The subject’s distinguished name or the public key or both differ from the original values provided when you first configured PKI Services. The original values cannot be changed without reconfiguring PKI Services.

Destination: Descriptor code is 6. Routing code is 2.

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. Make any required changes. Then, restart PKI Services. For more information, see Chapter 17, “RACF administration for PKI Services” on page 199 and *z/OS Security Server RACF Security Administrator’s Guide*.

System Action: PKI Services is stopped.

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 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 85.

System Action: The PKI Services daemon halts its initialization and stops after displaying this message to the standard output (STDOUT) of the process.

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 is stopped.

System Programmer Response: Look up the error code in either *z/OS Open Cryptographic Services Facility Application Programming* or *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming*. Diagnose the problem indicated by the return code. Restart PKI Services after corrections are made.

IKYS002I Error nnnn in OCSF-API-name

Explanation: PKI Services is calling an OCSF or OCEP API. The invocation has failed. The API name and error code encountered are displayed.

System Action: If the error occurs during PKI Services initialization, PKI Services is stopped. Otherwise, PKI Services continues processing. However, needed cryptographic services may 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 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 SecureWay Security Server 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 may indicate a problem with the certificate stored in the SAF key ring or it may be an internal error.

System Action: PKI Services is stopped.

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 17, "RACF administration for PKI Services" on page 199 and *z/OS Security Server RACF Security Administrator's Guide*.

**IKYS004I Error nnnn in opening key ring
key-ring-name**

Explanation: PKI Services is initializing and is calling OCSF to open the SAF key ring containing the CA certificate. The open failed. The key ring name and OCSF or OCEP error code encountered is displayed.

System Action: PKI Services is stopped.

System Programmer Response: Look up the error code in either *z/OS Open Cryptographic Services Facility Application Programming* or *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming*. Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

IKYS005I Error nnnn in closing key ring

Explanation: PKI Services is terminating and is invoking OCSF to close the SAF key ring containing the CA certificate. The close failed. The OCSF or OCEP error code encountered is displayed.

System Action: PKI Services continues termination.

System Programmer Response: Look up the error code in either *z/OS Open Cryptographic Services Facility Application Programming* or *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application 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 is stopped.

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 58.

Messages

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 may indicate a problem with the certificate and private key stored in the SAF key ring or it may be an internal error.

System Action: PKI Services is stopped.

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 17, "RACF administration for PKI Services" on page 199 and *z/OS Security Server RACF Security Administrator's Guide*.

IKYS009I Profile for key ring *key-ring-name* not found

Explanation: PKI Services is reading its configuration file to locate the value specified for "KeyRing" in the "SAF" section. The key ring specified is incorrect. No such key ring exists.

System Action: PKI Services is stopped.

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 58.

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 does not have a private key.
- User ID assigned to the PKI Services daemon does not have permission to read the key ring or private key.

System Action: PKI Services is stopped.

System Programmer Response: Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 5, "Running IKYSETUP to perform RACF administration" on page 37 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 is stopped.

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 is stopped.

System Programmer Response: Report the error to the IBM support center.

IKYS013I Cannot find the private key associated with the default certificate

Explanation: PKI Services is attempting to retrieve data from the SAF key ring specified by the "KeyRing" value in the "SAF" section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. Possible problems are:

- Key ring is empty.
- CA certificate in the key ring not connected as PERSONAL DEFAULT.
- CA certificate in key ring does not have a private key.
- User ID assigned to the PKI Services daemon does not have permission to read the key ring or private key.

System Action: PKI Services is stopped.

System Programmer Response: Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 5, "Running IKYSETUP to perform RACF administration" on page 37 and *z/OS Security Server RACF Security Administrator's Guide*.

IKYS014I Cannot find the default certificate with private key associated in key ring

Explanation: PKI Services is attempting to retrieve data from the SAF key ring specified by the "KeyRing" value in the "SAF" section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. Possible problems are:

- Key ring is empty.

- CA certificate in the key ring not connected as PERSONAL DEFAULT.
- CA certificate in key ring does not have a private key.
- User ID assigned to the PKI Services daemon does not have permission to read the key ring or private key.

System Action: PKI Services is stopped.

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 certificate and key ring” on page 203 and *z/OS Security Server RACF Security Administrator’s Guide*.

IKYS015I RACF callable service, R_datalib, with function code nnnn returns with SAF return code=nnnn, RACF return code=nnnn, RACF reason code=nnnn

Explanation: PKI Services is attempting to retrieve data from the SAF key ring specified by the “KeyRing”

value in the “SAF” section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. Possible problems are:

- Key ring is empty.
- CA certificate in the key ring not connected as PERSONAL DEFAULT.
- CA certificate in key ring does not have a private key.
- User ID assigned to the PKI Services daemon does not have permission to read the key ring or private key.

System Action: PKI Services is stopped.

System Programmer Response: Look up the return and reason code displayed in *z/OS Security Server RACF Callable Services*. Make corrections as needed. Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 17, “RACF administration for PKI Services” on page 199 and *z/OS Security Server RACF Security Administrator’s Guide*.

```

IKYP025I PKI SERVICES SETTINGS:
SUBCOMPONENT          MESSAGE LEVEL
LDAP                   {current-message-level-for-subcomponent}
SAF                     {current-message-level-for-subcomponent}
DB                     {current-message-level-for-subcomponent}
CORE                   {current-message-level-for-subcomponent}
PKID                   {current-message-level-for-subcomponent}
POLICY                 {current-message-level-for-subcomponent}
MESSAGE LOGGING SETTING: {STDERR_LOGGING | STDOUT_LOGGING}
CONFIGURATION FILE IN USE:
{full-UNIX-pathname-of-configuration-file-being-used}
    
```

Figure 37. Settings that IKYP025I displays

Chapter 22. File directory structure

This chapter discusses the location of files in:

- z/OS product libraries
- HFS directory `/usr/lpp/pkiserv/` and its subdirectories.

Product libraries

SMP/E installs PKI Services into the following product libraries:

- SAMPLIB/ASAMPLIB
 - IKYCVSAM
 - IKYRVSAM
 - IKYSETUP
 - IKYISMKD
 - IKYMKDIR
 - IKYALLOC
 - IKYDDDEF
- PROCLIB/APROCLIB
 - IKYSPROC with alias PKISERVD
- LINKLIB/ALINKLIB
 - IKYPKID - The PKI Services daemon
 - IKYPRTM - The Resource Termination Manager for the daemon

HFS directory and subdirectories

Additionally, unless you change the default, SMP/E installs PKI Services into the HFS directory `/usr/lpp/pkiserv`. The following table describes the directory structure and contents:

Table 63. Files contained in subdirectories

Subdirectory	Contains File
bin	Utilities executables: <ul style="list-style-type: none">• iclview — Utility for viewing issued certificate list (certificate database). (For more information, see Chapter 20, “Using PKI Services utilities” on page 231.)• pkitp_install — Program to register the PKI Services Trust Policy plug-in with OCSF. (For more information, see “Configuring and getting started with PKITP” on page 341.)• pkitp_ivp — Program to verify that the PKI Services Trust Policy plug-in installed successfully. (For more information, see “Configuring and getting started with PKITP” on page 341.)• vosview — Utility for viewing VSAM object store (request database). (For more information, see Chapter 20, “Using PKI Services utilities” on page 231.)
include	C header files: <ul style="list-style-type: none">• pkitp.h — C language header file for writing application programs that use the PKI Trust Policy Plug-in. (For more information, see “Files for PKITP” on page 340.)

File directory structure

Table 63. Files contained in subdirectories (continued)

lib	<p>Loadable files:</p> <ul style="list-style-type: none">• pkitp.so — OCSF Trust Policy plug-in for PKI Services. (For more information, see “Files for PKITP” on page 340.)• *.dll — Dynamic Link Libraries (DLLs) that the PKI Services daemon uses.• nls/msg/En_US.IBM-1047/*.cat - The PKI Services message catalogs. (These message catalogs are also symbolically linked in the /usr/lpp/pkiserv/lib/nls/msg/C directory as well as the /usr/lib/nls/msg/En_US.IBM-1047 and /usr/lib/nls/msg/C directories.)
PKIServ	<p>CGIs that make up the PKIServ Web application. (For information about CGIs, see “Relationship between CGIs and the pkiserv.tmpl file” on page 114 and Table 41 on page 133.)</p> <p>PKIServ contains the following subdirectories:</p> <ul style="list-style-type: none">• public-cgi — Public (non-SSL) directory• ssi-cgi-bin — SSL-protected<ul style="list-style-type: none">– 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).<ul style="list-style-type: none">– auth — SSL with client certificate protection. Work runs under administrator’s ID.

Table 63. Files contained in subdirectories (continued)

samples	<p>Various sample files, including:</p> <ul style="list-style-type: none"> • expiringmsg.form — The e-mail message sent to a user as notification about an certificate that will expire • httpd.conf — Contains z/OS HTTP Server directives. (For a code sample, see “z/OS HTTP Server configuration directives” on page 327.) • httpd2.conf — Contains z/OS HTTP Server directives for the second webserver. (For a code sample, see “z/OS HTTP Server configuration directives” on page 327.) • 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” on page 307.) • Makefile.pkixit — The makefile for the PKI Services exit. (For more information, see “Steps for updating the exit code sample” on page 142.) • Makefile.pkitpsamp — The makefile for pkitpsamp.c, which is a sample application to call the PKI Trust Policy plug-in. (For more information, see “Files for PKITP” on page 340.) • pkixit.c — The sample PKI Services exit, which PKI Services provides. (For more information, see “Steps for updating the exit code sample” on page 142.) • pkiserv.envars — The PKI Services environment variables file. (For more information, see “Optionally updating PKI Services environment variables” on page 55 and “The pkiserv.envars environment variables file” on page 307.) • pkiserv.tmpl — The PKI Services certificate templates file. (For more information, see Chapter 12, “Customizing the end-user Web application” on page 91. For a code sample, see Chapter 24, “The pkiserv.tmpl certificate templates file” on page 263.) • pkiserv.conf — The PKI Services configuration file. (For more information, see “(Optional) Steps for updating the configuration file” on page 58 and Chapter 23, “The pkiserv.conf configuration file” on page 261.) • pkitpsamp.c — Sample application to call the PKI Trust Policy plug-in. (For more information, see “Files for PKITP” on page 340 and “Providing the certificate validation service” on page 347.) • readymsg.form — The e-mail message sent to a user as notification a certificate is ready for retrieval • rejectmsg.form — The e-mail message sent to a user as notification a request for a certificate has been rejected
---------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

File directory structure

Chapter 23. The pkiserv.conf configuration file

This chapter 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 58 and Table 20 on page 58.

The example that follows might not be identical to the code shipped with the product. If you want to see the exact code, look at the pkiserv.conf file in the source directory /usr/lpp/pkiserv/samples/.

```
# Licensed Materials - Property of IBM
# 5694-A01
# (C) Copyright IBM Corp. 2001, 2002
# Status = HKY7707

[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
sha-1WithRSAEncryption=1.2.840.113549.1.1.5
id-dsa-with-sha1=1.2.840.10040.4.3
MyPolicy=1.2.3.4

[ObjectStore]
ObjectDSN='pkisrvd.vsam.ost'
ObjectTidDSN='pkisrvd.vsam.ost.path'
ICLDSN='pkisrvd.vsam.icl'
RemoveCompletedReqs=1w
RemoveInactiveReqs=4w
# 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
PolicyRequired=F
PolicyCritical=F
PolicyName1=MyPolicy
Policy1Org=MyOrganization
Policy1Notice1=3
Policy1Notice2=17
UserNoticeText1=This is some very lawyerly statement....
CPS1=http://www.mycompany.com/cps.html
```


Chapter 24. The pkiserv.tpl certificate templates file

This chapter includes a code sample of the pkiserv.tpl certificate templates file. (For a description of the main sections and subsections of pkiserv.tpl, see “Contents of the pkiserv.tpl certificates templates file” on page 91.) The example that follows might not be identical to the code shipped with the product. To view the most current code, see the pkiserv.tpl certificate template file in the source directory /usr/lpp/pkiserv/samples/.

```
#
# =====
#
# COMPONENT_NAME: pkiserv.tpl
#
# Licensed Materials - Property of IBM
# 5694-A01
# (C) Copyright IBM Corp. 2001, 2002
# Status = HKY7707
#
# =====
#
# Configuration file for interfacing with R_PKIServ. This file may be
# customized as required by the installation. Any line with an '#' in
# column 1 is considered a comment.
#
# Structure:
#
# The file contains a mixture of true HTML and HTML like tags. The
# main tags divide the file into sections, APPLICATION, TEMPLATE,
# and INSERT, where APPLICATION and TEMPLATE may contain various
# subsections, named fields, and substitution variables as explained
# below.
#
# <APPLICATION NAME=appl-name> ... </APPLICATION>
#
# This section identifies the applications that will make use of
# PKI Services for Z/OS. The product ships with one application
# defined, "PKISERV". This section may contain the following subsections:
#
# <CONTENT> ... </CONTENT>
#
# This subsection contains the HTML to be presented to the end
# user requesting and retrieving certificates
#
# The subsection should contain one or more named fields
# identifying certificate templates to be used for requesting
# or managing certificates through this application. (See below
# for a description of named fields.) These template names
# should match the HTML selection value associated with them.
#
# <RECONTENT> ... </RECONTENT>
#
# This subsection contains the HTML which will display the
# certificate details so that the end user may confirm that
# that is the certificate to be renewed or revoked. This will
# make use of a new substitution variable, [printablecert],
# which contains the data extracted from the ICL entry.
#
# <RESUCCESSCONTENT> ... </RESUCCESSCONTENT>
#
# This subsection contains the HTML to be presented to the end
# user when the certificate revoke request
# was submitted successfully.
# Any named fields in this subsection are interpreted as
# content inserts defined by INSERT sections. For PKISERV, the
```

The pkiserv.tmpl certificate templates file

```
#      INSERT sections are included as part of the HTML presented
#      to the end user.
#
#      <REFAILURECONTENT> ... </REFAILURECONTENT>
#
#      This subsection contains the HTML to be presented to the end
#      user when the certificate renew/revoke 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.
#
#      <ADMINHEADER> ... </ADMINHEADER>
#
#      This subsection contains the general insallation specific HTML
#      content for the header of all admin pages.
#
#      <ADMINFOOTER> ... </ADMINFOOTER>
#
#      This subsection contains the general insallation specific HTML
#      content for the footer of all admin pages.
#
#
#      <TEMPLATE NAME=tmp1-name> ... </TEMPLATE>
#      <TEMPLATE NAME=tmp1-name alias>
#      <NICKNAME=nick-name>
#
#      This section defines the certificate templates referenced in the
#      APPLICATION sections. You may refer to a single template by
#      more than one name using alias. Also since the template name
#      needs to be recalled in order to renew a certificate, it will
#      need to be stored with the certificate. The nick name of the
#      template will serve this purpose.
#
#      Applicable subsections are:
#
#      <CONTENT> ... </CONTENT>
#
#      This subsection contains the HTML to be presented to the end
#      user requesting certificates of this type. Any named fields
#      in this subsection are interpreted as certificate field names
#      defined by INSERT sections. (See below for a description of
#      named fields.) For PKISERV, the INSERT sections
#      are included as part of the HTML presented to the end user.
#      (i.e., the end user provides values for these fields.)
#      Named fields in this subsection are considered optional if
#      the named field contains more that one word within the %%
#      delimiters, e.g., %%AltName (Optional)%%. The user need not
#      supply a value for AltName
#
#      <APPL> ... </APPL>
#
#      This subsection identifies certificate fields that the
#      application itself should provide values for. This subsection
#      should contain named fields only, one per line. Currently,
#      the only supported named field allowed in this section is
#      "UserId"
#
#      <CONSTANT> ... </CONSTANT>
#
#      This subsection identifies certificate fields that have a
#      constant (hardcoded) value for everyone. This subsection
#      should contain named fields only, one per line. The syntax
#      for specifying the values is %%field-name=field-value%%,
#      e.g., %%KeyUsage=handshake%%
#
#      <SUCCESSCONTENT> ... </SUCCESSCONTENT>
#
```

The pkiserv.tmpl certificate templates file

```
# This subsection contains the HTML to be presented to the end
# user when 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 presented
# to the end user.
#
# <FAILURECONTENT> ... </FAILURECONTENT>
#
# This subsection contains the HTML to be presented to the end
# user 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.
#
# <RETRIEVECONTENT> ... </RETRIEVECONTENT>
#
# This subsection contains the HTML to be presented to the end
# user to enable certificate retrieval.
# 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.
#
# <RETURNCERT> ... </RETURNCERT>
#
# This subsection contains the HTML to be presented to the
# enduser upon successful certificate retrieval. For PKISERV, if
# the certificate being retrieved is a browser certificate, then
# this section must contain a single line containing a browser
# qualified INSERT name, e.g., %returnbrowsercert
# [browserstype]%. Additionally, INSERTs for Netscape
# (returnbrowsercertNS) and Internet Exploror (returnbrowsercertIE)
# containing browser specific HTML for returning certificates must
# be defined elsewhere in the configuration file. If the
# certificate being retrieved is a server certificate, this section
# should contain the HTML necessary to present the certificate
# to the user as text
#
# <INSERT NAME=insert-name> ... </INSERT>
# This section contains HTML that either describes a certificate
# field or defines other common HTML that may be referenced in
# the TEMPLATE sections. INSERTs are referenced elsewhere by
# using a named field of the form %%insert-name%%
#
# Named Fields - Delineated with %, e.g., %%Label%. Their meaning
# is specific to the section they are contained in. Named fields
# are case sensitive. Named fields are also usin to reference common
# includeable HTML. Note, PKISERV treats named fields that begin with
# a dash as just includeable code. Any special meaning a named field
# may have, given the section its contained in, is ignored if it
# begins with a dash. For example, if %%-pagefooter%% was specified
# in a TEMPLATE CONTENT section, -pagefooter would not be considered
# a certificate field name. However, the INSERT with the name
# -pagefooter would be included in the HTML page presented to the
# end user.
#
# Substitution Variables - Delineated with square brackets, e.g.,
# [base64cert]. They represent variables that get replaced with
# an actual value at run time. Substitution variables are case
# sensitive. The valid substitution variables are:
#
# transactionid - Unique value returned from a certificate request.
#
# tmplname - Certificate template name. Primed from the HTML tag
# <SELECT NAME="Template"> in the <APPLICATION NAME=PKISERV>
# section. This is selected by the end user on the first web page.
```

The pkiserv.tmpl certificate templates file

```
#
#   iecert - The requested certificate in a form the Microsoft
#           Internet Explorer accepts.
#
#   base64cert - The requested base64 encoded certificate.
#
#   browsertype - Special substitution variable to be used to qualify
#   named field only. Its use enables the different browsers,
#   Netscape and Internet Explorer, to perform browser specific
#   operations, i.e., Netscape uses a KEYGEN HTML tag to generate a
#   public/private key pair while Internet Explorer uses ACTIVEX
#   controls. For example, if %%PublicKey[browsertype]%% was
#   specified in a TEMPLATE CONTENT section referenced by a user
#   with the Netscape Navigator browser then INSERT PublicKeyNS
#   would be included. Likewise, if the users browser was the
#   Microsoft Internet Explorer, INSERT PublicKeyIE would be included.
#
#   optfield - Special substitution variable that should be placed in
#   any certificate field name INSERT where the value may be supplied
#   by the end user. It enables the field to be displayed as optional
#   if desired.
#
#   printablecert - Summary information about the certificate to be
#                   renewed/revoked, such as issuer's name, subject's
#                   name...
#
#   errorinfo - Information about the failing SAF call such as the
#               return code and reason code.
#
# Note, depending on where a substitution variable is used, it may
# not have a valid meaning, e.g., base64cert would be meaningless
# prior to the certificate being retrieved. The value of
# [base64cert] would be the empty string (aka NULL) in this case.
#
# =====
# Application - PKISERV
#
# The installation should customize the CONTENT and ADMINCONTENT
# subsections as appropriate
#
# =====
<APPLICATION NAME=PKISERV>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Application </TITLE>
%%-copyright%%
</HEAD>
<BODY>
| <H1>PKI Services Certificate Generation Application</H1>
<p>
<A HREF="/PKIServ/cacerts/cacert.der">Install
  our CA certificate into your browser </A>
<H2>Choose one of the following:</H2>
<ul>
<li><h3>Request a new certificate using a model</h3>
<FORM name=mainform METHOD=GET ACTION="/PKIServ/ssl-cgi/catmpl.rexx">
<p> Select the certificate template to use as a model
<SELECT NAME="Template">
  %%1 Year PKI SSL Browser Certificate%%
  <OPTION>1 Year PKI SSL Browser Certificate
  %%1 Year PKI S/MIME Browser Certificate%%
  <OPTION>1 Year PKI S/MIME Browser Certificate
  %%2 Year PKI Browser Certificate For Authenticating To z/OS%%
  <OPTION>2 Year PKI Browser Certificate For Authenticating To z/OS
  %%5 Year PKI SSL Server Certificate%%
  <OPTION>5 Year PKI SSL Server Certificate
```

The pkiserv.tmpl certificate templates file

```
%%5 Year PKI IPSEC Server (Firewall) Certificate%%
<OPTION>5 Year PKI IPSEC Server (Firewall) Certificate
%%5 Year PKI Intermediate CA Certificate%%
<OPTION>5 Year PKI Intermediate CA Certificate
%%1 Year SAF Browser Certificate%%
<OPTION>1 Year SAF Browser Certificate
%%1 Year SAF Server Certificate%%
<OPTION>1 Year SAF Server Certificate
</SELECT>
<p>
<INPUT TYPE="submit" VALUE="Request Certificate">
</FORM>
<li><h3>Pick up a previously requested certificate</h3>
<FORM name=selform METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<p> Enter the assigned transaction ID
<INPUT NAME="TransactionId" TYPE="text" SIZE=56 maxlength="56">
<br>Select the certificate return type
<SELECT NAME="Template">
%%PKI Browser Certificate%%
<OPTION>PKI Browser Certificate
%%PKI Server Certificate%%
<OPTION>PKI Server Certificate
%%SAF Browser Certificate%%
<OPTION>SAF Browser Certificate
%%SAF Server Certificate%%
<OPTION>SAF Server Certificate
</SELECT>
<p>
<INPUT TYPE="submit" VALUE="Pick up Certificate">
</FORM>
<li><h3>Renew or revoke a previously issued browser certificate</h3>
<FORM name=selform METHOD=GET ACTION="/PKIServ/clientauth-cgi/cadisplay.rexx">
<p>
<SCRIPT LANGUAGE="JavaScript">
<!--
function RenewRevokeAlert(){
var STRING_RenewRevokePrompt=
    "You will be prompted by the browser to select " +
    "the certificate you want to renew or revoke. " +
    "Once you select the certificate you will be " +
    "given the opportunity to confirm your selection. " +
    "Note that you can only renew or revoke a single " +
    "certificate per one browser session. If you wish " +
    "to renew or revoke another certificate, you must " +
    "close your browser and restart it.";
    alert(STRING_RenewRevokePrompt);
    return true;
}

function ValidateEntry(){
var STRING_MissingFieldPrompt=
    "Enter the required field."
var STRING_MissingConfirmPwdPrompt=
    "Reenter password."
var STRING_UnmatchPwdPrompt=
    "The passwords do not match. Enter again."
if(document.renform.PassPhrase.value=="") {
alert(STRING_MissingFieldPrompt);
document.renform.PassPhrase.focus();
return true;
}
else if(document.renform.ConfirmPassPhrase.value=="") {
alert(STRING_MissingConfirmPwdPrompt);
document.renform.ConfirmPassPhrase.focus();
return true;
}
else if(document.renform.PassPhrase.value!=
    document.renform.ConfirmPassPhrase.value){
```

The pkiserv.tmpl certificate templates file

```
alert(String_UnmatchPwdPrompt);
document.renform.ConfirmPassPhrase.focus();
return true;
}
else {
return false;
}
}
//-->
</SCRIPT>
<INPUT TYPE="submit" VALUE="Renew or Revoke Certificate"
onClick="return RenewRevokeAlert()">
</FORM>
<li><h3>Administrators click here</h3>
# 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 Administration Page">
</FORM>
</ul>
<p> %%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<RECONTENT>
<HTML><HEAD>
<TITLE> PKISERV Renew or Revoke a Browser Certificate </TITLE>
%%-copyright%%
</HEAD>
<BODY>
<H1>Renew or Revoke a Browser Certificate</H1>
<h3>Here is the certificate you selected:</h3>
<p>
[printablecert]
<h2>If this is the correct certificate, choose one of the following:</h2>
<b>(otherwise you need to restart your browser to pick another certificate)</b>
<ul>
<h3><li>Renew the above certificate</h3>
<FORM name=renform METHOD=POST
ACTION="/PKIServ/clientauth-cgi/camodify.rexx">

<SCRIPT LANGUAGE="JavaScript">
<!--

function ValidateEntry(){
var STRING_MissingFieldPrompt=
    "Enter the required field."
var STRING_MissingConfirmPwdPrompt=
    "Reenter password."
var STRING_UnmatchPwdPrompt=
    "The passwords do not match. Enter again."
if(document.renform.PassPhrase.value=="") {
alert(STRING_MissingFieldPrompt);
document.renform.PassPhrase.focus();
return true;
}
else if(document.renform.ConfirmPassPhrase.value=="") {
alert(STRING_MissingConfirmPwdPrompt);
document.renform.ConfirmPassPhrase.focus();
return true;
}
else if(document.renform.PassPhrase.value!=
document.renform.ConfirmPassPhrase.value){
alert(STRING_UnmatchPwdPrompt);
document.renform.ConfirmPassPhrase.focus();
```

```

return true;
}
else {
return false;
}
}
//-->
</SCRIPT>
<INPUT NAME="action" TYPE="hidden" VALUE="renew">
| %%NotifyEmail (optional)%%
%%PassPhrase%%
<p>
<INPUT TYPE="submit" VALUE="Renew" onClick=
"if(ValidateEntry()) return false; else return true;">
</FORM>
<h3><li>Revoke the above certificate</h3>
<FORM name=revform METHOD=POST
ACTION="/PKIServ/clientauth-cgi/camodify.rexx">
<INPUT NAME="action" TYPE="hidden" VALUE="revoke">
<INPUT TYPE="submit" VALUE="Revoke">
<SELECT NAME="reason">
<OPTION Selected VALUE="0">No Reason
<OPTION VALUE="1">User key was compromised
<OPTION VALUE="2">CA key was compromised
<OPTION VALUE="3">User changed affiliation
<OPTION VALUE="4">Certificate was superseded
<OPTION VALUE="5">Original use no longer valid
</SELECT>
</ul>
</FORM>
<p>
<FORM METHOD=GET ACTION="/PKIServ/public-cgi/camain.rexx">
<center>
<INPUT TYPE="submit" VALUE="Home Page">
</FORM>
</center>
<p> %%-pagefooter%%
</BODY>
</HTML>
</RECONTENT>
<RESUCCESSCONTENT>
%%-renewrevokeok%%
</RESUCCESSCONTENT>
<REFAILURECONTENT>
%%-renewrevokebad%%
</REFAILURECONTENT>
<ADMINHEADER>
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Administration </TITLE>
%%-copyright%%
</HEAD>
<BODY>
</ADMINHEADER>
<ADMINFOOTER>
<p> %%-pagefooter%%
</BODY>
</HTML>
</ADMINFOOTER>
</APPLICATION>

#
# =====
#
# Sample Templates - Browser and Server Certificate Requesting
#
# =====
#

```

The pkiserv.tmpl certificate templates file

```
# Template Name - 1 Year SAF Server Certificate
#
# Function - Allows end users to request certificates for servers
# using native SAF certificate generation facilities. The end user
# may provide values for any of the following fields:
#
# CommonName - optional
# OrgUnit - required
# Org - required
# Locality - optional
# StateProv - optional
# Country - required
# AltEmail - optional
# AltDomain - optional
# AltURI - optional
# AltIPAddr - optional
# Label - required
# PublicKey - required (This is the PKCS#10 request)
#
# PKISERV will provide the authenticated client UserId. The certificate
# will be used for handshaking only (e.g., SSL) and is good for 1
# year. The CERTAUTH certificate with Label "Local SAF CA" will be
# used for signing the certificate
#
# =====
#
<TEMPLATE NAME=1 Year SAF Server Certificate>
<TEMPLATE NAME=SAF Server Certificate>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based SAF Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingRequiredFAlert(){
var STRING_MissingRequiredFPrompt=
    "Enter the field value that's not optional."

if ((document.serverform.Label.value=="")||
    (document.serverform.OrgUnit.value=="")||
    (document.serverform.Org.value=="")||
    (document.serverform.Country.value=="")||
    (document.serverform.PublicKey.value==""))
{
    alert(STRING_MissingRequiredFPrompt);
    if (document.serverform.OrgUnit.value=="")
        document.serverform.OrgUnit.focus();
    else if (document.serverform.Org.value=="")
        document.serverform.Org.focus();
    else if (document.serverform.Country.value=="")
        document.serverform.Country.focus();
    else if (document.serverform.Label.value=="")
        document.serverform.Label.focus();
    else
        document.serverform.PublicKey.focus();

    return true;
}
else {
return false;
}
}
//-->
</SCRIPT>

</HEAD>
<BODY>
<H1> SAF Server Certificate 1 Year (Auto Approved)</H1>
```

The pkiserv.tmp certificate templates file

```
<p>
<H2>Choose one of the following:</H2>
<p>
<u1>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME=serverform 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=serverform 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=serverform METHOD=POST ACTION=
#          "/PKIServ/ssl-cgi-bin/careq.rexx" onSubmit=
#          "if(MissingRequiredFAlert()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<p> Enter values for the following field(s)
  %%CommonName (Optional)%%
  %%OrgUnit%%
  %%OrgUnit2 (Optional)%%
  %%Org%%
  %%Locality (Optional)%%
  %%StateProv (Optional)%%
  %%Country%%
  %%AltEmail (Optional)%%
  %%AltDomain (Optional)%%
  %%AltURI (Optional)%%
  %%AltIPAddr (Optional)%%
  %%Label%%
  %%PublicKey%%
<p>
<INPUT TYPE="submit" VALUE="Submit certificate request"
ONCLICK="if(MissingRequiredFAlert()) return false; else return true;">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>

<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</u1>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<APPL>
  %%UserId%%
</APPL>
<CONSTANT>
  %%KeyUsage=handshake%%
  %%NotAfter=365%%
  %%SignWith=SAF:CERTAUTH/taca%%
</CONSTANT>
<SUCCESSCONTENT>
  %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
  %%-requestbad%%
</FAILURECONTENT>
```

The pkiserv.tmpl certificate templates file

```
<RETRIEVECONTENT>
<HTML><HEAD>
<TITLE> Web Based SAF Certificate Generation Application Pg 3</TITLE>
%%-copyright%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingTransIdAlert(){
var STRING_MissingTransIdPrompt=
    "Enter the transaction ID assigned to the certificate.";
if(document.retrieveform.TransactionId.value==""){
    alert(STRING_MissingTransIdPrompt);
    document.retrieveform.TransactionId.focus();
    return true;
}
else {
    return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
<FORM NAME=retrieveform METHOD=GET ACTION=
    "/PKIServ/ssl-cgi/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=GET ACTION=
#    "/PKIServ/ssl-cgi/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
#<FORM NAME=retrieveform METHOD=GET ACTION=
#    "/PKIServ/ssl-cgi/cagetcert.rexx" onSubmit=
#        "if(MissingTransIdAlert()) return false; else return true;">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
%%TransactionId%%
<p>
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%-returnpkcs10cert%%
</RETURNCERT>
</TEMPLATE>

#
# =====
#
# Template Name - 1 Year SAF Browser Certificate
#
# Function - Allows end users to request certificates for their
# browsers using native SAF certificate generation facilities. The end
# user may provide values for any of the following fields:
#
```

The pkiserv.tmp certificate templates file

```
# Label - required
# PublicKey - required (Provided by the browser itself)
#
# PKISERV will provide the authenticated client UserId. The certificate
# will be used for handshaking only (e.g., SSL) and is good for 1
# year. The CERTAUTH certificate with Label "Local SAF CA" will be
# used for signing the certificate. The Subject's Distinguished Name
# will be formed as:
#
# C=US/O=The Firm/OU=SAF template certificate/
#       OU=Nuts and Bolts Division/CN=<determined by SAF>
#
# The presence of CommonName without a value tells SAF to determine
# the CN value from the PGMRNAME field of the user's USER profile.
# See the RACF Callable Services Guide for more information
#
# =====
#
<TEMPLATE NAME=1 Year SAF Browser Certificate>
<TEMPLATE NAME=SAF Browser Certificate>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based SAF Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
%%-AdditionalHead[browsertype]%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingRequiredFAlert(){
var STRING_MissingRequiredFPrompt=
    "Enter the field that's required."
if(document.CertReq.Label.value==""){
alert(STRING_MissingRequiredFPrompt);
document.CertReq.Label.focus();
return true;
}
else {
return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> SAF Browser Certificate 1 Year (Auto Approved)</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#       "/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=
#       "if(MissingRequiredFAlert()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmp1name]">
<p> Enter values for the following field(s)
```

The pkiserv.tmpl certificate templates file

```
%%Label%%
%%PublicKey[browsertype]%%
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<APPL>
%%UserId%%
</APPL>
<CONSTANT>
%%KeyUsage=handshake%%
%%NotAfter=365%%
%%OrgUnit=SAF template certificate%%
%%OrgUnit=Nuts and Bolts Division%%
%%Org=The Firm%%
%%Country=US%%
%%SignWith=SAF:CERTAUTH/taca%%
%%CommonName=%%
</CONSTANT>
<SUCCESSCONTENT>
%%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
%%-requestbad%%
</FAILURECONTENT>
<RETRIEVECONTENT>

<HTML><HEAD>
%%-copyright%%
<TITLE> Web Based SAF Certificate Generation Application Pg 3</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingTransIdAlert(){
var STRING_MissingTransIdPrompt=
"Enter the transaction ID assigned to the certificate.";
if(document.retrieveform.TransactionId.value==""){
alert(STRING_MissingTransIdPrompt);
document.retrieveform.TransactionId.focus();
return true;
}
else {
return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
<FORM NAME=retrieveform METHOD=GET ACTION=
```

