Tivoli® SecureWay® Public Key Infrastructure

System Administration Guide

Version 3  Release 7.0

SH09-4532-02
Third Edition (November 2000)

This edition applies to IBM SecureWay Trust Authority, program 5648-D09, version 3 release 7 modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 1999, 2000. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Tables ....................................................................... vii

Chapter 1. About Trust Authority ........................................... 1

Chapter 2. Overview ......................................................... 3

Chapter 3. How do I...? ...................................................... 5

  Administer Trust Authority .................................................. 5
  Enter commands on Windows NT ........................................... 5
  Change Trust Authority passwords ........................................ 5
  Start and stop the server components .................................... 7
  Use IniEditor to change configuration files .............................. 9
  Change Trust Authority IP addresses ..................................... 11
  Back up and restore the system ............................................ 12
  Track numbers of users and certificates .................................. 12
  Track certificate activity .................................................... 13

  Administer the WebSphere Application Server ........................... 15
  Check the status of the WebSphere Application Server .......... 15
  Check the WebSphere Application Server logs ....................... 16

  Administer the HTTP Server ............................................... 16
  Check the status of the HTTP Server ................................... 17
  Check the HTTP Server logs ............................................. 18

  Administer the CA server ................................................... 18
  Change the CA server listener port ...................................... 18
  Change the CA polling interval .......................................... 19
  Change CRL settings ....................................................... 19
  Change the ICL protection policy ........................................ 20
  Change the ICL protection key ............................................ 21
  Check the integrity of the CA server database ......................... 22
  Request a CA certificate using cross-certification ................... 22
  Request a CA certificate using the hierarchy model ................. 25
  Make updates after changing the CA hierarchy ..................... 27
  Grant a CA certificate to a third party CA ............................. 28
  Check the CA server logs ................................................ 29
  Check the status of the CA server ...................................... 29
  Activate CRL distribution points ......................................... 30

  Administer the RA server .................................................. 30
  Add registrars .............................................................. 31
Enable encryption for the RA database ........................................... 33
Change the RA server listener port ............................................. 34
Change the RA polling interval ................................................ 34
Change the RA retry interval .................................................. 35
Check the RA server logs .................................................... 35
Check the status of the RA server ............................................. 35
Change RA settings for communications with the Directory . . . . . . . . . 36
Enabling or disabling Trust Chain Delivery ....................................... 37
Installing Multiple RAs ...................................................... 37

Administer the Audit subsystem .................................................... 39
View audit records .................................................................... 40
Search audit records .................................................................. 41
Change the Audit server port on the Audit client .................... 42
Change how events are sent from the Audit client . . . . . . . . . . . . . 42
Change the port on which the Audit server listens ................ 43
Change binding attempts from the Audit client to the Audit server . 44
Change the interval between binding attempts ......................... 44
Change log settings .................................................................. 44
Generate audit reports ........................................................... 47
Archive and sign the audit log files ............................................. 47
Check the integrity of the Audit server database and archive files . . . . . 48
Check the status of the Audit server ............................................. 50
Check the Audit server logs ..................................................... 51

Administer the DB2 databases ..................................................... 51
Check the status of the DB2 databases ........................................... 51
Check the DB2 logs ................................................................ 53

Administer the Directory server .................................................. 53
Check the status of the Directory server ........................................ 54
Check the Directory server logs ................................................. 55

Administer the 4758 Cryptographic Coprocessor .................................. 55

Chapter 4. Tell me about .......................................................... 57
Trust Authority security .......................................................... 57
Access control lists ............................................................. 57
Certificate authorities ............................................................ 57
CA hierarchies ......................................................................... 58
Certificate extensions ........................................................... 59
Certificate revocation lists ..................................................... 61
Cross-certification ............................................................... 61
<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit Archive and Sign utility</td>
<td>99</td>
</tr>
<tr>
<td>Audit Integrity Check utility</td>
<td>100</td>
</tr>
<tr>
<td>Create New RA utility</td>
<td>101</td>
</tr>
<tr>
<td>MRA Enrollment Authorization utility</td>
<td>101</td>
</tr>
<tr>
<td>Audit event fields</td>
<td>102</td>
</tr>
<tr>
<td>Audit events</td>
<td>102</td>
</tr>
<tr>
<td>Audit database data</td>
<td>105</td>
</tr>
<tr>
<td>Keys table</td>
<td>106</td>
</tr>
<tr>
<td>Event Severities table</td>
<td>106</td>
</tr>
<tr>
<td>Event Control table</td>
<td>106</td>
</tr>
<tr>
<td>Sources table</td>
<td>106</td>
</tr>
<tr>
<td>Authorized Entities table</td>
<td>107</td>
</tr>
<tr>
<td>Authorized Roles table</td>
<td>107</td>
</tr>
<tr>
<td>Affected Entity Types table</td>
<td>107</td>
</tr>
<tr>
<td>Component Types table</td>
<td>108</td>
</tr>
<tr>
<td>Audit Log table</td>
<td>108</td>
</tr>
<tr>
<td>System table</td>
<td>109</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>110</td>
</tr>
<tr>
<td>Basic troubleshooting</td>
<td>110</td>
</tr>
<tr>
<td>Troubleshooting with debug-level messaging enabled</td>
<td>110</td>
</tr>
<tr>
<td>Notices</td>
<td>113</td>
</tr>
<tr>
<td>Trademarks and service marks</td>
<td>114</td>
</tr>
<tr>
<td>Related information</td>
<td>117</td>
</tr>
<tr>
<td>Glossary</td>
<td>119</td>
</tr>
<tr>
<td>Index</td>
<td>137</td>
</tr>
</tbody>
</table>
Tables

1. Trust Authority configuration files ................................................................. 10
2. WebSphere Application Server logs from transactions with the Setup Wizard ........... 16
3. WebSphere Application Server logs from transactions with the registration facility ........ 16
4. HTTP server logs ....................................................................................... 18
5. CA server logs ............................................................................................. 29
6. RA server logs ............................................................................................. 35
7. Column descriptions for the Trust Authority Audit database views .......................... 40
8. Audit server logs .......................................................................................... 51
9. Database locations ...................................................................................... 51
10. Directory server logs ................................................................................... 55
11. Certificate extensions .................................................................................. 59
12. Three-server, three-port model for IBM HTTP Servers ..................................... 67
13. CA server configuration file ......................................................................... 73
14. RA server configuration file ......................................................................... 82
15. Audit server configuration file ...................................................................... 91
16. Audit client configuration file ...................................................................... 95
17. Audit event fields ......................................................................................... 102
18. Audit events .................................................................................................. 102
19. Keys table fields .......................................................................................... 106
20. Event Severities table fields ....................................................................... 106
21. Event Control table fields ........................................................................... 106
22. Sources table fields ..................................................................................... 107
23. Authorized Entities table fields ................................................................... 107
24. Authorized Roles table fields ....................................................................... 107
25. Affected Entity Types table fields .................................................................. 107
26. Component Types table fields ..................................................................... 108
27. Audit Log table fields .................................................................................. 108
28. System table fields ...................................................................................... 109
IBM® SecureWay® Trust Authority provides applications with the means to authenticate users and ensure trusted communications:

- It allows organizations to issue, publish, and administer digital certificates in accordance with their registration and certification policies.
- Support for Public Key Infrastructure for X.509 version 3 (PKIX) and Common Data Security Architecture (CDSA) cryptographic standards allows for vendor interoperability.
- Digital signing and secure protocols provide the means to authenticate all parties in a transaction.
- Browser- and client-based registration capabilities provide maximum flexibility.
- Encrypted communications and secure storage of registration information help ensure confidentiality.

A Trust Authority system can run on IBM® AIX/6000® and Microsoft® Windows NT® server platforms. It includes the following key features:

- A trusted Certificate Authority (CA) manages the life cycle of digital certification. To vouch for the authenticity of a certificate, the CA digitally signs each one it issues. It also signs certificate revocation lists (CRLs) to vouch for the fact that a certificate is no longer valid. To further protect its signing key, you can use cryptographic hardware, such as the IBM SecureWay® 4758 PCI Cryptographic Coprocessor.
- A Registration Authority (RA) handles the administrative tasks behind user registration. The RA provides that only certificates that support your business activities are issued, and that they are issued only to authorized users. The administrative tasks can be handled through automated processes or human decision-making.
- A Web-based enrollment interface makes it easy to obtain certificates for browsers, servers, and other purposes, such as virtual private network (VPN) devices, smart cards, and secure e-mail.
- A Windows® application, the Trust Authority Client, enables end users to obtain and manage certificates without using a Web browser.
- A Web-based administration interface, the RA Desktop, enables authorized registrars to approve or reject enrollment requests and administer certificates after they have been issued.
- An Audit subsystem computes a message authentication code (MAC) for each audit record. If audit data is altered or deleted after it has been written to the audit database, the MAC enables you to detect the intrusion.
- Policy exits enable application developers to customize the registration processes.
Integrated support for a cryptographic engine. To authenticate communications, the core Trust Authority components are signed with a factory-generated private key. Security objects, such as keys and MACs, are encrypted and stored in protected areas called KeyStores.


Integrated support for IBM WebSphere™ Application Server and IBM HTTP Server. The Web server works with the RA server to encrypt messages, authenticate requests, and transfer certificates to the intended recipient.

Integrated support for the award-winning IBM DB2® Universal Database.
Overview

This document provides the information required to operate and administer the Trust Authority system. It assumes that you have knowledge or experience in the following:

- AIX® or UNIX® operating system
- Windows NT operating system
- System architecture
- Network administration
- Database administration
- Web server administration
- Directory administration

“How do I...?” on page 5 provides procedural information on how to operate and administer the Trust Authority system. It tells you how to start and stop the system, how to change passwords, how to use the configuration file editor, how to back up and restore the system, and how to administer the system components:

- IBM WebSphere™ Application Server
- IBM HTTP Server
- CA server
- RA server
- Audit subsystem
- IBM DB2 Universal Database™ (UDB)
- IBM SecureWay Directory
- 4758 Cryptographic Coprocessor

“Tell me about...” on page 57 provides more detailed technical information about the subjects that are raised in the procedures.

“Reference” on page 71 provides reference information such as configuration file parameters, syntax for command-line utilities, and tabular information on the Audit subsystem.
How do I...?

The topics in this chapter tell you how to administer IBM SecureWay Trust Authority and its components.

Administer Trust Authority

This section describes the tools and processes that you can use to administer the Trust Authority system as a whole.

They are as follows:

- Entering commands on the Windows NT command line
- Changing Trust Authority passwords
- Starting and stopping the server components
- Using the IniEditor to change configuration files
- Checking Trust Authority configuration logs
- Changing Trust Authority IP addresses
- Backing up and restoring the system
- Tracking certificate activity
- Tracking numbers of users and certificates

Enter commands on Windows NT

If you installed Trust Authority on Windows NT, there are several tasks that do not have program icons and that must be run from a DOS command line. For example, you must manually run add_rauser to define registrars.

When you specify a command that requires a directory path as a parameter, and the path contains embedded spaces, you must enclose the path in double quotation marks ("). For example, you would specify the path parameter in the add_rauser command as follows:

```
add_rauser "c:\Program Files\IBM\Trust Authority\bin"
```

Change Trust Authority passwords

IBM Trust Authority provides a Change Password utility. You can use it to change the default passwords set during configuration of the system. You should change the passwords at least once after initial system configuration and before the system becomes available to your users. These passwords control access to the following functional components:

- Secure start-up mechanism
The Control Program password enables the system to auto-start or shut down all Trust Authority components. It provides access to the secure start-up encryption key.

- **Directory**
  The Directory Administrator password controls access to the elements that you can change in the Directory.

- **Audit server**
  The Audit Administrator password provides access to the audit logs and the audit administration tools.

- **4758 CA profile**
  The 4758 CA Profile password controls access to the 4758 Cryptographic Coprocessor CA profile.

To change passwords, you can use one of a variety of approaches. If you installed Trust Authority on an AIX platform, you must use the command-line procedure described in [Command-line procedure](#). If you are running on a Windows NT platform, you can use either the procedure described in [Windows NT program icon procedure](#) or the command-line procedure to run the Change Password utility.

**Command-line procedure**

1. Go to the Trust Authority bin directory by using one of the following paths:
   - On AIX, the default value for the path is: `/usr/lpp/iau/bin`
   - On Windows NT, the default value for the path is: `c:\Program Files\IBM\Trust Authority\bin`

2. Depending on your operating system, enter one of the following commands:
   - On AIX, the command is: `changePWD.sh`
   - On Windows NT, the command path is: `changePWD.bat`

   The following menu appears:
   
   `------------- Change Trust Authority Passwords -------------`
   
   Enter the option number for the component you want to change. You will be prompted to enter the current password and then the new password.

   `----------------------------------------------------------------------------`
   
   Change password for
   1) Quit
   2) Control Program
   3) Directory Administrator
   4) Audit Administrator
   5) 4758 CA Profile
   
   Enter Option:

3. Select the option for the password you want to change by entering the corresponding number in the Enter Option field.

4. When the system prompts you to do so, supply and confirm the current password for the option you selected.
Notes:

a. The first time you use this tool, the current password is the password set at configuration time. For the Control Program and Audit Administrator passwords, specify the Trust Authority server password as the current password. For the Directory Administrator password, specify the Directory Administrator password that was created during configuration as the current password. For the 4758 CA Profile, specify IBMCA001.

b. If you installed Trust Authority on an AIX server and did not change the configuration user (cfguser) password before configuring the system, the default password is Secure99.

5. When the system prompts you to do so, supply and confirm the new password for the option you selected.
   The password must be less than or equal to eight characters in length (the 4758 CA Profile password must be exactly eight).
   The utility displays a message to let you know whether your changes were successful.