The pkiserv.tmp certificate templates file

```
"/PKIServ/ssl-cgi/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=GET ACTION=
# "/PKIServ/ssl-cgi/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
#<FORM NAME=retrieveform METHOD=GET ACTION=
# "/PKIServ/ssl-cgi/cagetcert.rexx" onSubmit=
# "if(MissingTransIdAlert()) return false; else return true;">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
  %%TransactionId%%
<p>
<INPUT TYPE="submit" VALUE="Retrieve and Install Certificate">
</FORM>
<p>
<H2>To check that your certificate installed properly, follow the
procedure below:</h2>
<p><B>Netscape V6</B> - Click Edit->Preferences, then Privacy and Security->
Certificates. Click the Manage Certificates button to start the Certificate Manager.
Your new certificate should appear in the Your Certificates list.
Select it then click View to see more information.
<p><B>Netscape V4</B> - Click the Security button, then Certificates->
Yours. Your certificate should appear in the list. Select it then
click Verify.
<p><B>Internet Explorer V5</B> - Click Tools->Internet Options, then
Content, Certificates.
Your certificate should appear in the Personal list. Click Advanced to
see additional information.
<p>
<FORM METHOD=GET ACTION="/PKIServ/public-cgi/camain.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Home page">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>

#
#
# =====
#
# Template Name - 1 Year PKI SSL Browser Certificate
#
# Function - Creates a 1 year certificate good for general SSL client
# authentication using a browser. If approved, the
# certificate becomes valid after it's requested.
# (You may delay the valid date by specifying a non zero
# number for the value of 'NotBefore',
# eg. NotBefore=5. That means if the request is approved,
# the certificate will become valid 5 days after it's
# requested.)
#
# These certificates will be stored in LDAP if The O= and
# OU= suffixes have already been created
#
# Other than the user input fields, all other information is hard coded.
#
# User input fields:
# CommonName - required
# Requestor - optional
# PassPhrase - required
```

The pkiserv.tmpl certificate templates file

```
# PublicKey - required (Provided by the browser itself)
# NotifyEmail - optional
# Email - optional
#
# RACF userid/password authentication : not require
# Administrator approval             : require
#
# =====
#
<TEMPLATE NAME=1 Year PKI SSL Browser Certificate>
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=1YBSSL>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
%%-AdditionalHead[browsertype]%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(){
var STRING_MissingFieldPrompt=
    "Enter the required field."
var STRING_MissingConfirmPwdPrompt=
    "Reenter password."
var STRING_UnmatchPwdPrompt=
    "The passwords do not match. Enter again."
var STRING_UnmatchEmailPrompt=
    "The Email addresses for Distinguished name and notification do not match. Enter again."
if(document.CertReq.CommonName.value=="") {
alert(STRING_MissingFieldPrompt);
document.CertReq.CommonName.focus();
return true;
}
else if(document.CertReq.PassPhrase.value=="") {
alert(STRING_MissingFieldPrompt);
document.CertReq.PassPhrase.focus();
return true;
}
else if(document.CertReq.ConfirmPassPhrase.value=="") {
alert(STRING_MissingConfirmPwdPrompt);
document.CertReq.ConfirmPassPhrase.focus();
return true;
}
else if(document.CertReq.PassPhrase.value!=
    document.CertReq.ConfirmPassPhrase.value){
alert(STRING_UnmatchPwdPrompt);
document.CertReq.ConfirmPassPhrase.focus();
return true;
}
else if((document.CertReq.Email.value != "" &&
document.CertReq.NotifyEmail.value != "") &&
(document.CertReq.Email.value!=
document.CertReq.NotifyEmail.value)){
alert(STRING_UnmatchEmailPrompt);
document.CertReq.NotifyEmail.focus();
return true;
}
else {
return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1>1 Year SSL Browser Certificate</H1>
<p>
```

The pkiserv.tmp certificate templates file

```
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#          "/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=
    "if(ValidateEntry()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
  %%CommonName%%
|  %%Email (optional)%%
|  %%Requestor (optional)%%
|  %%NotifyEmail (optional)%%
  %%PassPhrase%%
  %%PublicKey2[browsertype]%%
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<CONSTANT>
  %%NotBefore=0%%
  %%NotAfter=365%%
  %%KeyUsage=handshake%%
  %%OrgUnit=Class 1 Internet Certificate CA%%
  %%Org=The Firm%%
  %%SignWith=PKI:%%
</CONSTANT>
<ADMINAPPROVE>
  %%CommonName (Optional)%%
  %%OrgUnit (Optional)%%
  %%OrgUnit (Optional)%%
  %%Org (Optional)%%
  %%NotBefore (optional)%%
  %%NotAfter (Optional)%%
  %%KeyUsage (Optional)%%
  %%HostIdMap (Optional)%%
  %%HostIdMap (Optional)%%
  %%HostIdMap (Optional)%%
  %%HostIdMap (Optional)%%
</ADMINAPPROVE>
<SUCCESSCONTENT>
  %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
  %%-requestbad%%
</FAILURECONTENT>
```

The pkiserv.tmpl certificate templates file

```
<RETRIEVECONTENT>
<HTML><HEAD>
%%-copyright%%
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingTransIdAlert(){
var STRING_MissingTransIdPrompt=
    "Enter the transaction ID assigned to the certificate.";
if(document.retrieveform.TransactionId.value==""){
    alert(STRING_MissingTransIdPrompt);
    document.retrieveform.TransactionId.focus();
    return true;
}
else {
    return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#    "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#    "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
<FORM NAME=retrieveform METHOD=POST ACTION=
    "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
    "if(MissingTransIdAlert()) return false; else return true;">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
    %%TransactionId%%
    %%ChallengePassPhrase (optional)%%
<p>
<INPUT TYPE="submit" VALUE="Retrieve and Install Certificate">
</FORM>
<p>
<H2>To check that your certificate installed properly, follow the
procedure below:</h2>
<p><B>Netscape V6</B> - Click Edit->Preferences, then Privacy and Security->
Certificates. Click the Manage Certificates button to start the Certificate Manager.
Your new certificate should appear in the Your Certificates list.
Select it then click View to see more information.
<p><B>Netscape V4</B> - Click the Security button, then Certificates->
Yours. Your certificate should appear in the list. Select it then
click Verify.
<p><B>Internet Explorer V5</B> - Click Tools->Internet Options, then
Content, Certificates.
Your certificate should appear in the Personal list. Click Advanced to
see additional information.
<p>
<FORM METHOD=GET ACTION="/PKIServ/public-cgi/camain.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
```

The pkiserv.tmp certificate templates file

```
<INPUT TYPE="submit" VALUE="Home page">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>

#
#
# =====
#
# Template Name - 1 Year PKI S/MIME Browser Certificate
#
# Function - Creates a 1 year certificate good for S/MIME
#             authentication using a browser. If approved, the
#             certificate becomes valid after it's requested.
#             (You may delay the valid date by specifying a non zero
#             number for the value of 'NotBefore',
#             eg. NotBefore=5. That means if the request is approved,
#             the certificate will become valid 5 days after it's
#             requested.)
#
#             These certificates will be stored in LDAP if The 0= and
#             OU= suffixes have already been created
#
# Other than the user input fields, all other information is hard coded.
#
# User input fields:
# CommonName - required
# AltEmail - required
# Requestor - optional
# PassPhrase - required
# PublicKey - required (Provided by the browser itself)
# NotifyEmail- optional
#
# RACF userid/password authentication : not required
# Administrator approval             : required
#
# =====
#
<TEMPLATE NAME=1 Year PKI S/MIME Browser Certificate>
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=1YBSM>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
%%-AdditionalHead[browsertype]%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(){
var STRING_MissingFieldPrompt=
    "Enter the required field."
var STRING_MissingConfirmPwdPrompt=
    "Reenter password."
var STRING_MissingPwdPrompt=
    "Need to enter password before confirm it."
var STRING_UnmatchPwdPrompt=
    "The passwords do not match. Enter again."
if(document.CertReq.CommonName.value=="") {
alert(STRING_MissingFieldPrompt);
document.CertReq.CommonName.focus();
return true;
```

The pkiserv.tmpl certificate templates file

```
}
if(document.CertReq.AltEmail.value=="") {
alert(String_MissingFieldPrompt);
document.CertReq.AltEmail.focus();
return true;
}
else if(document.CertReq.PassPhrase.value=="") {
alert(String_MissingFieldPrompt);
document.CertReq.PassPhrase.focus();
return true;
}
else if(document.CertReq.ConfirmPassPhrase.value=="") {
alert(String_MissingConfirmPwdPrompt);
document.CertReq.ConfirmPassPhrase.focus();
return true;
}
else if(document.CertReq.PassPhrase.value!=
document.CertReq.ConfirmPassPhrase.value){
alert(String_UnmatchPwdPrompt);
document.CertReq.ConfirmPassPhrase.focus();
return true;
}
else {
return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1>1 Year S/MIME Browser Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
# "/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=
"if(ValidateEntry()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
%%CommonName%%
%%AltEmail%%
%%Requestor (optional)%%
%%NotifyEmail (optional)%%
%%PassPhrase%%
%%PublicKey2[browsertype]%%
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
```

```

</FORM>
</u1>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<CONSTANT>
  %%NotBefore=0%%
  %%NotAfter=365%%
  %%KeyUsage=handshake%%
  %%OrgUnit=Class 1 Internet Certificate CA%%
  %%Org=The Firm%%
  %%SignWith=PKI:%%
</CONSTANT>
<ADMINAPPROVE>
  %%CommonName (Optional)%%
  %%OrgUnit (Optional)%%
  %%Org (Optional)%%
  %%NotBefore (optional)%%
  %%NotAfter (Optional)%%
  %%KeyUsage (Optional)%%
</ADMINAPPROVE>
<SUCCESSCONTENT>
  %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
  %%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT>
<HTML><HEAD>
%%-copyright%%
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingTransIdAlert(){
var STRING_MissingTransIdPrompt=
  "Enter the transaction ID assigned to the certificate.";
if(document.retrieveform.TransactionId.value==""){
  alert(STRING_MissingTransIdPrompt);
  document.retrieveform.TransactionId.focus();
  return true;
}
else {
  return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#   "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#   "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=

```

The pkiserv.tmpl certificate templates file

```
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
<FORM NAME=retrieveform METHOD=POST ACTION=
  "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
    "if(MissingTransIdAlert()) return false; else return true;">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
  %%TransactionId%%
  %%ChallengePassPhrase (optional)%%
<p>
<INPUT TYPE="submit" VALUE="Retrieve and Install Certificate">
</FORM>
<p>
<H2>To check that your certificate installed properly, follow the
procedure below:</h2>
<p><B>Netscape V6</B> - Click Edit->Preferences, then Privacy and Security->
Certificates. Click the Manage Certificates button to start the Certificate Manager.
Your new certificate should appear in the Your Certificates list.
Select it then click View to see more information.
<p><B>Netscape V4</B> - Click the Security button, then Certificates->
Yours. Your certificate should appear in the list. Select it then
click Verify.
<p><B>Internet Explorer V5</B> - Click Tools->Internet Options, then
Content, Certificates.
Your certificate should appear in the Personal list. Click Advanced to
see additional information.
<p>
<FORM METHOD=GET ACTION="/PKIServ/public-cgi/camain.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Home page">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>
#
#
# =====
#
# Template Name - 2 Year PKI Browser Certificate For Authenticating
#                 to z/OS
#
# Function - Creates a 2 year certificate good for authenticating to
#            z/OS. If approved, the certificate becomes valid after
#            it's requested.
#            (You may delay the valid date by specifying a non zero
#            number for the value of 'NotBefore',
#            eg. NotBefore=5. That means if the request is approved,
#            the certificate will become valid 5 days after it's
#            requested.)
#            HostidMap is formed by putting %%Userid%% and
#            %%HostIdMap=@host-name in the APPL section.
#
#            These certificates will be stored in LDAP if The O= and
#            OU= suffixes have already been created
#
# Other than the user input fields, all other information is hard coded.
#
# User input fields:
# Requestor - optional
# PassPhrase - required
# PublicKey - required (Provided by the browser itself)
# NotifyEmail - optional
#
# The presence of CommonName without a value tells SAF to determine
```

The pkiserv.tmpl certificate templates file

```
# the CN value from the PGMRNAME field of the user's USER profile.
# See the RACF Callable Services Guide for more information
#
# RACF userid/password authentication : require
# Administrator approval           : not require
#
#
# =====
#
<TEMPLATE NAME=2 Year PKI Browser Certificate For Authenticating To z/OS>
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=2YBZOS>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
%%-AdditionalHead[browsertype]%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(){
var STRING_MissingFieldPrompt=
    "Enter the required field."
var STRING_MissingConfirmPwdPrompt=
    "Reenter password."
var STRING_MissingPwdPrompt=
    "Need to enter password before confirm it."
var STRING_UnmatchPwdPrompt=
    "The passwords do not match. Enter again."
if(document.CertReq.PassPhrase.value=="") {
alert(STRING_MissingFieldPrompt);
document.CertReq.PassPhrase.focus();
return true;
}
else if(document.CertReq.ConfirmPassPhrase.value=="") {
alert(STRING_MissingConfirmPwdPrompt);
document.CertReq.ConfirmPassPhrase.focus();
return true;
}
else if(document.CertReq.PassPhrase.value!=
    document.CertReq.ConfirmPassPhrase.value){
alert(STRING_UnmatchPwdPrompt);
document.CertReq.ConfirmPassPhrase.focus();
return true;
}
else {
return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1>2 Year Browser Certificate For Authenticating To z/OS</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#     "/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=
```

The pkiserv.tmpl certificate templates file

```
# 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=
#      "if(ValidateEntry()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
  %%Requestor (optional)%%
  %%NotifyEmail (optional)%%
  %%PassPhrase%%
  %%PublicKey2[browsertype]%%
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<APPL>
  %%UserId%%
  %%HostIdMap=@host-name%%
</APPL>
<CONSTANT>
  %%NotBefore=0%%
  %%NotAfter=730%%
  %%KeyUsage=handshake%%
  %%OrgUnit=Class 1 Internet Certificate CA%%
  %%Org=The Firm%%
  %%SignWith=PKI:%%
  %%CommonName=%%
</CONSTANT>
<SUCCESSCONTENT>
  %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
  %%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT>
<HTML><HEAD>
%%-copyright%%
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingTransIdAlert(){
var STRING_MissingTransIdPrompt=
  "Enter the transaction ID assigned to the certificate.";
if(document.retrieveform.TransactionId.value==""){
  alert(STRING_MissingTransIdPrompt);
  document.retrieveform.TransactionId.focus();
  return true;
}
else {
  return false;
}
}
//-->
</SCRIPT>
</HEAD>
```

The pkiserv.tmpl certificate templates file

```
<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
#         "if(MissingTransIdAlert()) return false; else return true;">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
  %%TransactionId%%
  %%ChallengePassPhrase (optional)%%
<p>
<INPUT TYPE="submit" VALUE="Retrieve and Install Certificate">
</FORM>
<p>
<H2>To check that your certificate installed properly, follow the
procedure below:</h2>
<p><B>Netscape V6</B> - Click Edit->Preferences, then Privacy and Security->
Certificates. Click the Manage Certificates button to start the Certificate Manager.
Your new certificate should appear in the Your Certificates list.
Select it then click View to see more information.
<p><B>Netscape V4</B> - Click the Security button, then Certificates->
Yours. Your certificate should appear in the list. Select it then
click Verify.
<p><B>Internet Explorer V5</B> - Click Tools->Internet Options, then
Content, Certificates.
Your certificate should appear in the Personal list. Click Advanced to
see additional information.
<p>
<FORM METHOD=GET ACTION="/PKIServ/public-cgi/camain.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Home page">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>
#
#
# =====
# Template Name - 5 Year PKI SSL Server Certificate
#
# Function - Creates a 5 year Server certificate. If approved, the
# certificate becomes valid after it's requested.
# (You may delay the valid date by specifying a non zero
# number for the value of 'NotBefore',
# eg. NotBefore=5. That means if the request is approved,
# the certificate will become valid 5 days after it's
# requested.)
```

The pkiserv.tmpl certificate templates file

```
#
#           These certificates will be stored in LDAP if The O= and
#           OU= suffixes have already been created
#
# Other than the user input fields, all other information is hard coded.
#
# User input fields:
| # Email - optional
# CommonName - optional
# OrgUnit - optional
# Org - optional
| # Street - optional
# Locality - optional
# StateProv - optional
| # PostalCode - optional
# Country - optional
# AltEmail - optional
# AltDomain - optional
# AltURI - optional
# AltIPAddr - optional
# PassPhrase - required
| # PublicKey - required (This is the PKCS#10 request)
# NotifyEmail - optional
#
# RACF userid/password authentication : not require
# Administrator approval           : require
#
#
# =====
#
<TEMPLATE NAME=5 Year PKI SSL Server Certificate>

<TEMPLATE NAME=PKI Server Certificate>

<NICKNAME=5YSSSL>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingRequiredFAlert(){
var STRING_MissingRequiredFPrompt=
    "Enter the field value that's not optional."
var STRING_MissingConfirmPwdPrompt=
    "Reenter password."
var STRING_MissingPwdPrompt=
    "Need to enter password before confirm it."
var STRING_UnmatchPwdPrompt=
    "The passwords do not match. Enter again."
| var STRING_UnmatchEmailPrompt=
|     "The Email addresses for Distinguished name and notification do not match. Enter again."

if (document.serverform.PassPhrase.value=="")
{
    alert(STRING_MissingRequiredFPrompt);
    document.serverform.PassPhrase.focus();
    return true;
}
else if(document.serverform.ConfirmPassPhrase.value=="") {
alert(STRING_MissingConfirmPwdPrompt);
document.serverform.ConfirmPassPhrase.focus();
return true;
}
else if(document.serverform.PassPhrase.value!=
    document.serverform.ConfirmPassPhrase.value){
alert(STRING_UnmatchPwdPrompt);
document.serverform.ConfirmPassPhrase.focus();
```

```

return true;
}
else if(document.serverform.PublicKey.value=="") {
alert(String_MissingRequiredFPrompt);
document.serverform.PublicKey.focus();
return true;
}
else if((document.serverform.Email.value != "" &&
document.serverform.NotifyEmail.value != "") &&
(document.serverform.Email.value !=
document.serverform.NotifyEmail.value)){
alert(String_UnmatchEmailPrompt);
document.serverform.NotifyEmail.focus();
return true;
}
else {
return false;
}
}
//-->
</SCRIPT>

</HEAD>
<BODY>
<H1> 5 Year PKI SSL Server Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME=serverform 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=serverform 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=serverform METHOD=POST ACTION=
"/PKIServ/ssl-cgi-bin/careq.rexx" onSubmit=
"if(MissingRequiredFAlert()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
| %%Email (Optional)%%
| %%CommonName (Optional)%%
| %%OrgUnit (Optional)%%
| %%OrgUnit2 (Optional)%%
| %%Org (Optional)%%
| %%Street (Optional)%%
| %%Locality (Optional)%%
| %%StateProv (Optional)%%
| %%PostalCode (Optional)%%
| %%Country (Optional)%%
| %%AltEmail (Optional)%%
| %%AltDomain (Optional)%%
| %%AltURI (Optional)%%
| %%AltIPAddr (Optional)%%
| %%Requestor (Optional)%%
| %%NotifyEmail (Optional)%%
| %%PassPhrase%%
| %%PublicKey%%
<p>
<INPUT TYPE="submit" VALUE="Submit certificate request"

```

The pkiserv.tmpl certificate templates file

```
ONCLICK="if(MissingRequiredFAlert()) return false; else return true;">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>

<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<CONSTANT>
%%NotBefore=0%%
%%NotAfter=1825%%
%%KeyUsage=handshake%%
%%SignWith=PKI:%%
</CONSTANT>
<ADMINAPPROVE>
%%CommonName (Optional)%%
%%OrgUnit (Optional)%%
%%OrgUnit (Optional)%%
%%Org (Optional)%%
%%Locality (Optional)%%
%%StateProv (Optional)%%
%%Country (Optional)%%
%%AltEmail (Optional)%%
%%AltDomain (Optional)%%
%%AltURI (Optional)%%
%%AltIPAddr (Optional)%%
%%NotBefore (optional)%%
%%NotAfter (Optional)%%
%%KeyUsage (Optional)%%
</ADMINAPPROVE>
<SUCCESSCONTENT>
%%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
%%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
%%-copyright%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingTransIdAlert(){
var STRING_MissingTransIdPrompt=
    "Enter the transaction ID assigned to the certificate.";
if(document.retrieveform.TransactionId.value==""){
    alert(STRING_MissingTransIdPrompt);
    document.retrieveform.TransactionId.focus();
    return true;
}
else {
    return false;
}
}
//-->
</SCRIPT>
</HEAD>

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

The pkiserv.tmpl certificate templates file

```
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
<FORM NAME=retrieveform METHOD=POST ACTION=
     "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
     "if(MissingTransIdAlert()) return false; else return true;">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
%%TransactionId%%
%%ChallengePassPhrase (optional)%%
<p>
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%-returnpkcs10cert%%
</RETURNCERT>
</TEMPLATE>

#
# =====
#
# Template Name - 5 Year PKI IPSEC Server (Firewall) Certificate
#
# Function - Creates a 5 year Server certificate. If approved, the
#           certificate becomes valid after it's requested.
#           (You may delay the valid date by specifying a non zero
#           number for the value of 'NotBefore',
#           eg. NotBefore=5. That means if the request is approved,
#           the certificate will become valid 5 days after it's
#           requested.)
#
#           These certificates will be stored in LDAP if The O= and
#           OU= suffixes have already been created
#
#
# Other than the user input fields, all other information is hard coded.
#
# User input fields:
| # Email - optional
| # CommonName - optional
| # OrgUnit - optional
| # Org - optional
| # Street - optional
| # Locality - optional
| # StateProv - optional
| # PostalCode - optional
| # Country - optional
| # AltEmail - optional
| # AltDomain - optional
| # AltURI - optional
```

The pkiserv.tmpl certificate templates file

```
# AltIPAddr - optional
# PassPhrase - required
# PublicKey - required (This is the PKCS#10 request)
# NotifyEmail - optional
#
# RACF userid/password authentication : not require
# Administrator approval           : require
#
#
# =====
#
<TEMPLATE NAME=5 Year PKI IPSEC Server (Firewall) Certificate>
<TEMPLATE NAME=PKI Server Certificate>
<NICKNAME=5YSIPS>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingRequiredFAlert(){
var STRING_MissingRequiredFPrompt=
    "Enter the field value that's not optional."
var STRING_MissingConfirmPwdPrompt=
    "Reenter password."
var STRING_MissingPwdPrompt=
    "Need to enter password before confirm it."
var STRING_UnmatchPwdPrompt=
    "The passwords do not match. Enter again."
var STRING_UnmatchEmailPrompt=
    "The Email addresses for Distinguished name and notification do not match. Enter again."

if (document.serverform.PassPhrase.value=="")
{
    alert(STRING_MissingRequiredFPrompt);
    document.serverform.PassPhrase.focus();
    return true;
}
else if(document.serverform.ConfirmPassPhrase.value=="") {
alert(STRING_MissingConfirmPwdPrompt);
document.serverform.ConfirmPassPhrase.focus();
return true;
}
else if(document.serverform.PassPhrase.value!=
    document.serverform.ConfirmPassPhrase.value){
alert(STRING_UnmatchPwdPrompt);
document.serverform.ConfirmPassPhrase.focus();
return true;
}
else if(document.serverform.PublicKey.value=="") {
alert(STRING_MissingRequiredFPrompt);
document.serverform.PublicKey.focus();
return true;
}
else if((document.serverform.Email.value != "" &&
document.serverform.NotifyEmail.value != "" ) &&
(document.serverform.Email.value!=
document.serverform.NotifyEmail.value)){
alert(STRING_UnmatchEmailPrompt);
document.serverform.NotifyEmail.focus();
return true;
}
else {
return false;
}
}
//-->
</SCRIPT>
```

The pkiserv.tmp certificate templates file

```
</HEAD>
<BODY>
<H1> 5 Year PKI IPSEC Server (Firewall) Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME=serverform 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=serverform 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=serverform METHOD=POST ACTION=
          "/PKIServ/ssl-cgi-bin/careq.rexx" onSubmit=
          "if(MissingRequiredFAlert()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
|   %%Email (Optional)%%
|   %%CommonName (Optional)%%
|   %%OrgUnit (Optional)%%
|   %%OrgUnit2 (Optional)%%
|   %%Org (Optional)%%
|   %%Street (Optional)%%
|   %%Locality (Optional)%%
|   %%StateProv (Optional)%%
|   %%PostalCode (Optional)%%
|   %%Country (Optional)%%
|   %%AltEmail (Optional)%%
|   %%AltDomain (Optional)%%
|   %%AltURI (Optional)%%
|   %%AltIPAddr (Optional)%%
|   %%Requestor (Optional)%%
|   %%NotifyEmail (Optional)%%
|   %%PassPhrase%%
|   %%PublicKey%%
<p>
<INPUT TYPE="submit" VALUE="Submit certificate request"
ONCLICK="if(MissingRequiredFAlert()) return false; else return true;">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>

<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<CONSTANT>
%%KeyUsage=handshake%%
%%KeyUsage=dataencrypt%%
%%NotBefore=0%%
%%NotAfter=1825%%
%%SignWith=PKI:%%
```

The pkiserv.tmpl certificate templates file

```
</CONSTANT>
<ADMINAPPROVE>
  %%CommonName (Optional)%%
  %%OrgUnit (Optional)%%
  %%OrgUnit (Optional)%%
  %%Org (Optional)%%
  %%Locality (Optional)%%
  %%StateProv (Optional)%%
  %%Country (Optional)%%
  %%AltEmail (Optional)%%
  %%AltDomain (Optional)%%
  %%AltURI (Optional)%%
  %%AltIPAddr (Optional)%%
  %%NotBefore (optional)%%
  %%NotAfter (Optional)%%
  %%KeyUsage (Optional)%%
</ADMINAPPROVE>
<SUCCESSCONTENT>
  %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
  %%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
%%-copyright%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingTransIdAlert(){
var STRING_MissingTransIdPrompt=
  "Enter the transaction ID assigned to the certificate.";
if(document.retrieveform.TransactionId.value==""){
  alert(STRING_MissingTransIdPrompt);
  document.retrieveform.TransactionId.focus();
  return true;
}
else {
  return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#   "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#   "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
<FORM NAME=retrieveform METHOD=POST ACTION=
  "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
  "if(MissingTransIdAlert()) return false; else return true;">
```

The pkiserv.tpl certificate templates file

```
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<p> Enter values for the following field(s)
  %%TransactionId%%
  %%ChallengePassPhrase (optional)%%
<p>
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%-returnpkcs10cert%%
</RETURNCERT>
</TEMPLATE>

#
# =====
#
# Template Name - 5 Year PKI Intermediate CA Certificate
#
# Function - Creates a 5 year CA certificate. If approved, the
#            certificate becomes valid after it's requested.
#            (You may delay the valid date by specifying a non zero
#            number for the value of 'NotBefore',
#            eg. NotBefore=5. That means if the request is approved,
#            the certificate will become valid 5 days after it's
#            requested.)
#
#            These certificates will be stored in LDAP if The O= and
#            OU= suffixes have already been created
#
#
# Other than the user input fields, all other information is hard coded.
#
# User input fields:
| # Email - optional
| # CommonName - optional
| # OrgUnit - optional
| # Org - optional
| # Street - optional
| # Locality - optional
| # StateProv - optional
| # PostalCode - optional
| # Country - optional
| # AltEmail - optional
| # AltDomain - optional
| # AltURI - optional
| # AltIPAddr - optional
| # PassPhrase - required
| # PublicKey - required (This is the PKCS#10 request)
| # NotifyEmail - optional
#
# RACF userid/password authentication : require
# Administrator approval : not require
#
#
# =====
#
<TEMPLATE NAME=5 Year PKI Intermediate CA Certificate>
<TEMPLATE NAME=PKI Server Certificate>
<NICKNAME=5YSCA>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
<SCRIPT LANGUAGE="JavaScript">
<!--
```

The pkiserv.tmpl certificate templates file

```
function MissingRequiredFAlert(){
var STRING_MissingRequiredFPrompt=
    "Enter the field value that's not optional."
var STRING_MissingConfirmPwdPrompt=
    "Reenter password."
var STRING_MissingPwdPrompt=
    "Need to enter password before confirm it."
var STRING_UnmatchPwdPrompt=
    "The passwords do not match. Enter again."
| var STRING_UnmatchEmailPrompt=
|     "The Email addresses for Distinguished name and notification do not match. Enter again."

if (document.serverform.PassPhrase.value=="")
{
    alert(STRING_MissingRequiredFPrompt);
    document.serverform.PassPhrase.focus();
    return true;
}
else if(document.serverform.ConfirmPassPhrase.value=="") {
alert(STRING_MissingConfirmPwdPrompt);
document.serverform.ConfirmPassPhrase.focus();
return true;
}
else if(document.serverform.PassPhrase.value!=
    document.serverform.ConfirmPassPhrase.value){
alert(STRING_UnmatchPwdPrompt);
document.serverform.ConfirmPassPhrase.focus();
return true;
}
else if(document.serverform.PublicKey.value=="") {
alert(STRING_MissingRequiredFPrompt);
document.serverform.PublicKey.focus();
return true;
}
| else if((document.serverform.Email.value != "" &&
|     document.serverform.NotifyEmail.value != "") &&
|     (document.serverform.Email.value!=
|     document.serverform.NotifyEmail.value)){
| alert(STRING_UnmatchEmailPrompt);
| document.serverform.NotifyEmail.focus();
| return true;
| }
else {
return false;
}
}
//-->
</SCRIPT>

</HEAD>
<BODY>
<H1> 5 Year PKI Intermediate CA Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
<FORM NAME=serverform 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=serverform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
```

The pkiserv.tpl certificate templates file

```
# surrogate ID
#<FORM NAME=serverform METHOD=POST ACTION=
#           "/PKIServ/ssl-cgi-bin/careq.rexx" onSubmit=
#           "if(MissingRequiredFAlert()) return false; else return true;">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<p> Enter values for the following field(s)
|  %%Email (Optional)%%
  %%CommonName (Optional)%%
  %%OrgUnit (Optional)%%
  %%OrgUnit2 (Optional)%%
  %%Org (Optional)%%
|  %%Street (Optional)%%
  %%Locality (Optional)%%
  %%StateProv (Optional)%%
|  %%PostalCode (Optional)%%
  %%Country (Optional)%%
  %%AltEmail (Optional)%%
  %%AltDomain (Optional)%%
  %%AltURI (Optional)%%
  %%AltIPAddr (Optional)%%
  %%Requestor (Optional)%%
|  %%NotifyEmail (Optional)%%
  %%PassPhrase%%
  %%PublicKey%%
<p>
<INPUT TYPE="submit" VALUE="Submit certificate request"
ONCLICK="if(MissingRequiredFAlert()) return false; else return true;">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>

<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</u1>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<APPL>
  %%UserId%%
</APPL>
<CONSTANT>
  %%NotBefore=0%%
  %%NotAfter=1825%%
  %%KeyUsage=certsign%%
  %%SignWith=PKI:%%
</CONSTANT>
<SUCCESSCONTENT>
  %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
  %%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
%%-copyright%%
<SCRIPT LANGUAGE="JavaScript">
<!--
function MissingTransIdAlert(){
var STRING_MissingTransIdPrompt=
  "Enter the transaction ID assigned to the certificate.";
if(document.retrieveform.TransactionId.value==""){
```

The pkiserv.tpl certificate templates file

```
alert(String_MissingTransIdPrompt);
document.retrieveform.TransactionId.focus();
return true;
}
else {
    return false;
}
}
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
<FORM NAME=retrieveform METHOD=POST ACTION=
    "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#    "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#    "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
    "if(MissingTransIdAlert()) return false; else return true;">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<p> Enter values for the following field(s)
%%TransactionId%%
%%ChallengePassPhrase (optional)%%
<p>
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%-returnpkcs10cert%%
</RETURNCERT>
</TEMPLATE>

#
# =====
#
# Sample INSERTS
#
# =====
#
<INSERT NAME=-AdditionalHeadIE>
<OBJECT
    classid="clsid:43F8F289-7A20-11D0-8F06-00C04FC295E1"
    CODEBASE="xenroll.cab"
    id="certmgr"
>
</OBJECT>
</INSERT>

<INSERT NAME=-requestok>
<HTML><HEAD>
```

The pkiserv.tmp certificate templates file

```
<TITLE> Web Based Certificate Generation Success</TITLE>
</HEAD>
<BODY>
<H1> Request submitted successfully</H1>
[errorinfo]
<p> Here's your transaction ID. You will need it to retrieve your
certificate. Press 'Continue' to retrieve the certificate.
<p> <TABLE BORDER><TR><TD>[transactionid]</TD></TR></TABLE>
<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<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>
<H1> Request was not successful</H1>
<p> Please correct the problem or report the error to your Web admin
person<br>
<PRE>
[errorinfo]
</PRE>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>

<INSERT NAME=-renewrevokeok>
<HTML><HEAD>
<TITLE> Web Based Certificate Renew/Revoke Success</TITLE>
</HEAD>
<BODY>
<H1> Request submitted successfully</H1>
<FORM METHOD=GET ACTION="/PKIServ/public-cgi/camain.rexx">
<INPUT TYPE="submit" VALUE="Home Page">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>

<INSERT NAME=-renewrevokebad>
<HTML><HEAD>
<TITLE> Web Based Certificate Renew/Revoke Failure</TITLE>
</HEAD>
<BODY>
<H1> Request was not successful</H1>
<p> Please correct the problem or report the error to your Web admin
person<br>
<PRE>
[errorinfo]
</PRE>
<FORM METHOD=GET ACTION="/PKIServ/public-cgi/camain.rexx">
<INPUT TYPE="submit" VALUE="Home Page">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>

<INSERT NAME=-returnpkcs10cert>
```

The pkiserv.tmpl certificate templates file

```
<HTML><HEAD>
<TITLE> Web Based SAF Certificate Generation Application Pg 4</TITLE>
</HEAD>
<BODY>
<H1> Here's your Certificate. Cut and paste it to a file</H1>
<TABLE BORDER><TR><TD>
<PRE>
[base64cert]
</PRE>
</TD></TR></TABLE>
<p>%-pagefooter%</p>
</BODY>
</HTML>
</INSERT>

<INSERT NAME=returnbrowsercertNS>
[base64cert]
</INSERT>

<INSERT NAME=returnbrowsercertIE>
<HTML>
<HEAD>
<TITLE>MSIE Certificate Install</TITLE>
<OBJECT
  classid="clsid:43F8F289-7A20-11D0-8F06-00C04FC295E1"
  CODEBASE="xenroll.cab"
  id="certmgr"
>
</OBJECT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="VBScript">
<!--
  Sub INSTALL_OnClick
    Dim pkcs7data, errmsg, rc
    On Error Resume Next
    certmgr.DeleteRequestCert = false
    err.clear
    certmgr.WriteCertToCSP = true
    pkcs7data = "[iecert]"
    certmgr.acceptPKCS7(pkcs7data)
    if err.number <> 0 then
certmgr.WriteCertToCSP = false
      err.clear
      certmgr.acceptPKCS7(pkcs7data)
    end if
    if err.number <> 0 then
errmsg = "Your new certificate failed to install. " & _
  "Please ensure that you are using the same browser " & _
  "that you used when making the certificate request. " & _
rc = MsgBox (errmsg, 48, "Certificate Installation")
    else
errmsg = "Your new certificate installed successfully."
rc = MsgBox (errmsg, 64, "Certificate Installation")
    end if
  End Sub
  // -->
</SCRIPT>
<h1>Internet Explorer certificate install</h1>
<p> Click &quot;Install Certificate&quot; to store your new
certificate into your browser
<TABLE>
<TR> <br>
<TD><INPUT TYPE="BUTTON" VALUE="Install Certificate" NAME="INSTALL" >
<FORM METHOD=GET ACTION="/PKIServ/public-cgi/camain.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Home page">
</FORM>
```

```

</TD>
</TR>
</TABLE>
</BODY>
</HTML>
</INSERT>
#
# =====
#
# X.509 fields (INSERTs) valid for certificate requests
#
# =====
#
<INSERT NAME=KeyUsage>
<p> Indicate the intended purpose for the certificate [optfield] <BR>
<SELECT NAME="KeyUsage" MULTIPLE>
  <OPTION VALUE="handshake">Protocol handshaking (e.g., SSL)
  <OPTION VALUE="dataencrypt">Data encryption
  <OPTION VALUE="certsign">Certificate signing
  <OPTION VALUE="docsign">Document signing (nonrepudiation)
</SELECT>
</INSERT>

<INSERT NAME=NotBefore>
<p> Number of days after today before the certificate becomes current
  [optfield] <BR>
<SELECT NAME="NotBefore">
  <OPTION> 0
  <OPTION> 30
</SELECT>
</INSERT>

<INSERT NAME=NotAfter>
<p> Length of time that the certificate is current [optfield] <BR>
<SELECT NAME="NotAfter">
  <OPTION value="365">1 Year
  <OPTION value="730">2 Years
</SELECT>
</INSERT>

<INSERT NAME=Country>
<p> Country [optfield] <BR>
<INPUT NAME="Country" TYPE="text" SIZE=2 maxLength="2">
</INSERT>

<INSERT NAME=Org>
<p> Organization [optfield] <BR>
<INPUT NAME="Org" TYPE="text" SIZE=64 maxLength="64">
</INSERT>

<INSERT NAME=OrgUnit>
<p> Organizational Unit [optfield] <BR>
<INPUT NAME="OrgUnit" TYPE="text" SIZE=64 maxLength="64">
</INSERT>

<INSERT NAME=OrgUnit2>
<p> Organizational Unit [optfield] <BR>
<INPUT NAME="OrgUnit2" TYPE="text" SIZE=64 maxLength="64">
</INSERT>

<INSERT NAME=Locality>
<p> Locality [optfield] <BR>
<INPUT NAME="Locality" TYPE="text" SIZE=64 maxLength="64">
</INSERT>

<INSERT NAME=StateProv>
<p> State or Province [optfield] <BR>
<INPUT NAME="StateProv" TYPE="text" SIZE=64 maxLength="64">

```

The pkiserv.tmpl certificate templates file

```
</INSERT>

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

<INSERT NAME=Title>
<p> Title [optfield] <BR>
<INPUT NAME="Title" TYPE="text" SIZE=64 maxlength="64">
</INSERT>

<INSERT NAME=AltIPAddr>
| <p> IP address for alternate name in dotted decimal form [optfield] <BR>
<INPUT NAME="AltIPAddr" TYPE="text" SIZE=15 maxlength="15">
</INSERT>

<INSERT NAME=AltEmail>
| <p> Email address for alternate name [optfield] <BR>
<INPUT NAME="AltEmail" TYPE="text" SIZE=100 maxlength="100">
</INSERT>

<INSERT NAME=AltURI>
| <p> Uniform Resource Identifier for alternate name [optfield] <BR>
<INPUT NAME="AltURI" TYPE="text" SIZE=100 maxlength="255">
</INSERT>

<INSERT NAME=AltDomain>
| <p> Domain name for alternate name [optfield] <BR>
<INPUT NAME="AltDomain" TYPE="text" SIZE=100 maxlength="100">
</INSERT>

| <INSERT NAME=Street>
| <p> Street address [optfield] <BR>
| <INPUT NAME="Street" TYPE="text" MAXLENGTH=64 SIZE=64>
| </INSERT>

| <INSERT NAME=PostalCode>
| <p> Zipcode or postal code [optfield] <BR>
| <INPUT NAME="PostalCode" TYPE="text" MAXLENGTH=64 SIZE=64>
| </INSERT>

| <INSERT NAME=Email>
| <p> Email address for distinguished name [optfield] <BR>
| <INPUT NAME="Email" TYPE="text" MAXLENGTH=64 SIZE=64>
| </INSERT>

<INSERT NAME=SignWith>
<p> Component:/key-Label used to sign this certificate [optfield] <BR>
<p> e.g., "SAF:CERTAUTH/Local CA Cert" sign by CERTAUTH certificate
"Local CA Cert"
<INPUT NAME="SignWith" TYPE="text" SIZE=45 maxlength="45">
</INSERT>

<INSERT NAME=PublicKey>
<p> Base64 encoded PKCS#10 certificate request [optfield] <BR>
<TEXTAREA NAME="PublicKey"
  COLS="70"
  ROWS="12"
  WRAP="OFF">
</TEXTAREA>
</INSERT>

<INSERT NAME=PublicKeyNS>
<p> Select a key size
<KEYGEN NAME="PublicKey">
<p>
```

The pkiserv.tmp certificate templates file

```
<INPUT TYPE="Submit" VALUE="Submit certificate request">
</INSERT>

<INSERT NAME=PublicKey2NS>
<p> Select a key size
<KEYGEN NAME="PublicKey">
<p>
<INPUT TYPE="Submit" VALUE="Submit certificate request">
</INSERT>

<INSERT NAME=PublicKeyIE>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub SendReq

    On Error Resume Next
    Dim pkcs10data,DN,i,Message

        DN= ""
    CommonName= "Unspecified Distinguished Name"
    DN= "CN=" + CommonName + ";"
    certmgr.KeySpec = 1
    KeyUsage = "1.3.6.1.5.5.7.3.2"
    i = document.all.CSP.options.selectedIndex
    certmgr.providerName = document.all.CSP.options(i).text
    certmgr.providerType = document.all.CSP.options(i).value
    If document.CertReq.KeyProt.value = 1 Then
        certmgr.GenKeyFlags = 3
    Else
        certmgr.GenKeyFlags = 1
    End If

    pkcs10data = ""
    pkcs10data = certmgr.CreatePKCS10(DN, KeyUsage)
    document.CertReq.PublicKey.value = pkcs10data

    If Len(pkcs10data) > 0 Then
        document.CertReq.submit()
        return True
    Else
        return MsgBox ("PKCS10 Creation Failed",48,"Certificate request")
    End If

End Sub
// -->
</SCRIPT>

<p> Select the following key information
<p> Cryptographic Service Provider
<select name="CSP">
<script language="VBScript">
    On Error Resume Next
        Dim i, csp, sv

    certmgr.providerType = 1
    i = 0
    csp = ""
    csp = certmgr.enumProviders(i,0)
    sv = "SELECTED"
    While Len(csp) <> 0
        document.write("<OPTION VALUE=1 " & sv & ">" & csp & "</OPTION>")
        i = i + 1
        csp = ""
        csp = certmgr.enumProviders(i,0)
        selvalue = ""
    Wend
</script>
```

The pkiserv.tmpl certificate templates file

```
</select>

<p> Enable strong private key protection?
<select name="KeyProt">
  <option value="1">Yes</option>
  <option value="0" selected>No</option>
</select>
<input type="hidden" name="PublicKey" value="">
<p>
<INPUT TYPE="Button" VALUE="Submit certificate request" ,
  ONCLICK="if (MissingRequiredFAlert()) return false; SendReq()">
</INSERT>

<INSERT NAME=PublicKey2IE>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub SendReq

  On Error Resume Next
  Dim pkcs10data, DN, i, Message

      DN= ""
  CommonName= "Unspecified Distinguished Name"
  DN= "CN=" + CommonName + ";"
  certmgr.KeySpec = 1
  KeyUsage = "1.3.6.1.5.5.7.3.2"
  i = document.all.CSP.options.selectedIndex
  certmgr.providerName = document.all.CSP.options(i).text
  certmgr.providerType = document.all.CSP.options(i).value
  If document.CertReq.KeyProt.value = 1 Then
    certmgr.GenKeyFlags = 3
  Else
    certmgr.GenKeyFlags = 1
  End If

  pkcs10data = ""
  pkcs10data = certmgr.CreatePKCS10(DN, KeyUsage)
  document.CertReq.PublicKey.value = pkcs10data

  If Len(pkcs10data) > 0 Then
    document.CertReq.submit()
    return True
  Else
    return MsgBox ("PKCS10 Creation Failed",48,"Certificate request")
  End If

End Sub
// -->
</SCRIPT>

<p> Select the following key information
<p> Cryptographic Service Provider
<select name="CSP">
<script language="VBScript">
  On Error Resume Next
  Dim i, csp, sv

  certmgr.providerType = 1
  i = 0
  csp = ""
  csp = certmgr.enumProviders(i,0)
  sv = "SELECTED"
  While Len(csp) <> 0
    document.write("<OPTION VALUE=1 " & sv & ">" & csp & "</OPTION>")
    i = i + 1
    csp = ""
    csp = certmgr.enumProviders(i,0)
```

```

    selvalue = ""
Wend
</script>
</select>

<p> Enable strong private key protection?
<select name="KeyProt">
  <option value="1">Yes</option>
  <option value="0" selected>No</option>
</select>
<input type="hidden" name="PublicKey" value="">
<p>
<INPUT TYPE="Button" VALUE="Submit certificate request" ,
  ONCLICK="if (ValidateEntry()) return false; SendReq()">
</INSERT>

#
# =====
#
# non-X.509 certificate request fields (INSERTs)
#
# =====
#
<INSERT NAME=UserId>
<p> Owing SAF User ID [optfield] <BR>
<INPUT NAME="UserId" TYPE="text" SIZE=8 maxlength="8">
</INSERT>

<INSERT NAME=Label>
<p> Label assigned to certificate being requested [optfield] <BR>
<INPUT NAME="Label" TYPE="text" SIZE=32 maxlength="32">
</INSERT>

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

<INSERT NAME=PassPhrase>
<p> Pass phrase for securing this request. You will need to supply
this value when retrieving your certificate [optfield] <BR>
<INPUT NAME="PassPhrase" TYPE="password" SIZE=32 maxlength="32"> <BR>
<p> Reenter your pass phrase to confirm <BR>
<INPUT NAME="ConfirmPassPhrase" TYPE="password" SIZE=32
  maxlength="32">
</INSERT>

<INSERT NAME=ChallengePassPhrase>
<p> If you specified a pass phrase when submitting the certificate
request, type it here, exactly as you typed it on the
request form <BR>
<INPUT NAME="ChallengePassPhrase" TYPE="password" SIZE=32
  maxlength="32">
</INSERT>

<INSERT NAME=HostIdMap>
<p> HostIdMapping Extension value in subject-id@host-name form
  [optfield] <BR>
<INPUT NAME="HostIdMap" TYPE="text" SIZE=100 maxlength="100">
</INSERT>

<INSERT NAME=TransactionId>
<p> Enter the assigned transaction ID [optfield] <BR>
<INPUT NAME="TransactionId" TYPE="text" SIZE=56 maxlength="56"
  VALUE="[transactionid]">
</INSERT>

| <INSERT NAME=NotifyEmail>

```

The pkiserv.tmpl certificate templates file

```
| <p> Email address for notification purposes [optfield] <BR>
| <INPUT NAME="NotifyEmail" TYPE="text" SIZE=64 MAXLENGTH="64">
| </INSERT>

#####
#
#           Additional section           #
#                                           #
#####

<INSERT NAME=-copyright>
<!--
/*****
/*
/* Licensed Materials - Property of IBM
/* 5694-A01
/* (C) Copyright IBM Corp. 2001
/*
/*****
-->
<META HTTP-EQUIV="Content Type" content="text/html; charset=ISO-8859-1">
</INSERT>

<INSERT NAME=-pagefooter>
<A HREF="mailto:webmaster@your-company">
email: webmaster@your-company.com</A>
</INSERT>
```

Chapter 25. Environment variables

This chapter describes the environment variables that PKI Services uses and their possible values. It also includes a code sample of the environment variables file, `pkiserv.envars` (see “The `pkiserv.envars` environment variables file” on page 307). For information about `PKISERV` proc, which specifies the pathname of the environment variables file, see “`PKISERV` sample procedure to start PKI Services daemon” on page 335.

Environment variables in the environment variables file

The environment variables contained in `pkiserv.envars` and their values are:

`_PKISERV_MSG_LOGGING`

Values include:

`STDOUT_LOGGING`

Indicates writing all messages (verbose, debug, 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, debug, 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 the entries with commas (,). 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 (Debug).

Example:

```
_PKISERV_MSG_LEVEL=*.W,PKID.D
```

Environment variables

The subcomponents are:

Table 64. Subcomponents for message level

Subcomponent	Meaning
*	This is the wildcard character, which represents all subcomponents.
CORE	The core functions of PKI Services that are not specific to the other subcomponents.
DB	Activity related to the request or issued certificate VSAM data stores.
LDAP	LDAP posting operations.
PKID	The PKI Services daemon address setup and infrastructure.
POLICY	Certificate creation and revocation policy processing.
SAF	SAF key ring, OCEP, and R_data lib calls.

The message levels are:

Table 65. Message levels

Debug level (hierarchically listed)	Meaning
S	This indicates logging only Severe messages.
E	This indicates logging Severe and Error messages.
W	This indicates logging Severe, Error, and Warning messages. This is the default message level for all subcomponents if you do not set the environment variable.
I	This indicates logging Severe, Error, Warning, and Informational messages.
D	This indicates logging Severe, Error, Warning, Informational and Diagnostic.
V	This indicates logging ALL messages, including Verbose Diagnostic messages. This is very verbose. Recommendation: Do not use this level unless IBM support personnel request you to do so.

_PKISERV_CONFIG_PATH

Specifies the pathname for the directory containing the configuration file, pkiserv.conf, and the certificate template file, pkiserv.tmpl. The default value (if you do not set the environment variable) is /etc/pkiserv.

Recommendation: 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 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's 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.

Environment variables

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 Web page interface calls. (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 exit processing is disabled.

Note: 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's environment variables file. The z/OS HTTP Server environment variables file is usually `/etc/httpd.envvars`. PKI Services uses two instances of the z/OS HTTP Server. Therefore, if the two servers are using different environment variables files, you need to update both files.

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 55.) The code sample that follows might not be identical to the code shipped with the product. To see the most current code, look at 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
# (C) Copyright IBM Corp. 2001
# Status = HKY7706
#
#-----#
#
# Language and Path configurations
#
LANG=En_US.IBM-1047
| PATH=/usr/sbin' after LANG
LIBPATH=/usr/lpp/pkiserv/lib:/usr/lib
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/pkiserv/lib/nls/msg/%L/%N
#
# Configuration File location and Message configuration Options
#
_PKISERV_CONFIG_PATH=/etc/pkiserv
_PKISERV_MSG_LOGGING=stdout_logging
_PKISERV_MSG_LEVEL=*.w
#
# Location of the OCSF Registry (/var/ocsf is the default location)
#
OCSFREGDIR=/var/ocsf
```

Environment variables

Chapter 26. The IKYSETUP REXX exec

IKYSETUP is a REXX exec that issues RACF commands to perform RACF administration. This chapter 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
- Creates the CA certificate, private key, and key ring
- Creates the z/OS HTTP Server certificate, private key, and key ring
- Enables surrogate operation for the z/OS HTTP Server
- Enables the PKI Services daemon to call OCSF functions

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 UNIX System Services. 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.

Recommendation: 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 PKISRVD.* 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 may 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.

IKYSETUP

Recommendation: Although under certain circumstances it may 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 TSO command to create the surrogate user ID (PKISERV). Give it an OMVS segment because it needs access to z/OS UNIX.

Recommendation: 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*, where *function* is one of the following:

- GENCERT
- EXPORT
- REQCERT
- VERIFY
- REVOKE
- GENRENEW
- REQRENEW.

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 66. Access required if you plan to use an administrator

Resource	Access
IRR.DIGTCERT.REQCERT	READ
IRR.DIGTCERT.VERIFY	READ
IRR.DIGTCERT.REVOKE	READ
IRR.DIGTCERT.REQRENEW	READ
IRR.DIGTCERT.EXPORT	(If your end-users will always provide a passphrase) READ (Otherwise) UPDATE

If your clients request certificates that are auto-approved without action by an administrator, PKISERV needs the following access:

Table 67. Access required if you plan to use auto-approval

Resource	Access
IRR.DIGTCERT.GENCERT	CONTROL

Table 67. Access required if you plan to use auto-approval (continued)

Resource	Access
IRR.DIGTCERT.ADD	UPDATE
IRR.DIGTCERT.VERIFY	READ
IRR.DIGTCERT.REVOKE	READ
IRR.DIGTCERT.GENRENEW	READ
IRR.DIGTCERT.EXPORT	(If your end-users will always provide a passphrase), READ (Otherwise) UPDATE

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 68. FACILITY class access needed for protecting administrative functions

Resource	Access	(Purpose)
IRR.RPKISERV.PKIADMIN	READ	(For list and query operations)
	UPDATE	(To act on certificate requests and issued certificates)

To grant user ID ADMINID authority to administer PKI Services, use the following RACF TSO commands:

```
RDEFINE FACILITY (IRR.RPKISERV.PKIADMIN) UACC(NONE)
PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY) ACCESS(UPDATE) ID(ADMINID)
SETROPTS RACLIST (FACILITY) REFRESH
```

Creating the CA certificate, private key, and key ring

To create and sign digital certificates for others, you need to define a CA certificate and associated private key. The RACF RACDCERT GENCERT TSO command does this.

Before issuing the command, you need to know what the CA's distinguished name will be and where it will be located (under CERTAUTH or under the PKI Services daemon user ID). Typically, CAs have distinguished names in the following form:

OU=your-CA's-friendly-name.O=your-organization.C=your-two-letter-country-abbreviation

Example:

The RACDCERT GENCERT TSO command to create a 20-year CERTAUTH certificate with a distinguished name of OU=Human Resources Certificate Authority.O=Your Company, Inc.C=US is:

IKYSETUP

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(OU('Human Resources Certificate Authority')
O('Your Company, Inc') C('US')) WITHLABEL('Local PKI CA')
NOTBEFORE(DATE(2001/05/07)) NOTAFTER(DATE(2021/05/06))
```

To back up the certificate and private key to a password-protected data set and migrate the private key to ICSF, issue:

```
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA')) DSN('PKISRVD.PRIVATE.KEY.P12BIN')
FORMAT(PKCS12DER) PASSWORD('your-passphrase')
```

```
RACDCERT CERTAUTH ADD('PKISRVD.PRIVATE.KEY.P12BIN')
PASSWORD('your-passphrase') ICSF
```

Note: The preceding example assumes you want to use ICSF for private key protection and signing. For this to succeed, ICSF must be running and configured for RSA operations. (For additional information, see *z/OS ICSF Administrator's Guide*.) If you do not want to use ICSF, omit the RACDCERT ADD command.

After your CA certificate is created, you must place it in a key ring so that PKI Services can access it. This is also done using the RACF RACDCERT TSO command with sub keywords ADDRING and CONNECT. For example, the RACDCERT TSO commands to create a key ring called CAring for User ID PKISRVD and connect the preceding certificate to it are:

Example:

```
RACDCERT ADDRING(CAring) ID(PKISRVD)
RACDCERT ID(PKISRVD) CONNECT(CERTAUTH LABEL('Local PKI CA') RING(CAring)
USAGE(PERSONAL) DEFAULT)
```

Note: Make sure your CA certificate is marked TRUSTed in RACF. (Otherwise PKI Services will not be able to use the certificate.) Use the RACDCERT LIST command to check this and the RACDCERT ALTER command to change it if needed.

To use RACF's certificate services, the PKISRVD user ID needs access to the following FACILITY class resources:

Table 69. Access PKISERV needs to use RACF's certificate services

Resource	Access
IRR.DIGTCERT.GENCERT	(If the CA certificate was created under CERTAUTH) CONTROL (Otherwise) READ
IRR.DIGTCERT.LISTRING	READ

Configuring the z/OS HTTP Server for SSL mode

The PKISERV application requires the z/OS HTTP Server to operate in three modes. That is why PKI Services requires two z/OS HTTP Servers. The modes are:

- Normal
- SSL without client authentication
- SSL with client authentication.

For SSL, your server needs to obtain a digital certificate. You can:

- Purchase one from an external source
- Create one using RACF

Note: If your server is already operating in SSL mode, you can skip the following section, “Using RACF to obtain a certificate for the Web server”.

Using RACF to obtain a certificate for the Web server

The z/OS HTTP Server supports using either gskkyman key databases (.kdb files) or RACF (SAF) key rings for the server’s certificate store. You are expected to use SAF key rings if setting up their Web server for the first time.

Note: If you have already set up your Web server using gskkyman, you can continue to use it.

Use RACDCERT to generate the server certificate signed by the new Certificate Authority.

Example:

```
RACDCERT GENCERT ID(WEBSRV) SIGNWITH(CERTAUTH LABEL('Local PKI CA'))
WITHLABEL('SSL Cert') SUBJECTSDN(CN('www.yourcompany.com') O('Your Company Inc')
L('Millbrook') SP('New York') C('US'))
```

The Web server needs a key ring containing its new certificate and any trusted CA certificate. The RACDCERT command with operands ADDRING and CONNECT also sets this up. For example, the RACDCERT commands to create a key ring called SSLring for user ID WEBSRV and to connect the Web server and CA certificates to it are:

Example:

```
RACDCERT ADDRING(SSLring) ID(websrv)
RACDCERT ID(websrv) CONNECT(CERTAUTH LABEL('Local PKI CA')) RING(SSLring)
USAGE(PERSONAL) DEFAULT)
RACDCERT ID(websrv) CONNECT(ID(websrv) LABEL('SSL Cert') RING(SSLring)
USAGE(PERSONAL) DEFAULT)
```

Export the CA certificate to an MVS data set. Then OPUT it to an HFS file so that it can be made available to your clients.

Example:

```
RACDCERT EXPORT(LABEL('Local PKI CA'))
CERTAUTH DSN('pkisrvd.private.cacert.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.

Enabling the PKI Services daemon to call OCSF functions

For access to OCSF, the PKI Services daemon needs READ access to BPX.SERVER. If program control is in effect (for example, z/OS UNIX level security), then the daemon needs READ access to BPX.DAEMON as well.

Code Sample: IKYSETUP

IKYSETUP contains the commands to perform the RACF administrator tasks of adding groups and user IDs, setting up access control, creating CA and SSL certificates, and setting up daemon security. The example that follows might not be identical to the code shipped with the product. To see the exact code, look at SYS1.SAMPLIB member IKYSETUP.

```

/* REXX */
/*****
/*
/* DESCRIPTIVE NAME:  PKI Services RACF setup CLIST
/*
/* Licensed Materials - Property of IBM
/* 5694-A01
/* (C) Copyright IBM Corp. 2001
/* Status = HKY7706
/*
/*01* EXTERNAL CLASSIFICATION: OTHER
/*01* END OF EXTERNAL CLASSIFICATION:
/*
/* FUNCTION:
/*
/* This CLIST will issue the RACF TSO commands necessary to set up
/* security for PKI Services. It must be run from TSO by a user ID
/* that is RACF SPECIAL.
/*
/* USAGE:
/*
/* 1) Read accompanying PKI Services post installation
/* instructions.
/* 2) Perform necessary prerequisite product installation for
/* the webserver (websphere), LDAP, etc.
/* 3) Make note of any predetermined values such as the LDAP
/* suffix, webserver fully qualified domain name, and the
/* settings contained in the pkiserv.conf file.
/* 4) Copy the CLIST to a data set where you can edit it.
/* 5) Examine the entire CLIST, in particular, the configurable
/* section.
/* 6) Modify the values in the configurable section as needed for
/* your installation.
/* 7) Run the CLIST. Syntax:
/*
/* EX 'data-set-name(IKYSETUP)' 'RUN(YES | NO | PROMPT)'
/*
/* where: YES - indicates to run CLIST as is
/* NO - indicates to display the commands only
/* PROMPT - indicates to prompt the user prior
/* to invoking each command
/*
/* DISCLAIMER:
/*
/* This CLIST is not intended to cover every possible customer
/* scenario. Modification of the actual commands to be issued
/* may be required
/*
*****/

trace value('0')

/*-----*/
/* configurable section
/*-----*/

/*-----*/
/* Part 1 - Things you must change */

```

```

/*-----*/

/*****/
/* This exec will create the certificate, private key, and */
/* keyring needed for your certificate authority. */
/* */
/* You must update the distinguished name of your certificate */
/* authority defined below. The suffix of this DN must match */
/* the suffix set up for your LDAP directory (suffix value from */
/* your slapd.conf file). */
/* */
/* Typically, Certificate Authorities have distinguished names */
/* in the following form: */
/* */
/* OU=<your-CA's-friendly-name>,O=<your-organization>, */
/* C=<your-2-letter-country-abbreviation> */
/* */
/* e.g., OU=Human Resources Certificate Authority.O=IBM,C=US */
/* */
/* If you already have your CA certificate and private key set */
/* up in RACF, set ca_dn="" and update the ca_label variable to */
/* equal your CA certificate's label. Note, it must reside */
/* under CERTAUTH */
/*****/

ca_dn=,
  "OU('Human Resources Certificate Authority')",
  "O('Your Company')",
  "C('Your Country 2 Letter Abbreviation')"
ca_label = "Local PKI CA" /* Label for CA certificate */

/*****/
/* This exec will create the certificate, private key, and */
/* keyring needed for your webserver. (Required for SSL.) */
/* */
/* You must update the distinguished name of your */
/* webserver. The Common Name (CN) must match your webserver's */
/* fully qualified domain name. */
/* */
/* e.g., CN=www.ibm.com,O=IBM,C=US */
/* */
/* If you already have your webserver configured for SSL, set */
/* web_dn="" */
/*****/

web_dn=,
  "CN('www.YourCompany.com')",
  "O('Your Company')",
  "L('Your City')",
  "SP('Your Full State or Province Name')",
  "C('Your Country 2 Letter Abbreviation')"

/*****/
/* The sample web server protection directives supplied by PKI */
/* use SSLring for the web server's SAF key ring. If you change */
/* the value below, you will need to modify the "KeyFile" */
/* directive in the samples/httpd.conf and samples/httpd2.conf */
/* files when configuring the web server. */
/* */
/* If you already have your webserver configured for SSL and */
/* are using a SAF key ring (vs a gskkyman keyfile), then set */
/* web_ring equal to your webserver's SAF key ring name. If you */
/* are using a gskkyman keyfile, then set web_ring="". Note, */
/* you will have to add the CA's certificate to the webserver's */
/* keyfile manually */
/*****/

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? */
/* */
/* if ICSF key protection is desired for your CA's private key, */
/* set use_icsf=1. ICSF must be configured for PKA support and */
/* running for this to be successful. Note, you can defer this */
/* until later if you wish. Read the next paragraph before */
/* making this decision */
/*-----*/
use_icsf=0

/*-----*/
/* If you set use_icsf=1 above, you will need to restrict access */
/* to the CA's private key. Unless you indicate otherwise, this */
/* exec will activate the CSFKEYS class, create a profile in the */
/* CSFKEYS class to protect the CA's private key, and permit */
/* the PKI Services daemon to use it. */
/* */
/* If you are already using ICSF, then you may have profiles in */
/* the CSFSERV class protecting ICSF services. The PKI Services */
/* daemon would need access to the profile that covers the */
```

```

/* CSFDSV and CSFDSG services. Also, the PKI Services surrogate */
/* ID would need access to the profile that covers the */
/* CSFENC and CSFDEC services. You may also have a RACF group */
/* for authorized ICSF users. Both of these user IDs */
/* would need to be added to this group. */
/* */
/* Set the following variables as needed: */
/* */
/* csfkeys_profile - Profile to be created in the CSFKEYS class */
/* Set the value to '' if you don't want the profile */
/* csfserv_profile - Profile to be created in the CSFSERV class */
/* e.g., 'CSF*' */
/* csfusers_grp - Group name for authorized ICSF users */
/* e.g., 'ICSFUGRP' */
/*****
csfkeys_profile='IRR.DIGTCERT.CERTIFAUTH.*'
csfserv_profile='CSF*'
csfusers_grp=''

/*****
/* Question 3 - Back up your private key? */
/* */
/* The exec will prompt you to enter a pass phrase to encrypt a */
/* backup copy of your CA's certificate and private key. */
/* Caution, the text you enter at the prompt WILL be displayed */
/* at the terminal. Backup is highly recommended. If you do not */
/* wish to back up your CA's certificate and private key to a */
/* pass phrase encrypted data set, set key_backup=0. The back up */
/* may be done later if the key is not stored in ICSF. */
/* */
/* Note, back up is not performed if the CA certificate was not */
/* created by this exec */
/* *****/
key_backup=1

/*****
/* Question 4 - Set up z/OS UNIX level security? */
/* */
/* z/OS UNIX may be set up to operate with a higher level of */
/* security than traditional UNIX. While we recommend this, it */
/* difficult to set up. You may want to defer this until later. */
/* */
/* If you don't want to set up UNIX security now, leave */
/* unix_sec=0. */
/* */
/* If you already have UNIX level security established and wish */
/* to continue it, set unix_sec=1. */
/* */
/* If you don't have UNIX level security established and wish */
/* to establish it now, set unix_sec=2. Note additional manual */
/* configuration probably will be required. This can be done */
/* by adding, removing, updating members of the two stem */
/* variables below. The pgmctl_dsn stem contains the data set */
/* names of load libraries that need program control. The */
/* bpx_userid stem contains the user IDs of your server daemons. */
/* (These need access to BPX.SERVER and BPX.DAEMON in the */
/* FACILITY class.) Again, you can defer this until later by */
/* leaving unix_sec=0
/*****
unix_sec=0
pgmctl_dsn.0=9 /* Number of program controlled data sets below */
pgmctl_dsn.1='CEE.SCEERUN'
pgmctl_dsn.2='CBC.SCLBDLL'
pgmctl_dsn.3='GLD.SGLDLNK'
pgmctl_dsn.4='GSK.SGSKLOAD'
pgmctl_dsn.5='SYS1.CSSLIB'
pgmctl_dsn.6='TCP/IP.SEZALINK'

```

IKYSETUP

```
pgmctl_dsn.7="'SYS1.LINKLIB'"
pgmctl_dsn.8="'CSF.SCSFMODE'"
pgmctl_dsn.9="'CSF.SCSFMODE1'"
bpx_userid.0=1 /* Number of additional bpx server ids below */
bpx_userid.1="OMVSKERN"

/*-----*/
/* Part 3 - Things you can change */
/*-----*/

/*****
/* 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)*/
/*****
log_dsn="PRIVATE.IKYSETUP.LOG" /* Under your ID */

/*****
/* 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 */
ca_ring="CAring" /* keyring name for PKI Srvs */

/*****
/* Data set to contain the backup copy of the CA certificate */
/* and private key. (pass phrase encrypted PKCS#12 format) */
/*****
backup_dsn = "" || daemon || ".PRIVATE.KEY.BACKUP.P12BIN"

/*****
/* Data set to contain the exported copy of the CA certificate */
/* (DER encoded). This is to be OPUT to an HFS file later to */
/* enable easy downloading by clients. */
/*****
export_dsn = "" || daemon || ".PRIVATE.CACERT.DERBIN"

/*****
/* This EXEC expects the web server to be set up. If this is */
/* not the case, please refer to: */
/* z/OS HTTP Server Planning, Installing and Using. */
/* If the user ID assigned to the IBM HTTP Server Daemon is not */
/* WEBSRV, please update the assignment below. */
/*****
webserver="WEBSRV"

/*-----*/
/* End of configurable section */
/*-----*/

parse upper arg "RUN(" runopt ")"

if runopt = '' then
```

```

    runopt="NO"
if runopt ^= "YES" & runopt ^= "PROMPT" & runopt ^= "NO" then do
    say "syntax ex 'data-set-name(IKYSETUP)' 'run(yes | no | prompt)'"
    return 8
end
if runopt ^= "YES" & runopt ^= "PROMPT" then
    runopt="NO"

say 'IKYSETUP EXEC invoked ...'
return_code= '0'
max_return_code= '0'
logdata.0=0

if log_dsn ^= "" then do
    say "Allocating log data set" log_dsn "...
    x = OUTTRAP(MSGS.)
    "FREE FI(IKYLOGDD)"
    "FREE DA(||log_dsn||)"
    "DELETE" log_dsn
    x = OUTTRAP('OFF')
    "ALLOCATE DA(||log_dsn||) FILE(IKYLOGDD) RECFM(V B)" ,
    " LRECL(256) DSORG(PS) BLKSIZE(2560) SP(1,1) TRACKS "
    al_rc= rc
    IF al_rc ^= 0 THEN
        do
            say 'Allocation of log data set failed.'
            return 8
        end
    end
end

call logsay "RUN("runopt") requested"
if runopt="NO" then
    call logsay "Running in test mode. Commands are not being invoked"
    call logsay " "
    /*****
    /* 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 logsay "Creating users and groups ..."
call tsoserv "ADDUSER " daemon "name('PKI Srvs Daemon')",
    " nopassword",
    " omvs(uid("daemon_uid"))",
    " assize(256000000)",
    " threads(512)"

if restricted_surrog=1 then
    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)"

if pkigroup ^= "" then do
    call tsoserv "ADDGROUP " pkigroup "OMVS(GID("pki_gid"))"
    do i = 1 to pkigroup_mem.0
        call tsoserv "CONNECT" pkigroup_mem.i "GROUP("pkigroup)"
    end
end
end