When processing is complete, the utility returns you to the main menu.

Windows NT program icon procedure
If you are running Trust Authority on a Windows NT platform, and prefer to select program icons instead of entering the command in a DOS window, use the following procedure to run the Change Password utility:

1. Select Start → Programs → IBM SecureWay Trust Authority → Change Passwords Tool
   The Change Trust Authority Passwords menu appears.

2. Follow the steps starting with step 5 on page 6 in “Command-line procedure” on page 6.

Start and stop the server components
IBM Trust Authority uses a secure, automated start-up mechanism called the Trust Authority Control program to start or stop all the components on a given machine. The Trust Authority Control (one per machine in the case of remote machines) has its own encryption/decryption key in which component passwords are encrypted or decrypted, as needed. If you installed any Trust Authority server components on remote machines, refer to the section “Start and stop remote servers” on page 9.

Start and stop server components locally
To start or stop all the components on a given machine, you can use any of the following methods:

- Enter the following commands on AIX to start or stop the server components. Follow the online instructions to enter the Control Program password. Note that to start the system, you must log in as the Trust Authority configuration user (cfguser). To stop the system, you can log in as root or cfguser.

  `cd /usr/lpp/iau/bin`
  `Start_TA.sh`
  or
  `Stop_TA.sh`

- If you installed Trust Authority on a Windows NT platform, you should select the appropriate icon from the NT program menu. For example, select:
If you need to use the command approach on NT, enter the following commands on a DOS command line to start or stop the server components. Follow the online instructions to enter the Control Program password.

- `Start_TA.bat`
- `Stop_TA.bat`

**Usage guidelines on AIX**

Guidelines for starting and stopping the system on AIX are as follows:

- If for some reason the CA and RA servers do not stop when you run the Control Program on AIX, you should use the ps command to list the running httpd processes, and then use the kill command to stop them. Shown below are examples of how to use these commands along with sample output.

  ```bash
  $ ps -ef | grep irgAutoCA
  root  18886  1  0 Oct 24 - 76:34 /usr/lpp/iau/bin/irgAutoCa
  /usr/lpp/iau/etc/TrustAuthority/irgAutoCA.ini
  cfguser 23316 32400 2 11:58:20 pts/4 0:00 grep irgAutoCA
  $ kill 18886
  $ ps -ef | grep irgrasvr
  root  23526  1 12 Oct 24 - 207:46 /usr/lpp/iau/bin/irgrasvr -c
  /usr/lpp/iau/pkrf/etc/domain.cfg -d YourDomain
  cfguser 26016 12488 3 11:57:16 pts/0 0:00 grep irgrasvr
  $ kill 23526
  ```

- Stopping Trust Authority should automatically stop the Web server programs on AIX. If they did not stop for some reason, you can use the ps command to identify the process ID of the instance to be stopped for your domain, and then use the kill command to stop it. For example:

  ```bash
  ps -ef | grep OutOfProc | grep YourDomain
  kill process_ID
  ```

- Starting and stopping Trust Authority will automatically start and stop the Directory server. If you do not want the Directory to be automatically started or stopped, edit the [AutoStopN] section of the TrustAuthControl.cfg file to remove the entry for the Directory component.

**Usage guidelines on Windows NT**

Guidelines for starting and stopping the system on Windows NT are as follows:

- Stopping Trust Authority does not automatically stop the Web server programs on Windows NT. After stopping Trust Authority you must manually stop the WebSphere instance that was started for the registration domain. To do this, use the NT Task Manager to end the java.exe and reqdbagent.exe processes.

- For optimum performance, before you restart the server components on a Windows NT system, you should reboot the machine.

- When you use the Control Program to start the Trust Authority server components on Windows NT, the window in which the program is launched cannot be closed. This is because some components are attached to this console window. After the Trust Authority server components are stopped, you will be able to close this window.
Starting and stopping Trust Authority automatically starts and stops the Directory server. If you do not want the Directory to be automatically started or stopped, edit the [AutoStopN] section of the TrustAuthControl.cfg file to remove the entry for the Directory component.

Start and stop remote servers
If you installed any Trust Authority components on remote machines, you must use the Control Program to start or stop the components that are running on each machine. To start remote components, use the following order:
1. Directory Server
2. CA and Audit Server
3. RA Server (including IBM HTTP Server and WebSphere Application Server)
To stop the system, stop the server programs in the reverse order from which they were started.

Use IniEditor to change configuration files
This section describes how to invoke and use the configuration file editor, IniEditor. The IniEditor tool allows you to add, update, and delete parameters and sections in each of the IBM Trust Authority configuration files. It displays each parameter as a name=value pair in an editing field so that you can easily locate and edit what you need. Refer to “Configuration files” on page 71 for configuration parameter descriptions and usage.

Notes:
1. The IniEditor is a simple editor designed to update configuration files. The tool does not provide validity checking of data.
2. For your protection, be sure to back up the configuration file you plan to edit before making any changes to it.

Run the editor
You can start the editor at the command prompt with or without specifying the name of the file you want to edit.

To run the editor using a file name such as myfile.ini, enter one of the following commands:

- In the AIX environment, at the command line:
  
  cd /usr/lpp/iau/bin
  run_IniEditor myfile.ini

- In the Windows NT environment, at the DOS command line:
  
  cd c:\Program Files\IBM\Trust Authority\bin
  IniEditor myfile.ini

To run the editor without using a file name, enter one of the following commands:

- In the AIX environment, on the command line:
  run_IniEditor

- In the Windows NT environment, on the DOS command line:
  IniEditor

If you do not enter a file name, the system prompts you to specify whether the file you want to edit is new or existing. If the file is new, the system prompts you to specify the type of
file you want to create. Each file type has a corresponding .ini file template that is read in when you specify the type. The options are as follows:

<table>
<thead>
<tr>
<th>Table 1. Trust Authority configuration files</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AIX default file location</strong></td>
</tr>
<tr>
<td>/usr/lpp/iau/etc/TrustAuthority/jonahca.ini</td>
</tr>
<tr>
<td>/usr/lpp/iau/pkrf/Domains/YourDomain/etc/jonahra.ini</td>
</tr>
<tr>
<td>/usr/lpp/iau/etc/AuditClient.ini</td>
</tr>
<tr>
<td>/usr/lpp/iau/etc/TrustAuthority/AuditServer.ini</td>
</tr>
</tbody>
</table>

The IniEditor attempts to locate the template files by searching the directory where the Trust Authority files reside.

**Use the editor**

After you have loaded the IniEditor with a configuration file, the existing parameters appear on the left-hand side of your screen in a tree structure. This structure contains the sections and keys of the file. To select a section, click on the name of the section. To expand the section, click on the plus (+) sign. If a section contains keys, the parameter name=value pair appears on the right-hand side of your screen in an editing field.
**Editing parameters**
You can perform the following editing tasks:

- To change the value of a parameter, type over the text in the editing field.
- To undo the change, select **Edit → Undo**.
- To delete a parameter or a section, select the items you want to delete and select **Edit → Delete**.

**Adding a section**
To add a section to the .ini file, follow these steps:

1. Select **Object → New section**.
   
   The Add a New Section dialog box appears.
2. Specify the name of the section in the dialog box.
3. Click **OK**.
   
   The new section appears at the bottom of the tree structure.

**Adding a parameter**
To add a parameter to a section in the .ini file, follow these steps:

1. Select **Object → New Parameter**.
   
   The Create a New Parameter dialog appears. This provides a drop-down menu from which you can select the section to which you want to add the parameter.
2. Specify the name of the parameter in the **Parameter** field.
3. Specify the value of the parameter in the **Value** field.
4. Click **OK**.
   
   The new parameter appears in the tree structure under the selected section, and on the right-hand side of the window in an editing field.

**Saving the file**
The IniEditor allows you to save the current .ini file, save the file with a different name, or exit the program. If the file has changed, the editor prompts you to save your changes. Both the Save option and the Exit (with a save) option save the current file as a backup with the extension .save and write the new file to the current file name.

**Change Trust Authority IP addresses**
If the Trust Authority system has been installed and configured on a single machine using a single hostname, then you can change the IP address of that machine using the following procedure:

1. Stop the Trust Authority system.
2. Change the IP address of the machine according to your organization’s policy.
3. Shut down and restart the machine.
4. Start the Trust Authority system.

This document does not address other scenarios involving multi-machine or multi-hostname IP address or hostname changes. Refer to the Library page of the IBM SecureWay Trust Authority Web site for the most current procedures.
Back up and restore the system
This section addresses backup and restore issues for IBM SecureWay Trust Authority. Specifically, the chapter covers the following:

- Guidelines for backing up and recovering the Trust Authority databases.
- Backing up and recovering the AIX and Windows NT operating systems.
- Cloning the IBM 4758 Cryptographic Coprocessor.

Back up and restore on AIX and Windows NT
To back up Trust Authority correctly, whether you are using AIX or Windows NT, you must first shut down all Trust Authority processes in the following order:

1. Trust Authority
2. DB2® databases

You must do this regardless of which backup utility you plan to use. The Directory server and DB2 use asynchronous writes, and unless these processes are fully shut down, there is no guarantee that data in the memory buffers will be written to disk when you back up the system. The closer you can get the system to a single-user maintenance mode, the better your backup will be. Refer to “Start and stop the server components” on page 7 for instructions on stopping and starting Trust Authority. Refer to IBM DB2 Universal Database Command Reference, Version 6.1 for stopping and starting the databases.

For backup utilities, IBM recommends the following:

- For AIX, use the built-in system backup utilities, mksysb and savevg or ADSM (another IBM product). If you use the mksysb and savevg utilities, consult your AIX documentation for correct procedures. If you use ADSM, use it in conjunction with mksysb.

- For Windows NT, use an aftermarket utility that makes a system image from which you can recover the entire system. If you use ADSM, use it in conjunction with a system image utility.

Follow the restore procedures appropriate to the backup utility you choose.

Clone the 4758 Cryptographic Coprocessor
To back up the 4758 Cryptographic Coprocessor, you must clone the master key, from the machine on which you have installed the Trust Authority system, to a separate machine. You can find the procedures for doing this on the Library page of the IBM SecureWay Trust Authority Web site at the following location:


Track numbers of users and certificates
Your license agreement with IBM requires you to monitor number of Trust Authority users. To help you obtain this information, Trust Authority provides a usage reporting tool, the Usage Report. This tool tracks the number of active and inactive users in your system and the number of active and revoked certificates signed by the Trust Authority CA since the product was installed. The specific Usage Report data items are described as follows:

Active users
Users with at least one active certificate, where user is defined as an entity that has a unique distinguished name (DN).
Inactive users
Users that have no active certificates; that is, certificates previously issued to these users have expired or been revoked.

Active certificates
Certificates signed by the Trust Authority CA that have not expired or been revoked.

Revoked certificates
Certificates that have been revoked.

Windows NT program icon procedure
If you are running Trust Authority on a Windows NT platform, and prefer to select program icons instead of entering the command in a DOS command window, use the following procedure to generate a usage report. See “Sample Usage Report output” for an example of the report output.

1. Log in as the Trust Authority configuration user on the machine where you installed the Trust Authority CA server component. The default and suggested user name is `cfguser`; your organization may have used a different value.


Command-line procedure
If you are running Trust Authority on an AIX platform, or if you prefer to enter the command instead of selecting program icons on Windows NT, use the following procedure to generate a usage report. See “Sample Usage Report output” for an example of the report output.

1. Log in as the Trust Authority configuration user on the machine where you installed the Trust Authority CA server component. The default and suggested user name is `cfguser`. If you installed Trust Authority on Windows NT, your organization may have used a different value.

2. Change to the bin subdirectory of the Trust Authority installation directory. The following examples show the default directory path:

   - On AIX: `/usr/lpp/iau/bin`
   - On Windows NT: `c:\Program Files\IBM\Trust Authority\bin`

3. Enter the following command:

   `TAUsageReport`

Sample Usage Report output

```
Trust Authority Usage Report
Report generated on : Sep 28, 1999 16:10:20 EDT
Total active users = 42
Total active certificates = 53
Total inactive users = 6
Total revoked certificates = 3
```

Track certificate activity
You can use the Trust Authority Activity Report utility to monitor the number of certificates that are approved, rejected, or revoked on a specific day or over the course of a week. The Activity Report is divided into the following two parts:
Part 1
The first part of the report lists the registrars (RA administrators) in your system and shows the certificate activity on a per registrar (administrator) basis.

Part 2
The second part of the report lists the total number of certificates that were approved, rejected, or revoked during the requested time period, whether manually by a registrar or automatically through policy exits.

Windows NT program icon procedure
If you are running Trust Authority on a Windows NT platform, and prefer to select program icons instead of entering the command in a DOS command window, use the following procedure to generate an Activity Report. See "Sample Activity Report output" on page 15 for an example of the report output.

1. Log in as the Trust Authority configuration user on the machine where you installed the Trust Authority RA server component. The suggested user name is cfguser; your organization may have used a different value.

2. Select Start → Programs → IBM SecureWay Trust Authority → Trust Authority Activity Report.

3. When the system prompts you for the type of report, enter 1 for a weekly report or 2 to view a summary of the activity that occurred on a specific day.

4. When the system prompts you for the date, enter the starting date for the week for which you want a report, or the exact date for which you want a daily report. You must type the date in yyyy-mm-dd format; for example, 1999-10-01.

Command-line procedure
If you are running Trust Authority on an AIX platform, or if you prefer to enter the command instead of selecting program icons on Windows NT, use the following procedure to generate an Activity Report. See "Sample Activity Report output" on page 15 for an example of the report output.