```

IKYSETUP

```
/******
 * Give the administrators access to the VSAM data sets
 * identified in the [ObjectStore] section of
 * the pkiserv.conf file.
*****/
call logsay "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 logsay "Creating the CA certificate ..."
  certcmd = "RACDCERT GENCERT CERTAUTH SUBJECTSDN('ca_dn'",
    " WITHLABEL('ca_label') NOTAFTER(DATE('ca_expires'))"
  if use_icsf=1 & key_backup=0 then
    certcmd= certcmd || " ICSF"

  call tsoserv certcmd

  if key_backup=1 then do
    /******/
    /* Export certificate and key to PKCS#12 dataset */
    /******/
    say ""
    say "Enter a passphrase to protect the key. You will need"
    say " this value later if you need to restore the key."
    say ""
    say "Attention, the value will be displayed in the screen:"
    parse pull pp
    call logsay "Backing up the CA certificate ..."
    certcmd = "RACDCERT CERTAUTH EXPORT(LABEL('ca_label'))",
      " DSN('backup_dsn') FORMAT(PKCS12DER)",
      " PASSWORD('pp')"

    call tsoserv certcmd
  end

  if use_icsf=1 & key_backup=1 then do
    /******/
    /* If ICSF was requested and key backup, reload the certificate */
    /* to get the key migrated to ICSF */
    /******/
    call logsay "Migrating the CA's private key to ICSF ..."
    certcmd = "RACDCERT CERTAUTH ADD('backup_dsn'",
      " PASSWORD('pp') ICSF"

    call tsoserv certcmd
  end

end /* ca_dn ^= "" */
/******/
/* Mark the CA certificate as HIGHTRUST so HostIdMappings */
/* are honored */
/******/
call logsay "Marking CA certificate as HIGHTRUST ..."
certcmd = "RACDCERT CERTAUTH ALTER(LABEL('ca_label')) HIGHTRUST"
call tsoserv certcmd

/******/
/* The CA certificate must be saved to a data set so that it may */
```

```

/* be OPUT to an HFS file. */
/*****/
call logsay "Saving the CA certificate to a data set for OPUT ..."
certcmd = "RACDCERT CERTAUTH EXPORT(LABEL('ca_label'))",
" DSN("export_dsn") FORMAT(CERTDER)"
call tsoserv certcmd

/*****/
/* The CA certificate must be placed in a key ring so that */
/* PKI Services can access it. */
/*****/
call logsay "Creating the PKI Services keyring ..."
call tsoserv "RACDCERT ADDRING("ca ring") ID("daemon)"
call tsoserv "RACDCERT ID("daemon") CONNECT(CERTAUTH",
" LABEL('ca_label')",
" RING("ca_ring") USAGE(PERSONAL) DEFAULT) "

/*****/
/* Create the certificate for the webserver signed by your new CA */
/*****/

if web_dn ^= "" then do
call logsay "Creating the Webserver SSL certificate and keyring ..."
call tsoserv "RACDCERT GENCERT ID("webserver") SIGNWITH(CERTAUTH",
" LABEL('ca_label'))",
" WITHLABEL('web_label') SUBJECTSDN("web_dn)",
" NOTAFTER(DATE("web_expires"))"

/*****/
/* Add the certificate to the webserver's RACF (SAF) key ring */
/*****/
call tsoserv "RACDCERT ADDRING("web_ring") ID("webserver)"
call tsoserv "RACDCERT ID("webserver") CONNECT(ID("webserver"),
" LABEL('web_label') RING("web_ring") USAGE(PERSONAL) DEFAULT)"
end /* web_dn ^= "" */

/*****/
/* Add the CA certificate to the webserver's RACF (SAF) key ring*/
/*****/
if web_ring ^= "" then
call tsoserv "RACDCERT ID("webserver") CONNECT(CERTAUTH",
" LABEL('ca_label') RING("web_ring)")"

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 logsay "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 logsay "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"

```

IKYSETUP

```
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 pgmctl_dsn.0
    call tsoserv "ALTER PROGRAM * ADDMEM("pgmctl_dsn.i"//NOPADCHK)",
      " UACC(READ)"
  end
  call tsoserv "SETROPTS WHEN(PROGRAM)"
end
call tsoserv "PERMIT * CLASS(PROGRAM)",
  " ID("surrog") ACCESS(READ)"
call tsoserv "SETROPTS WHEN(PROGRAM) REFRESH"
end /* unix_sec ^= 0 */

/*****/
/* Allow the daemon to be a certificate authority */
/*****/
call logsay "Allowing the PKI Services daemon to act as a CA ..."
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.GENCERT"
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.LISTRING"
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.LIST"
call tsoserv "PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY)",
  " ID("daemon") ACCESS(CONTROL)"
call tsoserv "PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)",
  " ID("daemon") ACCESS(READ)"
call tsoserv "PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)",
  " ID("daemon") ACCESS(READ)"

/*****/
/* Allow the webserver to access its keyring */
/*****/
call logsay "Allowing the Webserver to access its keyring ..."
call tsoserv "PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)",
  " ID("webserver") ACCESS(READ)"
call tsoserv "PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)",
  " ID("webserver") ACCESS(READ)"

/*****/
/* Permit the webserver daemon User ID to switch identity to the */
/* surrogate Id */
/*****/
```

```

call logsay "Allowing the Webserver to switch identity to "surrog" ..."
call tsoserv "SETROPTS CLASSACT(SURROGAT)"
call tsoserv "RDEFINE SURROGAT BPX.SRV."surrog
call tsoserv "PERMIT BPX.SRV."surrog" CLASS(SURROGAT)",
" ID("webserver") ACCESS(READ)"
call tsoserv "SETROPTS RACLIST(SURROGAT) REFRESH"

if use_icsf then do
/*****/
/* Allow the daemon authorization to use ICSF */
/*****/
call logsay "Allowing the PKI Services daemon to use ICSF ..."
if csfkeys_profile ^= '' | csfserv_profile ^= '' then do
call tsoserv "SETROPTS GENERIC(CSFKEYS CSFSERV)"
call tsoserv "SETROPTS GENERIC(CSFKEYS CSFSERV) REFRESH"
end
if csfkeys_profile ^= '' then do
call tsoserv "RDEFINE CSFKEYS" csfkeys_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 csfserv_profile ^= '' then do
call tsoserv "RDEFINE CSFSERV" csfserv_profile "UACC(NONE)"
call tsoserv "PERMIT" csfserv_profile "CLASS(CSFSERV)",
" ID("daemon") ACCESS(READ)"
call tsoserv "PERMIT" csfserv_profile "CLASS(CSFSERV)",
" ID("surrog") ACCESS(READ)"
call tsoserv "SETROPTS CLASSACT(CSFSERV) RACLIST(CSFSERV)"
call tsoserv "SETROPTS RACLIST(CSFSERV) REFRESH"
end
if csfusers_grp ^= '' then do
call tsoserv "CONNECT" daemon "GROUP(" csfusers_grp ")"
call tsoserv "CONNECT" surrog "GROUP(" csfusers_grp ")"
end
end

/*****
* Tie the daemon user ID to PKI Services started procedure
*****/
call logsay "Creating the STARTED class profile for the daemon ..."
call tsoserv "RDEFINE STARTED PKISERVD.* STDATA(USER("daemon"))"
call tsoserv "SETROPTS CLASSACT(STARTED) RACLIST(STARTED)"
call tsoserv "SETROPTS RACLIST(STARTED) REFRESH"

/*****/
/* Give the surrogate user ID authority to request certificate */
/* generation functions. */
/*****/
call logsay "Allowing "surrog" to request certificate functions ..."
call tsoserv "SETR GENERIC(FACILITY)"
call tsoserv "RDEFINE FACILITY IRR.RPKISERV.**"
call tsoserv "PERMIT IRR.RPKISERV.** CLASS(FACILITY) ID("surrog")",
" ACCESS(CONTROL)"

/*****/
/* The administrative functions of PKI Services are protected */
/* by the IRR.RPKISERV.PKIADMIN FACILITY class resource. */
/* The following commands give UPDATE access to the PKI */
/* services group to allow them to act on certificate */
/* requests and issued certificates. */
/*****/
call logsay "Creating the profile to protect PKI Admin functions ..."

```

IKYSETUP

```
call tsoserv "RDEFINE FACILITY IRR.RPKISERV.PKIADMIN"
call tsoserv "PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY)",
  " ID("pkigroup") ACCESS(UPDATE)"
call tsoserv "PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY)",
  " ID("surrog") ACCESS(NONE)"
call tsoserv "SETROPTS RACLIST(FACILITY) REFRESH"

/*****/
/* Done. Now write to the log */
/*****/
upper daemon vsamhql export_dsn
call logsay " "
call logsay "-----"
call logsay "Information needed for PKI Services UNIX set up:"
call logsay "-----"
call logsay " "
call logsay "The daemon user ID is:"
call logsay " " daemon
call logsay " "
call logsay "The VSAM high level qualifier is:"
call logsay " " vsamhql
call logsay,
"This is needed for the [ObjectStore] section in pkiserv.conf"
call logsay " "
call logsay "The PKI Services' DER encoded certificate is in data set:"
call logsay " " export_dsn
call logsay,
"This must be OPUT to /var/pkiserv/cacert.der with the BINARY option"
call logsay " "
call logsay "The fully qualified PKI Services' SAF keyring is:"
call logsay " " daemon/"ca_ring"
call logsay,
"This is needed for the [SAF] section in pkiserv.conf"
call logsay " "
if ca_dn ^= "" then do
  call logsay "The PKI Services CA DN is:"
  call norm_dn ca_dn
  call logsay " " dn
  call logsay "The suffix must match the LDAP suffix in slapd.conf"
end
else
  call logsay "CA certificate not created by this exec"
call logsay " "
if web_dn ^= "" then do
  call logsay "The webserver's SAF keyring is:"
  call logsay " " web_ring
  call logsay,
  "This is needed for the KeyFile directive in httpd*.conf files"
  call logsay " "
  call logsay "The Webserver's DN is:"
  call norm_dn web_dn
  call logsay " " dn
  call logsay "The left most RDN must be the webserver's fully",
    "qualified domain name"
end
else
  call logsay,
  "Webserver certificate and keyring not created. You must add the CA",
  "certificate as a 'trusted root' manually"
call logsay " "
if log_dsn ^= "" then do
  x = OUTTRAP(MSGS.)
  'EXECIO' logdata.0 'DISKW IKYLOGDD (FINIS STEM LOGDATA.'
  'FREE FI(IKYLOGDD)'
  x=OUTTRAP('OFF')
  say "Commands complete. Results written to log data set" log_dsn
end
```

```

end
/*****/
/* Exit */
/*****/
say 'The IKYSETUP EXEC has completed.'
Exit max_return_code

/*****/
/* tsoserv - echo rc and commands and track highest rc */
/*****/
tsoserv:
Parse arg cmd
return_code = 0
skipit= 0
if runopt = "NO" | runopt = "PROMPT" then
  call logsay cmd
  if runopt = "PROMPT" then do
    say "Run command (y/n) ?"
    parse pull ans
    if substr(ans,1,1) ^= 'Y' & substr(ans,1,1) ^= 'y' then
      skipit= 1
  end
end
if skipit = 0 then
  if runopt = "YES" | runopt = "PROMPT" then do
    msg_status= MSG('ON')
    x=OUTTRAP('rac_ret.')
    Address TSO cmd
    return_code=rc
    y=OUTTRAP('OFF')
    call logsay 'Return code' return_code 'from->' cmd
    If return_code\=0 then do
      Do j=1 to rac_ret.0
        call logsay rac_ret.j
      end
    end
  end
end
max_return_code= max(max_return_code,return_code)
return return_code
return 0

/*****/
/* logsay - echo messages to the terminal and logdata stem */
/*****/
logsay:
Parse arg cmd
parse var cmd leftpart " PASSWORD(' pw ') " rightpart
if pw ^= "" then
  cmd= leftpart "PASSWORD('*****') " rightpart
say cmd
k= logdata.0 + 1
logdata.k= cmd
logdata.0= k

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

```

IKYSETUP

```
dns.= ""
do i = 1 to 7
  q= strip(q.i)
  upper q
  if q = "" then
    leave
  if q = "CN" then
    dns.1= "CN=" || v.i
  else
    if q = "T" then
      dns.2= "T=" || v.i
    else
      if q = "OU" then
        dns.3= "OU=" || v.i
      else
        if q = "O" then
          dns.4= "O=" || v.i
        else
          if q = "L" then
            dns.5= "L=" || v.i
          else
            if q = "SP" then
              dns.6= "ST=" || v.i
            else
              dns.7= "C=" || v.i
            end
          end
        dn= ""
        do i = 1 to 7
          if dns.i ^= "" then
            if dn = "" then
              dn= dns.i
            else
              dn= dn || "," || dns.i
            end
          end
        end
      return 0
    end
  end
end
```

Chapter 27. Other code samples

This chapter 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 a sample IDCAMS JCL to create VSAM data sets (regardless of whether you are using a sysplex or non-sysplex). (See “IKYCVSAM” on page 330.)
- `IKYRVSAM`, which is a sample IDCAMS JCL to use if you are migrating from z/OS Version 1 Release 3 and want sysplex support. `IKYRVSAM` reallocates your z/OS Version 1 Release 3 VSAM data sets in preparation for sharing in a sysplex. (See “IKYRVSAM” on page 332.)
- `PKISERVD`, which is a sample procedure to start PKI Services daemon. (See “PKISERVD sample procedure to start PKI Services daemon” on page 335.)

Note: Other important programs are contained in other chapters:

- `IKYSETUP`, a REXX exec to set up RACF profiles — see Chapter 26, “The IKYSETUP REXX exec” on page 309
- `pkiserv.envars`, the PKI Services environment variables file — see “The `pkiserv.envars` environment variables file” on page 307
- `pkiserv.conf`, the PKI Services configuration file — see Chapter 23, “The `pkiserv.conf` configuration file” on page 261
- `pkiserv.tmpl`, the PKI Services certificate template file — see Chapter 24, “The `pkiserv.tmpl` certificate templates file” on page 263.

z/OS HTTP Server configuration directives

The example that follows might not be identical to the code shipped with the product. If you want to see the exact code, look at 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
# Status = HKY7706

# For a secure system, set the default User ID to %%CLIENT%%
UserId      %%CLIENT%%

# SSL support using a SAF keyring
keyfile SSLring SAF
#          OR
# May use a gskkyman key database instead of SAF keyring
#keyfile /etc/key.kdb

sslmode on
sslport 443
Normalmode on
Protection PublicUser {
    ServerId      PublicUser
    UserID        PKISERV
    Mask          Anyone
}
Protect /PKIServ/public-cgi/* PublicUser
Protect /PKIServ/ssl-cgi-bin/* PublicUser
Protect /PKIServ/* PublicUser
```

Other code samples

```
Protection AuthenticatedUser {
    ServerId      AuthenticatedUser
    AuthType      Basic
    PasswdFile    %%SAF%%
    UserID        %%CLIENT%%
    Mask          All
}
Protect /PKIServ/ssl-cgi-bin/auth/* AuthenticatedUser

Protection SurrogateUser {
    ServerId      SurrogateUser
    AuthType      Basic
    PasswdFile    %%SAF%%
    UserID        PKISERV
    Mask          All
}

Protect /PKIServ/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 /PKIServ/clientauth-cgi/* \ https://<server-domain-name>:1443/PKIServ/clientauth-cgi/*

Exec      /PKIServ/public-cgi/*      <application-root>/PKIServ/public-cgi/*
Exec      /PKIServ/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.

```
# Licensed Materials - Property of IBM
# 5694-A01
# (C) Copyright IBM Corp. 2001
# Status = HKY7706

# For a secure system, set the default User ID to %%CLIENT%%
UserId      %%CLIENT%%

# SSL support using a SAF keyring
keyfile SSLring SAF
#          OR
# May use a gskkyman key database instead of SAF keyring
#keyfile /etc/key.kdb

sslmode on
sslport 1443
Normalmode off
SSLClientAuth strong
SSLX500CARoots local_and_x500
SSLX500Host <ldap-server-name>
SSLX500Port <ldap-port-number>
SSLX500UserID <ldap-distinguished-name>
SSLX500Password <ldap-password>

Protection RenewRevokeUser {
    ServerId      RenewRevokeUser
    AuthType      Basic
    UserID        PKISERV
    SSL_CLIENTAUTH Client
    Mask          Anyone
}

Protect /PKIServ/clientauth-cgi/* RenewRevokeUser

Protection AuthenticatedAdmin {
    ServerId      AuthenticatedAdmin
    AuthType      Basic

    UserID        %%CERTIF%%
    SSL_CLIENTAUTH Client
    Mask          Anyone
}
Protect /PKIServ/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/*

Exec /PKIServ/clientauth-cgi/* <application-root>/PKIServ/clientauth-cgi-bin/*
```

IKYCVSAM

IKYCVSAM is sample IDCAMS JCL to create VSAM data sets. (You 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 and are migrating from z/OS Version 1 Release 3, see "IKYRVSAM" on page 332.) IKYCVSAM is installed as a member of SYS1.SAMPLIB.

Note: The example that follows might not be identical to the code shipped with the product. To view the most current code, see SYS1.SAMPLIB member IKYCVSAM.

```
//IKYCVSAM JOB <job card parameters>
//*****
//* SAMP:      IKYCVSAM                      *
//*          *                               *
//* Licensed Materials - Property of IBM    *
//* 5694-A01                                  *
//* (C) Copyright IBM Corp. 2001, 2002     *
//* Status = HKY7707                        *
//*                                          *
//*****
//*
//* This sample JCL may be used to create the VSAM data sets *
//* PKI Services utilizes to store certificate requests and *
//* issued certificates.                                  *
//*                                          *
//*****
//*
//* Caution: This is neither a JCL procedure nor a complete job. *
//* Before using this job step, you will have to make the following *
//* modifications:                                       *
//*                                          *
//* 1) Change the job card to meet your system requirements. *
//*                                          *
//* 2) If you wish to change the data set qualifiers from the *
//* default value change all occurrences of "PKISRVD.VSAM" *
//* to a preferred value. If you choose to modify this value, be *
//* be sure to also modify the sample configuration file *
//* appropriately(/etc/pkiserv/pkiserv.conf). *
//*                                          *
//* 3) If you are using VSAM record level sharing (RLS), perform *
//* the following steps: *
//*                                          *
//* a) Replace the VOL(vvvvvv) statements with *
//* STORCLAS(class-name) where class-name is the name of the *
//* storage class defined for VSAM RLS. *
//*                                          *
//* b) Remove the SPANNED and CISIZE statements. *
//*                                          *
//* If not using VSAM RLS, change vvvvvv to the VOLSER value *
//* appropriate for the system this job is to be run on. Do not *
//* remove the SPANNED and CISIZE statements. *
//*                                          *
//*                                          *
//* 4) If you wish to change the default userid to own the VSAM *
//* data set, change the OWNER(PKISRVD) operand to the userid you *
//* want to own the data sets. If you choose to modify this value *
//* ensure you have modified the sample setup REXX exec (IKYSETUP) *
//* to account for this change. *
//*                                          *
//*                                          *
//* 5) If you wish to change either the primary or secondary record *
//* allocation sizes for either the OST or ICL datasets from the *
//* default value, update the RECORDS(50 50) operands on the *
//*
```

```

/**      DEFINE CLUSTER or DEFINE ALTERNATE INDEX commands.      *
/**      **Note, do not change any of the numeric values other than *
/**      RECORDS(50 50)                                          *
/**-----*
/** Change Activity:                                           *
/**      $L1=PKIS3   HKY7707 020314 PDJWS1: VSAM RLS           @L1A*
/**      $P1=MG00719 HKY7707 020416 PDJWS1: VSAM RLS 2       @P1A*
/**      Change Description:                                     *
/**      C: Added STORCLAS instructions, LOG. Removed VOLUME DDs @L1C*
/**      D: Removed FILE(VOLUME) statements                    @P1A*
/**-----*
/**      *
/**-----*
/** Delete cluster, AIX, and PATH if they already exist      *
/**-----*
//DELCLUST EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN DD *
DELETE -
    PKISRVD.VSAM.OST -
    CLUSTER -
    PURGE -
    ERASE
DELETE -
    PKISRVD.VSAM.ICL -
    CLUSTER -
    PURGE -
    ERASE
    IF LASTCC 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) -
    SPANNED -
    CISZ(512) -
    RECORDS(50 50) -
    LOG(NONE) -
    OWNER(PKISRVD) ) -
DATA -
    (NAME(PKISRVD.VSAM.OST.DA)) -
INDEX -
    (NAME(PKISRVD.VSAM.OST.IX))

DEFINE CLUSTER -
    (NAME(PKISRVD.VSAM.ICL) -
    VOL(vvvvvv) -
    RECSZ(1024 32756) -
    INDEXED -
    NOREUSE -
    KEYS(4 0) -
    SHR(2) -
    SPANNED -

```

Other code samples

```
        CISZ(512) -
        RECORDS(50 50) -
        LOG(NONE) -
        OWNER(PKISRVD) ) -
DATA -
        (NAME(PKISRVD.VSAM.ICL.DA)) -
INDEX -
        (NAME(PKISRVD.VSAM.ICL.IX))
/*
//*-----*
//*  Repro record of all binary zeros into KSDS  *
//*-----*
//REPROKSD EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSDATA DD   *
```

IKYRVSAM

IKYRVSAM is sample IDCAMS JCL to create VSAM data sets that you use if you intend to use parallel sysplex support and are migrating from z/OS Version 1 Release 3. IKYRVSAM is installed as a member of SYS1.SAMPLIB.

Note: The example that follows might not be identical to the code shipped with the product. To view the most current code, see SYS1.SAMPLIB member IKYRVSAM.

```
//IKYRVSAM JOB <job card parameters>
//*****
//*  SAMP:      IKYRVSAM                               *
//*                                                    *
//*  Licensed Materials - Property of IBM             *
//*  5694-A01                                         *
//*  (C) Copyright IBM Corp. 2002                   *
//*  Status = HKY7707                                 *
//*                                                    *
//*****
//*
//*  This sample JCL may be used to reallocate the VSAM data sets *
//*  in a storage class acceptable to VSAM record level sharing (RLS). *
//*  This is a prerequisite to using PKI Services SYSPLEX support. *
//*                                                    *
//*****
//*
//*  Caution: This is neither a JCL procedure nor a complete job. *
//*  Before using this job step, you will have to make the following *
//*  modifications: *
//*                                                    *
//*  1) Change the job card to meet your system requirements. *
//*                                                    *
//*  2) Change the STORCLAS statements to provide the name of the *
//*  storage class defined for use with VSAM RLS. *
//*                                                    *
//*  3) This job assumes you are using the default VSAM data set *
//*  names (all have high level qualifiers "PKISRVD.VSAM"). If *
//*  you have changed these data set names, you will need to *
//*  modify the source data set names in the ALTER *
//*  statements of the RENAMEDS step. *
//*                                                    *
//*  4) This job creates destination data sets with the same default *
//*  names as the source data sets. (The source data sets are *
//*  renamed.) If you wish to use different destination data set *
//*  names, you will need to modify the data set names in all *
//*  steps except the RENAMEDS step. If you modify these names, *
//*  be sure to also modify your configuration file appropriately *
//*  (/etc/pkiserv/pkiserv.conf). *
//*                                                    *
```

```

/**
/** 5) This job renames the source data sets to begin with high
/** level qualifiers "PKISRVD.OLDVSAM". If you wish to change
/** these names, you will need to do so in the RENAMEDS and
/** and REPROCL steps.
/**
/** 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
/** RECORDS(50 50)
/**-----*
/** Change Activity:
/**
/** $L0=PKIS3 HKY7707 020314 PDJWS1: VSAM RLS
/** $P1=MG00719 HKY7707 020416 PDJWS1: VSAM RLS 2 @P1A*
/**
/** Change Description:
/**
/** C: Removed SPANNED, CISIZE, and FILE(VOLUME) statements @P1A*
/**-----*
/**
/**-----*
/** 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.AIX.IX -
    NEWNAME(PKISRVD.OLDVSAM.AIX.IX)
/*
/**-----*
/** Define destination Clusters
/**-----*
//DEFKSDS EXEC PGM=IDCAMS,COND=(8,LE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DEFINE CLUSTER -
        (NAME(PKISRVD.VSAM.OST) -
        STORCLAS(class-name) -
        RECSZ(1024 32756) -
        INDEXED -
        NOREUSE -
        KEYS(4 0) -
        SHR(2) -
        RECORDS(50 50) -
        LOG(NONE) -
        OWNER(PKISRVD) ) -
    DATA -

```

Other code samples

```
        (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) -
        RECORDS(50 50) -
        OWNER(PKISRVD) ) -
DATA -
        (NAME(PKISRVD.VSAM.ICL.DA)) -
INDEX -
        (NAME(PKISRVD.VSAM.ICL.IX))
/*
/*-----*
/* Repro source cluster to destination cluster *
/*-----*
//REPROCL EXEC PGM=IDCAMS,COND=(8,LE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        REPRO INDATASET(PKISRVD.OLDVSAM.OST) -
                OUTDATASET(PKISRVD.VSAM.OST)
        REPRO INDATASET(PKISRVD.OLDVSAM.ICL) -
                OUTDATASET(PKISRVD.VSAM.ICL)
/*
/*-----*
/* Define ALTERNATE INDEX AND PATH *
/*-----*
//DEFALTDX EXEC PGM=IDCAMS,COND=(8,LE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE ALTERNATEINDEX -
                (NAME(PKISRVD.VSAM.OST.AIX) -
                RELATE(PKISRVD.VSAM.OST)-
                RECORDS(50 50) -
                KEYS(24 44) ) -
DATA -
        (NAME(PKISRVD.VSAM.OST.AIX.DA)) -
INDEX -
        (NAME(PKISRVD.VSAM.AIX.IX))
        DEFINE PATH -
                (NAME(PKISRVD.VSAM.OST.PATH) -
                PATHENTRY(PKISRVD.VSAM.OST.AIX))
/*
/*-----*
/* BUILD ALTERNATE INDEX *
/*-----*
//BLDINDEX EXEC PGM=IDCAMS,COND=(8,LE)
//BASEDD DD DSNAME=PKISRVD.VSAM.OST,DISP=OLD
//AIXDD DD DSNAME=PKISRVD.VSAM.OST.AIX,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        BLDINDEX INFILE(BASEDD) -
                OUTFILE(AIXDD)
/*
```

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 (timezone) environment variable, which is the environment variable most likely to change. You need to specify any other environment variables that PKI Services needs in an environment variables file, by default pkiserv.envars. PKISERVD contains FN (file name) and DIR (directory) parameters, to specify the pathname of the environment variables file. You can make any needed changes in PKISERVD, such as updating this pathname.

Recommendation: By default, the pathname for the pkiserv.envars environment variables file is /usr/lpp/pkiserv/samples/pkiserv.envars. If you need to make changes in the environment variables file, you need to copy it from the samples directory to another directory. IBM recommends that you specify your environment variables using an environment variables file under the /etc directory, for example /etc/pkiserv/pkiserv.envars.

The code sample that follows might not be identical to the code shipped with the product. To see the most current code, see 'SYS1.PROCLIB' member PKISERVD.

```
//*****
//*
//*          Licensed Materials - Property of IBM          *
//*          5694-A01                                       *
//*          (C) Copyright IBM Corp. 2001                 *
//*          Status=HKY7706                                 *
//*
//*****
//*****
//*****
//* Procedure for starting the PKI Services Daemon      *
//*
//*****
//PKISERVD PROC REGSIZE=256M,                               X
//          OUTCLASS='A',                                   X
//          TZ='EST5EDT',                                    X
//          FN='pkiserv.envars',                             X
//          DIR='/usr/lpp/pkiserv/samples',                  X
//          STDO='1>DD:STDOUT',                              X
//          STDE='2>DD:STDERR'
//*-----
//GO      EXEC  PGM=IKYPKID,REGION=&REGSIZE,TIME=1440,
//  PARM=('ENVAR("_CEE_ENVFILE=&DIR/&FN", "TZ=&TZ") / &STDO &STDE')
//STDOUT  DD   SYSOUT=&OUTCLASS
//STDERR  DD   SYSOUT=&OUTCLASS
//SYSOUT  DD   SYSOUT=&OUTCLASS
//CEEDUMP DD   SYSOUT=&OUTCLASS
```

Other code samples

Chapter 28. The certificate validation service

This chapter:

- Provides an overview of PKITP, the PKI Services Trust Policy plug-in for OCSF
- Describes certificate policies and extensions
- Explains how to perform additional OCEP configuration needed for PKITP
- Describes the Trust Policy API, `CSSM_TP_PassThrough`

Overview

The PKI Services Trust Policy (PKITP) is an OCSF plug-in to perform certificate validation. It 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 (users, other servers, and so forth) present. PKI Services or other certificate authorities may have issued these certificates.

Before using this plug-in, the server administrator must create a SAF key ring containing the certificates of trusted CAs (anchor certificates). (See *z/OS Security Server RACF Command Language Reference* for how to create a SAF key ring.) This key ring can also contain trusted site certificates if appropriate.

The server application must attach to and open this key ring using the OCEP DL plug-in. (See *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming* for more information on OCEP and the use of SAF key rings.) 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.

The certificate validation service

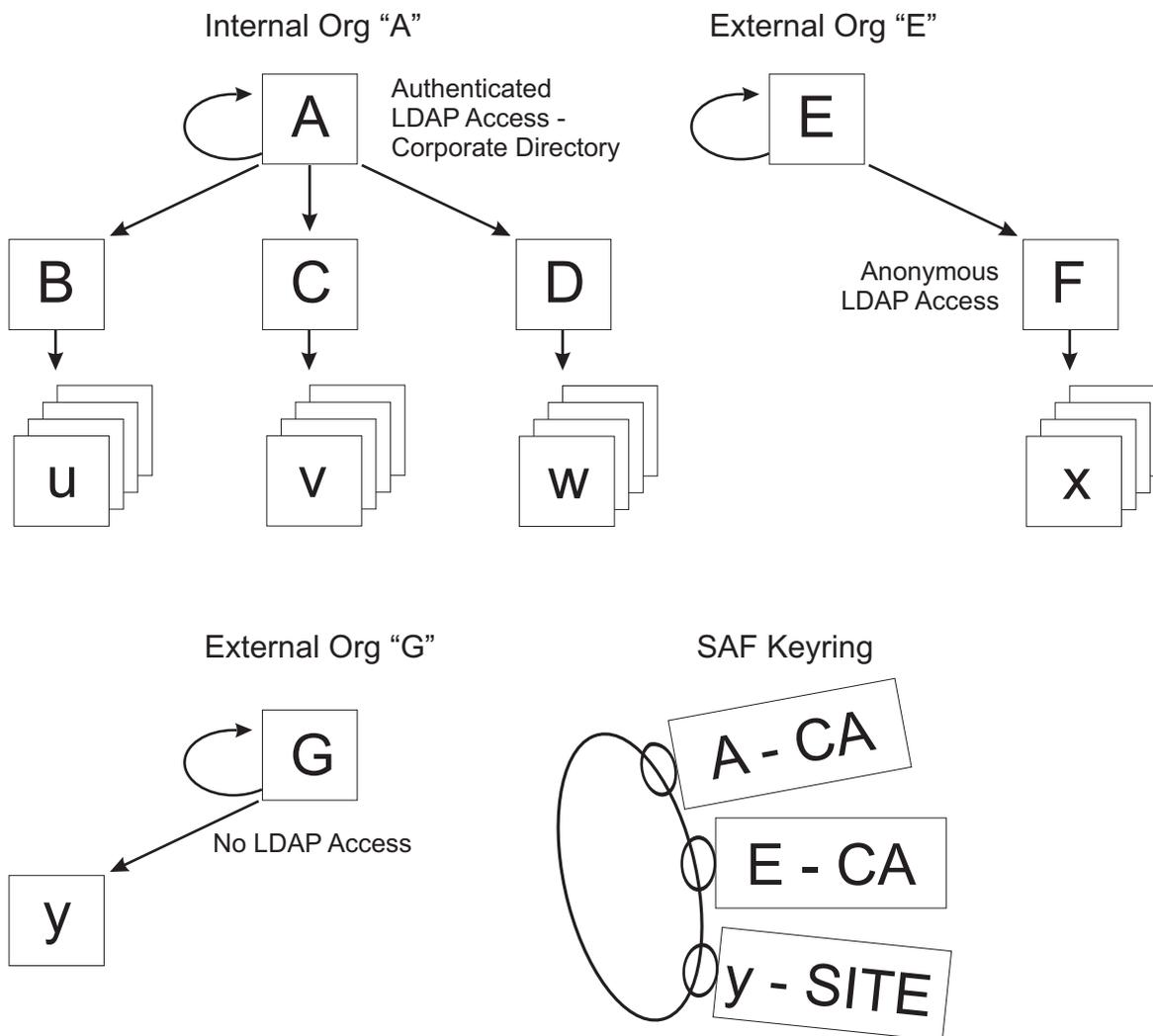


Figure 38. Examples of organizations, certificates, and chains

Organization A represents the local (corporate) certificate hierarchy. It contains one self-signed root certificate 'A.' Perhaps RACF or the Tivoli PKI created this. 'B', 'C', and 'D' are intermediate CAs. They could be separate instances of PKI Services. Certificates issued within this hierarchy are stored in an LDAP directory accessible to corporate server applications.

Organization E represents a public or business partner's certificate hierarchy with an LDAP directory that allows anonymous access. Organization G represents some other certificate hierarchy, in which either the directory does not exist or it is not accessible. The key ring contains three anchor certificates. Certificates 'A' and 'E' are trusted CAs, and there is a business need to trust end-entity certificate y, even though it cannot be verified. If each of these CAs has posted current CRLs to their default locations (PKI Services has no support for CRL distribution points) and all certificate chains to be verified are genuine, the PKITP CertGroupVerify function can validate the following input chains:

- Single certificates x, u, v, or w (PKITP can extract the missing links from the directories.)
- Chains u-B, v-C, w-D, u-B-A, v-C-A, w-D-A, x-F, or x-F-E (These chains have no missing links.)

The certificate validation service

- Any chain beginning with certificate *y* (As Figure 38 on page 338 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.

Recommendation: 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.

Note: For certificates that PKI Services generates, the PKI Services configuration file parameters PolicyRequired and PolicyCritical define whether the extension exists and whether it is marked critical, respectively. By default the certificate policies extension is *not* included in a certificate and the critical flag is *not* turned on. (For more information, see Table 20 on page 58.)

Likewise, 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.

Certificate extensions

PKITP supports the following certificate extensions:

- SubjectKeyIdentifier — Checked for form only.
- 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.
- IssuerAltName — Checked for form only. Must be marked critical if the Issuer DN is empty.

The certificate validation service

- BasicConstraints — For CA certificate, cA flag must be on. Also checked for certification path length.
- CertificatePolicies — See preceding description.
- AuthorityKeyIdentifier — Checked for form only.
- HostIdMappings — 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
- InvalidationDate

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 70. Summary of information about important files for PKITP

File	Description	Source location (default)
Makefile.pkitpsamp	Makefile for pkitpsamp.c.	/usr/lpp/pkiserv/samples/
install_pkitp	Program that registers the PKI Services Trust Policy plug-in with OCSF.	/usr/lpp/pkiserv/bin
pkitp_ivp	This program verifies that the plug-in installed successfully.	/usr/lpp/pkiserv/bin
pkitp.h	Contains #defines for applications calling the PKI Services OCSF Trust Policy.	/usr/lpp/pkiserv/include/
pkitp.so	This is the OCSF Trust Policy plug-in for PKI Services.	/usr/lpp/pkiserv/lib
pkitpsamp.c	Sample application program (in the C language) to call the PKI Trust Policy plug-in.	/usr/lpp/pkiserv/samples

Configuring and getting started with PKITP

If you have not already installed and configured OCEP, you need to do so now. See “Installing and configuring OCSF and OCEP” on page 29 for more information. To install PKITP, you need to follow all of the configuration instructions in *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming* and then perform the following post-installation instructions. The PKITP must be registered with OCSF before being used.

Steps for configuring PKITP

Before you begin: If you have not already done so, run the OCSF and OCEP install and verification scripts.

Perform the following steps to install and configure PKITP:

1. Run the PKITP post installation script by entering the following command:

```
/usr/lpp/pkiserv/bin/install_pkitp
```

The program prompts you for certain information. Assuming PKI Services has been installed in its default location, answer the prompts as follows:

Prompt	Response
addin directory?	/usr/lpp/pkiserv/lib
addin filename?	pkitp.so
action? [install uninstall]	install

You know you are done and that the installation was successful when you see the following:

```
Installing IBMPKITP...
Addin successfully installed.
```

2. To verify that the installation was successful, run the verification program (/usr/lpp/pkiserv/bin/pkitp_ivp).

You know you are done and that the verification program ran successfully when you see the following:

```
Starting pkitp IVP
Initializing CSSM
CSSM Initialized
Attaching pkitp
Attach successful, Detaching pkitp
Detach of pkitp successful
Completed pkitp IVP
```

Trust Policy API

Programming Interface information

PKITP supports only one API, CSSM_TP_PassThrough. The Globally Unique Identifier (GUID) for this plug-in is: {01EBC8AC-CC6F-450c-83B4-F0BE0FBE78F9}. (Before an application can use a module, an installation application must register the module’s name, location, and description with OCSF. The name given to a

The certificate validation service

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:

CSSM_TP_PassThrough

```
typedef struct tp_verify_extra {
    /* similar parameters as TP_CertGroupVerify */
    CSSM_CL_HANDLE CLHandle; - Not used. Set to 0. Ignored by PKITP
    CSSM_DL_DB_LIST_PTR DBList; - List of
    CSSM_DL_DB_HANDLE, see below
    CSSM_CSP_HANDLE CSPHandle; - Handle to IBMSWCSP module
    CSSM_FIELD_PTR PolicyIdentifiers; - Not used. Set to 0. Ignored by PKITP
    uint32 NumberOfPolicyIdentifiers; - Not used. Set to 0. Ignored by PKITP
    CSSM_TP_STOP_ON VerificationAbortOn; - Must be set to CSSM_TP_STOP_ON_POLICY
    CSSM_CERTGROUP_PTR CertToBeVerified; - Address of cert group struct to verify
    CSSM_DATA_PTR AnchorCerts; - Not used. Set to 0. Ignored by PKITP
    uint32 NumberOfAnchorCerts; - Not used. Set to 0. Ignored by PKITP

    /* extra parameters: input */
    TP_INITIALPOLICY_PTR InitialPolicy; - Address of policy struct or 0 , see below
    time_t CurrentTime; - Not used. Set to 0. Ignored by PKITP
    time_t ValidationTime; - Time to use for validation, e.g., time(0)

    /* extra parameters: output */
    CSSM_BOOL result; - Success indicator
    uint32 DLStatusCode - Status code from DL failures
    uint32 DLIndex - Index (from 0) into DBList of failing DL
    TP_EVIDENCE_PTR Evidence; - Address of evidence struct or 0, see below
} 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 SAF key ring). The key ring is used to declare the list of trusted CA and SITE certificates. Like the OCEP Trust Policy, certificate chains to verify must originate from one of these trusted CAs (anchors) or the end-entity certificate must be one of the SITE certificates. Also like the OCEP Trust Policy, if the security product (SAF) marks any certificate in the candidate chain NOTRUST, the certificate chain fails validation.

The other entries in the list are used for LDAPDL DB stores. PKITP runs through these to locate CRLs and intermediate CA certificates. For each item PKITP requests, the LDAPDLs are queried in the order in which they appear in the list. The search stops the first time an LDAPDL returns an item or when the OCEPDL is reached. No query is made to the OCEPDL to locate CRLs or intermediate CA certificates.

The initial policy

The following optional, caller-provided and initialized structure defines InitialPolicy. PKITP uses the default values if the structure is not provided:

```

typedef struct tp_initialpolicy {
  /* initial-policy-set - To be used if your application is policy specific */
  uint32 NumberOfPolicyIdentifiers;           - Number of application specific policy OIDs
                                              (defaults to 0)
  CSSM_OID_PTR PolicyIdentifiers;           - Address of array of policy OIDs or 0

  /* check certificates against CRLs */
  uint32 useCRLs;                            - 0 - no CRL processing, 1 -Search for CRLs
                                              but, continue if search fails, 2 - Strong CRL
                                              checking (defaults to 2). Must be set to 0 if
                                              DBList contains no LDAPDL DB stores.

  /* initial-explicit-policy indicator */
  CSSM_BOOL initialExplicitPolicy;           - Indicates that PKITP should consider the
                                              policy set critical (defaults to false)

  /* initial-policy-mapping-inhibit indicator *
  CSSM_BOOL initialPolicyMappingInhibit;    - Not used. Ignored by PKITP
} TP_INITIALPOLICY, *TP_INITIALPOLICY_PTR;

```

The evidence

The following optional, caller-provided structure defines the evidence. This structure is used to return information relative to the validation decision PKITP makes. The caller must free the data areas returned. (The FreeEvidence pass-through function is provided for this.)

```

typedef struct tp_evidence {
  /* valid certification path if validation succeeds */
  CSSM_CERTGROUP_PTR CompleteCertGroup;     - Cert group from EE to anchor CA

  /* relevant CRL if validation fails */
  CSSM_DATA_PTR CRL;                        - CRL for revoked cert or incorrect CRL

  /* relevant certificate if validation fails */
  CSSM_DATA_PTR Cert;                       - Certificate causing the failure

  /* authority-constrained-policy */
  CSSM_BOOL authAnyPolicy;                  - false - critical certificatePolicies found

  uint32 NumberOfAuthCertPolicyIdentifiers; - Nonzero if authAnyPolicy is false
  CSSM_OID_PTR AuthCertPolicyIdentifiers;   - Array of policy OIDs

  /* list of policy mappings that occurred */
  uint32 NumberOfMappedPolicies;           - Not used. Ignored by PKITP
  TP_CSSM_OID_PAIR_PTR mappedPolicies;     - Not used. Ignored by PKITP
} TP_EVIDENCE, *TP_EVIDENCE_PTR;

```

Error codes

Table 71 lists the error codes that are unique to PKI Services OCSF Trust Policy (PKITP).

Table 71. PKI Services OCSF Trust Policy (PKITP) error codes

Decimal Value	Error Description
8001	Certificate encoding error. Incorrect CertificatePolicies extension.
8002	Certificate policies violation.
8003	Incorrect certificate distinguished name chaining.
8004	Certificate encoding error. Subject name missing.
8006	Incorrect certificate BasicConstraints extension - cA flag off in signing certificate.

CSSM_TP_PassThrough

Table 71. PKI Services OCSF Trust Policy (PKITP) error codes (continued)

Decimal Value	Error Description
8008	Incorrect certificate keyUsage extension - keyCertSign flag off in signing certificate.
8010	Unsupported AltName form in certificate.
8013	Certificate or CRL encoding error. Signature algorithm mismatch.
8014	Certificate encoding error. Incorrect version.
8015	CRL encoding error. Incorrect version.
8016	Unsupported critical extension in certificate.
8017	Unsupported critical extension in CRL.
8018	Unsupported critical entry extension in CRL.
8019	Certificate encoding error. Duplicate extension.
8020	CRL encoding error. Duplicate extension.
8021	Certificate signature failed verification.
8022	CRL signature failed verification.
8023	Incorrect date range in certificate or CRL. NotAfter earlier than NotBefore.
8024	Certificate's date range is in the future.
8025	Certificate has expired.
8026	CRL's date range is in the future.
8027	CRL has expired.
8028	DBList incorrect, no LDAPDL DBs or non-LDAPDL specified.
8029	CRL not found.
8030	Certificate is revoked.
8031	Unable to build certificate chain.
8033	Certificate not trusted.
8501	Unexpected status code returned from accessing LDAPDL.
8502	Unexpected status code returned from accessing OCEPDL.
8503	DBList incorrect, no OCEPDL DB or DB empty.

Providing the certificate validation service

To perform certificate validation, your server application calls the CSSM_TP_PassThrough API (see CSSM_TP_PassThrough on page 343), 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 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 “Example of using PKITP program” on page 348), 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).

Note: If you have more than one LDAP server, you need to provide this information for each LDAP server.

 - c. Specify the user ID and keyname for the SAF key ring containing trusted CA or site certificates (in the sample, this is `G9VEMER/myring`).
 - d. If necessary, change the value of the `CSSM_GUID`:
 - IBMSWCSP_GUID** This is the value in the sample.
 - IBMWKCSPP_GUID** This GUID is for use when the IBMSWCSP plug-in is not available (for example, US export-controlled locations).
 - e. 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.)
 - f. 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.
 - g. 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:

CSSM_TP_PassThrough

make

-
4. Run the pkitsamp.c in your own directory by entering the following command:

pkitsamp

Example of using PKITP program

Note: The example that follows might not be identical to the code shipped with the product. If you want to see the most current code, look in the /usr/lpp/pkiserv/samples directory.

```
/* **** */
/* This file contains sample code. IBM provides this code on an */
/* 'as is' basis without warranty of any kind, either express or */
/* implied, including but not limited to, the implied warranties */
/* of merchantability or fitness for a particular purpose. */
/* **** */
/* **** */
/* Licensed Materials - Property of IBM */
/* 5694-A01 */
/* (C) Copyright IBM Corp. 2001 */
/* Status = HKY7706 */
/* **** */
/* **** */
/* Sample use of IBM PKITP program */
/* **** */
/* Purpose: Program attaches needed CSSM modules, then prompts */
/* the user for filename(s) containing DER encoded */
/* certificates. The certificate(s) are read from the */
/* file, then passed to PKITP for verification. */
/* A summary of the results are printed to stdout. */
/* **** */
/* Caution: In order to run this sample program, modification MUST */
/* BE MADE to several values assigned to the following */
/* variables that are defined between the block comment */
/* containing the text "Start of application specific */
/* options" and the block comment containing the text */
/* "End of application specific options"(without the */
/* quotation marks): */
/* **** */
/* #define NUM_LDAPS 1 */
/* Define the number of LDAP servers that PKITP should */
/* query for certificates, CRLs and ARLs. This can be 0, */
/* if entire certificate chain will be passed as input to */
/* PKITP AND caller requests to NOT process CRLs/ARLs (see */
/* useCRLs option below). */
/* **** */
/* struct ldap_info ldapservers[NUM_LDAPS] = */
/* { "@LDAPSERVERNAME:PORTNUMBER@", */
/* "@LDAPUSER@", */
/* "@LDAPUSERPASSWORD@"}; */
/* If NUM_LDAPS > 0, then ldapservers 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.                                             */
/*
/*      CSSM_GUID csp_guid = IBMSWCSP_GUID;
/*      The CSSM GUID (globally unique id) for Cryptographic
/*      Service Provider. PKITP will call the specified CSP to
/*      verify signatures in the certificate chain. The
/*      csp_guid variable is set to use the software CSP,
/*      IBMSWCSP_GUID, but may to set to either IBMWKCSP_GUID
/*      or IBMCCA_GUID.
/*
/*      #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
/*      cheching, a valid CRL must be found for each CA
/*      certificate in the chain.)
/*
/*      #define NUM_POLICIES 2
/*      static unsigned char my_policy1[5] =
/*      {0x06,0x03,0x2a,0x03,0x04}; // DER encoded 1.2.3.4
/*      static unsigned char my_policy2[7] =
/*      {0x06,0x05,0x2a,0x03,0x03,0x02,0x01}; // DER 1.2.3.3.2.1
/*      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 certicate
/*      policy error.
/*
/*      #define INITIALExplicitPolicy FALSE
/*      Set to true if you want PKITP to require that all
/*      certificates in chain to have at least one policy
/*      listed by the policydata defined above.
/*
/*
/*
/*****
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <cssm.h>
#include <ibmocepd1.h>
#include <ibmswcsp.h>
#include <cssmapi.h>
#include <cssmtype.h>
#include <pkitp.h>
#include <ldapdl.h>

struct ldap_info
{
    char * ldapserver;
    char * ldapauthuser;
    char * ldapauthpass;
};

//-----

```

CSSM_TP_PassThrough

```
// storage function definitions needed to talk to CSSM
//-----

#ifdef __cplusplus
extern "C"
#endif
void * OurMalloc(size_t size, void * allocRef)
{
    return malloc(size);
}

#ifdef __cplusplus
extern "C"
#endif
void OurFree(void* memPtr, void * allocRef)
{
    free(memPtr);
}

#ifdef __cplusplus
extern "C"
#endif
void * OurRealloc(void * memPtr,
                  size_t size,
                  void * allocRef)
{
    return realloc(memPtr, size);
}

#ifdef __cplusplus
extern "C"
#endif
void * OurCalloc(size_t num,
                 size_t size,
                 void* allocRef)
{
    return calloc(num, size);
}

static CSSM_API_MEMORY_FUNCS memoryFuncs; // used to pass function addresses to CSSM
static CSSM_CSP_HANDLE      ibm_csp_handle;

//-----
// internal function declarations
//-----
int connectTP(char * ringname,
              int number_ldap,
              struct ldap_info *,
              CSSM_DL_DB_LIST *,
              CSSM_TP_HANDLE *);

void disconnectTP(CSSM_DL_DB_LIST *, CSSM_TP_HANDLE);

int buildCertGroup(CSSM_CERTGROUP *, char * [], uint32);
void verifyCertGroup(CSSM_CERTGROUP certgroup,
                    CSSM_DL_DB_LIST * datasources_ptr,
                    CSSM_TP_HANDLE tphandle);

void
reportCertGroupVerify
    (TP_VERIFY_EXTRA extraVerifyInfo);

void printEvidence(TP_EVIDENCE_PTR evidence_ptr);

void freeCertGroup(CSSM_CERTGROUP * certGroupPtr);
////////////////////////////////////
//
```

```

// Start of application specific options
//
// The defines and declarations that follow should be altered to fit the
// particular application calling PKITP.
//
////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////
// Define the number of LDAP servers that PKITP should query for certificates,
// CRLs and ARLs. This can be 0, if entire certificate chain will be passed as
// input to PKITP AND caller requests to NOT process CRLs/ARLs (see useCRLs
// option below).
//
// If NUM_LDAPS > 0, then ldapserver array should define the LDAP server:port,
// user and password for each LDAP server, as this example shows.
////////////////////////////////////////////////////////////////
#define NUM_LDAPS 1

struct ldap_info ldapserver[NUM_LDAPS] =
    { "@LDAPSERVERNAME:PORTNUMBER@", // LDAP server:port
      "@LDAPUSER@", // user
      "@LDAPUSERPASSWORD@"}; // password

////////////////////////////////////////////////////////////////
// Define the SAF keyring containing trusted CA and/or site certificates.
// Format is "USERID/keyname"
////////////////////////////////////////////////////////////////
char keyring[] = "@USERID@/KEYRINGNAME@";

////////////////////////////////////////////////////////////////
// The CSSM GUID (globally unique id) for Cryptographic Service Provider.
// PKITP will call the specified CSP to verify signatures in the certificate
// chain. Must be either: IBMSWCSP_GUID, IBMWKCSP_GUID or IBMCCA_GUID
////////////////////////////////////////////////////////////////
CSSM_GUID csp_guid = IBMSWCSP_GUID;

////////////////////////////////////////////////////////////////
// Define how the useCRLs option should be set.
// Set to 0 if no CRL processing to be done
// Set to 1, if we are to query LDAP for CRLs and process those found
// Set to 2, for strong CRL checking -- must find CRLs in LDAP.
////////////////////////////////////////////////////////////////
#define USECRLS 1

////////////////////////////////////////////////////////////////
// Define the policies that the application calling PKITP uses.
//
// These become important if a certificate in the certificate chain has a
// critically marked policy extension. At least one policy
// that is listed in such a critically marked policy extension, must appear
// in the list defined here or PKITP will return certificate policy error.
////////////////////////////////////////////////////////////////
#define NUM_POLICIES 2

static unsigned char my_policy1[5] = {0x06,0x03,0x2a,0x03,0x04}; // DER encoded 1.2.3.4
static unsigned char my_policy2[7] = {0x06,0x05,0x2a,0x03,0x03,0x02,0x01}; // DER 1.2.3.3.2.1

CSSM_DATA policydata[NUM_POLICIES] = {{sizeof(my_policy1),(unsigned char *)my_policy1},
                                       {sizeof(my_policy2),(unsigned char *)my_policy2}};

#define 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

```

CSSM_TP_PassThrough

```
////////////////////////////////////
//
// End of application specific options
//
////////////////////////////////////

//-----
// main
//-----
int
main(int argc, char* argv[])
{

    CSSM_DL_DB_LIST datasources;
    CSSM_TP_HANDLE tphandle = 0;
    CSSM_CERTGROUP certGroup;
    int repeating = 1;
    char buffer[1024];
    int num_certs = 0;
    char * cert_files[25];
    char * next_file;
    char * input;

    int rc;

    rc = connectTP(keyring,NUM_LDAPS,ldapservers, &datasources, &tphandle);
    if (rc == 0)
    {
        //////////////////////////////////////
        // prompt for certificates to verify
        //////////////////////////////////////
        do
        {
            num_certs = 0;
            printf("Enter filename(s) of certificate(s). (List EE first). ");
            printf("Blank line to quit.\n");

            if ((input = gets(buffer)) != NULL) // get input line
            {
                next_file = strtok(input," ");
                while ((next_file != NULL) && (num_certs < 25)) // tokenize it
                {
                    cert_files[num_certs] = next_file;
                    num_certs++;
                    next_file = strtok(NULL," ");
                }
            }

            //////////////////////////////////////
            // If we were given a list of files containing certificates, input them to TP
            //////////////////////////////////////
            if (num_certs > 0)
            {
                rc = buildCertGroup(&certGroup, cert_files, num_certs);
                if (rc == 0)
                {
                    verifyCertGroup(certGroup, &datasources, tphandle);
                    freeCertGroup(&certGroup);
                }
            }
        } while (num_certs > 0);
    }
    disconnectTP(&datasources, tphandle);
}

//-----
```

```

// connectTP
//
// Purpose: connect to the datasources PKITP needs
// then connect to the PKITP
//
// Input: ringname - string containing "USERID/ringname" of SAF
//         keyring containing trusted CA and/or SITE certificates
//         number_ldap - number of ldap servers
//         ldaps - array of ldap_info structures
//
// Output: The CSSM_DL_DB_LIST structure addressed by datasources will have
//         been initialized with the various handles that CSSM_ModuleAttach
//         and CSSM_DL_DbOpen calls have returned
//         The CSSM_TP_HANDLE addressed by tphandle_ptr will have been initialized.
//         int returned will be 0 if successful, -1 if not successful.
//-----
int connectTP(char * ringname,
             int number_ldap,
             struct ldap_info * ldaps,
             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_VERSION CSP_version = {IBMSWCSP_MAJOR_VERSION, IBMSWCSP_MINOR_VERSION};
    CSSM_DB_ACCESS_TYPE access = { CSSM_TRUE,
                                   CSSM_FALSE,
                                   CSSM_FALSE,
                                   CSSM_FALSE};

    CSSM_VERSION DL_version;
    CSSM_DL_HANDLE LDAP_dhhandle;
    CSSM_MODULE_INFO* moduleInfoPtr;
    void * voidptr;
    CSSM_DB_ACCESS_TYPE accessRequest = { CSSM_TRUE,    // ReadAccess
                                         CSSM_TRUE,    // WriteAccess
                                         CSSM_FALSE,   // PrivilegedMode
                                         CSSM_FALSE }; // Asynchronous

    memoryFuncs.malloc_func = OurMalloc;
    memoryFuncs.free_func = OurFree;
    memoryFuncs.realloc_func = OurRealloc;
    memoryFuncs.calloc_func = OurCalloc;
    memoryFuncs.AllocRef = NULL;

    DL_version.Major = IBMOCEPDL_MAJOR_VERSION;
    DL_version.Minor = IBMOCEPDL_MINOR_VERSION;

    datasources_ptr->NumHandles = number_ldap + 1;
    voidptr = malloc(sizeof(CSSM_DL_DB_HANDLE)*(number_ldap +1)); // get storage for DBlist
    memset(voidptr,0,(sizeof(CSSM_DL_DB_HANDLE)*(number_ldap +1))); // zero it
    datasources_ptr->DLDBHandle = (CSSM_DL_DB_HANDLE *)voidptr;

    if (CSSM_Init(&cssm_version, &memoryFuncs, NULL) != CSSM_OK)
    {
        printf("Failed CSSM_Init: %d, line %d\n",CSSM_GetError()->error,__LINE__);
        return -1;
    }

    // attach to LDAP and open each LDAP DB
    if (number_ldap > 0) // if we have any LDAP sources
    {
        moduleInfoPtr = CSSM_GetModuleInfo((CSSM_GUID*)&LDAPDL_GUID,

```

CSSM_TP_PassThrough

```

                                CSSM_SERVICE_DL,
                                CSSM_ALL_SUBSERVICES,
                                CSSM_INFO_LEVEL_ALL_ATTR);
if (!moduleInfoPtr)
{
    printf("Failed CSSM_GetModuleInfo: %d, line %d\n",CSSM_GetError()->error,__LINE__);
    return -1;
}

LDAP_dlhandle = CSSM_ModuleAttach((CSSM_GUID*)&LDAPDL_GUID,
                                &moduleInfoPtr->Version,
                                &memoryFuncs,
                                0,
                                0,
                                0,
                                NULL,
                                NULL);

if (!LDAP_dlhandle)
{
    printf("Failed CSSM_ModuleAttach: %d, line %d\n",CSSM_GetError()->error,__LINE__);
    return -1;
}

// connect to multiple database instances

//-----
// fill in LDAP DL authentication information:
// necessary only if user is supplying a name and password
//-----
for (z = 0; z < number_ldap; z++)          // for each LDAP source
{
    LDAP_BIND_PARMS bindParms;
    CSSM_USER_AUTHENTICATION userAuthentication = {0,0};
    CSSM_DATA_userCredential = {0,0};
    CSSM_USER_AUTHENTICATION_PTR userAuthenticationPtr = 0;

    datasources_ptr->DLDBHandle[z].DLHandle = LDAP_dlhandle;
    if (ldaps[z].ldapauthuser && ldaps[z].ldapauthpass)
    {
        //-----
        // fill in LDAP DL specific data structure: LDAP_BIND_PARMS
        //-----
        bindParms.DN = ldaps[z].ldapauthuser;
        bindParms.SASL = 0;
        bindParms.credentials.Data = (uint8 *)ldaps[z].ldapauthpass;
        bindParms.credentials.Length = strlen(ldaps[z].ldapauthpass)+1;
        userCredential.Length = sizeof(LDAP_BIND_PARMS);
        userCredential.Data = (unsigned char*)&bindParms;
        userAuthentication.Credential = &userCredential;
        userAuthenticationPtr = &userAuthentication;
    }

    //-----
    // Open LDAP DL Database
    //-----
    datasources_ptr->DLDBHandle[z].DBHandle = CSSM_DL_DbOpen(LDAP_dlhandle,
                                                            ldaps[z].ldapservers,
                                                            &accessRequest,
                                                            userAuthenticationPtr,
                                                            (void *)0);
    if (!datasources_ptr->DLDBHandle[z].DBHandle)
    {
        printf("Failed CSSM_DL_DbOpen %d, line %d\n", CSSM_GetError()->error,__LINE__);
        return -1;
    }
}

// end of for each each LDAP source
```

```

if (CSSM_FreeModuleInfo(moduleInfoPtr) == CSSM_FAIL)
{
printf("Failed CSSM_FreeModuleInfo, line %d, error %d\n", __LINE__,
CSSM_GetError()->error);
// This is not a catastrophic error, we'll continue
}
} // end if we have any LDAP sources

////////////////////////////////////
// Attach to OCEP DL (to access RACF keyring)
////////////////////////////////////

datasources_ptr->DLDBHandle[number_ldap].DLHandle =
    CSSM_ModuleAttach(&IBMOCEPDL_GUID,
        &DL_version,
        &memoryFuncs,
        0,
        0,
        0,
        NULL,
        NULL);

if (!(datasources_ptr->DLDBHandle[number_ldap].DLHandle))
{
printf("Failed CSSM_ModuleAttach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
return -1;
}

datasources_ptr->DLDBHandle[number_ldap].DBHandle =
    CSSM_DL_DbOpen(datasources_ptr->DLDBHandle[number_ldap].DLHandle,
        ringname,
        &access,
        NULL,
        NULL);

if (!(datasources_ptr->DLDBHandle[number_ldap].DBHandle))
{
printf("Failed CSSM_DL_DbOpen %d, line %d\n", CSSM_GetError()->error, __LINE__);
return -1;
}

////////////////////////////////////
// Attach to cryptographic service provider - PKITP uses for signature checking
////////////////////////////////////
ibm_csp_handle = CSSM_ModuleAttach(&csp_guid, &CSP_version,
    &memoryFuncs, 0, 0, 0, NULL, NULL);

if (!ibm_csp_handle)
{
printf("Failed CSSM_ModuleAttach %d, line %d\n", CSSM_GetError()->error, __LINE__);
return -1;
}

////////////////////////////////////
// Attach to PKITP
////////////////////////////////////
moduleInfoPtr = CSSM_GetModuleInfo((CSSM_GUID*)&PKITP_GUID,
    CSSM_SERVICE_TP,
    CSSM_ALL_SUBSERVICES,
    CSSM_INFO_LEVEL_ALL_ATTR);

if (!moduleInfoPtr)
{
printf("Failed CSSM_GetModuleInfo: %d, line %d\n", CSSM_GetError()->error, __LINE__);
return -1;
}

```

CSSM_TP_PassThrough

```
*(tphandle_ptr) = CSSM_ModuleAttach((CSSM_GUID*)&PKITP_GUID,
                                     &moduleInfoPtr->Version,
                                     &memoryFuncs,
                                     0,
                                     0,
                                     0,
                                     NULL,
                                     NULL);

if (!(*tphandle_ptr))
{
    printf("Failed CSSM_ModuleAttach: %d, line %d\n",CSSM_GetError()->error,__LINE__);
    return -1;
}

if (CSSM_FreeModuleInfo(moduleInfoPtr) == CSSM_FAIL)
{
    printf("Failed CSSM_FreeModuleInfo, line %d, error %d\n",__LINE__,
          CSSM_GetError()->error);
    // This is not a catastrophic error, we'll continue
}

return 0;
}

//-----
// disconnectTP
//
// Purpose: to close any open databases and detach any CSSM modules
//          that connectTP attached
//
// Input: The CSSM_DL_DB_LIST structure, CSSM_TP_HANDLE,
//        ibm_csp_handle (static variable referenced both places)
//        that were initialized by connectTP.
//
// Output: None
//-----

void disconnectTP(CSSM_DL_DB_LIST * datasources_ptr, CSSM_TP_HANDLE tphandle)
{
    int x;
    int status;

    ////////////////////////////////////////////////////////////////////
    // Sever ties to LDAP
    // For each LDAP database opened -- call CSSM_DL_DbClose
    ////////////////////////////////////////////////////////////////////

    for (x = 0; x < datasources_ptr->NumHandles - 1; x++)
    {
        // we close each ldap database separately
        if (datasources_ptr->DLDBHandle[x].DBHandle) // if we opened database
        {
            status = CSSM_DL_DbClose(datasources_ptr->DLDBHandle[x]);
            if (status != 0)
            {
                printf("Failed CSSM_DL_DbClose %d, line %d\n", CSSM_GetError()->error,__LINE__);
                // we continue trying to close other stuff
            }
        }
    }

    ////////////////////////////////////////////////////////////////////
    // Now detach the LDAP module
    ////////////////////////////////////////////////////////////////////
    if (datasources_ptr->DLDBHandle[0].DLHandle)
    {
```

```

if ((status = CSSM_ModuleDetach(datasources_ptr->DLDBHandle[0].DLHandle)) != 0)
{
    printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
    // we continue trying to close other stuff
}
datasources_ptr->DLDBHandle[0].DLHandle = 0; // clear handle
}

////////////////////////////////////
// Say goodbye to OCEP
////////////////////////////////////
status = CSSM_DL_DbClose(datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1]);
if (status != 0)
{
    printf("Failed CSSM_DL_DbClose %d, line %d\n", CSSM_GetError()->error, __LINE__);
    // we continue trying to close other stuff
}

if (datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle)
{
    if ((status = CSSM_ModuleDetach(
        datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle)) != 0)
    {
        printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        // we continue trying to close other stuff
    }
    datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle = 0;
}

////////////////////////////////////
// Say goodbye to cryptographic service provider (CSP)
////////////////////////////////////
if (ibm_csp_handle)
{
    if (status = CSSM_ModuleDetach(ibm_csp_handle) != 0)
    {
        printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        // we continue trying to close other stuff
    }
}

////////////////////////////////////
// Farewell PKITP
////////////////////////////////////
if (tphandle)
{
    if (status = CSSM_ModuleDetach(tphandle) != 0)
    {
        printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        // we continue trying to close other stuff
    }
}

return;
}

/*****
* name: buildCertGroup - read certificates from files, set up
*       CSSM_CERTGROUP to reference input certificates
*
* input: CSSM_CERTGROUP * -- addresses uninitialized CSSM_CERTGROUP
*        certFile - array of strings containing names of files that
*        have DER encoded certificates to be verified by PKITP
*        certCount - number of elements (strings) in certFile
*
* output: returns CSSM_OK if all certificates read
*         - CSSM_CERTGROUP will have NumCerts set and CertList
*****/

```

CSSM_TP_PassThrough

```
*          will be the address of array of certificates
*          returns CSSM_FALSE if error reading a file
*****/

int buildCertGroup(CSSM_CERTGROUP * certGroupPtr,
                  char * certFile[], uint32 certCount)
{
    FILE * inFile;
    CSSM_DATA * certArray = (CSSM_DATA *) calloc(certCount, sizeof(CSSM_DATA));
    uint32 i, certSize;

    certGroupPtr->NumCerts = certCount;
    certGroupPtr->CertList = certArray;

    for (i=0; i < certCount; i++)
    {
        inFile = fopen(certFile[i], "rb");
        if (!inFile)
        {
            printf("File %s could not be opened\n", certFile[i]);
            if (i > 1) // if we've read any certs before this
            {
                certGroupPtr->NumCerts = i - 1; // indicate how many read
                freeCertGroup(certGroupPtr); // free alloc'd storage
            }
            return(CSSM_FAIL);
        }
        /* Find size of certificate file */
        fseek(inFile, 0L, SEEK_END);
        certSize = ftell(inFile);
        rewind(inFile);

        /* Read in certificate data*/
        certArray[i].Length = certSize;
        certArray[i].Data = (uint8 *)calloc(certSize, sizeof(char));
        fread(certArray[i].Data, 1, certSize, inFile);
        fclose(inFile);
    }
    return(CSSM_OK);
}

/*****
* name: verifyCertGroup - call the Trust Policy (FINALLY)
*
* purpose: call CSSM_TP_PassThrough (PKITP) to verify certificate(s)
*          call reportCertGroupVerify (internal routine to display
*          results to stdout
*          call CSSM_TP_PassThrough (PKITP) to free storage related to
*          results
*
* input:   CSSM_CERTGROUP containing number of and array of certificates
*          CSSM_DL_DB_LIST containing CSSM handles for LDAP and OCEP
*          CSSM_TP_HANDLE CSSM handle for PKITP
*
* output:  none
*
*****/

void verifyCertGroup(CSSM_CERTGROUP certgroup,
                    CSSM_DL_DB_LIST * datasources_ptr,
                    CSSM_TP_HANDLE tphandle)
{
    ////////////////////////////////////////////////////////////////////
    //
    // While there are only 3 parameters on CSSM_TP_PassThrough call to PKITP:
    // - the CSSM_TP_HANDLE,
```

```

// - the function code "TP_VERIFY_PASSTHROUGH" and
// - a pointer to the TP_VERIFY_EXTRA structure.
// TP_VERIFY_EXTRA structure contains many parameters, including the address of
// TP_INITIALPOLICY structure that can be used to override the default
// policy settings and the address of TP_VERIFY_EXTRA which PKITP can use
// to pass back more detailed results.
////////////////////////////////////////////////////////////////////
TP_INITIALPOLICY initialPolicyPreferences;
TP_EVIDENCE pkixEvidence;
TP_VERIFY_EXTRA extraVerifyInfo;

////////////////////////////////////////////////////////////////////
// The field initialPolicyMappingInhibit in TP_INITIALPOLICY is not used
// by PKITP, therefore we do not set it.
////////////////////////////////////////////////////////////////////
initialPolicyPreferences.NumberofPolicyIdentifiers = NUM_POLICIES;
initialPolicyPreferences.PolicyIdentifiers = policydata;
initialPolicyPreferences.initialExplicitPolicy = INITIALExplicitPolicy;
initialPolicyPreferences.initialPolicyMappingInhibit = CSSM_FALSE;
initialPolicyPreferences.useCRLs = USECRLS;