1. Log in as the Trust Authority configuration user on the machine where you installed the Trust Authority RA server component. The default and suggested user name is cfguser. If you installed Trust Authority on Windows NT, your organization may have used a different value.

2. Change to the bin subdirectory of the Trust Authority installation directory. The following examples show the default directory path:
   - On AIX: /usr/lpp/iau/bin
   - On Windows NT: c:\Program Files\IBM\Trust Authority\bin

3. Enter the following command:
   TAActivityReport

4. When the system prompts you for the type of report, enter 1 for a weekly report or 2 to view a summary of the activity that occurred on a specific day.

5. When the system prompts you for the date, enter the starting date for the week for which you want a report, or the exact date for which you want a daily report. You must type the date in yyyy-mm-dd format; for example, 1999-10-01.
Note: To bypass the prompts, you can specify the following command parameters on the command line:

\texttt{TAAActivityReport [-t reportType(1/2)] [-d startDate(yyyy-mm-dd)]}

For example: \texttt{TAAActivityReport -t 2 -d 1999-09-27}

**Sample Activity Report output**
The following example shows a daily activity report. The weekly report looks the same but provides a summary of weekly totals.

**Trust Authority Activity Report**

Report generated on : Sep 28, 1999 15:59:23 EDT
Date of daily report : 1999-09-27

**Daily Activity Report for RA Administrators**

<table>
<thead>
<tr>
<th>RA Administrator DN</th>
<th>Approved</th>
<th>Rejected</th>
<th>Revoked</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN=RAadmin1, OU=PKI, O=IBM, C=US</td>
<td>23</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>CN=RAadmin2, OU=PKI, O=IBM, C=US</td>
<td>14</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>CN=RAadmin3, OU=PKI, O=IBM, C=US</td>
<td>55</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

**Daily Activity Report for the System**

<table>
<thead>
<tr>
<th>Approved</th>
<th>Rejected</th>
<th>Revoked</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>16</td>
<td>5</td>
</tr>
</tbody>
</table>

**Administer the WebSphere Application Server**

IBM WebSphere is a set of software products that help organizations develop and manage high-performance Web sites. The WebSphere Application Server, along with the IBM HTTP Server, helps provide the infrastructure for Web server functionality in Trust Authority.

**Check the status of the WebSphere Application Server**

Depending on your environment, you can check the status of the WebSphere Application Server by performing one of the following sets of procedures:

- **On AIX:**
  1. Log in to AIX as root.
  2. Check the process table and look for:
     \texttt{OutofProcEngine}

     \texttt{If you see this process, go to step \textbf{3}. If you do not see this process, refer to \text"Troubleshooting" on page \textbf{110} for instructions.}

  3. Refer to the WebSphere documentation for accessing the WebSphere administration pages and determining the status of the server.

- **On Windows NT:**
  1. Log into Windows NT as the system administrator.
  2. Start the Task Manager by pressing the Ctrl Alt and Delete keys.
  3. Select the Processes tab.
  4. Look for at least one instance of the process java.exe.
If you see this process, go to step 5. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.

5. Refer to the WebSphere documentation for accessing the WebSphere administration pages and determining the status of the server.

Check the WebSphere Application Server logs

You can check the logs that record transactions between the WebSphere Application Server and the IBM Trust Authority Setup Wizard at the locations shown in Table 2.

Table 2. WebSphere Application Server logs from transactions with the Setup Wizard

<table>
<thead>
<tr>
<th>AIX default file location</th>
<th>Windows NT default file location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/WebSphere/AppServer/logs/TivoliPKICfg.stderr.log</td>
<td>c:\Program Files\IBM\Trust Authority\etc\logs\jvm_stderr.log</td>
<td>The standard error output from the Java Virtual Machine (JVM).</td>
</tr>
<tr>
<td>/usr/WebSphere/AppServer/logs/TivoliPKICfg.stdout.log</td>
<td>c:\Program Files\IBM\Trust Authority\etc\logs\jvm_stdout.log</td>
<td>The standard output from the JVM.</td>
</tr>
</tbody>
</table>

You can check the logs that record transactions between the WebSphere Application Server and your registration facility at the locations shown in Table 3.

Table 3. WebSphere Application Server logs from transactions with the registration facility

<table>
<thead>
<tr>
<th>AIX default file location</th>
<th>Windows NT default file location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/WebSphere/AppServer/logs/TivoliPKIstderr.log</td>
<td>c:\Program Files\IBM\Trust Authority\pkrf\Domains\YourDomain\etc\logs\jvm_stderr.log</td>
<td>The standard error output from the Java Virtual Machine (JVM).</td>
</tr>
<tr>
<td>/usr/WebSphere/AppServer/logs/TivoliPKIstdout.log</td>
<td>c:\Program Files\IBM\Trust Authority\pkrf\Domains\YourDomain\etc\logs\jvm_stdout.log</td>
<td>The standard output from the JVM.</td>
</tr>
<tr>
<td>/usr/lpp/iau/pkrf/Domains/YourDomain/logs/hostname_ssl-port-ssl-error.log</td>
<td>c:\Program Files\IBM\Trust Authority\pkrf\Domains\YourDomain\etc\logs\hostname_ssl-port-ssl-error.log</td>
<td>Error messages from transactions over secure HTTP connections. There can be two of these: Client authenticated and non-client authenticated.</td>
</tr>
<tr>
<td>/usr/lpp/iau/pkrf/Domains/YourDomain/logs/hostnameerror.log</td>
<td>c:\Program Files\IBM\Trust Authority\pkrf\Domains\YourDomain\logs\hostname_public-porterror.log</td>
<td>Error messages from transactions over public HTTP connections.</td>
</tr>
</tbody>
</table>

Administer the HTTP Server

IBM HTTP Server is the Web server product that handles Web-based communications with browsers and other programs. The HTTP daemon (HTTPD) is the persistent process that is the heart of the HTTP Server. Trust Authority employs multiple HTTP daemons. These HTTP daemon instances handle public, encrypted, and client-authenticated and encrypted connections. Refer to “Web servers” on page 67 for detailed information on Web servers and the different types of connections used in Trust Authority to handle different types of transactions.
Check the status of the HTTP Server

Depending on your environment, you can check the status of the HTTP Server by performing one of the following sets of procedures:

- On AIX:
  1. Log in to AIX as root.
  2. Check the process table and look for these process instances:
     * 1 of sidd
     * at least 2 of httpd

     If you see all three of these processes, go to §. If you do not see all three of these instances, refer to “Troubleshooting” on page 110 for instructions.
  3. Go to the bin directory which has the following default path:
     `/usr/lpp/iau/bin`
  4. Check the ports by entering this command for each port:
     
     ```
     checkSrvPortStatus -p port -s server -r1 -w1
     ```

     where `port` is the number of the port you want to check, and `server` is the name of the HTTP Server associated with it. Refer to Table 12 on page 67 for a summary of the Trust Authority default HTTP Server and port configuration.

     If the port you are checking is responding successfully, the system displays the following message:

     ```
     Port:port on server is bound.
     ```

     where `port` is the port you are checking and `server` is the name of the corresponding server.

- On Windows NT:
  1. Log into Windows NT as the system administrator.
  2. Start the Task Manager by pressing the Ctrl, Alt and Delete keys.
  3. Select the Processes tab.
  4. Look for two instances of the process Apache.exe.

     If you see these two processes, go to §. If you do not see these two instances, refer to “Troubleshooting” on page 110 for instructions.
  5. At the MS DOS prompt, go to the bin directory, which has the following default path:
     `c:\Program Files\IBM\Trust Authority\bin`
  6. Check the ports by entering this command for each port:

     ```
     checkSrvPortStatus -p port -s server -r1 -w1
     ```

     where `port` is the number of the port you want to check, and `server` is the name of the HTTP Server associated with it. Refer to Table 12 on page 67 for a summary of the Trust Authority default HTTP Server and port configuration.

     If the port you are checking is responding successfully, the system displays the following message:
Port: \textit{port} on server is bound.

where \textit{port} is the port you are checking and \textit{server} is the name of the corresponding server.

Check the HTTP Server logs

You can check the logs that record transactions between the IBM HTTP server and the Trust Authority Setup Wizard. Refer to the WebSphere documentation for names and locations of its logs.

You can check the logs that record transactions between the IBM HTTP server and your registration facility at the locations shown in Table 4.

\textbf{Table 4. HTTP server logs}\n
<table>
<thead>
<tr>
<th>AIX default file location</th>
<th>Windows NT default file location</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/HTTPServer/logs/error.log</td>
<td>c:\Program Files\IBM\Trust Authority\Domains\YourDomain \logs\apache.log.ERROR.PID</td>
<td>The IBM HTTP Server error messages.</td>
</tr>
<tr>
<td>/usr/HTTPServer/logs/access.log</td>
<td>c:\Program Files\IBM\Trust Authority\Domains\YourDomain \logs\apache.log.INFORM.PID</td>
<td>The IBM HTTP Server access messages.</td>
</tr>
</tbody>
</table>

Administer the CA server

This section provides operational and administrative procedures for the Trust Authority certificate authority (CA) server. The CA server handles the server-side tasks for the Trust Authority CA. It resides, along with its DB2 database instance, on either a local or remote machine.

The tasks that you are likely to perform to administer the CA server are as follows:

- Use the IniEditor to make these changes to the jonahca.ini configuration file:
  - Change the TCP port on which the CA listens.
  - Change the polling interval for PKIX messages.
  - Change the CRL settings.
  - Change the ICL protection policy..
- Change the ICL protection key.
- Check the integrity of the CA server database.
- Use the CA Certification utility for cross-certification and CA hierarchy.
- Check the CA server logs.
- Check the status of the CA server.
- Activate CRL distribution points.

Change the CA server listener port

The CA server listener port is where the CA listens for PKIX messages. To change the value of this port, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.

3. Start the IniEditor and load the jonahca.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.

4. Select and expand the Transport section and select the TCPPort parameter.

5. In the displayed editing field, change the value of the port number.

6. Save the file and exit the program.

7. Restart the IniEditor and load the jonahra.ini configuration file (this file could be on a local or remote machine, depending on your installation).

8. Select the URLs section.

9. In the displayed editing field, change the value of the port number.

10. Select and expand the General section and select the Issuer1URL1 parameter.

11. In the displayed editing field, change the value of the port number.

12. Save the file and exit the program.

13. Start the Trust Authority system.

Change the CA polling interval
The CA polling interval is the amount of time, in seconds (s), minutes (m), or hours (h), between interrogations of the workflow queue by the CA server. Elements on the queue whose dispatch times have elapsed are dispatched for processing. To change the polling interval, follow these steps:

1. Log in to your operating system as a system administrator.

2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.

3. Start the IniEditor and load the jonahca.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.

4. Select and expand the Transport section and select the PollInterval parameter.

5. In the displayed editing field, change the value of the polling interval.

6. Save the file and exit the program.

7. Start the Trust Authority system.

Change CRL settings
A certificate revocation list (CRL) is a digitally signed, time-stamped list of certificates that have been revoked by a CA. The following values can be changed in the CA server configuration file to affect the way the CRL is handled:

- The time between scheduled CRL creation
- The lifetime of a CRL

Change the time allowed between CRL creation
To change the policy on certificates and cross-certificates with respect to the time allowed between scheduled publication of a CRL, follow these steps:
1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to "Start and stop the server components" on page 7 for instructions.
3. Start the IniEditor and load the jonahca.ini configuration file. If necessary, refer to "Use IniEditor to change configuration files" on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.
4. Select and expand the CertPolicy section and select the TimeBetweenCRLs parameter.
5. In the displayed editing field, change the value of the time between the creation of new CRLs.
   The value will be an interval in minutes (m), hours (h), or days (d). For example, 1d. The value must be less than the value of the CRL duration.
6. Select and expand the CrossCertPolicy section and select the TimeBetweenCRLs parameter.
7. In the displayed editing field, change the value of the time between the creation of new CRLs.
   The value will be an interval in minutes (m), hours (h), or days (d). For example, 1d. The value must be less than the value of the CRL duration.
8. Save the file and exit the program.
9. Start the Trust Authority system.

**Change the lifetime of a CRL**
To change the policy on certificates and cross-certificates with respect to the duration, or lifetime, of a CRL, follow these steps:
1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to "Start and stop the server components" on page 7 for instructions.
3. Start the IniEditor and load the jonahca.ini configuration file. If necessary, refer to "Use IniEditor to change configuration files" on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.
4. Select and expand the CertPolicy section and select the CRLDuration parameter.
5. In the displayed editing field, change the value of the amount of time a CRL is valid.
   The value will be an interval in minutes (m), hours (h), or days (d). For example, 2d.
6. Select and expand the CrossCertPolicy section and select the CRLDuration parameter.
7. In the displayed editing field, change the value of the amount of time a CRL is valid.
   The value will be an interval in hours, minutes, or days. For example, 2d.
8. Save the file and exit the program.
9. Start the Trust Authority system.

**Change the ICL protection policy**
The Issued Certificate List (ICL) is a list of certificates that have been issued and their status. The ICL protection key is an encrypted key stored in the Trust Authority KeyStore and used to compute a message authentication code (MAC) value on all the columns of the ICL. In cases where the computed MAC value does not agree with the stored MAC value, the system consults the ICL protection policy and takes appropriate action.
To change the ICL protection policy, follow these steps:

1. Log into your operating system as a system administrator.

2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.

3. Start the IniEditor and load the jonahca.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.

4. Select and expand the ICL section and select the IclProtectionPolicy parameter.

5. In the displayed editing field, change the value of the ICL protection policy to one that conforms to the needs of your organization. The value will be one of the following:
   - **Ignore** — The system ignores the mismatch and does not display any messages.
   - **ContinueWithMessage** — The system displays a warning message but continues with the transaction.
   - **TerminateTransaction** — The system displays an error message and terminates the transaction.