////////////////////////////////////////////////////////////////////
// The following fields in TP_VERIFY_EXTRA are not used by PKITP.
// CLHandle, PolicyIdentifiers and NumberofPolicyIdentifiers
// (not to be confused with fields of same name in TP_INITIALPOLICY structure),
// AnchorCerts and NumberofAnchorCerts.
// Therefore we do not set these fields below.
////////////////////////////////////////////////////////////////////
extraVerifyInfo.DBList = datasources_ptr;
extraVerifyInfo.CSPHandle = ibm_csp_handle;
extraVerifyInfo.VerificationAbortOn = CSSM_TP_STOP_ON_POLICY;
extraVerifyInfo.CertToBeVerified = &certgroup;
extraVerifyInfo.InitialPolicy = &initialPolicyPreferences;
extraVerifyInfo.Evidence = &pkixEvidence;
extraVerifyInfo.ValidationTime = time(0);

(void*)CSSM_TP_PassThrough(tphandle,
                          0,
                          0,
                          0,
                          0,
                          TP_VERIFY_PASSTHROUGH,
                          (void *)&extraVerifyInfo);

reportCertGroupVerify(extraVerifyInfo);
(void*)CSSM_TP_PassThrough(tphandle,
                          0,
                          0,
                          0,
                          0,
                          TP_FREE_EVIDENCE,
                          (void *)&extraVerifyInfo);
}

//=====
// function: reportCertGroupVerify
//=====

void
reportCertGroupVerify
(TP_VERIFY_EXTRA extraVerifyInfo)
{
//-----
// report success or failure
//-----
unsigned int reported_err = CSSM_GetError()->error;

```

CSSM_TP_PassThrough

```
printf("TP_VERIFY_PASSTHROUGH : ");
if (CSSM_FALSE == extraVerifyInfo.result)
{
    printf("FAILED. Error code: %d\n",reported_err);
}
else
{
    printf("PASSED\n");
}

//-----
// report evidence
//-----
printEvidence(extraVerifyInfo.Evidence);

}

void printEvidence(TP_EVIDENCE_PTR evidence_ptr)
{
    if (evidence_ptr == NULL) return;
    if (evidence_ptr->CompleteCertGroup)
    {
        printf("CompleteCertGroup was returned containing %d certificates at address %x\n",
            evidence_ptr->CompleteCertGroup->NumCerts,
            evidence_ptr->CompleteCertGroup->CertList);
    }
    else printf("CompleteCertGroup was NULL.\n");

    if (evidence_ptr->CRL)
    {
        printf("CRL was returned of %d bytes (decimal) at address %x\n",
            evidence_ptr->CRL->Length,
            evidence_ptr->CRL->Data);
    }
    else printf("CRL was NULL.\n");

    if (evidence_ptr->Cert)
    {
        printf("Cert (failed certificate) was returned of %d bytes (decimal) at address %x\n",
            evidence_ptr->Cert->Length,
            evidence_ptr->Cert->Data);
    }
    else printf("Cert was NULL.\n");
}

/*****
 * name: freeCertGroup - Free certificate data storage
 *****/

void freeCertGroup(CSSM_CERTGROUP * certGroupPtr)
{
    CSSM_DATA * certArray = certGroupPtr->CertList;
    uint32 i;
    uint32 certCount = certGroupPtr->NumCerts;

    for (i=0; i <= certCount-1; i++)
    {
        free(certArray[i].Data);
    }
    free(certArray);
    return;
}
```

_____ End of Programming Interface information _____

Part 8. Appendixes

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 is recommended 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 72. Table of LDAP objectclasses and attributes that PKI Services sets

End Entity or branch node?	Visible RDN attribute	Objectclasses used	Additional attributes set (other than visible RDN attribute)
Creating a branch node	C=	country	none
Creating a branch node	L=	locality	none
Creating a branch node	O=	organization	none
Creating a branch node	OU=	organizationalUnit	none
Creating a branch node	Any supported value other than the preceding	organizationalUnit and extensibleObject	ou — ou value from CreateOUValue in LDAP section of pkiserv.conf file
Creating a user End Entity	Any supported value	account, pkiUser, and extensibleObject	userCertificate and uid — hardcoded to "NoUid"
Creating a CA End Entity	O=	organization and pkiCA	cACertificate
Creating a CA End Entity	OU=	organizationalUnit and pkiCA	cACertificate
Creating a CA End Entity	Any supported value other than O or OU	account, pkiCA, and extensibleObject	cACertificate and uid — hardcoded to "NoUid"
User End Entity that already exists	Any supported value	pkiUser	userCertificate
CA End Entity that already exists	Any supported value	pkiCA	cACertificate

The R_PKIServ SAF callable service supports specifying the subject's DN through named fields in the CertPlist. The CGIs invoke the R_PKIServ SAF callable service. For more information, see *z/OS Security Server RACF Callable Services*. PKI Services supports the subject's DN fields, plus some additional ones: postal code, street, and mail. They are mapped to LDAP attributes as the following table indicates:

Table 73. Relationship of named fields to LDAP attributes and object identifiers

Named field	Visible RDN attribute	OID
CommonName	CN	2.5.4.3
Title	TITLE	2.5.4.12
OrgUnit	OU	2.5.4.11
Org	O	2.5.4.10
Locality	L	2.5.4.7
StateProv	ST	2.5.4.8
Country	C	2.5.4.6
PostalCode	POSTALCODE	2.5.4.17

LDAP directory server requirements

Table 73. Relationship of named fields to LDAP attributes and object identifiers (continued)

Named field	Visible RDN attribute	OID
Street	STREET	2.5.4.9
Email	MAIL Note: 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).	0.9.2342.19200300.100.1.3

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 — Create new key pair and certificate request. Answer the prompts for file name, label, key size (1024 recommended), 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 RACDCERT 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 HFS files.

Example:

```
RACDCERT EXPORT(LABEL('SSL Cert')) ID(WEBSRV) DSN(cert.arm)  
FORMAT(CERTB64)  
OPUT cacert.der '/var/pkiserv/cacert.der' BINARY
```

8. You can optionally delete both certificate TSO data sets (but not the HFS files).

Using gskkyman

-
9. In the UNIX shell, cd to /etc and invoke /usr/lpp/gskssl/bin/gskkyman.

 10. Choose option 2 to open the key database (created earlier). Reply to the name and password prompts.

 11. Choose option 6 to store a CA certificate and specify the '/var/pkiserv/cacert.der' file.

 12. When asked to "exit gskkyman?" Enter 0 for No.

 13. Choose option 4 to receive a certificate issued for your request and specify the '/etc/cert.arm' file. Again enter 0 when asked to "exit gskkyman?"

 14. Choose option 11 to store encrypted database password.

 15. Exit gskkyman.

 16. You can optionally remove the /etc/cert.arm file.

Appendix C. 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 it 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 Volume 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.

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.

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 chapter or section or by the following marking:

Programming Interface information

End of Programming Interface information

Trademarks

The following terms are trademarks of the IBM Corporation in the United States, or other countries, or both:

| AIX
| BookManager
| DB2
| DFS
| IBM
| IBMLink
| Library Reader
| MVS
| OS/390
| RACF
| Redbooks
| Resource Link
| S/390
| SecureWay
| TalkLink

|
|
|

z/OS
z/OS.e
zSeries

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.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Tivoli is a trademark of International Business Machines Corporation or Tivoli Systems, Inc. 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, and service names may be trademarks or service marks of others.

Bibliography

The following lists titles and numbers of documents referenced in this publication.

- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS DFSMS Access Method Services for Catalogs*, SC26-7394
- | • *z/OS DFSMS Introduction*, SC26-7397
- | • *z/OS DFSMSdfp Storage Administration Reference*, SC26-7402
- *z/OS Distributed File Service DFS Administration*, SC24-5915
- *z/OS HTTP Server Planning, Installing, and Using*, SC34-4826
- *z/OS ICSF Administrator's Guide*, SA22-7521
- *z/OS ICSF Application Programmer's Guide*, SA22-7522
- *z/OS ICSF System Programmer's Guide*, SA22-7520
- | • *z/OS MVS Programming: Sysplex Services Guide*, SA22-7617
- | • *z/OS Security Server LDAP Client Programming*, SC24-5924
- *z/OS Security Server LDAP Server Administration and Use*, SC24-5923
- *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*, SA22-7612
- *z/OS MVS Programming: Sysplex Services Reference*, SA22-7618
- *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming*, SC24-5925
- *z/OS Open Cryptographic Services Facility Application Programming*, SC24-5899
- *z/OS Security Server RACF Callable Services*, SA22-7691
- *z/OS Security Server RACF Command Language Reference*, SA22-7687
- *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683
- *z/OS 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

- _PKISERV_CONFIG_PATH environment variable 306
- _PKISERV_EXIT
 - adding to environment variables file 142
 - description 307
- _PKISERV_MSG_LEVEL
 - description 305
 - message levels 229
 - subcomponents 229
- _PKISERV_MSG_LOGGING
 - STDERR_LOGGING 305
 - STDOUT_LOGGING 305
- AdditionalHeadIE 94
- renewrevokebad 94
- renewrevokeok 94
- requestbad 94
- requestok 94
- return10cert 94
- /etc/pkiserv 9
- /usr/lpp/pkiserv 9
 - description 9
 - subdirectories 257
- /var/pkiserv 9
 - description 9
 - setting up 67
- []
 - for substitution variables 91
- %
 - in named fields 92
- %%-renewrevokebad%% named field
 - pkiserv.tpl 99
- %%-renewrevokeok%% named field
 - pkiserv.tpl 99
- %%-requestok%% 104
- %%dn%% 131
- %%notafter%% 131
- %%requestor%% 131
- %%transactionid%% 131

Numerics

- 1YBSM 101
- 1YBSSL 101
- 2YBZOS 101
- 4758 coprocessor (PCICC) 199
- 5YSCA 101
- 5YSIPS 101
- 5YSSSL 101

A

- abends, recording 219
- access
 - READ, authorizing 201
 - required for administrator 310
 - required for PKI Services request 212
 - to administration pages 173

- access (*continued*)
 - changing 136
 - to end-user Web pages 157
 - to OCSF datasets, providing 50
 - to RACF group 37
 - to VSAM data sets 37
- access control, setting up 37, 309
- accessibility 369
- accessing
 - administration home page 173
 - end-user Web pages 157
- actions
 - on certificate requests 178
 - on certificates 189
- activating KEYSMSTR class profile 214
- active (status of certificate) 188
- adding
 - certificate template 127
 - members to group 200
- adtype directive 73
- admactcert.rexx 134
- admactid.rexx 134
- admactid2.rexx 134
- admicl.rexx 133
- admiclall.rexx 134
- admiclcert.rexx 133
- ADMINAPPROVE subsection (in TEMPLATE section of pkiserv.tpl) 102
- adminDN keyword 69, 72, 77
- ADMINFOOTER subsection (in APPLICATION section of pkiserv.tpl) 98
- ADMINFOOTER subsection (of APPLICATION section of pkiserv.tpl) 135
- ADMINHEADER subsection (in APPLICATION section of pkiserv.tpl) 98
- ADMINHEADER subsection (of APPLICATION section of pkiserv.tpl) 135
- administering
 - HostIdMappings extension 201
 - PKI Services 173
 - VSAM Record-Level Sharing 82
- administration
 - approving certificate requests 183
 - changing log options 229
 - deleting certificate requests 184
 - deleting certificates 190
 - displaying log options settings 230
 - log options
 - changing 229
 - displaying 230
 - modifying certificate requests 181
 - processing
 - certificate requests using searches 184
 - certificates using searches 190
 - multiple certificate requests 185
 - multiple certificates 191
 - selected certificate requests 186
 - selected certificates 193

- administration (*continued*)
 - processing (*continued*)
 - single certificate 189
 - single certificate request 180
 - RACF 199
 - ongoing administration 199
 - running IKYSETUP 37
 - rejecting certificate requests 183
 - revoking certificates 190
 - searching
 - certificate requests 184
 - certificates 190
 - selected certificate requests 186
 - selected certificates 193
 - starting PKI Services 85
 - stopping PKI Services daemon 86
- administration home page
 - accessing 173
 - using 178
- administration tasks
 - PKI Services 173
 - processing certificate requests 177
 - processing certificates 188
 - RACF
 - ongoing administration 199
 - running IKYSETUP 37
- administration Web application
 - PKI Services component description 4
- administration Web pages
 - alternate access 136
 - changing access to 136
 - customizing 133, 134
 - steps 135
 - fields 177
 - removing link to 136
 - using 173
- administrative functions
 - protecting 37, 311
 - R_PKIServ 213
- administrator
 - access required 310
- adminPW
 - slapd.conf file 69, 72
- admmain.rexx 133
- admmodtid.rexx 133
- admpend.rexx 133
- admpendall.rexx 134
- admpendtid.rexx 133
- advanced customization 139
- alias
 - file (for sendmail) 33
 - for certificate template
 - changing 117
 - description 101
 - list 101
 - for database entry 153
 - for IKYSPROC 335
 - PKISERVD 335
- ALINKLIB 257
- AltDomain (named field in pkiserv.tmpl) 94
- AltEmail (named field in pkiserv.tmpl) 94
- alternate name
 - domain name 160
 - e-mail address 161
 - IP address 161
 - Uniform resource identifier (URI) 162
- AltIPAddr (named field in pkiserv.tmpl) 94
- AltURI (named field in pkiserv.tmpl) 94
- APPL subsection (in TEMPLATE section of pkiserv.tmpl) 102
- APPLICATION section of pkiserv.tmpl
 - ADMINFOOTER subsection 98, 135
 - ADMINHEADER subsection 98, 135
 - CONTENT subsection 98, 99
 - examining 107
 - RECONTENT subsection 98, 99
 - REFAILURECONTENT subsection 98, 99
 - RESUCCESSCONTENT subsection 98, 99
 - subsections 97
- approve (action on certificate request) 178
- approve with modifications (action on certificate request) 178
- approved (status of certificate request) 177
- approving certificate requests
 - multiple 186
 - selected 186
 - single 183
- APROCLIB 257
- ASAMPLIB 257
- associating
 - user ID with PKI Services started procedure 37, 309
 - Web server and CA certificate to key ring 37
- attributes
 - HIGHTRUST 202
 - LDAP, that PKI Services requires 365
 - NOPASSWORD 309
 - OU 78, 80
 - PROTECTED 310
 - RDN 365
 - RESTRICTED 41, 310
- AuthName1 (parameter in pkiserv.conf) 77
- AuthorityKeyIdentifier
 - certificate extension 340
 - CRL extension 340
- authorization checking, using PKI exit 141
- authorizing
 - groups 201
 - PKI Services daemon user ID for CA functions 37
 - users for inquiry access 200
- AuthPwd1 (parameter in pkiserv.conf) 78
- auto-approval
 - access required 310
 - of certificates 99, 100
- automatic deletion from ObjectStore 59

B

- backing up
 - CA certificate and private key 37, 312
 - PKDS 199
- backup_dsn (variable in IKYSETUP) 46

- base64-encoded
 - #10 certificate request 96
 - certificate 92, 114, 115
 - response 6
- base64-encoded PKCS (field in end-user Web pages) 160
- base64cert substitution variable 92, 94, 114
- BasicConstraints (certificate extension) 340
- bibliography 375
- bin subdirectory 257
- bind passwords for LDAP
 - encrypted 214
 - in the clear 214
- binding
 - distinguished name for LDAP 77
 - passwords for LDAP servers 78
- BindProfile1 (parameter in pkiserv.conf) 79
- bpx_userid. (variable in IKYSETUP) 44
- brackets (in substitution variables) 91
- browser certificates
 - aliases 101
 - installing 167, 169
 - one-year PKI S/MIME browser certificate 100
 - one-year PKI SSL browser certificate 100
 - one-year SAF browser certificate 99
 - requesting 162
 - retrieving 167, 169
 - supported types 7
 - two-year PKI browser certificate for authenticating to z/OS 100
- browsertype substitution variable 92

C

- CA certificate
 - backing up 37
 - creating 311
 - using IKYSETUP 37
 - using PCICC 199
 - exporting 37
 - installing 158, 174
 - renewing 206
- CA certificate profile, recovering 207
- CA functions
 - authorizing PKI Services daemon user ID for 37
- CA signing key pair, creating 199
- ca_dn (variable in IKYSETUP) 39
- ca_expires (variable in IKYSETUP) 46
- ca_label (variable in IKYSETUP) 39
- ca_ring (variable in IKYSETUP) 46
- cadisplay.rexx 115
- cagetcert.rexx 115
- callable service, R_PKIServ (IRRSPX00) 211
- camain.rexx 114
- camodify.rexx 115
- capturing certificates 141
- careq.rexx 115
- caretrieve.rexx 115
- CAring (default name of SAF key ring) 46, 63
- catmpl.rexx 115
- CBC.SCLBDLL 45

- CDSA 3, 4
- CEE.SCEERUN 45
- CERTDETAILS 213
- CertGroupVerify 337
- certificate authority (CA)
 - certificate
 - backing up 37
 - creating 37
 - exporting 37
 - installing 158, 174
 - renewing 206
 - definition 3
 - overview 3
- certificate data set, editing 205
- certificate extensions
 - customizing 8
 - host identity mapping 7
 - in PKI Services 8
 - standard 7
 - supported by PKITP 339
- certificate policies
 - PKITP supports 339
 - using 139
- certificate profile, recovering 207
- certificate requests
 - actions on 178
 - approving 183
 - changing 181
 - deleting 184
 - modifying 181
 - processing 173
 - multiple 185
 - selected 186
 - single 180
 - using searches 184
 - rejecting 183
 - relationship with certificates 195
 - searching 184
 - states 177
 - statuses 177
 - updating 181
- certificate revocation list (CRL)
 - revoked certificates on 188
 - time interval between issuances 61
 - validity period 61
- certificate serial number incremter, restoring 209
- certificate store, using gskkyman for 367
- certificate templates
 - adding 127
 - alias 101
 - file 263
 - customization, additional first-time 117
 - customization, minimal 116, 117
 - retrofitting release changes 121
 - name 101
 - nickname 101
 - pkiserv.tmpl 99
 - subsections, summary 105
 - true name 101
- certificate validation service 337
- CertificateIssuer (CRL entry extension) 340

- CertificatePolicies extension
 - creating 139
 - in certificate 61
 - organization name for 62
 - PolicyCritical (parameter in pkiserv.conf) 61
 - supported by PKITP 340
 - using certificate policies 139
- certificates
 - actions on 189
 - auto-approval 99, 100
 - capturing 141
 - deleting 190
 - extensions 8
 - locating 203
 - processing 173, 188
 - relationship with certificate requests 195
 - renewing 169
 - requesting 162
 - retrieving
 - from bookmarked page 167
 - from home page 169
 - revoking
 - by administrator 190
 - by user 172
 - searching 190
 - single 189
 - standard extensions 7
 - states 188
 - statuses 188
 - supported types 7
 - types of 157
 - uses 7
 - X.509v3 support 7
- Certification Practice Statement
 - Uniform Resource Identifier 62
- CertPolicy section (of pkiserv.conf)
 - default values 59
 - description 57
 - excerpt 57
 - information needed 59
- CF lock structure
 - defining in SMS base configuration 82
 - defining to MVS 82
- CGIs
 - admactcert.rexx 134
 - admactid.rexx 134
 - admactid2.rexx 134
 - admicl.rexx 133
 - admiclall.rexx 134
 - admiclcert.rexx 133
 - admmain.rexx 133
 - admmodtid.rexx 133
 - admpend.rexx 133
 - admpendall.rexx 134
 - admpendtid.rexx 133
 - cadisplay.rexx 115
 - cagetcrt.rexx 115
 - camain.rexx 114
 - camodify.rexx 115
 - careq.rexx 115
 - caretrieve.rexx 115
- CGIs (*continued*)
 - catmpl.rexx 115
 - summary 133
- chains 337
- challenge passphrase (field in end-user Web pages) 160
- ChallengePassPhrase (named field in pkiserv.tmpl) 94
- changes
 - interface 23
 - summary xxi
- changing
 - access to administration pages 136
 - certificate request 181
 - configuration file
 - overview 57
 - steps 58
 - e-mail notifications 129, 132
 - environment variables
 - overview 55
 - steps 56
 - exit 142
 - expiringmsg.form 132
 - forms for e-mail notifications 129
 - LDAP section of pkiserv.conf 79
 - log options 229
 - notification forms 129
 - parameters 141
 - pkiexit.c 142
 - pkiserv.conf
 - overview 57
 - steps 58
 - pkitpsamp.c 347
 - readymsg.form 132
 - rejectmsg.form 132
 - runtime user ID 127
 - for requesting certificates 128
 - for retrieving certificates 129
 - signature algorithm 141
 - z/OS HTTP Server configuration files 71
- check boxes 186, 193
- CISIZE statements 83
- clear, LDAP bind passwords in 214
- client user ID 127
- code samples
 - certificate template file 263
 - changed in z/OS Version 1 Release 4 xxi
 - configuration directives 327, 329
 - configuration file 261
 - environment variables file 307
 - expiringmsg.form 131
 - httpd.conf 327
 - httpd2.conf 329
 - IKYCVSAM 330
 - IKYRVSAM 332
 - IKYSETUP 314
 - JCL to create VSAM data sets
 - not using RLS 330
 - using RLS 332
 - pkiserv.conf 261
 - pkiserv.envars 307
 - pkiserv.tmpl 263

- code samples (*continued*)
 - APPLICATION section 107
 - INSERT section 112
 - TEMPLATE section 109
 - PKISERVD 335
 - pkitpsamp.c 348
 - procedure to start PKI Services daemon 335
 - readymsg.form 130
 - rejectmsg.form 131
 - summary of interface changes 24
- command
 - syntax conventions xvi
- Common Data Security Architecture (CDSA) 3, 4
- common name (field in end-user Web pages) 160
- CommonName (named field in pkiserv.tmpl) 95
- completed (status of certificate request) 177
- components
 - diagram 5
 - in message numbers 239
- configurable section of IKYSETUP 37
- configuration directives
 - example 327, 329
- configuration file
 - example 261
 - for SSL traffic 71
 - pathname 306
 - updating 64
 - overview 57
 - steps 58
- configuring
 - ICSF 32
 - LDAP 30
 - tailoring for PKI Services 69
 - OCSF and OCEP 29
 - PKITP 341
 - prerequisite products 27
 - sendmail 33
 - system for PKI Services 35
 - UNIX runtime environment 53
 - z/OS HTTP Server 27
- connecting
 - members to group 200
 - members to new group 201
- CONSTANT subsection (in TEMPLATE section of pkiserv.tmpl) 102
- CONTENT subsection
 - in APPLICATION section of pkiserv.tmpl 98
 - in TEMPLATE section of pkiserv.tmpl 101
- CONTENT subsection (in APPLICATION section of pkiserv.tmpl) 99
- CONTROL access for IRR.DIGTCERT.GENCERT 310, 312
- controlling applications that invoke R_PKIServ 211
- copying
 - e-mail notifications files 55
 - expiringmsg.form 55
 - IKYCVSAM 83
 - IKYRVSAM 84
 - pkiserv.conf 54
 - pkiserv.tmpl certificate templates file 54
 - readymsg.form 55
- copying (*continued*)
 - rejectmsg.form 55
- core function 306
- CORE subcomponent 306
- country (field in end-user Web pages) 160
- Country (named field in pkiserv.tmpl) 95
- CPS in URI 62
- CPS1 (parameter in pkiserv.conf) 62, 140
- CreateInterval (parameter in pkiserv.conf) 60
- CreateOUValue (parameter in pkiserv.conf) 78
- creating
 - CA certificate 311
 - using IKYSETUP 37
 - using PCICC 199
 - CA signing key pair 199
 - CertificatePolicies extension 139
 - daemon user ID 37, 309
 - ICL data sets 83
 - implementation plan 14
 - IRR.PROXY.DEFAULTS profile 214
 - key ring 37
 - LDAPBIND class profile 214
 - PKI Services daemon user ID 37, 309
 - private key 37, 311
 - SAF key ring 37, 311
 - SSL certificate 37, 313
 - surrogate user ID 37
 - user ID
 - PKI Services daemon 37, 309
 - surrogate 37
 - VSAM data sets 81
 - not using RLS 81, 83, 330
 - space considerations 81
 - using RLS 84, 332
 - VSAM object store 83
- critical (marking of extension) 339
- critical flag 61
- CRL
 - entry extensions 340
 - extensions 340
 - revoked certificates on 188
- CRLDuration (parameter in pkiserv.conf) 61
- CRLNumber (CRL extension) 340
- CRLReason (CRL entry extension) 340
- cryptographic service provider (field in end-user Web pages) 160
- cryptology
 - standards supported 6
- CSECTs
 - IKY8B 221
 - IKYAPIMS 219
 - IKYP0N 221
 - IKYP81 221
 - IKYP8A 221, 222
 - IKYP8B 221
 - IKYSCHDR 220
 - IKYTIMER 220
- CSF.SCSFMOD0 45
- CSF.SCSFMOD1 45
- csfkeys_profile (variable in IKYSETUP) 44
- csfserv_profile (variable in IKYSETUP) 44

- csfusers_grp (variable in IKYSETUP) 44
- CSP 107
- CSSM_TP_PassThrough
 - DBList 344
 - evidence 345
 - format 343
 - functions
 - CertGroupVerify 337
 - FreeEvidence 337
 - initial policy 344
 - parameters 343
 - performing certificate validation 347
 - purpose 343
 - return codes 345
- customizing
 - administration Web pages 133, 134
 - steps 135
 - advanced 139
 - certificate extensions 8
 - certificate templates file
 - additional first-time 117
 - minimal 116
 - retrofitting release changes 121
 - e-mail notifications 129, 132
 - end-user Web pages 91
 - additional first-time 117
 - minimal 116
 - retrofitting release changes 121
 - expiringmsg.form 132
 - forms for e-mail notifications 129
 - notification forms 129
 - pkiserv.tmpl
 - additional first-time 117
 - minimal 116
 - retrofitting release changes 121
 - readymsg.form 132
 - rejectmsg.form 132

D

- daemon
 - description of PKI Services component 5
 - enabling to call OCSF functions 313
 - PKI Services component
 - description 5
 - sample procedure for starting 335
 - starting 85
 - stopping 86
 - user ID
 - creating 37, 309
 - PKISRVD 46
 - WEBSRV 47
 - variable (user ID for PKI Services) 46
 - daemon (variable in IKYSETUP) 46
 - daemon_uid (variable in IKYSETUP) 39
 - data set name
 - certificate expiring message form 63
 - certificate ready message form 63
 - certificate reject message form 63
 - data sharing environment, for VSAM RLS 82
 - DB subcomponent 306

- DB2 30
- decision tables
 - key_backup in IKYSETUP 42
 - restrict_surrog in IKYSETUP 41
 - unix_sec in IKYSETUP 43
 - use_icsf in IKYSETUP 42
- default
 - /etc/pkiserv 9
 - /usr/lpp/pkiserv 9
 - /var/pkiserv 9
 - binding information 214
 - bpx user ID 44
 - CAring (SAF key ring) 46
 - certificate template nicknames 233
 - certificate, dataset for backup copy of 46
 - configuration file for z/OS HTTP Server 72
 - daemon user ID
 - PKI Services 46
 - Web server 47
 - data set for backup copy of PKI Services certificate, private key 46
 - dataset
 - for copy of PKI Services certificate and private key 46
 - for copy of PKI Services certificate for copying to HFS 46
 - environment variables file 55
 - file locations 237
 - HFS files location 67
 - high-level qualifier 58
 - ICSF
 - profile to protect ICSF services 44
 - profile to protect PKI Services key 44
 - installation directory 9
 - key, dataset for backup copy of 46
 - message level 229, 305
 - OMVSKERN (bpx user ID) 44
 - PKI Services
 - administration group 47
 - configuration file 85
 - daemon user ID 46, 63
 - surrogate user ID 47
 - PKIGRP 47
 - PKISERV 47
 - pkiserv.conf file values 58, 59, 60, 61, 62, 63
 - PKISRVD (PKI Services daemon user ID) 46
 - primary and secondary extent allocations (in JCL) 81
 - registry directory for OCSF and OCEP 29
 - runtime directory 9
 - SAF key ring 63
 - sendmail location 55
 - sha-1WithRSA Encryption 60
 - STDOUT_LOGGING 305
 - surrogate user ID for PKI Services 47
 - time zone 86
 - variables directory 9
 - VSAM data set name
 - ICL data 59
 - ObjectStore alternate index 59
 - ObjectStore data 58

- default *(continued)*
 - Web server's daemon user ID 47
 - WEBSRV 47
 - defining
 - IRR.PROXY.DEFAULTS profile 214
 - KEYSMSTR class profile 214
 - LDAPBIND class profile 214
 - delete (action for certificate) 189
 - delete (action on certificate request) 178
 - deleting
 - certificate requests
 - multiple 186
 - selected 186
 - single 184
 - certificates
 - multiple 193
 - selected 193
 - single 190
 - groups 201
 - members 200, 201
 - diagnosing problems 219
 - Diagnostic messages, logging 306
 - diagram, PKI Services system 5
 - DIR parameter 55
 - directives
 - adddtype 73
 - example 327, 329
 - exec 73
 - FastCGI 73
 - keyfile 73
 - Log 74
 - normalmode 72, 73
 - pass 73
 - protect 72, 73
 - protection 72, 73
 - redirect 72, 73
 - sslclientauth 73
 - sslmode 72, 73
 - sslport 72, 73
 - SSLX500CARoots 73
 - SSLX500Host 73
 - SSLX500Password 73
 - SSLX500Port 73
 - SSLX500UserID 73
 - userId 72, 73
 - directories
 - /etc/pkiserv 9
 - /usr/lpp/pkiserv 9, 257
 - /var/pkiserv
 - description 9
 - setting up 67
 - for installation 9
 - for runtime 9
 - for variables 9
 - runtime 85
 - structure 257
 - directory server, LDAP
 - planning for 10
 - requirements 365
 - disability 369
 - displaying
 - information about LDAPBIND class 215
 - log options 230
 - distinguished name
 - e-mail address 160
 - for LDAP binding 77
 - LDAP administrator's 69, 72
 - qualifiers
 - areas affected 21
 - coexistence considerations 21
 - dependencies 21
 - Email 95
 - migration summary 20
 - migration tasks 21
 - PostalCode 96
 - Street 97
 - DN fields
 - mapping to LDAP attributes 365
 - documents
 - bibliography 375
 - configuring UNIX runtime environment 54
 - installing prerequisite products
 - ICSF 32
 - LDAP 30
 - OCSF and OCEP 29
 - sendmail 33
 - z/OS HTTP Server 27
 - RACF administration 38
 - UNIX programmer 54
 - documents, licensed xviii
 - domain name
 - field in end-user Web pages 160
 - fully qualified, for LDAP 69, 72, 76
- ## E
- e-mail
 - applications 3
 - secure 3
 - e-mail address
 - for alternate name (field in end-user Web pages) 161
 - for distinguished name (field in end-user Web pages) 160
 - for notifications 161
 - e-mail notifications
 - adding PATH statement 56
 - copying files for 55
 - customizing
 - overview 129
 - steps for 132
 - editing 132
 - environment variables, updating for 55
 - ExpireWarningTime
 - description 60
 - updating 65
 - ExpiringMessageForm
 - description 63
 - updating 66
 - expiringmsg.form
 - copying 55

- e-mail notifications (*continued*)
 - expiringmsg.form (*continued*)
 - description 54
 - forms for
 - expiringmsg.form 131
 - readymsg.form 130
 - rejectmsg.form 131
 - migration
 - areas affected 19
 - coexistence considerations 20
 - dependencies 20
 - documents 20
 - summary 19
 - tasks 20
 - NotifyEmail 96
 - PATH statement, adding 56
 - pkiserv.conf
 - updating, overview 57
 - updating, steps 64
 - ReadyMessageForm
 - description 63
 - updating 66
 - readymsg.form
 - copying 55
 - description 54
 - RejectMessageForm
 - description 63
 - updating 66
 - rejectmsg.form
 - copying 55
 - description 54
 - retrieving your certificate 167
 - updating
 - environment variables 55
 - ExpireWarningTime 65
 - ExpiringMessageForm 66
 - expiringmsg.form 132
 - ReadyMessageForm 66
 - readymsg.form 132
 - RejectMessageForm 66
 - rejectmsg.form 132
 - variables 131
 - %%dn%% 131
 - %%notafter%% 131
 - %%requestor%% 131
 - %%transactionid%% 131
- editing (*continued*)
 - administration Web pages 136
 - certificate data set 205
 - certificate templates file
 - administration Web pages 136
 - end-user Web pages 116
 - configuration file
 - for configuring PKI Services 58
 - to change signature algorithm 141
 - to create CertificatePolicies extension 139
 - to test configuration 85
 - e-mail notifications 132
 - end-user Web pages 116
 - expiringmsg.form 132
 - httpd.envvars 142
- editing (*continued*)
 - IKYSETUP 48
 - Log directives in httpd1443.conf 74
 - pkiserv.conf
 - for configuring PKI Services 58
 - to change signature algorithm 141
 - to create CertificatePolicies extension 139
 - to test configuration 85
 - pkiserv.tpl
 - administration Web pages 136
 - end-user Web pages 116
 - pkitpsamp.c 347
 - readymsg.form 132
 - rejectmsg.form 132
 - schema.user.ldif 70
 - Email (named field in pkiserv.tpl) 95
 - encrypted passwords for LDAP servers 79
 - BindProfile1
 - description 79
 - updating 79
 - migration
 - affected areas 22
 - coexistence considerations 22
 - dependencies 22
 - documents 22
 - summary 21
 - tasks 22
 - RACF administration 214
 - storing information for 75
 - updating LDAP section of pkiserv.conf 76
 - encryption 141
 - end-user functions
 - protecting 37, 309
 - R_PKIServ 211
 - end-user Web application
 - PKI Services component
 - description 4
 - end-user Web pages
 - accessing 157
 - code locations 125
 - customizing 91
 - additional first-time 117
 - minimal 116
 - fields 160
 - using 157
 - environment variables
 - _PKISERV_CONFIG_PATH 306
 - _PKISERV_EXIT 142, 307
 - _PKISERV_MSG_LEVEL 305
 - _PKISERV_MSG_LOGGING 305
 - changing
 - overview 55
 - steps 56
 - description 305
 - file
 - code sample 307
 - file name
 - DIR parameter 55
 - FN parameter 55
 - in PKISERVD 335
 - OCSFREGDIR 56

environment variables (*continued*)

- TZ 55
- updating
 - overview 55
 - steps 56

error messages

- changed in z/OS Version 1 Release 4 xxi
- list 240
- new in z/OS Version 1 Release 4 xxi
- summary of interface changes 25

Error messages, logging 306

EST5EDT 86

establishing

- PKI Services as an intermediate CA 205
- RLS
 - enabling VSAM data sets for 84
 - preliminary steps 82

event file 23

examples

- _PKISERV_MSG_LEVEL 305
- certificate template file 263
- configuration directives 327, 329
- configuration file 261
- environment variables file 307
- expiringmsg.form 131
- httpd.conf 327
- httpd2.conf 329
- IKYCVSAM 330
- IKYRVSAM 332
- IKYSETUP 314
- JCL
 - certificate serial number incrementer, restoring 209
 - IKYCVSAM 330
 - IKYRVSAM 332
 - PKISERVD 335
- ldapmodify 70
- log options settings 230
- LOGREC data 222
- named field 92
- output from displaying log options settings 230
- pkiserv.conf 261
- pkiserv.envars 307
- pkiserv.tmpl 263
 - APPLICATION section 107
 - INSERT section 112
 - TEMPLATE section 109
- PKISERVD 335
- pkitpsamp.c 348
- procedure to start PKI Services daemon 335
- readymsg.form 130
- rejectmsg.form 131
- substitution variable 92

excerpt

- pkiserv.conf
 - CertPolicy section 57
 - General section 57
 - LDAP section 75
 - ObjectStore section 57
 - OIDs section 57
 - SAF section 58

excerpt (*continued*)

- pkiserv.tmpl 107

exec directive 73

exit

- _PKISERV_EXIT environment variable 307
- arguments 142
- description of PKI Services component 5
- environment variable 307
- pathname 307
- PKI Services component
 - description 5
- post-processing 145, 147
 - EXPORT 149
 - GENRENEW 145
 - REQRENEW 147
 - REVOKE 151
- preprocessing 146
 - EXPORT 148
 - GENCERT 144
 - GENRENEW 144
 - REQRENEW 146
 - REVOKE 150
- scenarios 152
- updating sample code 142
- using 141

expired (status of certificate) 188

ExpireWarningTime (parameter in pkiserv.conf) 60

expiring message form for certificate 63

ExpiringMessageForm (parameter in pkiserv.conf) 63

expiringmsg.form

- code sample 131
- copying 55
- customizing 132
- in samples directory 259
- purpose 54

EXPORT

- accesses required 212
- description 153
- parameters
 - post-processing 149
 - preprocessing 148
- R_PKIServ function 310
- return codes
 - post-processing 149
 - preprocessing 148

export_dsn (variable in IKYSETUP) 46

exporting CA certificate 37

extensions

- CertificatePolicies 61
- supported by PKITP 339
- X.509 version 3 standard 7

extent allocations in IKYCVSAM 81

F

FACILITY class profile

- IRR.PROXY.DEFAULTS 214, 215
- IRR.RPKISERV.PKIADMIN 213

FAILURECONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 104

FastCGI directive 73

- fields
 - administration Web pages 177
 - end-user Web pages 160
 - in IKYSETUP REXX exec
 - change based on setup 41
 - change optionally 46
 - change required 39
 - modifiable by administrator 183
 - X.509 version 3 standard 7
- file directory structure 257
- files
 - CGIs
 - administrator Web pages 133
 - end-user Web pages 114
 - copying
 - for configuring PKI Services 54
 - exit 141
 - expiringmsg.form
 - copying 55
 - customizing 132
 - httpd.conf
 - code sample 327
 - updating 72
 - httpd1443.conf
 - source for 329
 - updating 73
 - httpd2.conf
 - code sample 329
 - copying from 73
 - IKYCVSAM
 - code sample 330
 - copying 83
 - extent allocations 81
 - updating 83
 - IKYRVSAM
 - code sample 332
 - copying 84
 - updating 84
 - IKYSETUP
 - code sample 314
 - running 37
 - Makefile.pkiexit 141
 - pkiexit.c 141
 - pkiserv.conf
 - code sample 261
 - copying 54
 - updating 57, 76, 139
 - pkiserv.envars
 - code sample 307
 - copying 55
 - updating 56
 - pkiserv.tmpl
 - code sample 263
 - contents 91
 - copying 54
 - customizing 116, 117, 121
 - PKISERVD
 - code sample 335
 - updating 56
 - PKITP 340
 - pkitp.h 257
 - files (*continued*)
 - pkitp.so 258
 - readymsg.form
 - code sample 130
 - copying 55
 - customizing 132
 - rejectmsg.form
 - code sample 131
 - copying 55
 - customizing 132
 - firewall certificate
 - description 100
 - fields 106
 - five-year PKI intermediate CA certificate
 - description 100
 - fields 106
 - five-year PKI IPSEC server (firewall) certificate
 - description 100
 - fields 106
 - five-year PKI SSL server certificate
 - description 100
 - fields 106
 - FN parameter 55
 - forms for e-mail notifications
 - copying 55
 - customizing 129, 132
 - expiringmsg.form 131
 - readymsg.form 130
 - rejectmsg.form 131
 - variables 131
 - %%dn%% 131
 - %%notafter%% 131
 - %%requestor%% 131
 - %%transactionid%% 131
 - FreeEvidence 337
 - fully qualified domain name
 - LDAP 69, 72
- G**
 - GENCERT
 - accesses required 212
 - exit scenario use 152, 153
 - parameters
 - post-processing 145
 - preprocessing 144
 - R_PKIServ function 310
 - return codes
 - post-processing 145
 - preprocessing 144
 - General section (of pkiserv.conf)
 - default values 62
 - description 57
 - excerpt 57
 - information needed 62
 - generating
 - CA signing key pair 199
 - server certificate 37
 - GENRENEW
 - accesses required 212
 - exit scenario use 153

- GENRENEW *(continued)*
 - parameters
 - post-processing 145
 - preprocessing 144
 - R_PKIServ function 310
 - return codes
 - post-processing 145
 - preprocessing 144
- GLD.SGLDLNK 45
- groups
 - authorizing 201
 - deleting 201
- GSK.SGSKLOAD 45
- gskkyman 367

H

- HFS
 - installation directory 9
 - runtime directory 9
 - subdirectories 257
- HFS-install-dir 9
- HIGHTRUST attribute 201
- HoldInstructionCode (CRL entry extension) 340
- host identity mapping 7
- HostIdMap (named field in pkiserv.tmpl) 95
- HostIdMappings extension
 - administering 201
 - field (on end-user Web pages) 161
 - PKITP support 340
- httpd.conf 327
- httpd.envvars 142
- httpd1443.conf 71
 - editing Log directives in 74
- httpd2.conf 329

I

- ICL
 - certificates maintained in 188
 - data
 - VSAM data set name for 59
 - space considerations 81
- ICL data sets and indexes
 - creating 83
- ICLDSN (parameter in pkiserv.conf) 59
- iclview
 - examples 235
 - format 235
 - parameters 235
 - purpose 235
 - summary of interface changes 25
- ICSF
 - authorizing PKI Services 37
 - configuring 32
 - description of PKI Services component 5
 - installing 32
 - PKI Services component
 - description 5
 - Public Key Data Set (PKDS) 32

- ICSF programmer
 - installing and configuring ICSF 32
 - skills 12
 - tasks 12
 - team member 11
- IDCAMS 81
- iecert substitution variable 92
- IKY8B CSECT 221
- IKYALLOC 257
- IKYAPIMS CSECT 219
- IKYCVSAM
 - copying 83
 - creating VSAM data sets 81
 - extent allocations 81
 - in SAMPLIB 257
 - sample 330
 - updating 83
- IKYDDDEF 257
- IKYISMKD 257
- IKYMKDIR 257
- IKYP0N CSECT 221
- IKYP81 CSECT 221
- IKYP8A CSECT 221, 222
- IKYP8B CSECT 221
- IKYPKID 257
- IKYPRTM 257
- IKYRVSAM
 - copying 84
 - in SAMPLIB 257
 - sample 332
 - updating 84
- IKYSCHDR CSECT 220
- IKYSETUP REXX exec
 - actions 309
 - code sample 314
 - decision tables
 - for key_backup 42
 - for restrict_surrog 41
 - for unix_sec 43
 - use_icsf 42
 - in SAMPLIB 257
 - parts 37
 - RACF administration 37
 - actions 309
 - steps for 47
 - sample log data set 50
 - structure and divisions 38
 - variables
 - backup_dsn 46
 - bpx_userid. 44
 - ca_dn 39
 - ca_expires 46
 - ca_label 39
 - ca_ring 46
 - changes based on setup 41
 - changes optional 46
 - changes required 39
 - csfkeys_profile 44
 - csfserv_profile 44
 - csfusers_grp 44
 - daemon 46

IKYSETUP REXX exec (continued)

variables (continued)

- daemon_uid 39
- export_dsn 46
- key_backup 42, 44
- log_dsn 47
- pgmcntl_dsn. 45
- pki_gid 40
- pkigroup 47
- pkigroup_mem. 40
- restrict_surrog 41, 45
- surrog 47
- surrog_uid 40
- unix_sec 43, 45
- use_icsf 42, 46
- vsamhlq 47
- web_dn 41
- web_expires 47
- web_label 47
- web_ring 41
- webserver 47

IKYSPROC 257

IKYTIMER CSECT 220

implementation plan

- creating 14
- tasks 14

IMWEBSRV started procedure 74

include subdirectory 257

INETD 86

Informational messages, logging 306

InitialThreadCount (parameter in pkiserv.conf) 63

inquiry access, authorizing users for 200

INSERT sections of pkiserv.tmpl 93, 112

INSERTs

- AdditionalHeadIE 94
- renewrevokebad 94
- renewrevokeok 94
- requestbad 94
- requestok 94
- return10cert 94
- AltDomain 94
- AltEmail 94
- AltIPAddr 94
- AltURI 94
- ChallengePassPhrase 94
- CommonName 95
- Country 95
- Email 95
- HostIdMap 95
- KeyProt 95
- KeyUsage 95
- Label 95
- Locality 95
- NotAfter 96
- NotBefore 96
- NotifyEmail 96
- Org 96
- OrgUnit 96
- OrgUnit2 96
- PassPhrase 96
- PostalCode 96

INSERTs (continued)

- PublicKey 96
- PublicKeyIE 96
- PublicKeyNS 96
- Requestor 97
- returnbrowsercertIE 94
- returnbrowsercertNS 94
- SignWith 97
- StateProv 97
- Street 97
- Title 97
- TransactionId 97
- UserId 97
- install_pkitsp 340, 341
- installation directory 9
- installing
 - CA certificate 158, 174
 - ICSF 32
 - LDAP 30
 - OCSF and OCEP 29
 - PKI Services
 - skills 12
 - SMP/E 9
 - prerequisite products
 - directions 27
 - skills 11
 - z/OS HTTP Server 27
- intermediate CA certificate
 - description 100
 - description of template 100
 - fields 106
 - template description 100
 - establishing PKI Services as 205
- Internet Explorer
 - key protection field on end-user Web page 161
 - requesting a certificate 163
 - selecting a key size 164
 - supported standard 6
 - verifying certificate installed correctly 168, 169
- Internet Protocol Security standard (IPSEC) 3
- interval
 - before certificate expiration 60
 - between certificate revocation lists 61
 - scanning database for approved requests 60
 - warning message about certificate expiration 60
- InvalidityDate (CRL entry extension) 340
- IP address
 - AltIPAddr field 94
 - field in end-user Web pages 161
 - format 94
 - LDAP fully qualified domain name 69
- IPSEC
 - certificate format 6
 - certificates 7
 - supported standard 3
- IRR.DIGTCERT.ADD 212, 311
- IRR.DIGTCERT.CERTIFAUTH.* 44
- IRR.DIGTCERT.EXPORT 212, 310, 311
- IRR.DIGTCERT.GENCERT 212, 310, 312
- IRR.DIGTCERT.GENRENEW 212, 311

- IRR.DIGTCERT.LISTRING 312
- IRR.DIGTCERT.REQCERT 212, 310
- IRR.DIGTCERT.REQRENEW 213, 310
- IRR.DIGTCERT.REVOKE 213, 310, 311
- IRR.DIGTCERT.VERIFY 213, 310, 311
- IRR.PROXY.DEFAULTS 76, 214, 215
- IRR.RPKISERV.PKIADMIN 213, 311
- IRRSPX00 SAF callable service 142
 - controlling applications that call 211
 - PKI Services component description 5
- issued certificate list (ICL) 188
- IssuerAltName
 - certificate extension 339
 - CRL extension 340
- IssuingDistributionPoint (CRL extension) 340

J

- JCL
 - creating VSAM data sets
 - not using RLS 330
 - using RLS 332
 - example
 - certificate serial number incrementer, restoring 209
 - IKYCVSAM 330
 - IKYRVSAM 332
 - PKISERVD 335
 - EXEC card, PARM= operand limitation 56
 - VSAM data sets
 - not using RLS 330
 - using RLS 332
- JOB card 83

K

- key protection (field in end-user Web pages) 161
- key ring
 - associating Web server and CA certificates with 37
 - creating 37
 - locating 203
- key size (field in end-user Web pages) 161
- key usage (field in end-user Web pages) 161
- key_backup (variable in IKYSETUP)
 - decision table 42
 - default value 44
 - description 44
- keyboard 369
- keyfile directive 73
- KeyProt (named field in pkiserv.tmpl) 95
- KeyRing (parameter in pkiserv.conf) 63
- KEYSMSTR class
 - activating 214
 - profile, defining 214
- KeyUsage (certificate extension) 339
- KeyUsage (named field in pkiserv.tmpl) 95

L

- label (field in end-user Web pages) 161
- Label (named field in pkiserv.tmpl) 95
- LDAP
 - adminDN keyword 77
 - administrator's distinguished name
 - description 69, 72
 - administrator's password
 - description 69, 72
 - attributes
 - mapped to DN fields 365
 - mapped to object identifiers 365
 - PKI Services requires 365
 - backend 30
 - bind passwords
 - encrypted 214
 - in the clear 214
 - configuring 30
 - description of PKI Services component 5
 - directory server requirements 365
 - distinguished name
 - administrator's 69, 72
 - for binding 77
 - domain name
 - description 69, 72
 - Server1 parameter 76
 - encrypted passwords
 - BindProfile1 description 79
 - BindProfile1, updating 79
 - LDAPBIND class profile 75
 - migration summary 21
 - RACF administration tasks for 214
 - storing information for 75
 - updating LDAP section of pkiserv.conf 76
 - fully qualified domain name
 - description 69, 72
 - for LDAP server 76
 - Server1 parameter 76
 - installing 30
 - IP address
 - for LDAP server 76
 - IP address and port 76
 - objectclasses PKI Services requires 365
 - OU attribute 78, 80
 - password
 - administrator's 69, 72
 - encrypted 214
 - for binding 78
 - in the clear 214
 - PKI Services component
 - description 5
 - PKI Services objectclasses and attributes requirements 365
 - port
 - description 69, 72
 - for LDAP server 76
 - profile name 79
 - retrying post requests 78
 - servers available (number of) 76
 - standard 7
 - subcomponent for message logging 306

- LDAP (*continued*)
 - suffix, description 69
 - tailoring configuration for PKI Services 69
 - TDBM DB2 backend 30
 - time interval for scanning for items to post 76
 - version 7
- LDAP programmer
 - skills 12, 13
 - tasks
 - configuration, tailoring LDAP 69
 - configuring LDAP 30
 - installing LDAP 30
 - LDAP configuration, tailoring 69
 - LDAP, installing and configuring 30
 - schema.user.ldif, updating 69
 - summary 12, 13
 - tailoring LDAP configuration 69
 - updating schema.user.ldif 69
 - team member 11
- LDAP section (of pkiserv.conf)
 - default value 76, 77, 78, 79
 - description 58
 - excerpt 75
 - information needed 76, 77, 78, 79
 - tailoring 76
- LDAPBIND class
 - displaying information about 215
 - profile
 - creating 214
 - specifying name when configuring 76
- ldapmodify 70
- ldif2tdbm 31, 69, 72
- legal statement about certificate issuance and use 62
- lib 258
- libraries
 - ALINKLIB 257
 - APROCLIB 257
 - ASAMPLIB 257
 - LINKLIB 257
 - PROCLIB 257
 - SAMPLIB 257
- licensed documents xviii
- link to administration pages, removing 136
- LINKLIB 257
- load libraries 45
- load utility (ldif2tdbm) 31, 69, 72
- loading
 - schema.user.ldif 70
 - sendmail configuration file 33
- Local PKI CA 39
- locality (field in end-user Web pages) 161
- Locality (named field in pkiserv.tmpl) 95
- locating
 - key ring 203
 - PKI Services certificate 203
- log data set
 - from running IKYSETUP 50
- Log directives in httpd1443.conf
 - editing 74
- log options
 - changing 229

- log options (*continued*)
 - displaying 230
- log_dsn (variable in IKYSETUP) 47
- logging message level 305
- LOGREC
 - description 219
 - sample data 222
- logs
 - changing options for 229
 - IKYSETUP data set sample 50
 - using information from 225
- LookAt message retrieval tool xvii

M

- MAIL distinguished name qualifier
 - migration
 - areas affected 21
 - coexistence considerations 21
 - dependencies 21
 - documents 21
 - summary 20
 - tasks 21
- Makefile.pkiexit 142
- Makefile.pkitpsamp 141, 340
- mapping
 - access control for certificates to CGI directories 128
 - DN fields to LDAP attributes 365
 - host identity 7
- MD-2 141
- MD-5 6, 141
- members
 - connecting
 - to group 200
 - to new group 201
 - deleting 200, 201
- message form
 - certificate expiring 63
 - certificate ready 63
 - certificate rejected 63
- message levels
 - _PKISERV_MSG_LEVEL 229
 - for logging 305
 - logging
 - Diagnostic 306
 - Error 306
 - Informational 306
 - Severe 306
 - Verbose Diagnostic 306
 - Warning 306
- message logging
 - CORE subcomponent 306
 - DB subcomponent 306
 - LDAP subcomponent 306
 - PKID subcomponent 306
 - POLICY subcomponent 306
 - SAF subcomponent 306
- message numbers
 - components identified 239
- message retrieval tool, LookAt xvii
- message types 239

- messages
 - changed in z/OS Version 1 Release 4 xxi
 - new in z/OS Version 1 Release 4 xxi
 - summary of interface changes 25
- Microsoft Internet Explorer
 - key protection field on end-user Web page 161
 - requesting a certificate 163
 - selecting a key size 164
 - supported standard 6
 - verifying certificate installed correctly 168, 169
- migrating
 - private key 37
- migrating from z/OS V1R3
 - e-mail notifications
 - areas affected 19
 - coexistence considerations 20
 - customizing forms 132
 - dependencies 20
 - documents 20
 - ExpireWarningTime, updating 65
 - ReadyMessageForm, updating 66
 - summary 19
 - tasks 20
 - updating ExpireWarningTime 65
 - updating ReadyMessageForm 66
 - encrypted passwords for LDAP servers
 - affected areas 22
 - coexistence considerations 22
 - dependencies 22
 - documents 22
 - LDAPBIND class profile 75
 - overview 21
 - RACF administration 214
 - storing information for 75
 - tasks 22
 - updating LDAP section of pkiserv.conf 76
 - event file, storing in VSAM
 - summary 23
 - interface changes
 - code samples 24
 - messages 25
 - samples 24
 - summary of 23
 - SYS1.SAMPLIB members 25
 - utilities 25
 - LDAP servers, encrypted passwords for
 - LDAPBIND class profile 75
 - RACF administration 214
 - storing information for 75
 - updating LDAP section of pkiserv.conf 76
 - MAIL distinguished name qualifier
 - areas affected 21
 - coexistence considerations 21
 - dependencies 21
 - documents 21
 - summary 20
 - tasks 21
 - overview 17
 - POSTALCODE distinguished name qualifier
 - areas affected 21
 - coexistence considerations 21
 - migrating from z/OS V1R3 *(continued)*
 - POSTALCODE distinguished name qualifier *(continued)*
 - dependencies 21
 - documents 21
 - summary 20
 - tasks 21
 - release summary 17
 - serial number file, storing in VSAM
 - summary 23
 - storing serial number and event files
 - areas affected 23
 - coexistence considerations 23
 - dependencies 23
 - documents 23
 - summary 23
 - tasks 23
 - STREET distinguished name qualifier
 - areas affected 21
 - coexistence considerations 21
 - dependencies 21
 - documents 21
 - summary 20
 - tasks 21
 - sysplex support
 - areas affected 17
 - coexistence considerations 18
 - dependencies 18
 - documents 19
 - RLS, enabling VSAM data sets for 84
 - RLS, setting up, preliminary steps 82
 - SharedVSAM, updating 65
 - summary 17
 - tasks 18
 - updating SharedVSAM 65
 - user notifications
 - customizing forms 132
 - summary 19
 - MODIFY command
 - change log options 229
 - display logging options 230
 - stop PKI Services daemon 86
 - MODIFYCERTS 213
 - modifying
 - certificate request 181
 - MODIFYREQS 213
 - MVS programmer
 - installation of PKI Services 9
 - skills 13
 - tasks
 - creating VSAM data sets 81
 - enabling VSAM data sets for RLS 84
 - establishing RLS, preliminary steps 82
 - RLS, enabling VSAM data sets for 84
 - RLS, preliminary steps for establishing 82
 - starting PKI Services daemon 85
 - stopping PKI Services daemon 86
 - VSAM data sets, creating 81
 - team member 11
 - MyPolicy (parameter in pkiserv.conf) 58, 140

N

name

- certificate templates
 - alias names 101
 - nicknames 101
 - short names 101
 - table summarizing 101
 - true names 101
- field (in end-user Web pages) 162

named fields (in pkiserv.tpl) 92

Netscape

- key size field on end-user Web page 161
- requesting a certificate 163
- selecting a key size 164
- supported standard 6
- verifying certificate installed correctly 168, 169

nickname

- certificate template 101, 233

NOPASSWORD attribute 309

normal operating mode of z/OS HTTP Server 312

normalmode directive 72, 73

not after date (field in end-user Web pages) 161

not before date (field in end-user Web pages) 161

NotAfter (named field in pkiserv.tpl) 96

NotBefore (named field in pkiserv.tpl) 96

notice

- legal 62
- number 62

Notices 371

notification e-mail address (field in end-user Web pages) 161

notification forms

- copying 55
- customizing 129

notifications

- customizing 132
- retrieving your certificate 167
- summary of migration support 19

NotifyEmail

- adding 122
- deleting 117
- migration summary information 19
- must match MAIL 171

NotifyEmail (named field in pkiserv.tpl)

- description 96

notifying users

- copying files for 55
- customizing forms 132
- forms for
 - expiringmsg.form 131
 - readymsg.form 130
 - rejectmsg.form 131
- migration summary 19

NumServers (parameter in pkiserv.conf) 76

O

Object ID

- for policy 62
- signing algorithm 60

object identifiers

- mapping to LDAP attributes 365

object store

- space considerations 82

objectclasses

- LDAP, that PKI Services requires 365

ObjectDSN (parameter in pkiserv.conf) 58

ObjectStore

- alternate index
 - VSAM data set name for 59
- DB subcomponent for message logging 306
- enabled for sysplex 59
- section of pkiserv.conf
 - default value 58
 - description 57
 - excerpt 57
 - information needed 58
- sysplex enabled 59
- time period before automatic deletion
 - completed requests 59
 - inactive requests 59
 - incomplete requests 59
 - unsuccessful requests 59

ObjectTidDSN (parameter in pkiserv.conf) 59

OCEP programmer

- installing OCSF and OCEP 29
- team member 11

OCSF

- datasets, providing access to 50
- functions, enabling PKI Services daemon to call 313
- programmer 11
- Trust Policy
 - module 341
 - overview 337
 - plug-in 337

OCSF and OCEP

- configuring 29
- installing 29
- programmer
 - installing and configuring OCSF and OCEP 29
 - skills 12

OCSF programmer

- installing OCSF 29

OCSFREGDIR environment variable 56

OIDs section (of pkiserv.conf)

- default value 58
- description 57
- excerpt 57
- information needed 58

OMVSKERN 44

one-year PKI S/MIME browser certificate

- description 100
- fields 106

one-year PKI SSL browser certificate

- description 100
- fields 106

one-year SAF browser certificate

- description 99
- fields 106

- one-year SAF server certificate
 - description 99
 - fields 106
- optfield substitution variable 92
- Org (named field in pkiserv.tpl) 96
- organization (field in end-user Web pages) 161
- organization name
 - for CertificatePolicies extension 62
- organizational unit (field in end-user Web pages) 161
- organizationalUnit objectclass 78
- OrgUnit (named field in pkiserv.tpl) 96
- OrgUnit2 (named field in pkiserv.tpl) 96
- OU attribute 78, 80

P

- parallel sysplex support
 - prerequisites 9
 - requirements 9
- parameters
 - changing 141
 - validating 141
- pass directive 73
- pass phrase (field in end-user Web pages) 161
- PassPhrase (named field in pkiserv.tpl) 96
- passwords
 - binding 214
 - encrypted, for LDAP servers
 - LDAPBIND class profile 75
 - migration summary 21
 - RACF administration for 214
 - for LDAP binding 78
 - for LDAP servers, encrypted
 - storing information for 75
 - updating LDAP section of pkiserv.conf 76
 - LDAP
 - encrypted 214
 - in the clear 214
 - LDAP administrator's 69, 72
 - RACF administration for 214
- PATH statement 56
- pathname
 - certificate expiring message form 63
 - certificate ready message form 63
 - certificate reject message form 63
 - configuration file 306
 - exit program 307
- PCICC 199
 - keys
 - CA certificate, creating 199
 - creating CA certificate 199
- PDS 257
- pending approval (status of certificate request) 177
- pgmcntl_dsn. (variable in IKYSETUP) 45
- PKCS #10 certificate request 160
- PKCS#10 browser certificate format 6
- PKCS#10 server certificate format 6
- PKDS 32
- PKI
 - definition 4
 - PKI browser certificate for authenticating to z/OS
 - description 100
 - fields 106
 - PKI exit
 - arguments 142
 - PKI Services component
 - description 5
 - post-processing
 - EXPORT 149
 - GENCERT 145
 - GENRENEW 145
 - REQCERT 147
 - REQRENEW 147
 - REVOKE 151
 - preprocessing
 - EXPORT 148
 - GENCERT 144
 - GENRENEW 144
 - REQCERT 146
 - REQRENEW 146
 - REVOKE 150
 - scenarios 152
 - using 141
 - PKI intermediate CA certificate
 - description 100
 - fields 106
 - PKI IPSEC server (firewall) certificate
 - description 100
 - fields 106
 - PKI S/MIME browser certificate
 - description 100
 - fields 106
 - PKI Services
 - administering RACF 199
 - administration
 - changing log options 229
 - displaying log options settings 230
 - log options, changing 229
 - log options, displaying 230
 - starting PKI Services 85
 - stopping PKI Services daemon 86
 - using Web pages 173
 - administration group PKIGRP 47
 - administration Web application
 - component 4
 - authorizing for ICSF 37
 - CA 3
 - certificate
 - locating 203
 - certificate authority 3
 - certificate authority certificate, renewing 206
 - certificate types 7
 - changing
 - environment variables 56
 - changing log options 229
 - component diagram 5
 - components
 - administration Web application 4
 - diagram 5
 - end-user Web application 4
 - exit 5

- PKI Services *(continued)*
 - components *(continued)*
 - ICSF 5
 - IRRSPX00 5
 - LDAP 5
 - list 4
 - PKI Services daemon 5
 - R_PKIServ callable service 5
 - RACF 5
 - z/OS HTTP Server 5
 - configuration file
 - overview 57
 - updating 58, 64
 - configuration, testing 85
 - cryptographic standards 6
 - customizing
 - administration Web pages 133
 - advanced 139
 - end-user Web pages 91
 - daemon
 - component 5
 - daemon user ID
 - authorizing for CA functions 37
 - creating 37
 - PKISRVD 46
 - directory structure 257
 - end-user Web application 4
 - environment variables
 - updating 56
 - exit 5
 - extensions supported 7
 - fields supported 7
 - file directory structure 257
 - ICSF
 - component 5
 - installing 32
 - implementation plan 14
 - installing
 - skills 12
 - SMP/E 9
 - intermediate certificate authority 204
 - introduction 3
 - IRRSPX00 5
 - key ring
 - locating 203
 - LDAP
 - attributes requirements 365
 - component 5
 - objectclasses requirements 365
 - tailoring configuration for PKI Services 69
 - tailoring pkiserv.conf 75
 - log options, changing 229
 - logs 225
 - OCSF Trust Policy plug-in 337
 - overview 3
 - PKI exit
 - component 5
 - using 141
 - planning 9
 - prerequisite products
 - ICSF 10

- PKI Services *(continued)*
 - prerequisite products *(continued)*
 - installing and configuring 27
 - LDAP server 10
 - OCSF and OCEP 10
 - planning for 10
 - z/OS HTTP Server 10
 - protecting administrative and end-user functions 37, 309
 - R_PKIServ callable service (IRRSPX00)
 - component 5
 - RACF
 - administration 199
 - component 5
 - using IKYSETUP 37
 - related products 4
 - renewing certificate authority certificate 206
 - SAF key ring 63
 - skill requirements 11
 - standards 6
 - starting 85
 - stopping 85, 86
 - subordinate certificate authority 204
 - surrogate user ID PKISERV 47
 - task roadmap 14
 - team members 11
 - testing configuration 85
 - updating
 - certificate templates file 116, 117, 122
 - configuration file 64
 - environment variables 56
 - uses 3
 - using
 - administration Web pages 173
 - end-user Web pages 157
 - utilities 231
 - iclview 235
 - vosview 232
 - Web pages
 - customizing 91, 133
 - using 157, 173
 - z/OS HTTP Server
 - component 5
 - installing 27
 - updating configuration 71
 - z/OS product libraries 257
- PKI Services administration
 - approving certificate requests 183
 - deleting certificate requests 184
 - deleting certificates 190
 - group, setting up 37
 - modifying certificate request 181
 - processing certificate requests
 - multiple 184
 - selected 186
 - single 180
 - using searches 184
 - processing certificates 188
 - multiple 191
 - selected 193
 - single 189

- PKI Services administration (*continued*)
 - rejecting certificate requests 183
 - revoking certificates 190
 - searching
 - certificate requests 184
 - certificates 190
 - selected certificate requests 186
 - selected certificates 193
 - setting up group 37
- PKI Services daemon
 - enabling OCSF functions 313
 - starting 85, 335
 - user ID 63
- PKI Services daemon user ID
 - creating 309
- PKI Services OCSF Trust Policy
 - API
 - CSSM_TP_PassThrough 341
 - overview 337
- PKI Services started procedure
 - associating user ID with 37
- PKI SSL browser certificate
 - description 100
 - fields 106
- PKI SSL server certificate
 - description 100
 - fields 106
- pk_i_gid (variable in IKYSETUP) 40
- PKID
 - subcomponent for message logging 306
- pkixit.c
 - description 141
 - scenarios 152
 - updating sample code 142
- pkigroup (variable in IKYSETUP) 47
- pkigroup_mem. (variable in IKYSETUP) 40
- PKIGRP 47
- PKISERV
 - application name 97, 107
 - runtime user ID 127
 - surrogate user ID 47, 310
 - z/OS HTTP Server operating modes required 312
- PKISERV certificate generation application Web
 - page 99
- PKIServ subdirectory 258
- pkiserv.conf
 - CertPolicy section
 - default values 59
 - description 57
 - excerpt 57
 - information needed 59
 - changing
 - signature algorithm 141
 - code sample 261
 - copying 54
 - editing
 - for configuring PKI Services 58
 - to change signature algorithm 141
 - to create CertificatePolicies extension 139
 - to test configuration 85

- pkiserv.conf (*continued*)
 - General section
 - default values 62
 - description 57
 - excerpt 57
 - information needed 62
 - LDAP section
 - default value 76, 77, 78, 79
 - description 58
 - excerpt 75
 - information needed 76, 77, 78, 79
 - ObjectStore section
 - default value 58
 - description 57
 - excerpt 57
 - information needed 58
 - OIDs section
 - default value 58
 - description 57
 - excerpt 57
 - information needed 58
 - parameters
 - AuthName1 77
 - AuthPwd1 78
 - BindProfile1 79
 - CPS1 62, 140
 - CreateInterval 60
 - CreateOUValue 78
 - CRLDuration 61
 - ExpireWarningTime 60
 - ExpiringMessageForm 63
 - ICLDSN 59
 - InitialThreadCount 63
 - KeyRing 63
 - MyPolicy 58, 140
 - NumServers 76
 - ObjectDSN 58
 - ObjectTidDSN 59
 - Policy1Notice1 62, 140
 - Policy1Notice2 62
 - Policy1Org 62, 140
 - PolicyCritical 61, 139
 - PolicyName1 62, 140
 - PolicyRequired 61, 139
 - PostInterval 76
 - ReadyMessageForm 63
 - RejectMessageForm 63
 - RemoveCompletedReqs 59
 - RemoveInactiveReqs 59
 - RetryMissingSuffix 78
 - Server1 76
 - SharedVSAM 59
 - SigAlg1 60, 141
 - TimeBetweenCRLs 61
 - UserNoticeText1 62, 140
 - passwords for LDAP servers, storing in clear
 - text 21
 - purpose 53
 - SAF section
 - default value 63
 - description 58

- pkiserv.conf *(continued)*
 - SAF section *(continued)*
 - excerpt 58
 - information needed 63
 - signature algorithm
 - changing 141
 - steps for updating LDAP section 79
 - storing passwords for LDAP servers in clear text 21
 - updating 64
 - overview 57
 - steps 58
- pkiserv.envars
 - code sample 307
 - purpose 53
 - updating 55
- pkiserv.tmpl
 - APPLICATION section 107
 - ADMINFOOTER subsection 98, 135
 - ADMINHEADER subsection 98, 135
 - CONTENT subsection 98, 99
 - RECONTENT subsection 98, 99
 - REFAILURECONTENT subsection 98, 99
 - RESUCCESSCONTENT subsection 98, 99
 - subsections 97
 - code sample 263
 - copying 54
 - customizing
 - customization, additional first-time 117
 - minimally 116
 - retrofitting release changes 121
 - description 91
 - editing
 - administration Web pages 136
 - end-user Web pages 116
 - INSERT sections 93, 112
 - INSERTs
 - AdditionalHeadIE 94
 - renewrevokebad 94
 - renewrevokeok 94
 - requestbad 94
 - requestok 94
 - return10cert 94
 - AltDomain 94
 - AltEmail 94
 - AltIPAddr 94
 - AltURI 94
 - ChallengePassPhrase 94
 - CommonName 95
 - Country 95
 - Email 95
 - HostIdMap 95
 - KeyProt 95
 - KeyUsage 95
 - Label 95
 - Locality 95
 - NotAfter 96
 - NotBefore 96
 - NotifyEmail 96
 - Org 96
 - OrgUnit 96
 - OrgUnit2 96
- pkiserv.tmpl *(continued)*
 - INSERTs *(continued)*
 - PassPhrase 96
 - PostalCode 96
 - PublicKey 96
 - PublicKeyIE 96
 - PublicKeyNS 96
 - Requestor 97
 - returnbrowsercertIE 94
 - returnbrowsercertNS 94
 - SignWith 97
 - StateProv 97
 - Street 97
 - Title 97
 - TransactionId 97
 - UserId 97
 - named fields 92
 - %%-renewrevokebad%% 99
 - %%-renewrevokeok%% 99
 - %%-requestok%% 104
 - purpose 53
 - sections 91
 - substitution variables 91
 - TEMPLATE section 109
 - ADMINAPPROVE subsection 102
 - APPL subsection 102
 - CONSTANT subsection 102
 - CONTENT subsection 101
 - FAILURECONTENT subsection 104
 - RETRIEVECONTENT subsection 104
 - RETURNCERT subsection 105
 - subsections 101
 - SUCCESSCONTENT subsection 104
 - updating
 - customization, additional first-time 117
 - minimally 116
 - retrofitting release changes 121
- PKISERVD
 - code sample 335
 - in PROCLIB 257
 - updating environment variables 55
- PKISRVD
 - PKI Services daemon user ID 46, 63
- PKITP
 - API
 - CSSM_TP_PassThrough 341
 - certificate extensions supported 339
 - certificate policies supported 339
 - configuring 341
 - files 340
 - overview 337
 - PKI Services Trust Policy plug-in for OCSF 337
 - pkitp_ivp 340, 341
 - pkitp.h 340
 - pkitp.so 340
 - pkitpsamp.c
 - description and directory 340
 - editing 347
 - sample code 348
- PKIX
 - compliant certificates 211

- PKIX *(continued)*
 - support for interoperability 4
 - supported by PKI Services 3
- planning
 - for PKI Services 9
- policy
 - notice number 62
 - Object ID for 62
 - usage 58
- POLICY
 - subcomponent for message logging 306
- Policy1Notice1 (parameter in pkiserv.conf) 62, 140
- Policy1Notice2 (parameter in pkiserv.conf) 62
- Policy1Org (parameter in pkiserv.conf) 62, 140
- PolicyCritical (parameter in pkiserv.conf) 61, 139, 339
- PolicyName1 (parameter in pkiserv.conf) 62, 140
- PolicyRequired (parameter in pkiserv.conf) 61, 139, 339
- ports
 - 1443 71
 - 443 71
 - 80 71
 - for HTTP traffic 71
 - for SSL traffic 71
 - LDAP 69, 72
- post requests
 - retrying for LDAP 78
- post-processing
 - exit 142
 - EXPORT 149
 - GENCERT 145
 - GENRENEW 145
 - REQCERT 147
 - REVOKE 151
- postal code
 - DN field supported 365
 - field in end-user Web pages 162
- PostalCode (named field in pkiserv.tmpl) 96
- POSTALCODE distinguished name qualifier migration
 - areas affected 21
 - coexistence considerations 21
 - dependencies 21
 - documents 21
 - summary 20
 - tasks 21
- PostInterval (parameter in pkiserv.conf) 76
- preprocessing
 - exit 142
 - EXPORT 148
 - GENCERT 144
 - GENRENEW 144
 - REQCERT 146
 - REVOKE 150
- prerequisite products
 - configuring 27
 - installing 27
 - skills 11
 - planning for 10
- prerequisites
 - for sysplex support 9
- prerequisites *(continued)*
 - products 27
- printablecert substitution variable 92
- private key
 - backing up 37
 - creating 37, 311
 - migrating to ICSF 37
 - storing
 - in ICSF 32
 - in RACF 5
- problems, diagnosing 219
- processing
 - certificate requests
 - actions 178
 - introduction 173
 - multiple 185
 - selected 186
 - single 180
 - using searches 184
 - certificates
 - actions 189
 - introduction 173
 - multiple 191
 - overview 188
 - selected 193
 - single 189
 - using searches 190
- PROCLIB 257
- product libraries 257
- profile
 - CA certificate, recovering 207
 - FACILITY class
 - IRR.PROXY.DEFAULTS 214, 215
 - IRR.DIGTCERT.ADD 212
 - IRR.DIGTCERT.EXPORT 212
 - IRR.DIGTCERT.GENCERT 212
 - IRR.DIGTCERT.GENRENEW 212
 - IRR.DIGTCERT.REQCERT 212
 - IRR.DIGTCERT.REQRENEW 213
 - IRR.DIGTCERT.REVOKE 213
 - IRR.DIGTCERT.VERIFY 213
 - IRR.RPKISERV.PKIADMIN 213
 - KEYSMSTR class
 - LDAP.BINDPW.KEY 214
 - LDAPBIND class profile
 - defining 214
- protect directive 72, 73
- PROTECTED attribute 310
- protection directive 72, 73
- protocols
 - supported in PKI Services 6
- province 162
- public key cryptography
 - standards supported 6
- Public Key Data Set (PKDS) 32
- Public Key Infrastructure for X.509 version 3 3
- public-private key pair, creating 199
- publications
 - on CD-ROM xvii
 - softcopy xvii
- PublicKey (named field in pkiserv.tmpl) 96

PublicKeyIE (named field in pkiserv.tmpl) 96
PublicKeyNS (named field in pkiserv.tmpl) 96

Q

qualifiers for distinguished name, migration
summary 20
QUERYCERTS 213
QUERYREQS 213

R

R_PKIServ callable service
administrative functions 213
controlling applications that invoke 211
description of PKI Services component 5
end-user functions 211
PKI Services component
description 5
protected by FACILITY class resources 310
RACF
administering PKI Services 199
authorizing
READ access 201
users for inquiry access 200
connecting members
to group 200
to new group 201
deleting groups 201
deleting members 200, 201
description of PKI Services component 5
PKI Services component
description 5
publications
on CD-ROM xvii
softcopy xvii
setting up PKI Services 37
RACF administration
for PKI Services, ongoing 199
for setting up PKI Services using IKYSETUP 37
steps for 47
using IKYSETUP 309
RACF administrator
ongoing administration for PKI Services 199
running IKYSETUP
overview 37
steps 49
skills 13
tasks 13
IKYSETUP, running 37
ongoing administration for PKI Services 199
performed by IKYSETUP 309
running IKYSETUP 37
setting up PKI Services using IKYSETUP 47
team member 11
RACF group
providing access 37
RDB (request database) 195
RDN attribute 365
READ access
authorizing 201

READ access (*continued*)
IRR.DIGTCERT.EXPORT 310, 311
IRR.DIGTCERT.GENCERT 312
IRR.DIGTCERT.GENRENEW 311
IRR.DIGTCERT.LISTRING 312
IRR.DIGTCERT.REQCERT 310
IRR.DIGTCERT.REQRENEW 310
IRR.DIGTCERT.REVOKE 310, 311
IRR.DIGTCERT.VERIFY 310, 311
IRR.RPKISERV.PKIADMIN 311
ready message form for certificate 63
ReadyMessageForm (parameter in pkiserv.conf) 63
readymsg.form
code sample 130
copying 55
customizing 132
in samples directory 259
purpose 54
recent activity (field in administration Web pages) 177
RECONTENT subsection (in APPLICATION section of
pkiserv.tmpl) 98, 99
Record-Level Sharing (RLS) 82
recording
errors 219
recovering
CA certificate profile 207
redirect directive 72, 73
REFAILURECONTENT subsection (in APPLICATION
section of pkiserv.tmpl) 98, 99
registry directory for OCSF and OCEP 29
reject (action on certificate request) 178
reject message form for certificate 63
rejected (status of certificate request) 177
rejected, user notified (status of certificate
request) 177
rejecting
certificate requests
multiple 186
selected 186
single 183
RejectMessageForm (parameter in pkiserv.conf) 63
rejectmsg.form
code sample 131
copying 55
customizing 132
in samples directory 259
purpose 54
relationship between certificate requests and
certificates 195
RemoveCompletedReqs (parameter in pkiserv.conf) 59
RemoveInactiveReqs (parameter in pkiserv.conf) 59
removing
groups 201
members 201
renew (action for certificate) 189
Renew or revoke a browser certificate Web page 99
renewing
certificate
steps for 169
PKI Services certificate authority certificate 206

REQCERT
 accesses required 212
 exit scenario use 152, 153
 parameters
 post-processing 147
 preprocessing 146
 R_PKIServ function 310
 return codes
 post-processing 147
 preprocessing 146
REQDETAILS 213
REQRENEW
 accesses required 213
 exit scenario use 153
 parameters
 post-processing 147
 preprocessing 146
 R_PKIServ function 310
 return codes
 post-processing 147
 preprocessing 146
 request database (RDB) 195
 requesting
 certificate
 steps for 162
 Requestor (named field in pkiserv.tmpl) 97
 requestor name (field in administration Web pages) 177
 requirements
 access (for PKI Services request) 212
 LDAP directory server 365
 prerequisite products 10
 skills 11
 sysplex support 9
 restoring
 certificate serial number incremter 209
 restrict_surrog (variable in IKYSETUP)
 decision table 41
 default value 45
 description 45
RESTRICTED attribute 310
RESUCCESSCONTENT subsection (in APPLICATION section of pkiserv.tmpl) 98, 99
RETRIEVECONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 104
 retrieving
 certificate
 steps for 167, 169
 retrofitting release changes into pkiserv.tmpl 121
 retrying
 LDAP post requests 78
 RetryMissingSuffix (parameter in pkiserv.conf) 78
 return codes
 CSSM_TP_PassThrough 345
 EXPORT
 post-processing 149
 preprocessing 148
 GENCERT
 post-processing 145
 preprocessing 144
 return codes (*continued*)
 GENRENEW
 post-processing 145
 preprocessing 144
 recording 219
 REQCERT
 post-processing 147
 preprocessing 146
 REQRENEW
 post-processing 147
 preprocessing 146
 REVOKE
 post-processing 151
 preprocessing 150
 returnbrowsercertIE 94
 returnbrowsercertNS 94
 RETURNCERT subsection (in TEMPLATE section of pkiserv.tmpl) 105
 REVOKE
 accesses required 213
 parameters
 post-processing 151
 preprocessing 150
 R_PKIServ function 310
 return codes
 post-processing 151
 preprocessing 150
 revoke (action for certificate) 189
 revoked (status of certificate) 188
 revoked expired (status of certificate) 189
 revoking certificates
 by administrator
 multiple 193
 selected 193
 single 190
 by user 172
 RFC2587.ldif 70
RLS
 enabling VSAM data sets for 84
 MVS programmer task 13
 preliminary steps for establishing 82
 setting up, preliminary steps 82
 roadmap for implementing PKI Services 14
 roles 11
RSA
 signature algorithm
 SigAlg1 parameter 60
 updating 141
 standard supported 6
 runtime directory 85
 runtime environment
 configuring 53
 runtime user ID
 changing 127
 for requesting certificates 128
 for retrieving certificates 129
 runtime-dir 9

S

S/MIME

- certificate format 6
- description of certificate 100
- fields of certificate 106
- supported standard 3
- use of certificate 7

SAF

- browser certificate
 - description 99
 - fields 106
- key ring
 - creating 37, 311
 - KeyRing parameter 63
- section (of pkiserv.conf)
 - default value 63
 - description 58
 - excerpt 58
 - parameter description 63
- server certificate
 - description 99
 - fields 106
- subcomponent for message logging 306

samples

- _PKISERV_MSG_LEVEL 305
- certificate template file 263
- changed in z/OS Version 1 Release 4 xxi
- configuration directives 327, 329
- configuration file 261
- directives 327, 329
- environment variables file 307
- expiringmsg.form 131
- httpd.conf 327
- httpd2.conf 329
- IKYCVSAM 330
- IKYRVSAM 332
- IKYSETUP 314
- JCL

- certificate serial number incrementer,
 - restoring 209
- IKYCVSAM 330
- IKYRVSAM 332
- PKISERVD 335

- log data set from IKYSETUP 50

- LOGREC data 222

- pkiserv.conf 261

- pkiserv.envars 307

- pkiserv.tmpl 263

- APPLICATION section 107

- INSERT section 112

- TEMPLATE section 109

- PKISERVD sample proc 335

- pkitpsamp.c 348

- readymsg.form 130

- rejectmsg.form 131

- summary of interface changes 24

- samples subdirectory 259

- SAMPLIB 257

scenarios

- PKI exit 152

scenarios (continued)

- allowing only selected users to request certificates 152
- maintaining customized certificate repository 153
- providing customized TITLE 152
- renewal only within 30 days of expiration 153

schema.user.Idif

- editing 70
- loading 70

searching

- certificate requests 184
- certificates 190

secure

- e-mail 3

Secure Multipurpose Internet Mail Extensions (S/MIME) 3

Secure Sockets Layer (SSL) 3, 6

selected certificate requests

- processing 186

selected certificates

- processing 193

sendmail

- configuring 33
- not using default 55
- planning for 11

serial number

- field in administration Web pages 177
- incrementer, restoring 209

serial number file 23

SERVAUTH class 202

server certificate

- generating 37

server certificates

- aliases 101
- five-year PKI intermediate CA certificate 100
- five-year PKI IPSEC server (firewall) certificate 100
- five-year PKI SSL server certificate 100
- installing 167, 169
- one-year SAF server certificate 99
- retrieving 167, 169
- supported types 7

Server1 (parameter in pkiserv.conf) 76

setting up

- /var/pkiserv 67
- access control 37, 309
- PKI Services 35
- PKI Services administration group 37
- prerequisite products 27
- RLS
 - enabling VSAM data sets for 84
 - preliminary steps 82
- z/OS HTTP Server for surrogate operation 37, 313

settings

- contained in pkiserv.conf 53
- displaying log options 230
- IKYP025I displays 255
- log options, displaying 230
- permission, changing with chmod 67

Severe messages, logging 306

SHA-1 6, 141

SharedVSAM (parameter in pkiserv.conf) 59

- sharing control data sets (SHCDS) 82
- SHCDS 82
- shortcut keys 369
- SigAlg1 (parameter in pkiserv.conf) 60, 141
- signature algorithm
 - Object ID 60
 - updating 141
- signing key 32
- signing public-private key pair, creating 199
- SignWith (named field in pkiserv.tmpl) 97
- single certificate, processing 189
- single request, processing 180
- skill requirements 11
- skills
 - ICSF programmer 12
 - installing PKI Services 12
 - installing prerequisite products 11
 - LDAP programmer 12, 13
 - MVS programmer 13
 - OCSF and OCEP programmer 12
 - RACF administrator 13
 - UNIX programmer 12, 13
 - Web server programmer 12, 13
- slapd.conf file
 - adminPW 69, 72
- smart cards 3
- SMP/E 9
- SMS base configuration 82
- space considerations
 - for ICL 81
 - for object store 82
 - for VSAM data sets 81
- SPANNED statements 83
- square brackets (in substitution variables) 91
- SSL
 - certificate
 - creating 37, 313
 - use 7
 - delivering certificates through 3
 - enabled 85
 - supported standards 6
 - two modes 71
 - with client authentication operating mode of z/OS
 - HTTP Server 312
 - without client authentication operating mode of z/OS
 - HTTP Server 312
- sslclientauth directive 73
- sslmode directive 72, 73
- sslport directive 72, 73
- SSLring 41
- SSLX500CARoots directive 73
- SSLX500Host directive 73
- SSLX500Password directive 73
- SSLX500Port directive 73
- SSLX500UserID directive 73
- standards
 - certificate extensions supported 7
 - LDAP 7
 - public key cryptography, supported 6
- starting
 - PKI Services 85
 - starting (*continued*)
 - PKI Services daemon 85
 - z/OS HTTP Server 74
- state (field in end-user Web pages) 162
- StateProv (named field in pkiserv.tmpl) 97
- statuses
 - certificate requests 177
 - certificates 188
- STDERR_LOGGING 305
- STDOUT 142
 - EXPORT
 - preprocessing 148
 - GENCERT
 - post-processing 145
 - preprocessing 144
 - GENRENEW
 - post-processing 145
 - preprocessing 144
 - REQCERT
 - post-processing 147
 - preprocessing 146
 - REQRENEW
 - post-processing 147
 - preprocessing 146
- STDOUT_LOGGING 305
- steps
 - /var/pkiserv, setting up 67
 - access to administration pages, changing 136
 - accessing
 - administration home page 173
 - end-user Web pages 157
 - adding new certificate template 127
 - administering HostIdMappings extensions 201
 - administration Web pages
 - changing access to 136
 - customizing 135
 - alternative access to administration pages,
 - providing 136
 - approving single request 180
 - authorizing users for inquiry access 200
 - bind passwords encrypted for LDAP
 - IRR.PROXY.DEFAULTS profile 215
 - LDAPBIND class 214
 - building sample application 347
 - CA certificate
 - creating, using PCICC 199
 - renewing 206
 - CA certificate profile, recovering 208
 - certificate templates file
 - customization, additional first-time 117
 - customization, minimal 116
 - retrofitting release changes 121
 - certificate, locating 203
 - changing
 - administration Web pages 135
 - end-user Web pages 116, 117, 121
 - environment variables 56
 - fields in requests 180, 183
 - pkiserv.conf configuration file 58
 - runtime user ID for requesting certificates 128
 - runtime user ID for retrieving certificates 129

- steps (*continued*)
 - changing (*continued*)
 - signature algorithm 141
 - configuration file, updating 58, 64
 - configuring
 - ICSF 32
 - LDAP 30
 - OCSF and OCEP 29
 - PKITP 341
 - z/OS HTTP Server 27
 - copying files 54
 - expiringmsg.form 55
 - pkiserv.conf 54
 - pkiserv.envvars 55
 - pkiserv.tmpl 54
 - readymsg.form 55
 - rejectmsg.form 55
 - creating
 - CA certificate using PCICC 199
 - CertificatePolicies extension 139
 - ICL data sets 83
 - ICL indexes 83
 - VSAM object store 83
 - customizing
 - administration Web pages 135
 - e-mail notifications 132
 - end-user Web pages 116, 117, 121
 - pkiserv.tmpl 116, 117, 121
 - deleting
 - multiple certificates 191
 - selected certificates 193
 - single certificate 189
 - single request 180
 - e-mail notifications, customizing 132
 - encrypted passwords for LDAP servers 214
 - environment variables, updating 56
 - establishing PKI Services as an intermediate CA 205
 - gskkyman for certificate store 367
 - HostIdMappings extensions, administering 201
 - ICL data sets, creating 83
 - IKYSETUP, using 47
 - inquiry access, authorizing users for 200
 - installing
 - ICSF 32
 - LDAP 30
 - OCSF and OCEP 29
 - z/OS HTTP Server 27
 - intermediate certificate authority, making PKI Services 205
 - key ring, locating 203
 - LDAP
 - schema.user.ldif, updating 69
 - section of PKI Services configuration file, tailoring 76
 - updating schema.user.ldif 69
 - LDAP bind passwords, encrypted
 - IRR.PROXY.DEFAULTS profile 215
 - LDAPBIND class 214
 - LDAP servers, encrypted passwords for 214

- steps (*continued*)
 - locating
 - key ring 203
 - PKI Services certificate 203
 - modifying single request 180
 - passwords, encrypted LDAP binding
 - IRR.PROXY.DEFAULTS profile 215
 - LDAPBIND class 214
 - performing RACF administration using IKYSETUP 47
 - PKI Services certificate authority certificate, renewing 206
 - PKI Services certificate, locating 203
 - PKI Services daemon
 - starting 85
 - stopping 86
 - pkiserv.conf
 - copying 54
 - updating 58, 64
 - pkiserv.tmpl
 - copying 54
 - customization, additional first-time 117
 - customization, minimal 116
 - retrofitting release changes 121
 - PKITP, configuring 341
 - pkitpsamp.c 347
 - processing
 - multiple certificates 191
 - multiple requests through searches 184
 - selected certificates 193
 - selected requests 186
 - single certificate 189
 - single request 180
 - RACF administration using IKYSETUP 47
 - recovering a CA certificate profile 208
 - rejecting single request 180
 - removing administration page link 136
 - renewing
 - certificate 169
 - PKI Services certificate authority certificate 206
 - requesting a certificate 162
 - retrieving a certificate
 - from bookmarked Web page 167
 - from PKI Services home page 169
 - retrofitting changes into certificate templates 121
 - revoking
 - certificate (by user) 172
 - multiple certificates 191
 - selected certificates 193
 - single certificate 189
 - RLS
 - enabling VSAM data sets for 84
 - preliminary steps for establishing 82
 - running IKYSETUP 47
 - searching for requests 184
 - sendmail configuration, testing 33
 - setting up /var/pkiserv 67
 - starting
 - PKI Services 85
 - PKI Services daemon 85
 - z/OS HTTP Server 74

- steps (*continued*)
 - stopping PKI Services daemon 86
 - tailoring LDAP section of PKI Services configuration file 76
 - testing sendmail configuration 33
 - updating
 - configuration file 58, 64
 - environment variables 56
 - exit code sample 142
 - LDAP section of pkiserv.conf 79
 - pkixit.c 142
 - pkiserv.conf 58, 64
 - signature algorithm 141
 - single request 180
 - z/OS HTTP Server configuration files 71, 72
 - user ID for requesting certificates, changing 128
 - user ID for retrieving certificates, changing 129
 - using
 - gskkyman 367
 - IKYSETUP 47
 - using encrypted passwords for LDAP servers 214
 - creating IRR.PROXY.DEFAULTS profile 215
 - creating LDAPBIND class profile 214
 - viewing Web pages 85
 - VSAM object store, creating 83
 - z/OS HTTP Server configuration files, updating 71
- STOP command 86
- stopping
 - PKI Services 85, 86
- storage needs
 - for ICL 81
 - for object store 82
- STORCLAS statements 83
- store
 - creating 83
 - determining size requirements 82
- storing
 - binding information 21
 - certificate requests 195
 - certificate revocation lists 5
 - certificates 5
 - encrypted password information for LDAP servers 75
 - in LDAP 10
 - LDAP server encrypted password information 75
 - password for LDAP server
 - encrypted 21, 75
 - in clear text in pkiserv.conf 21
 - private key
 - in ICSF 5, 10, 32, 199
 - in RACF 5
 - serial number and event files in VSAM
 - areas affected 23
 - coexistence considerations 23
 - dependencies 23
 - documents 23
 - overview 23
 - tasks 23
- street (field in end-user Web pages) 162
- Street (named field in pkiserv.tmpl) 97
- STREET distinguished name qualifier migration
 - areas affected 21
 - coexistence considerations 21
 - dependencies 21
 - documents 21
 - summary 20
 - tasks 21
- subcomponent level
 - for logging 305
- subdirectory
 - bin 257
 - include 257
 - lib 258
 - PKIServ 258
 - samples 259
- SubjectAltName (certificate extension) 339
- SubjectKeyIdentifier (certificate extension) 339
- subordinate certificate authority
 - using PKI Services as 204
- subsections in certificate templates, summary 105
- substitution variables
 - base64cert 92, 94
 - browsertype 92
 - iecert 92
 - optfield 92
 - pkiserv.tmpl 91
 - printablecert 92
 - tmplname 92
 - transactionid 92
- SUCCESSCONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 104
- suffix
 - LDAP 69
- summary of changes xxi
- summary of interface changes 23
- superuser authority 54
- surrog (variable in IKYSETUP) 47
- surrog_uid (variable in IKYSETUP) 40
- surrogate operation
 - setting up 37, 313
- surrogate user ID 127
 - creating 37
 - PKISERV 47, 310
- syntax conventions
 - how to read xvi
- SYS1.CSSLIB 45
- SYS1.LINKLIB 45
- SYS1.LOGREC 219
- SYS1.SAMPLIB members
 - summary of interface changes 25
- SYS1.SAMPLIB(IKYSETUP) 37
- SYSOUT
 - records, contents 228
 - viewing information 225
- sysplex enabled
 - in ObjectStore 59
- sysplex support
 - daemon, PKI Services, starting 85
 - migration
 - areas affected 17

- sysplex support (*continued*)
 - migration (*continued*)
 - coexistence considerations 18
 - dependencies 18
 - documents 19
 - summary 17
 - tasks 18
 - PKI Services daemon, starting 85
 - pkiserv.conf
 - updating, overview 57
 - updating, steps for 64
 - prerequisites 9
 - requirements 9
 - RLS
 - enabling VSAM data sets for 84
 - preliminary steps for establishing 82
 - SharedVSAM
 - description 59
 - updating 65
 - starting PKI Services daemon 85
 - updating pkiserv.conf 57
 - updating SharedVSAM 65
- system architecture diagram 5
- time period
 - in ObjectStore before automatic deletion 59
- TimeBetweenCRLs (parameter in pkiserv.conf) 61
- Title 97
- title (field in end-user Web pages) 162
- tmplname substitution variable 92
- transaction ID
 - field in administration Web pages 177
 - field in end-user Web pages 162
- TransactionId (named field in pkiserv.tmpl) 97
- transactionid substitution variable 92
- true name of certificate templates 101
- Trust Policy
 - API — CSSM_TP_PassThrough 341
 - overview 337
- trusting PKI Services 201
- two-year PKI browser certificate for authenticating to z/OS
 - description 100
 - fields 106
- types of certificates 157
- TZ environment variable 55

T

- tailoring
 - LDAP configuration 69
 - LDAP section of PKI Services configuration file 76
- task roadmap for implementing PKI Services 14
- TCPIP.SEZALINK 45
- TDBM 10, 30
 - specifying password as entry 69, 72
- team members 11
- TEMPLATE section of pkiserv.tmpl
 - ADMINAPPROVE subsection 102
 - APPL subsection 102
 - CONSTANT subsection 102
 - CONTENT subsection 101
 - examining contents 109
 - FAILURECONTENT subsection 104
 - RETRIEVECONTENT subsection 104
 - RETURNCERT subsection 105
 - subsections 101
 - SUCCESSCONTENT subsection 104
- templates
 - adding 127
 - customizing
 - additional first-time changes 117
 - minimal 116
 - retrofitting release changes 121
- testing
 - PKI Services configuration 85
- threads
 - created at initialization 63
- time interval
 - before certificate expiration 60
 - between certificate revocation lists 61
 - for scanning for items to post 76
 - scanning database for approved requests 60
 - warning message about certificate expiration 60

U

- unencrypted, LDAP bind passwords 214
- Uniform resource identifier (URI)
 - field in end-user Web pages 162
- Uniform Resource Identifier (URI)
 - Certification Practice Statement 62
- UNIX programmer
 - skills 12, 13
 - tasks 12, 13
 - configuring sendmail 33
 - configuring UNIX runtime environment 53
 - LDAP section of pkiserv.conf, updating 75
 - runtime environment, configuring 53
 - sendmail, configuring 33
 - UNIX runtime environment, configuring 53
 - updating LDAP section of pkiserv.conf 75
 - team member 11
- UNIX runtime environment
 - configuring 53
- unix_sec (variable in IKYSETUP)
 - decision table 43
 - default value 45
 - description 45
- UPDATE access
 - IRR.DIGTCERT.ADD 311
 - IRR.DIGTCERT.EXPORT 310, 311
 - IRR.RPKISERV.PKIADMIN 311
- updating
 - access to administration pages 136
 - certificate request 181
 - certificate templates file
 - customization, additional first-time 117
 - minimal 116
 - retrofitting changes 121
 - configuration file
 - overview 57
 - steps 58

- updating (*continued*)
 - e-mail notifications 129, 132
 - environment variables
 - overview 55
 - steps 56
 - exit 142
 - expiringmsg.form 132
 - forms for e-mail notifications 129
 - IKYCVSAM 83
 - IKYRVSAM 84
 - LDAP section of pkiserv.conf 79
 - notification forms 129
 - pkixit.c 142
 - pkiserv.conf
 - overview 57
 - steps 58
 - pkiserv.tmpl
 - customization, additional first-time 117
 - minimal 116
 - pkitpsamp.c 347
 - readymsg.form 132
 - rejectmsg.form 132
 - runtime user ID 127
 - for requesting certificates 128
 - for retrieving certificates 129
 - signature algorithm 141
 - z/OS HTTP Server configuration files 71
- URI 162
 - containing CPS 62
- usage policy 58
- use_icsf (variable in IKYSETUP)
 - decision table 42
 - default value 46
 - description 46
- user ID
 - associating with PKI Services started procedure 37, 309
 - changing
 - requesting certificates 128
 - retrieving certificates 129
 - PKI Services daemon 63
 - runtime
 - changing 127
- user notifications
 - copying files 55
 - customizing forms 132
 - migration
 - summary 19
- Userld (named field in pkiserv.tmpl) 97
- userld directive 72, 73
- UserNoticeText1 (parameter in pkiserv.conf) 62, 140
- userPassword attribute 69, 72
- using
 - administration home page 178
 - administration Web pages 173
 - certificate policies 139
 - end-user Web pages 157
 - exit 141
- utilities
 - executables 257
 - PKI Services 231

- utilities (*continued*)
 - iclview 235
 - vosview 232
 - summary of interface changes 25

V

- validating
 - parameters 141
- variables
 - in IKYSETUP REXX exec
 - change based on setup 41
 - change optionally 46
 - change required 39
 - configurable section 38
 - in notification forms
 - %%dn%% 131
 - %%notafter%% 131
 - %%requestor%% 131
 - %%transactionid%% 131
- variables-dir 9
- Verbose Diagnostic messages, logging 306
- VERIFY
 - accesses required 213
 - R_PKIServ function 310
- viewing
 - SYSOUT information 225
 - VSAM ICL data set records 235
 - VSAM ObjectStore data set records 232
- virtual private network (VPN) devices 3
- VOL statements 83
- vosview
 - examples 232
 - format 232
 - parameters 232
 - purpose 232
 - summary of interface changes 25
- VPN devices 3, 7
- VSAM
 - data set name for ICL data 59
 - data set name for ObjectStore alternate index 59
 - data set name for ObjectStore data 58
 - event file, storing in
 - migration overview 23
 - RLS
 - enabling data sets for 84
 - preliminary steps for establishing 82
 - serial number file, storing in
 - migration overview 23
 - storing serial number and event files in
 - migration overview 23
- VSAM data sets
 - creating 81
 - not using RLS 330
 - using RLS 83, 84, 332
 - giving administrators access to 37
 - RLS, enabling for 84
- VSAM object store, creating 83
- vsamhlq (variable in IKYSETUP) 47

W

- warning message before certificate expiration 60
- Warning messages, logging 306
- Web pages
 - accessing 157, 173
- Web server
 - daemon user ID WEBSRV 47
- Web server programmer
 - installing and configuring z/OS HTTP Server 27
 - skills 12
 - starting the z/OS HTTP Server 74
 - tasks 12, 13
 - configuring z/OS HTTP Server 27
 - installing z/OS HTTP Server 27
 - starting z/OS HTTP Server 74
 - updating z/OS HTTP Server configuration files 71
 - z/OS HTTP Server, installing and configuring 27
 - z/OS HTTP Server, starting 74
 - z/OS HTTP Server, updating configuration files 71
 - team member 11
 - updating the z/OS HTTP Server's configuration files 71
- web_dn (variable in IKYSETUP) 41
- web_expires (variable in IKYSETUP) 47
- web_label (variable in IKYSETUP) 47
- web_ring (variable in IKYSETUP) 41
- websrv (variable in IKYSETUP) 47
- WEBSRV 47

X

- X.509v3 certificates 6, 7

Y

- your name (field in end-user Web pages) 162

Z

- z/OS
 - PKI browser certificate for authenticating to
 - description 100
 - fields 106
 - V1R3
 - migrating from 17
 - V1R4
 - event file, storing in VSAM 23
 - new information xxi
 - new messages xxi
 - serial number file, storing in VSAM 23
 - storing serial number and event files in VSAM 23
 - z/OS Version 1 Release 4
 - migrating to 17
- z/OS HTTP Server
 - configuration files
 - updating 71
 - configuring 27

- z/OS HTTP Server (*continued*)
 - description of PKI Services component 5
 - installing 27
 - operating modes PKISERV requires 312
 - PKI Services component
 - description 5
 - setting up for surrogate operation 37, 313
 - starting 74
- z/OS product libraries
 - ALINKLIB 257
 - APROCLIB 257
 - ASAMPLIB 257
 - LINKLIB 257
 - PROCLIB 257
 - SAMPLIB 257
- z/OS UNIX level security 43
- z/OS V1R4
 - e-mail notifications
 - areas affected 19
 - coexistence considerations 20
 - customizing forms 132
 - dependencies 20
 - documents 20
 - ExpireWarningTime, updating 65
 - ReadyMessageForm, updating 66
 - summary 19
 - tasks 20
 - updating ExpireWarningTime 65
 - updating ReadyMessageForm 66
 - encrypted passwords for LDAP servers
 - affected areas 22
 - coexistence considerations 22
 - dependencies 22
 - documents 22
 - LDAPBIND class profile 75
 - overview 21
 - RACF administration 214
 - storing information for 75
 - tasks 22
 - updating LDAP section of pkiserv.conf 76
 - event file, storing in VSAM
 - summary 23
 - LDAP servers, encrypted passwords for
 - LDAPBIND class profile 75
 - overview 21
 - RACF administration 214
 - storing information for 75
 - updating LDAP section of pkiserv.conf 76
 - MAIL distinguished name qualifier
 - areas affected 21
 - coexistence considerations 21
 - dependencies 21
 - documents 21
 - summary 20
 - tasks 21
 - notifications for users
 - customizing forms 132
 - summary 19
 - POSTALCODE distinguished name qualifier
 - areas affected 21
 - coexistence considerations 21

z/OS V1R4 *(continued)*
 POSTALCODE distinguished name qualifier
 (continued)
 dependencies 21
 documents 21
 summary 20
 tasks 21
 serial number file, storing in VSAM
 summary 23
 storing serial number and event files
 areas affected 23
 coexistence considerations 23
 dependencies 23
 documents 23
 tasks 23
 STREET distinguished name qualifier
 areas affected 21
 coexistence considerations 21
 dependencies 21
 documents 21
 summary 20
 tasks 21
 sysplex support
 areas affected 17
 coexistence considerations 18
 dependencies 18
 documents 19
 RLS, enabling VSAM data sets for 84
 RLS, preliminary steps for establishing 82
 SharedVSAM, updating 65
 summary 17
 tasks 18
 updating SharedVSAM 65
z/OS.e xxi
zip code 162

Readers' Comments — We'd Like to Hear from You

**z/OS
Security Server PKI Services
Guide and Reference**

Publication No. SA22-7693-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



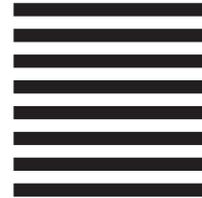
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5694-A01, 5655-G52

Printed in U.S.A.

SA22-7693-02