6. Save the file and exit the program.

7. Start the Trust Authority system.

---

### Change the ICL protection key

The ICL protection key is never divulged to a user or an administrator. Nevertheless, to guard against unauthorized tampering or inadvertent data inconsistencies, it is advisable to change the ICL protection key on a periodic basis. Trust Authority provides a command-line tool, CARemac, that generates a new ICL protection key to recompute a new MAC value. To change the ICL protection key, follow these steps:

1. Go to the machine where the CA server is installed.

2. Stop the Trust Authority system using the procedure described in “Start and stop the server components” on page 7.

3. Run the ICL protection key tool by entering the CARemac command (which includes your Trust Authority installation path) on the Trust Authority command line, for example:
   - On AIX, when you are using the default Trust Authority installation path:
     ```bash
     CARemac in /usr/lpp/iau/bin
     ```
   - On Windows NT, when you are using the default Trust Authority installation path:
     ```bash
     CARemac in "c:\Program Files\IBM\Trust Authority\bin"
     ```

   The tool prompts you for the Trust Authority Control Program password.

4. Enter the Trust Authority Control Program password.

5. Start the Trust Authority system using the procedure described in “Start and stop the server components” on page 7.
**Check the integrity of the CA server database**

Trust Authority provides a tool called the CAIntegrityCheck to detect tampering with the CA server database. Under normal circumstances, the CA performs protection checking only on the subset of ICL entries that it is currently accessing. In other words, it checks only those records that are selected during CRL creation and therefore have a status of *revoked*. Consequently, the majority of ICL entries are not integrity-checked during normal Trust Authority CA operations.

If you want to check the integrity of the entire CA database, you must use the CA integrity check tool. This executable reads and performs integrity checking on every entry in the ICL. It does not change the protection key but it still requires that the CA be shut down while it is running.

To check the integrity of the CA server database, follow these steps:

1. Go to the machine where the CA server is installed.
2. Stop the Trust Authority system using the procedure described in "Start and stop the server components" on page 7.
3. Run the CA database integrity checking tool by entering the following command (which includes your Trust Authority installation path) on the Trust Authority command line, for example:
   - On AIX when you are using the default Trust Authority installation path:
     
     ```
     CAIntegrityCheck in /usr/lpp/iau/bin
     ```
   - On Windows NT when you are using the default Trust Authority installation path:
     
     ```
     CAIntegrityCheck in "c:\Program Files\IBM\Trust Authority\bin"
     ```

   The tool prompts you for the Trust Authority Control Program password.
4. Enter the Trust Authority Control Program password.
5. Start the Trust Authority system using the procedure described in "Start and stop the server components" on page 7.

**Request a CA certificate using cross-certification**

You can request a CA certificate from another CA on behalf of the Trust Authority CA using the cross-certification trust model. Cross-certification allows CAs that trust each other to agree to accept one another’s certificates as proof of authenticity. Although cross-certification between CAs can be in both directions, in Trust Authority only one-way cross-certification requests are supported.

**Note:** Trust Authority supports cross-certification only between CAs that adhere to the PKIX Certificate Management Protocol (CMP).

To cross-certify with another CA, use the Trust Authority CA Certification utility (CACertRq). This utility is a command line tool. When starting this tool, you may specify options on the standard Name Constraints certificate extension. A lowercase letter option specifies inclusion in the permitted subtree list. An uppercase letter option specifies inclusion in the excluded subtree list. It is usual for excluded subtrees to be specified only for address types for which a permitted subtree list is also specified.
The following sections provide the steps you need to take and examples for obtaining a CA certificate using the cross-certification model.

**Steps for cross-certification**

To request a CA certificate using the cross-certification model, follow these steps:

1. On behalf of the CA requesting the cross-certification, perform the following steps according to the instructions on enrolling using the Web in the User's Guide:
   a. Perform preregistration, specifying that the request is for a CA.
      When completing the enrollment form, be sure that you enter a distinguished name that matches that of the requesting CA.
   b. Check the status of the enrollment request.
   c. Follow the note at the end of the section to save the preregistration file.

2. Transport the preregistration file to the machine where the CA requesting the cross-certification resides.

3. Log in to Trust Authority as the user that installed the system on the machine where the CA requesting cross-certification resides.

4. Execute the CaCertRq command from the command line if you are on AIX, or from the DOS prompt if you are on Windows NT.

   When you run the CaCertRq command, be sure to provide the absolute path and name of the preregistration file. Refer to "CA Certification utility" on page 96 for the command syntax and parameter descriptions of the command.

   **Note:** When you specify a directory path that contains embedded spaces as the parameter of a command, you must enclose the path in double quotation marks (" "). For example, for the CaCertRq command, you would specify a `-r preregistrationpath` parameter that contained embedded spaces as follows:

   ```
   CaCertRq -d .companyA.com -r "c:\Program Files\IBM\Trust Authority\ccprereg.reg" -P 1835 -W Secure99
   ```

**Specify an IP address mask**

You can request a cross-certified CA certificate that restricts the validity of the certificate to a range of IP addresses used by your company.

In the following example, the operating system is AIX, the range of IP addresses used by your company is 9.0.0.0 to 9.255.255.254, the preregistration path and filename is /tmp/ccprereg.reg, and the password is Secure99. This command places the specified range of addresses in the permitted subtree list:

```
CaCertRq -i 9.0.0.0/255.0.0.0 -r /tmp/ccprereg.reg -P 1835 -W Secure99
```

**Specify DNS addresses**

You can request a cross-certified CA certificate that restricts the validity of the certificate to a group of DNS addresses.

In the following example, you are running Windows NT, all Trust Authority CA hosts have DNS addresses ending in .companyA.com. This command places the specified DNS addresses in the permitted subtree list:

```
CaCertRq -d .companyA.com -r c:\temp\ccprereg.reg -P 1835 -W Secure99
```
Note: If a DNS address starts with a period, all hosts ending with that substring (including the ".") are intended to be placed in the subtree list. If it does not start with a period, only the host matching that string is intended.

For example, the constraint ".companyA.com" matches us.companyA.com, vnet.companyA.com, and w3.software.companyA.com, but not companyA.com (nor kidcompanyA.com) itself. The constraint companyA.com matches companyA.com but not us.companyA.com or the others. Such a subtree, without a leading ".", indicates only a single possible node. Those specifications without a leading "." are primarily useful for excluded subtrees.

Specify e-mail addresses
You can request a cross-certified CA certificate that restricts the validity of the certificate to a group of e-mail addresses.

In the following example, you are running Windows NT, all Trust Authority CA hosts have e-mail addresses ending in .us.companyA.com, the preregistration path and filename is a:\ccprereg.reg, and the password is Secure99. This command places the specified e-mail addresses in the permitted subtree list:

CaCertRq -m .us.companyA.com -r a:\ccprereg.reg -P 1835 -W Secure99

You can request a cross-certified CA certificate that restricts the validity of the certificate to all but one in a group of e-mail addresses.

In the following example, you are running Windows NT, the Trust Authority CA host has the e-mail address outCA.us.companyA.com, the preregistration path and filename is a:\ccprereg.reg, and the password is Secure99. This command places the specified e-mail addresses in the excluded subtree list:

CaCertRq -m .us.companyA.com -M outCA.us.companyA.com -r a:\ccprereg.reg -P 1835 -W Secure99

Specify URIs
You can request a cross-certified CA certificate that restricts the validity of the certificate to a group of Uniform Resource Identifiers (URIs, a category of identifiers of which URLs are the most common subset).

In the following example, you are running AIX, all Trust Authority CA hosts have host names ending in .xyz.com, the preregistration path and filename is /tmp/ccprereg.reg, and the password is Secure99. This command places the specified URI in the permitted subtree list:

CaCertRq -u .xyz.com -r /tmp/ccprereg.reg -P 1835 -W Secure99

Note: The node portion (which, in a tree structure, is a point at which subordinate items of data originate) of the URI is subject to the same rules as those described in “Specify DNS addresses” on page 23, unless it contains an IP address. In that case it is considered an exact match.

Specify Directory entries
You can request a cross-certified CA certificate that restricts the validity of the certificate to a group of Directory entries with the same relative distinguished names (RDNs).

In the following example, you are running AIX, all Trust Authority CA hosts have RDNs that match the supplied RDNs, /C=US/O=companyA/OU=departmentB, the preregistration
path and filename is /tmp/ccprereg.reg, and the password is Secure99. This command places
the specified RDN in the permitted subtree list:

```
CaCertRq -n "/C=US/O=companyA/OU=departmentB" -r /tmp/ccprereg.reg -P 1835
-W Secure99
```

**Request a CA certificate using the hierarchy model**

You can request a CA certificate from another CA using the hierarchy trust model.
Companies often request CA certificates from established and highly secure CAs to add
credibility to their CA. CAs trust the CAs that are above them in the hierarchy and accept
the certificates of those CAs as proof of authenticity. You can request a CA certificate from
either a third party CA or from a Trust Authority CA.

**Note:** Trust Authority supports hierarchical certification only between CAs that adhere to
the PKIX Certificate Management Protocol (CMP) and PKCS #7 and PKCS #10
protocols.

To set up a CA hierarchy, use the Trust Authority CA Certification utility (CACertRq). When
starting this tool, you must specify at least one option on the standard Name Constraints
certificate extension (refer to "Certificate extensions" on page 59 for a detailed discussion of
certificate extensions). A lowercase letter option specifies inclusion in the permitted subtree
list. An uppercase letter option specifies inclusion in the excluded subtree list. It is usual for
excluded subtrees to be specified only for address types for which a permitted subtree list is
also specified.

**Note:** If you are planning to download the hierarchical CA certificates into Netscape
Communicator as part of your overall hierarchy solution, then you should use the "-m .."
flag with the CA Certification utility in order to exclude the Name Constraints
extension from the resulting CA certificate. You should not use any of the Name
Constraints command line options other than "-m .." (in other words, not -i/I, -d/D,
-u/U, -n/N, or -m/-M) with the CA Certification utility. The syntax for such a
command looks like this:

```
CaCertRq -m .. -h -r preregistrationpath -P 1835 -W password
```

When you run the CA Certification command, the hierarchy that you establish is stored in
two files called CAChian.P7b and RAsCopyOfCACChain.P7b.

The following sections provide the steps you need to take and examples for obtaining a CA
certificate using the hierarchy model.

**Steps for requesting a CA certificate in an hierarchy**

To request a CA certificate using the hierarchy model, follow these steps:

1. On behalf of the CA requesting the cross-certification, perform the following steps
   according to the instructions on enrolling using the Web in the *User’s Guide*:

   a. Perform preregistration, specifying that the request is for a CA.
      When completing the enrollment form, be sure that you enter a distinguished name
      that matches that of the requesting CA.

   b. Check the status of the enrollment request.

   c. Follow the note at the end of the section to save the preregistration file.

2. Transport the preregistration file to the machine where the CA requesting the
   hierarchy-based CA certificate resides.
3. Log in to Trust Authority as the user that installed the system on the machine where the
   CA requesting the certificate resides.

4. Execute the CaCertRq command from the command line if you are on AIX, or from the
   DOS prompt if you are on Windows NT.

   When you run the CaCertRq command, be sure to provide the absolute path and name of
   the preregistration file. Refer to "CA Certification utility" on page 96 for the command
   syntax and parameter descriptions of the command.

   **Note:** When you specify a directory path that contains embedded spaces as the
   parameter of a command, you must enclose the path in double quotation marks (" "). For example, for the CaCertRq command, you would specify a -r
   `preregistrationpath` parameter that contained embedded spaces as follows:
   ```
   CaCertRq -d .companyA.com -h -r "c:\Program Files\IBM\Trust
   Authority\ccprereg.reg" -P 1835 -W Secure99
   ```

**Specify an IP address mask**
You can request a CA certificate that inherits the CA hierarchy of another CA and restricts
the validity of the certificate to a group of IP addresses used by your company.

   In the following example, you are running AIX, the range of IP addresses used by your
   company is 9.0.0.0 to 9.255.255.254, the preregistration path and filename is
   /tmp/ccprereg.reg, and the password is Secure99. This command places the specified range
   of addresses in the permitted subtree list:
   ```
   CaCertRq -i 9.0.0.0/255.0.0.0 -h -r /tmp/ccprereg.reg -P 1835 -W Secure99
   ```

**Specify DNS addresses**
You can request a CA certificate that inherits the CA hierarchy of another CA and restricts
the validity of the certificate to a group of DNS addresses.

   In the following example, you are running Windows NT, all Trust Authority CA hosts have
   DNS addresses ending in .companyA.com, the preregistration path and filename is
   a:\ccprereg.reg, and the password is Secure99. This command places the specified DNS
   addresses in the permitted subtree list:
   ```
   CaCertRq -d .companyA.com -h -r a:\ccprereg.reg -P 1835 -W Secure99
   ```

   **Note:** If a DNS address starts with a period, all hosts ending with that substring (including the ".") are intended to be placed in the subtree list. If it does not start with a period, only the host matching that string is intended.

   For example, the constraint ".companyA.com" matches us.companyA.com,
   vnet.companyA.com, and w3.software.companyA.com, but not companyA.com (nor
   kidcompanyA.com) itself. The constraint companyA.com matches companyA.com but
   not us.companyA.com or the others. Such a subtree, without a leading ".", indicates
   only a single possible node. Those specifications without a leading "." are primarily
   useful for excluded subtrees.

**Specify e-mail addresses**
You can request a CA certificate that inherits the CA hierarchy of another CA and restricts
the validity of the certificate to a group of e-mail addresses.
In the following example, you are running AIX, all Trust Authority CA hosts have e-mail addresses ending in .us.companyA.com, the preregistration path and filename is /tmp/ccprereg.reg, and the password is Secure99. This command places the specified e-mail addresses in the permitted subtree list:

CaCertRq -m .us.companyA.com -h -r /tmp/ccprereg.reg -P 1835 -W Secure99

You can request a CA certificate that inherits the CA hierarchy of another CA and restricts the validity of the certificate to all but one in a group of e-mail addresses. In the following example, the Trust Authority CA host has the e-mail address outCA.us.companyA.com, the preregistration path and filename is /tmp/ccprereg.reg, and the password is Secure99. This command places the specified e-mail address in the excluded subtree list:

CaCertRq -m .us.companyA.com -M outCA.us.companyA.com -h -r /tmp/ccprereg.reg -P 1835 -W Secure99

Specify URIs

You can request a CA certificate that inherits the CA hierarchy of another CA and restricts the validity of the certificate to a group of Uniform Resource Identifiers (URIs, a category of identifiers of which URLs are the most common subset). In the following example, you are running Windows NT, all Trust Authority CA hosts have URIs ending in .xyz.com, the preregistration path and filename is a:\ccprereg.reg, and the password is Secure99. This command places the specified URI in the permitted subtree list:

CaCertRq -u .xyz.com -h -r a:\ccprereg.reg -P 1835 -W Secure99

Note: The node portion of the URI is subject to the same rules as those described in “Specify DNS addresses” on page 23, unless it contains an IP address. In that case it is considered an exact match.

Specify Directory entries

You can request a CA certificate that inherits the CA hierarchy of another CA and restricts the validity of the certificate to a group of Directory entries with the same relative distinguished names (RDNs). In the following example, you are running AIX, all Trust Authority CA hosts have RDNs that match the supplied RDNs, /C=US/O=companyA/OU=departmentB, the preregistration path and filename is /tmp/ccprereg.reg, and the password is Secure99. This command places the specified RDN in the permitted subtree list:

CaCertRq -n "/C=US/O=companyA/OU=departmentB" -h -r /tmp/ccprereg.reg -P 1835 -W Secure99

Note: In Trust Authority, the following format is used for DNs:
/C=country/O=organization/OU=organizational_unit/CN=common_name

Exclude Name Constraints

If you are planning to download the hierarchical CA certificates into Netscape Communicator as part of your overall hierarchy solution, then you should use the "-m .." flag with the CA Certification utility in order to exclude the Name Constraints extension from the resulting CA certificate. In the following example, you are running Windows NT, using the default Trust Authority installation path, preregistration file name, and system password. For example, an instance of such a command might look like this:

CaCertRq -m .. -h -r "c:\Program Files\IBM\Trust Authority\ccprereg.reg" -P 1835 -W Secure99

Make updates after changing the CA hierarchy

Any time you change who signs your certificates, by making changes to your internal or upward hierarchy, you need to apply those changes to each of your RAs. When you change
your hierarchy, the new hierarchy structure will be written out to both the CACchain.p7b file and the RAsCopyOfCACChain.p7b file. The CACchain.p7b file will be ready to be used immediately, but the new RAsCopyOfCACChain.p7b file needs to be transferred to each RA you have running under that CA. The new RAsCopyOfCACChain.p7b file has to be transferred to each RA via a secure, off-line method of your choosing. You are responsible for the security and integrity of this file while being transferred. Once the old version of the RAsCopyOfCACChain.p7b file has been replaced on each RA system running under the CA that had its hierarchy changed, any certificates you have issued will accept the new hierarchy and will function as normal.

**Note:** If anyone tries to use a certificate between the time you change your hierarchy and you update the RAsCopyOfCACChain.p7b file on each of your RAs, the certificate authentication will fail. It is very important that you try to minimize this gap as much as possible.

**Grant a CA certificate to a third party CA**

Trust Authority allows you to grant a CA certificate to a third party CA. By granting or signing the third party CA certificate, you then place them in a downward hierarchy position form the Trust Authority CA. To grant a CA certificate to a third party CA, complete the steps below.

1. Obtain a request file from the CA requesting the certificate (subject CA). This request file for a certificate must be a PKCS #10 request in binary format. This request file will have to be transferred to you via a secure off-line method of your choosing. You and the third party who delivers the file to you are responsible for the security and integrity of this file while being transferred. Store the request file in a directory of your choosing.

   **Note:** If the requesting third party cannot deliver the file in binary format, you must convert the b64 file to binary before proceeding. This can be done by any number of simple tools found on the Internet.

2. Next you will run the CA Certification utility (CACertRq) and generate a response file for the third party CA. The response file will be a PKCS #7 response in either binary or base64 depending how you run the command. Run the CACertRq utility using the "-H" flag followed by the full path of the request file, and either the "-e" flag to output the response file in binary or the "-E" flag to output the file to base64.

   **Note:** You must use either the "-e" flag or the "-E" flag or the command will not execute.

   For example an instance of this command might look like this:
   
   CaCertRq -H "/usr/lpp/iau/request.P10" -e
   
   The response file will be written to the same directory where the request file is located.

3. The response file will have to be transferred to the third party via a secure off-line method of your choosing. You and the third party who receives the file are responsible for the security and integrity of this file while being transferred. At this point it up to the administrator of the third party CA to process the response file to make it take affect.

   Any certificates that you grant to a third party CA are not accessible via the RA Desktop like other certificates. The certificates for third party CAs are stored in the ICL Database and can only be revoked by accessing the ICL database and manually revoking the certificate.
Check the CA server logs

The CA server logs record all transactions with the CA server. You can check the CA server logs at the following location.

Table 5. CA server logs

<table>
<thead>
<tr>
<th>AIX default file location</th>
<th>Windows NT default file location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lpp/iau/etc/TrustAuthority</td>
<td>c:\Program Files\IBM\TrustAuthority\etc\TrustAuthority</td>
<td>There is one log filename stem, caSS.log. Each time a new log is created, the .xnnnnnn extension changes incrementally.</td>
</tr>
</tbody>
</table>

Check the status of the CA server

Depending on your environment, you can check the status of the CA Server by performing one of the following sets of procedures:

- **On AIX:**
  1. Log in to AIX as root.
  2. Check the process table and look for this process:
     `irgAutoCa`
     
     If you see this process, go to step 3. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.
  3. Go to the bin directory which has the following default path:
     `/usr/lpp/iau/bin`
  4. Check that port 1835 is responding successfully by entering this command:
     `ServerControl -i -c -k CA -n server -p 1835 -l "logfile"`
     where `server` is the name of the CA Server associated with port 1835, and `logfile` is the name of the log file where you want to record the results of the ServerControl command.
     
     If the port is responding successfully, the system displays the following message:
     `Verified that the [CA] service is running on Server: server, port: 1835.`
     where `server` is the server associated with port 1835.

- **On Windows NT:**
  1. Log into Windows NT as the system administrator.
  2. Start the Task Manager by pressing the **Ctrl**, **Alt** and **Delete** keys.
  3. Select the **Processes** tab.
  4. Look for the process `irgAutoCA.exe`.
     
     If you see this process, go to step 5. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.
  5. At the MS DOS prompt, go to the bin directory, which has the following default path:
     `c:\Program Files\IBM\Trust Authority\bin`
  6. Check that port 1835 is responding successfully by entering this command:
ServerControl -i -c -k CA -n server -p 1835 -I "logfile"

where `server` is the name of the CA Server associated with port 1835, and `logfile` is the name of the log file where you want to record the results of the ServerControl command.

If the port is responding successfully, the system displays the following message:
Verified that the [CA] service is running on Server: `server`, port: 1835.

where `server` is the server associated with port 1835.

**Activate CRL distribution points**

You can activate the CRL distribution points by performing the following steps:

1. Log into your operating system as administrator.
2. Stop the Trust Authority system.
3. Start the iniEditor and load the jonahca.ini configuration file.
4. Select and expand the **General** section and select the **CertperDP** parameter.
5. In the displayed editing field, change the value to a positive, non-zero integer.
6. While in the **General** section, select the **CRLDistName** parameter.
7. In the displayed editing field, change the value to a meaningful name; you must keep the '%d' in the name.
8. Save the file and exit the iniEditor.
9. Start the Trust Authority system.

**Administer the RA server**

This section provides operational and administrative procedures for the Trust Authority registration authority (RA) server. The RA server handles communications between the RA Desktop and the CA server. It resides, along with its DB2 database instance, on a local machine.

The tasks that you are likely to perform to administer the RA server are as follows:

- Use the Add RA User utility (add_rauser) to add users to the system as registrars (RA administrators).
- Use the Enable RA Database Encryption utility (iauEnableRADBSec) to enable encryption of the RA database.
- Use the IniEditor to make these changes to the jonahra.ini configuration file:
  - Change the RA server listener port.
  - Change the RA polling interval.
  - Change the RA retry interval.
  - Change the RA settings for communications with the Directory.
- Check the RA server logs.
- Check the status of the RA server.
Add registrars

You are responsible for adding new registrars to the system. A user that has been granted registrar status can perform the functions and access the restricted resources defined in one of the RA profiles specified in a Trust Authority registration facility configuration file. For you to add a new registrar, you must yourself be an RA administrator. This section provides the procedures for first, obtaining the RA administrator credentials, and then adding an RA administrator.

Add yourself as the first registrar

To add yourself to the Trust Authority system as a registrar, follow these steps:

1. If you do not already have one, apply for and obtain a browser certificate. Refer to the User’s Guide for instructions on doing this.

2. Record one of the following identification codes:
   - The request ID generated for you during browser enrollment. You may still have this if you recently enrolled for a browser certificate.
   - The credential unique user identification (credential UUID) generated as a result of successful browser enrollment.

   You can obtain your credential UUID by following these steps:
   a. Start a DB2 command line interactive session. Depending on your operating system, use one of the following sets of procedures:
      - On AIX:
        1) At the command prompt, enter the `su` command to switch to the administration account of the user that installed Trust Authority.
        2) Enter the password for that Trust Authority user.
        3) Enter `db2` to start the DB2 session.
        4) At the DB2 command prompt, connect to the database by entering: `connect to pkrfdb`
           In this case, the default registration database, pkrfdb, is assumed.
      - On Windows NT:
        1) On your Windows toolbar, select Start → Programs → DB2 for Windows NT → Command Line Processor
        2) At the prompt, enter `DB2`.
        3) Enter `connect to pkrfdb`.
           In this case, the default registration database, pkrfdb, is assumed.
   b. At the DB2 command prompt, enter an SQL command such as the following:
      ```sql
      select last_name, first_name, credential_uuid, created_on from requests 
      where last_name = 'yourlastname' and first_name = 'yourfirstname' and 
      profile_name like '%BrowserCert%'
      ```
      If successful, the system returns the requested last name, first name, credential UUID, and timestamp for all matching records. In this case, the returned information is from the request record for your approved browser certificate.
   c. Enter `quit` to end the DB2 session.
3. Follow the instructions in “Add a registrar” starting with step 3 to finish adding yourself to the system as the first registrar.

**Add a registrar**

Depending on the procedures set forth by your organization, a request to add a registrar could come in a variety of ways, from an informal telephone call to a more formalized application process. In most organizations, in order to be eligible for registrar status, a user must first have been issued an SSL browser certificate. Users can request browser certificates by following the standard, browser-based enrollment procedures outlined in the Trust Authority *User's Guide*.

Trust Authority provides the command-line utility, `add_rauser`, for adding registrars. The syntax for this command is documented in “Add RA User utility” on page 98.

After you have received a request, follow these steps to add a registrar to the system:

1. Check the status of the prospective registrar’s SSL browser certificate request.
   
   To do this, access the RA Desktop and follow the procedures documented in the *Registration Authority Desktop Guide* for submitting a query and viewing the query results.

2. If a certificate has been issued, obtain the credential UUID associated with the certificate.
   
   **Note:** Alternatively, you can use the request ID generated during a successful browser enrollment for a certificate.

   To obtain the credential UUID, locate the database record associated with the approved certificate and view the attributes of the record. You are looking for the credential UUID attribute which you will use as one of the parameters on the `add_rauser` command.

3. Depending on your environment, do one of the following:

   - **On AIX:**
     
     a. Go to the install root directory. If you use the default install root, the command looks like this:
     
        ```
        cd /usr/lpp/iau/bin
        ```
     
     b. Enter a command such as the following:
     
        ```
        ./add_rauser /usr/lpp/iau/pkrf/etc/domain.cfg YourDomain aBcXyZ== RAUser
        ```
     
        where:
        
        - `/usr/lpp/iau/pkrf/etc/domain.cfg` is the Trust Authority default install root path and domain configuration file.
        - `YourDomain` is the default registration domain.
        - `aBcXyZ==` is an example of either a request ID or credential UUID.
        - `RAUser` is the optional access profile.

   - **On Windows NT:**
     
     a. Go to the install root directory. If you use the default install root, the command looks like this:
cd "c:\Program Files\IBM\Trust Authority\bin"

b. Enter a command such as the following:
   add_rauser "c:\Program Files\IBM\Trust Authority\pkrf\etc\domaincfg"
   "YourDomain" aBcXyZ== RAUser

   where:
   • c:\Program Files\IBM\Trust Authority\pkrf\etc\domain.cfg is
     the Trust Authority default install root path and domain configuration file.
   
   **Note:** When you specify a directory path that contains embedded spaces as
   the parameter of a command, you must enclose the path in double
   quotation marks (" ").
   
   • YourDomain is the default registration domain.
   
   • aBcXyZ== is an example of a request ID or a credential UUID.
   
   • RAUser is the optional access profile.

   When the command has completed, the system displays a success or failure message.

### Enable encryption for the RA database

By default, information stored in the registration facility database is not encrypted. After
configuring the Trust Authority system, you can enable database encryption by running the
iauEnableRADBSec program. When you run iauEnableRADBSec, the program examines
the raconfig.cfg file to see whether database encryption is already turned on. If it is not enabled,
the iauEnableRADBSec program enables it; if it is enabled, the program issues a message
and terminates.

The iauEnableRADBSec program updates the RADATABASE and afservice.RADB entries
in the raconfig.cfg file with the generated encryption keys. It then encrypts and Base
64-encodes the values for these entries.

**Note:** A backup copy of the raconfig.cfg file will be created in the etc subdirectory of the
virtual root installation directory configured for your registration domain (the value
specified for the Root installation directory in the Setup Wizard). The backup copy is
named raconfig.cfg.ERADSBKUP.

Depending on your operating system, to enable RA database encryption, perform one of the
following procedures:

- **On AIX:**
  1. Log in as root:
     
     ```
     su - root
     ```
  2. Change to the bin subdirectory of your installation directory. This example uses the
default Trust Authority installation path:
     
     ```
     cd /usr/lpp/iau/bin
     ```
  3. Run the iauEnableRADBSec program as follows. This example uses the default Trust
Authority registration domain name and installation directory:
     
     ```
     ./iauEnableRADBSec -d YourDomain -r /usr/lpp/iau
     ```

- **On Windows NT:**
  1. Log in as the Trust Authority configuration user (typically cfguser).
2. Change to the bin subdirectory of your installation directory. This example uses the default Trust Authority installation path:
   ```bash
   cd "c:\Program Files\IBM\Trust Authority\bin"
   ```

3. Run the iauEnableRADSec program as follows. This example uses the default Trust Authority registration domain name and installation directory:
   ```bash
   iauEnableRADSec -d YourDomain -r "c:\Program Files\IBM\Trust Authority"
   ```

### Change the RA server listener port

The RA server listener port is where the RA listens for PKIX messages. To change the value of this port, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to "Start and stop the server components" on page 7 for instructions.
3. Start the IniEditor and load the jonahra.ini configuration file. If necessary, refer to "Use IniEditor to change configuration files" on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.
4. Select and expand the `Transport` section and select the `TCPPort` parameter.
5. In the displayed editing field, change the value of the port number.
6. Save the file and exit the program.
7. Restart the IniEditor and load the jonahca.ini configuration file (this file could be on a local or remote machine, depending on your installation).
8. Select the `URLs` section.
9. In the displayed editing field, change the value of the port number.
10. Save the file and exit the program.
11. Start the Trust Authority system.

### Change the RA polling interval

The RA polling interval is the amount of time, in seconds (s), minutes (m), or hours (h), between interrogations of the workflow queue by the RA server. Elements on the queue whose dispatch times have elapsed are dispatched for processing. To change the polling interval, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to "Start and stop the server components" on page 7 for instructions.
3. Start the IniEditor and load the jonahra.ini configuration file. If necessary, refer to "Use IniEditor to change configuration files" on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.
4. Select and expand the `Transport` section and select the `PollInterval` parameter.
5. In the displayed editing field, change the value of the polling interval.
6. Save the file and exit the program.
7. Start the Trust Authority system.
Change the RA retry interval

The RA retry interval is the amount of time, in seconds (s), minutes (m), or hours (h), between interrogations of the CA by the RA in case the poll time sent to the RA from the CA is earlier than the RA’s current clock time. To change the RA retry interval, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the jonahra.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.
4. Select and expand the Transport section and select the RetryInterval parameter.
5. In the displayed editing field, change the value of the retry interval.
6. Save the file and exit the program.
7. Start the Trust Authority system.

Check the RA server logs

You can check the RA server logs at the locations shown in Table 6.

<table>
<thead>
<tr>
<th>AIX default file location</th>
<th>Windows NT default file location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lpp/iau/pkrf/Domains/ YourDomain/logs/</td>
<td>c:\Program Files\IBM\Trust Authority\pkrf\Domains \YourDomain\logs\</td>
<td>Log files have the name irgrasvr and the extension .log.random_number</td>
</tr>
</tbody>
</table>

Note: As shipped, the RA server does not log operational activity. To enable RA server logging at the debug level, see “Troubleshooting with debug-level messaging enabled” on page 110 for instructions and recommended levels.

Check the status of the RA server

Depending on your environment, you can check the status of the RA Server by performing one of the following sets of procedures:

- On AIX:
  1. Log in to AIX as root.
  2. Check the process table and look for the following process: irgrasvr
     
     If you see this process, go to step 3. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.
  3. Go to the bin directory, which has the following default path: /usr/lpp/iau/bin
  4. Check that port 29783 is responding successfully by entering this command:
     ServerControl -i -c -k RA -n server -p 29783 -l "logfile"
where *server* is the name of the RA server associated with port 29783, and *logfile* is the name of the log file where you want to record the results of the ServerControl program.

If the port is responding successfully, the system displays the following message:

```
Verified that the [RA] service is running on Server: server, port: 29783.
```

where *server* is the server associated with port 29783.

- **On Windows NT:**
  1. Log into Windows NT as the system administrator.
  2. Start the Task Manager by pressing the **Ctrl**, **Alt** and **Delete** keys.
  3. Select the **Processes** tab.
  4. Look for the process irgrasvr.exe.

     **If you see this process, go to step 5. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.**
  5. At the MS DOS prompt, go to the bin directory, which has the following default path:

     ```c:\Program Files\IBM\Trust Authority\bin```

  6. Check that port 29783 is responding successfully by entering this command:

     ```ServerControl -i -c -k CA -n server -p 29783 -l "logfile"```

     where *server* is the name of the RA Server associated with port 29783, and *logfile* is the name of the log file where you want to record the results of the ServerControl command.

     If the port is responding successfully, the system displays the following message:

     ```
     Verified that the [RA] service is running on Server: server, port: 29783.
     ```

     where *server* is the server associated with port 29783.

**Change RA settings for communications with the Directory**

The RA server communicates with the IBM SecureWay Directory server to help manage the registration process. The following values can be changed in the RA server configuration file to affect the way the RA communicates with the Directory:

- Hostname and port of the Directory server

**Change RA settings to reflect hostname and port of the Directory server**

In order to communicate effectively with the Directory, the RA needs the correct hostname and the correct port on which the Directory listens. To change the hostname and port that the RA expects to use for its Directory communications, follow these steps:

1. Log into your operating system as a system administrator.

2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the jonahra.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.

4. Select and expand the LDAP section and select the Server1 parameter.

5. In the displayed editing field, change the values of the of the hostname and the port number.

6. Save the file and exit the program.

7. Start the Trust Authority system.

**Change RA settings to reflect the Directory server post interval**

The post interval is the time between checks by the RA to see if there is information to be posted to the Directory, such as new user certificates, or new CRLs.

To change the post interval of the RA, follow these steps:

1. Log into your operating system as a system administrator.

2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.

3. Start the IniEditor and load the jonahra.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT file locations.

4. Select and expand the LDAP section and select the PostInterval parameter.

5. In the displayed editing field, change the value of the PostInterval parameter.

6. Save the file and exit the program.

7. Start the Trust Authority system.

**Enabling or disabling Trust Chain Delivery**

When you install the CA Server, Trust Chain Delivery is disabled by default. Chaining can be turned on or off by changing the value of a parameter in the jonahra.ini file. This file can be changed with the IniEditor tool (see “Use IniEditor to change configuration files” on page 9 for instructions about using IniEditor). The parameter that must be changed is TrustChainDelivery. The TrustChainDelivery parameter is in the [General] section of the jonahra.ini file. The acceptable values for this parameter are as follows:

0 Off
1 On

If you change the jonahra.ini file, you must restart the RA server for the changes to take effect.

**Installing Multiple RAs**

Trust Authority version 3.7.0 supports the installation of multiple RA (MRA) servers under one Certificate Authority (CA). This allows you to set up different RA servers for different purposes. For example you may have a parent company that needs to issue different kinds of certificates for the different subsidiary companies under the parent company. This is possible by setting up multiple RAs. Each RA utilizes the same CA, but each RA can issue certificates based on a different policy. MRAs must be installed on a different server than
the original CA and RA. The installation of multiple RAs is very similar to the installation of the first RA. To install additional RAs, read and follow the instructions in the following sections.

Note: You can only install a MRA after the initial CA/RA installation.

Install Trust Authority on the MRA system
Trust Authority can be installed in any manner you want and all configuration options are allowed. Follow the instructions outlined in the Up and Running Guide. However, stop and return to these instructions once you have done the initial installation. You should not run any of the post installation or post configuration utilities at this time.

Create the mraConfig.dat file for the new MRA
Each installation of an MRA requires a mraConfig.dat file. You can create a new data file for each MRA you install, or you can use the same file each time. The data file contains the location, certificate, and certificate fingerprint of the CA. You can generate the data file on either the CA on or the initial RA by using the createNewRA utility. See “Command-line utilities” on page 96 for more information about the createNewRA utility. The createNewRA command creates the data file based on the mraTemplate.ini file. The name and destination of the output file may be specified by using the “-f” command line flag. Otherwise, the data file will be called mraConfig.dat and will be stored in the current working directory.

Transmit the mraConfig.dat file to the MRA server
Once the mraConfig.dat file is created, you must transfer the file via a secure off-line method to the system where you will be setting up the MRA. The contents of the file should be protected from tampering while in transit. However, because the data file contains the CA’s certificate, the CA should not be used to sign the file. It is your responsibility to protect the integrity of the file while being transferred to the MRA machine. The mraConfig.dat file should be stored in the following directory: /usr/lpp/iau/bin.

Run CfgPostInstall on the MRA server
After you get the mraConfig.dat file to the system where you will be setting up a new MRA, run the CfgPostInstall command with the “-m” flag followed by the name of the mraConfig.dat file. See “Command-line utilities” on page 96 for more information about the CfgPostInstall command. CfgPostInstall will use the information in the mraConfig.dat file to customize the panels in the Configuration Wizard and enable the creation of an MRA.

Run the Configuration Wizard on the MRA server
The next step is to run the Configuration Wizard on the MRA machine. To do this, you need to bring up a browser and enter the address of the MRA server. For example, https://webserver.tivoli.com/ The page that is displayed in the browser will be the same page used for the configuration of the primary CA/RA. Follow the instructions on the page to download the swingall.jar file, and then click the link to the Configuration Wizard. The Configuration Wizard will automatically detect a first stage MRA install, and provide you an opportunity to adjust and confirm the parameters of the MRA. When the Configuration Wizard is finished, it will create a pre-enrollment file for the MRA. The Configuration Wizard will display the path where the pre-enrollment file was stored.

Transmit the pre-enrollment file to the CA server
The pre-enrollment file contains information about the MRA you are setting up. This information will be processed by the CA and used to authorize the MRA. You must you must transfer the pre-enrollment file via a secure off-line method to the CA server. It is your
responsibility to protect the integrity of the file while being transferred to the CA. Store the file in the following directory on the CA server: /usr/lpp/iau/bin.

**Authorize the MRA enrollment on the CA server**
Once you have transferred the pre-enrollment file to the CA server, the CA server must approve the MRA. This is done using the `authRAEnrollment` command. This command requires the "-f" flag followed by the name of the pre-enrollment file. See "Command-line utilities" on page 96 for more information about the `authRAEnrollment` command. This will authorize the certificate from the MRA and prepare the CA to start accepting requests from the MRA.

**Run the Configuration Wizard again on the MRA server**
The final step is to run the Configuration Wizard on the MRA server again. If your browser is still open to the configuration page, you need to reload the browser by holding down the "Shift" key and clicking reload. This will reload the page from the server and not from the cache. You can also close the browser and reopen it to the configuration page. Click the link to the Configuration Wizard again. The Configuration Wizard will jump to the last panel automatically. Click "Finish" and wait for the configuration to complete. After the configuration is complete, your new MRA will be set up and ready to run.

**Administer the Audit subsystem**
To administer the Audit subsystem, you will be playing the role of the Audit Administrator. To perform some of the Audit administration tasks, you must have a password. Refer to "Change Trust Authority passwords" on page 5 if you need to change the Audit Administrator password.

The Audit server is a Trust Authority persistent process that receives audit events from Trust Authority components and writes them to an audit log. It is installed, along with its DB2 database, on either a local or remote machine. The Audit server must be located on the same machine as the CA server.

The tasks that you are likely to perform to administer the Audit subsystem are as follows:

- Use the two Trust Authority database views to view audit records in the audit database.
- Use DB2 UDB tools to search audit records or generate audit reports.
- Use the IniEditor to make these changes to the AuditClient.ini configuration file:
  - Change the port for the Audit server.
  - Change how events are sent from the Audit client.
- Use the IniEditor to make these changes to the AuditServer.ini configuration file:
  - Change the port on which the Audit server listens.
  - Change the retries for the Audit client to bind with the Audit server.
  - Change time between binding attempts.
  - Change log settings for the event, audit, trace and error logs.
- Use the Audit Archive and Sign utility to archive and sign audit log files.
- Use the Audit Integrity Check utility to check the integrity of the Audit server database and archive files.
Check the status of the Audit server.
Check the Audit server logs.

View audit records

You can view audit records in the Audit DB2 database using two Trust Authority views. Using these database views, you can see all the audit records currently stored in the database. The audit records are stored in the database table, audit_log. Many of the columns in this table are short integer codes that reference other tables. These other tables contain the full text description or names of the fields in the record (which in the database, correspond to the columns in the table). Refer to “Audit database data” on page 105 for the field names and descriptions of all the tables in the Audit database schema.

The two views are as follows:

- **viewar**
  This is the basic view that provides access to all the text descriptions without any truncations.

- **viewar_t**
  This view is the same as viewar, except that all the text columns are truncated to 40 characters.

**Note:** On Windows NT, if you query the viewar_t database by using the DB2 Control Center interface, the output of the sample contents option appears empty because of the truncation. The procedure documented in this section provides a workaround for this problem.

The views have the column names shown in Table 7.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>serial_num</td>
<td>The serial number of the audit record</td>
<td>integer</td>
</tr>
<tr>
<td>sourcetime</td>
<td>The timestamp when the audit event was generated by the client</td>
<td>timestamp</td>
</tr>
<tr>
<td>createtime</td>
<td>The timestamp when the audit record was created</td>
<td>timestamp</td>
</tr>
<tr>
<td>event</td>
<td>The name of the event</td>
<td>varchar</td>
</tr>
<tr>
<td>source</td>
<td>The audit client that generated the audit event</td>
<td>varchar</td>
</tr>
<tr>
<td>component</td>
<td>The component type of the audit client that generated the audit event</td>
<td>varchar</td>
</tr>
<tr>
<td>auth_entity</td>
<td>The entity that authorized the audit event</td>
<td>varchar</td>
</tr>
<tr>
<td>auth_role</td>
<td>The role of the entity that authorized the audit event</td>
<td>varchar</td>
</tr>
<tr>
<td>affected_entity</td>
<td>The identity of the entity affected by audit event</td>
<td>varchar</td>
</tr>
<tr>
<td>affected_entity_type</td>
<td>The type of the affected entity</td>
<td>varchar</td>
</tr>
</tbody>
</table>

Table 7. Column descriptions for the Trust Authority Audit database views
To view the Trust Authority Audit database records, follow these steps:

1. Log in as the Trust Authority user (the user who installed Trust Authority).
2. Start a DB2 command-line interactive session.

   **Note:** If you are running Windows NT, a codepage variable set during post-installation may have caused the char and varchar columns to display garbage characters in the DB2 command window. To correct this situation, enter the following commands before entering any DB2 SQL statements:

   ```
   set DB2CODEPAGE=
   db2 terminate
   ```

   Then re-start the DB2 session.

3. At the DB2 command-line, enter the following command to connect to the Audit database:

   ```
   connect to your_audit_database_name
   ```

   For example, for a database named adtdb, enter:

   ```
   connect to adtdb
   ```

4. Query the database using one of the Trust Authority views, as follows:

   - To query the viewvar view, enter this command:
     ```
     "select * from viewvar"
     ```

   - To query the viewvar_t view, enter this command:
     ```
     "select * from viewvar_t"
     ```

   The system displays all the columns described in Table 7 on page 40 for all the records in the Audit database.

5. To filter the view of the audit records, use the SQL `WHERE` clause. For example, to query records for a given date range, enter the following command:

   ```
   "select * from viewvar where sourcetime between '1999-07-01-08.00.00' and '1999-07-02-08.00.00'"
   ```

   Refer to the IBM DB2 Universal Database SQL Reference, Version 5.2 for details on the SQL `SELECT` statement. Refer to the IBM DB2 Universal Database Command Reference, Version 5.2 details on the `db2` command.

**Search audit records**

To search audit records, refer to the IBM DB2 Universal Database SQL Reference, Version 5.2 for instructions.
Change the Audit server port on the Audit client

The Audit server port is where the Audit server listens for new connections from the audit client. In order for the Audit client to communicate with the Audit server effectively, the Audit client must have the correct port for the Audit server. To change the values of the port of the Audit server in the Audit client configuration file, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the AuditClient.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the Connection section and select the Port parameter.
5. In the displayed editing field, change the value of the port for the Audit server.
6. Save the file and exit the program.
7. Start the Trust Authority system.
8. Ensure that the modified port value matches the port value of the acceptor.arg parameter in the AuditServer.ini file located on the Audit server.

Change how events are sent from the Audit client

The Audit client sends audit events to the Audit server. You can prevent some events from being sent to the Audit server by setting an audit mask in the AuditClient.ini file. Some events are mandatory, however. Refer to “Audit events” on page 102 to determine which events are mandatory. You can set the number of attempts after the first attempt that the Audit client will make to send events to the Audit server. You can also specify the amount of time to wait, in milliseconds, between retry attempts.

Specify an audit mask

To specify an audit mask, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the AuditClient.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the CA section and select the ExcludedEvents parameter.
5. In the displayed editing field, add the names of the audit events generated by the CA that you want the Audit client to withhold. The audit event names should be delimited by commas. Be sure that you specify only optional events.
6. Select and expand the RA section and select the ExcludedEvents parameter.
7. In the displayed editing field, add the names of the audit events generated by the RA that you want the Audit client to withhold. The audit event names should be delimited by commas. Be sure that you specify only optional events. Refer to “Audit events” on page 102 for names of audit events.
8. Save the file and exit the program.

9. Start the Trust Authority system.

**Change the number of retries for sending events**

To change the number of tries that the Audit client will make to send events to the Audit server after the initial attempt, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the AuditClient.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the **Connection** section and select the **Retries** parameter.
5. In the displayed editing field, change the value of the Retries parameter.
6. Save the file and exit the program.
7. Start the Trust Authority system.

**Change interval between retry attempts**

To specify the interval between retry attempts:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the AuditClient.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the **Connection** section and select the **Timer** parameter.
5. In the displayed editing field, change the value of the Timer parameter.
6. Save the file and exit the program.
7. Start the Trust Authority system.

**Change the port on which the Audit server listens**

The Audit server port is where the Audit server listens for new connections from the audit client. To change the values of the port of the Audit server in the Audit server configuration file, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the AuditServer.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the **Server Port** section and select the **acceptor.arg** parameter.
5. In the displayed editing field, change the value of the port number.
6. Save the file and exit the program.
7. Start the Trust Authority system.
8. Ensure that the modified server port value is propagated to all local and remote AuditClient.ini files. Refer to “Change the Audit server port on the Audit client” on page 42 for information.

**Change binding attempts from the Audit client to the Audit server**

To change the number of times the Audit client attempts to bind to the Audit server, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the AuditServer.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the **Server Port** section and select the **acceptor.init.retries** parameter.
5. In the displayed editing field, change the value of the binding attempts. The default is 3.
6. Save the file and exit the program.
7. Start the Trust Authority system.

**Change the interval between binding attempts**

To change the interval between attempts by the Audit client to bind to the Audit server, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the AuditServer.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the **Server Port** section and select the **acceptor.init.delay** parameter.
5. In the displayed editing field, change the value of the time, in seconds, between binding attempts. The default is 3 (the program assumes seconds).
6. Save the file and exit the program.
7. Start the Trust Authority system.

**Change log settings**

You can change settings in the AuditServer.ini configuration file for the following types of logs:

- Event log — For capturing audit events
- Audit log — For storing integrity-protected audit events in a database
- Trace log — For tracing program activity
- Error log — For error messages

**Change how audit events are captured in a file**

The event log is the file used for capturing events. You can change the configuration parameter values for the following:

- The filename and path of the event log
- Whether to append or overwrite the event log

To change settings that affect how events are captured, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to "Start and stop the server components" on page 7 for instructions.
3. Start the IniEditor and load the AuditServer.ini configuration file. If necessary, refer to "Use IniEditor to change configuration files" on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the **Event Log** section. Then:
   - To change the path or filename of the event log:
     a. Select the *event.log.filename* parameter.
     b. In the displayed editing field, change the value of the path or filename.
   - To change the flag indicating whether to append or overwrite the event log:
     a. Select the *event.log.append* parameter.
     b. In the displayed editing field, change the value of the parameter. It will be **true** for append or **false** for overwrite.
5. Save the file and exit the program.
6. Start the Trust Authority system.

**Change how audit events are logged to the database**

The audit log is a set of integrity-protected database tables used for storing audit events. The audit log contains one record per audit event. You can change the configuration parameter values for the following:

- Number of retries allowed for connecting to the audit log database
- Number of retries allowed for updating the audit log database
- Timeout value, in seconds, allowed for updating the audit log

To change settings that affect how audit events are logged, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to "Start and stop the server components" on page 7 for instructions.
3. Start the IniEditor and load the AuditServer.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.

4. Select and expand the Audit section. Then:
   - To change the number of retries allowed for connecting to the audit log:
     a. Select the audit.log.connect.retries parameter.
     b. In the displayed editing field, change the value of the parameter. It will be an integer.
   - To change the number of retries allowed for updating the audit log:
     a. Select the audit.log.update.retries parameter.
     b. In the displayed editing field, change the value of the parameter. It will be an integer.
   - To change the timeout value for updating the audit log:
     a. Select the audit.log.timeout parameter.
     b. In the displayed editing field, change the value of the parameter. It will be an interval.

5. Save the file and exit the program.

6. Start the Trust Authority system.

**Change trace log settings**

The trace log provides a record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. It is used primarily for debugging purposes. You can change the configuration parameter values for the following:

- Whether to enable or disable tracing
- The trace level
- The trace log filename and path
- Whether to append or overwrite the existing trace file

To change trace log settings, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 7 for instructions.
3. Start the IniEditor and load the AuditServer.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the Trace section. Then:
   - To enable or disable tracing:
     a. Select the trace.enable parameter.
b. In the displayed editing field, change the value of the parameter. It will be `true` for enable and `false` for disable.

- To change the trace level:
  a. Select the `trace.level.name` parameter.
  b. In the displayed editing field, change the value of the parameter. It will be a string. Refer to “Trace levels” on page 90 for a list of trace levels.

- To change the trace log filename or path:
  a. Select the `trace.log.filename` parameter.
  b. In the displayed editing field, change the value of the parameter. It will be a string.

- To change whether to append or overwrite the existing trace file:
  a. Select the `trace.log.append` parameter.
  b. In the displayed editing field, change the value of the parameter. It will be `true` for append or `false` for overwrite.

5. Save the file and exit the program.
6. Start the Trust Authority system.

**Change error log filename and path**
The error log contains error messages related to the Audit subsystem. To change the filename and path of the error log, follow these steps:

1. Log into your operating system as a system administrator.
2. Stop the Trust Authority system. If necessary, refer to “Start and stop the server components” on page 3 for instructions.
3. Start the IniEditor and load the AuditClient.ini configuration file. If necessary, refer to “Use IniEditor to change configuration files” on page 9 for instructions on starting and using the IniEditor. See Table 1 on page 10 for AIX and Windows NT configuration file locations.
4. Select and expand the Error section and select the `error.log.filename` parameter.
5. In the displayed editing field, change the value of the error.log.filename parameter.
6. Save the file and exit the program.
7. Start the Trust Authority system.

**Generate audit reports**
To generate audit reports, refer to the *IBM DB2 Universal Database SQL Reference, Version 6.1* for instructions.

**Archive and sign the audit log files**
You can archive and sign audit log records using the Trust Authority Audit Archive and Sign tool. This tool archives the Audit server database table that contains audit records. It archives them to a file using the DB2 export utility. It then signs the file. All audit records are deleted from the database after they are successfully archived, unless you specify the -n option.

**Note:** It is not necessary to shut down the Trust Authority system before running this utility.
For example, assume that you are using the default Trust Authority installation path. You want to archive the current Audit server database audit records to my.file and sign the archived audit log file, but you do not want to delete the audit records from the database. Follow these steps:

1. Log into Trust Authority as the configuration user on the machine where you installed the Trust Authority CA and Audit server components.
   The default and suggested username is **cfguser**; if you installed Trust Authority on Windows NT, your organization may have used a different value.

2. Enter the following on the command line for your environment:
   - On AIX:
     ```
     AuditArchiveAndSign -c /usr/lpp/iau/etc/TrustAuthority/AuditServer.ini -n /usr/lpp/iau/arc/my.file
     ```
   - On Windows NT:
     ```
     AuditArchiveAndSign -c "d:\Program Files\IBM\Trust Authority\etc\TrustAuthority\AuditServer.ini" -n "d:\Program Files\IBM\Trust Authority\arc\my.file"
     ```

3. When the system prompts you for a password (KeyStore PIN), enter the Audit Administrator password.
   The results are displayed as standard output. The archive file will have the .ixf extension.

Refer to [Audit Archive and Sign utility](#) on page 99 for the command syntax and parameter descriptions of the command.

### Check the integrity of the Audit server database and archive files

Trust Authority provides a command-line utility to detect tampering with the Audit server database and audit archive files. It is called the Audit Integrity Check utility.

**Note:** It is not necessary to shut down the Trust Authority system before running this utility.

You can use this utility to detect tampering with any of the following repositories of data. In each case, the system prompts you for the Audit Administrator password (KeyStore PIN).

- The Audit server database
- One or more Audit server archive files
- All the archive files under the specified directory

For example, assume that you are running AIX and using the default Trust Authority file path for your configuration files. Log into Trust Authority as the configuration user (cfguser) on the machine where you installed the Trust Authority CA and Audit server components. You can check for tampering in one of the following ways:

- To check the Audit server database:
  1. Enter this command on the AIX command line:
     ```
     AuditIntegrityCheck -c /usr/lpp/iau/etc/TrustAuthority/
     AuditServer.ini -d
     ```
  2. When the system prompts you for a password (KeyStore PIN), enter the Audit Administrator password
     The results are displayed to standard output.

- To check one or more of the Audit server archive files:
1. Enter this command on the AIX command line:
   ```bash
   AuditIntegrityCheck -c /usr/lpp/iau/etc/TrustAuthority/AuditServer.ini -a /usr/lpp/iau/arc/archive1_my.file
   ```

2. When the system prompts you for a password (KeyStore PIN), enter the Audit Administrator password.
   This command checks files with the pathname prefix /usr/lpp/iau/arc/archive1_my.file and extensions .ixf and .sig. The .ixf extension indicates a DB2-generated export format file. The .sig extension indicates a signature file generated by the Audit subsystem.

To check all the archive files under a specified directory, in this case, /usr/lpp/iau/arc/:

1. Enter this command on the AIX command line:
   ```bash
   AuditIntegrityCheck -c /usr/lpp/iau/etc/TrustAuthority/AuditServer.ini -A /usr/lpp/iau/arc/
   ```

2. When the system prompts you for a password (KeyStore PIN), enter the Audit Administrator password.
   This command checks all the files in the archive file directory with the extension .ixf.

Assume that you are running Windows NT and using the default Trust Authority file path for your configuration files. Log into Trust Authority as the configuration user on the machine where you installed the Trust Authority CA and Audit server components. You can check for tampering in one of the following ways:

To check the Audit server database:

1. Enter the following command at the DOS prompt (which in this case is c:):
   ```cmd
   AuditIntegrityCheck -c "c:\Program Files\IBM\Trust Authority\etc\TrustAuthority\AuditServer.ini" -d
   ```

2. When the system prompts you for a password (KeyStore PIN), enter the Audit Administrator password.
   The results are displayed to standard output.

To check one or more of the Audit server archive files:

1. Enter the following command at the DOS prompt (which in this case is c:):
   ```cmd
   AuditIntegrityCheck -c "c:\Program Files\IBM\Trust Authority\etc\TrustAuthority\AuditServer.ini" -a "c:\Program Files\IBM\Trust Authority\arc\archive1_my.file"
   ```

2. When the system prompts you for a password (KeyStore PIN), enter the Audit Administrator password.
   This command checks files with the pathname prefix \Program Files\IBM\Trust Authority\arc\archive1_my.file and extensions .ixf and .sig. The .ixf extension indicates a DB2-generated export format file. The .sig extension indicates a signature file generated by the Audit subsystem. In other words, the export-format files are signed and the signature of the export-format files are stored in the .sig file.

To check all the archive files under a specified directory, in this case, c:\Program Files\IBM\Trust Authority\arc:\

1. Enter the following command at the DOS prompt (which in this case is c:):
   ```cmd
   AuditIntegrityCheck -c c:\Program Files\IBM\Trust Authority\etc\TrustAuthority\AuditServer.ini -A c:\Program Files\IBM\Trust Authority\arc\
   ```
2. When the system prompts you for a password (KeyStore PIN), enter the Audit Administrator password.

This command checks all the files in the archive file directory, including those with the extensions .ixf and .sig.

Refer to "Audit Integrity Check utility" on page 100 for the command syntax and parameter descriptions of the command.

Check the status of the Audit server

Depending on your environment, you can check the status of the Audit server by performing one of the following sets of procedures:

- On AIX:
  1. Log in to AIX as root.
  2. Check the process table and look for the following process:
     ```
     java
     ```

     If you see this process, go to step 3. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.

     3. Go to the bin directory which has the following default path:
        ```
        /usr/lpp/iau/bin
        ```

     4. Check that the port (in this case, port 59998) is responding successfully by entering this command:
        ```
        ServerControl -i -c -k AUDIT -n server -p 59998 -l "logfile"
        ```

        where `server` is the name of the Audit server associated with the specified port (in this case, port 59998), and `logfile` is the name of the log file where you want to record the results of the ServerControl command.

        If the port is responding successfully, the system displays the following message:
        ```
        The Audit Server is currently running.
        ```

        where `server` is the server associated with port 59998.

- On Windows NT:
  1. Log into Windows NT as the system administrator.
  2. Start the Task Manager by pressing the **Ctrl**, **Alt** and **Delete** keys.
  3. Select the **Processes** tab.
  4. Look for the process java.exe.

     If you see this process, go to step 5. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.

     5. At the MS DOS prompt, go to the bin directory, which has the following default path:
        ```
        c:\Program Files\IBM\Trust Authority\bin
        ```

     6. Check that the port (in this case, port 59998) is responding successfully by entering this command:
        ```
        ServerControl -i -c -k AUDIT -n server -p 59998 -l "logfile"
        ```
where `server` is the name of the Audit server associated with port 59998, and `logfile` is the name of the log file where you want to record the results of the ServerControl command.

If the port is responding successfully, the system displays the following message:
The Audit Server is currently running.

where `server` is the server associated with the Trust Authority default port 59998.

**Check the Audit server logs**

The Audit server logs record all transactions with the Audit server. You can check the Audit server logs at the following location.

<table>
<thead>
<tr>
<th>Table 8. Audit server logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX default file location</td>
</tr>
<tr>
<td>/usr/lpp/iau/logs/smevents.log</td>
</tr>
<tr>
<td>/usr/lpp/iau/logs/iausmd.log</td>
</tr>
<tr>
<td>/usr/lpp/iau/logs/iausmd.err</td>
</tr>
</tbody>
</table>

**Administer the DB2 databases**

This section provides basic operational and administrative procedures for the DB2 databases. Refer to the DB2 UDB documentation for more detailed information.

Trust Authority uses the databases listed in Table 9.

<table>
<thead>
<tr>
<th>Table 9. Database locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>cfgusr</td>
</tr>
<tr>
<td>cfgusr</td>
</tr>
<tr>
<td>cfgusr</td>
</tr>
<tr>
<td>cfgusr</td>
</tr>
<tr>
<td>ldapInst</td>
</tr>
<tr>
<td>cfgusr</td>
</tr>
</tbody>
</table>

**Check the status of the DB2 databases**

Depending on your environment, you can check the status of the DB2 databases by performing one of the following sets of procedures:

- On AIX:
  1. Log into AIX as the system administrator.
2. At the command prompt, enter the `su` command to switch to the administration account of the user that installed Trust Authority.

3. Enter the password for that Trust Authority user.

4. Enter the following command:
   ```
   set DB2INSTANCE=TrustAuthority_instance
   ```
   where `TrustAuthority_instance` is the user ID of the user that installed Trust Authority.

5. Assuming the default configuration database, enter `db2 connect to cfgdb`
   If the connection is successful, the system returns messages such as the following:
   ```
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = cfgdb
   ```

6. Assuming the default CA database, enter `db2 connect to ibmdb`
   If the connection is successful, the system returns messages such as the following:
   ```
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = ibmdb
   ```

7. Assuming the default registration database, enter `db2 connect to pkrfdb`
   If the connection is successful, the system returns messages such as the following:
   ```
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = pkrfdb
   ```

8. Assuming the default audit database, enter `db2 connect to adtdb`
   If the connection is successful, the system returns messages such as the following:
   ```
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = adtdb
   ```

9. Exit back to root.

10. Use the `su` command to switch to the account of the Directory database administrator.

11. Assuming the default Directory database installed with the product, enter `db2 connect to ldapdb`
    If the connection is successful, the system returns messages such as the following:
    ```
    Database server = DB2/NT 6.1
    SQL authorization ID = TrustAuthority_instance
    Local database alias = ldapdb
    ```

12. Exit back to root.

   On Windows NT:
   1. Log into Windows NT as the system administrator.
   2. Start a DB2 command window.
   3. Enter the following command:
      ```
      set DB2INSTANCE=TrustAuthority_instance
      ```
      where `TrustAuthority_instance` is the user ID of the user that installed Trust Authority.
4. Assuming the default configuration database, enter `db2 connect to cfgdb`
   If the connection is successful, the system returns messages such as the following:
   
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = cfgdb

5. Assuming the default CA database, enter `db2 connect to ibmdb`
   If the connection is successful, the system returns messages such as the following:
   
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = ibmdb

6. Assuming the default registration database, enter `db2 connect to pkrfdb`
   If the connection is successful, the system returns messages such as the following:
   
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = pkrfdb

7. Assuming the default audit database, enter `db2 connect to adtdb`
   If the connection is successful, the system returns messages such as the following:
   
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = adtdb

8. Enter the following command:
   
   `set DB2INSTANCE=ldap_Instance`
   
   where `ldap_Instance` is ldapInst if you installed the Directory with the product.

9. Assuming the default Directory database installed with the product, enter `db2 connect to ldapInst`
   If the connection is successful, the system returns messages such as the following:
   
   Database server = DB2/NT 6.1
   SQL authorization ID = TrustAuthority_instance
   Local database alias = ldapInst

10. Exit the DB2 command window.

**Check the DB2 logs**

The DB2 logs are not used by Trust Authority from an operational standpoint. Refer to the IBM DB2 documentation for any other information about the logs.

**Administer the Directory server**

This section provides basic operational and administrative procedures for the IBM SecureWay Directory. To perform some of the Directory administration tasks, you must know the password. Refer to "Change Trust Authority passwords" on page 5 if you need to change the Directory Administrator password.

This Directory supports Lightweight Directory Access Protocol (LDAP) Directory standards. It incorporates a DB2 database, and can reside on either a local or remote machine. It can be either a preexisting server or one that is installed and configured especially for Trust Authority.
For more information about how Trust Authority interacts with the Directory, see *Using the SecureWay Directory With Trust Authority*. This document is available on the Library page of the IBM SecureWay Trust Authority Web site.

**Check the status of the Directory server**

Depending on your environment, you can check the status of the Directory Server by performing one of the following sets of procedures:

- **On AIX:**
  1. Log in to AIX as root.
  2. Check the process table and look for the following process:
     ```
     slapd
     ```
     If you see this process, go to step 3. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.
  3. Go to the bin directory which has the following default path:
     ```
     /usr/lpp/iau/bin
     ```
  4. Check that the Directory is responding successfully by entering this command:
     ```
     isdirup -h server -a port -p default_installation_path_of_Directory -t1
     ```
     where `server` is the name of the machine the Directory is running on, and `port` is the port on which the Directory server listens.
     
     If the Directory is responding successfully, the system displays the following message:
     ```
     isdirup: returning: 0
     ```

- **On Windows NT:**
  1. Log into Windows NT as the system administrator.
  2. Start the Task Manager by pressing the *Ctrl*, *Alt* and *Delete* keys.
  3. Select the **Processes** tab.
  4. Look for the process `slapd.exe`.
     If you see this process, go to step 5. If you do not see this process, refer to “Troubleshooting” on page 110 for instructions.
  5. At the MS DOS prompt, go to the bin directory, which has the following default path:
     ```
     c:\Program Files\IBM\Trust Authority\bin
     ```
  6. Check that the Directory is responding successfully by entering this command:
     ```
     isdirup -h server -a port -p default_installation_path_of_Directory -t1
     ```
     where `server` is the name of the machine the Directory is running on, and `port` is the port on which the Directory server listens.
     
     If the Directory is responding successfully, the system displays the following message:
     ```
     isdirup: returning: 0
     ```
Check the Directory server logs

You can check the Directory server logs at the locations shown in Table 10.

Table 10. Directory server logs

<table>
<thead>
<tr>
<th>AIX default file location</th>
<th>Windows NT default file location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/tmp/slapd.errors</td>
<td>c:\Directory’s installation \tmp\slapd.errors\</td>
<td>Error log for the Directory.</td>
</tr>
</tbody>
</table>

Administer the 4758 Cryptographic Coprocessor

The 4758 Cryptographic Coprocessor is an optional Trust Authority component. It is a programmable, tamper-detecting cryptographic PCI-bus card that provides high performance DES and RSA cryptographic processing. In Trust Authority, 4758 support is available only on AIX.

You can check the status of the 4758 using the csufcnm utility.

The 4758 Cryptographic Coprocessor does not generate logs. However, if the 4758 fails, a return code and a reason code usually accompany the failure. You can find a comprehensive list of the return and reason codes and their explanations in the IBM 4758 CCA Basic Services Reference and Guide document. These codes are usually sufficient for performing preliminary troubleshooting.

Refer to the IBM 4758 Cryptographic Coprocessor documentation for more information.
This chapter provides explanatory information that should help you to administer the Trust Authority system more effectively. Topics include general Trust Authority security issues as well as information about the Trust Authority certificate authorities (CAs), registration authorities (RAs), and the Audit subsystem.

**Trust Authority security**

The IBM Trust Authority security and trust model has the following characteristics:

- Trust Authority uses code signing to secure all software that performs key-based operations.
- Trust Authority uses KeyStores for storage of and access to component credentials such as keys and certificates. Credentials stored in the component KeyStores are encrypted using a key derived from a supplied password.
- For authentication purposes, inter-component communications are signed. For example, PKIX messages between the RA and the CA are signed.
- For AIX systems, Trust Authority provides 4758 hardware-based protection of signing keys.

**Access control lists**

Access control lists (ACLs) provide a mechanism for limiting the use of a specific resource to authorized users. The Trust Authority components that use ACLs are the CA, the RA, and the Directory.

The CA uses ACLs to restrict access to CA functions such as certificate creation. The RA uses ACLs to restrict access to RA functions such as request approval. The Directory uses ACLs to restrict access to various portions of the Directory that may contain sensitive information.

**Certificate authorities**

A certificate authority (CA) is the entity responsible for following an organization’s security policies and assigning secure electronic identities in the form of certificates. The certificates, signed by the CA’s own private key, contain identification and other information about the holder of the certificate.

The IBM Trust Authority CA does the following:

- It processes requests from RAs to issue, renew, and revoke certificates.
- It maintains an access control list (ACL), protected by file permission privileges, that contains the DNs of authorized RAs and administrative users.

- It maintains a list of issued certificates (ICL) in the CA database that provides information including certificate status, serial number, and CRL information. The CA communicates ICL information to the Trust Authority RA, which, in turn, publishes certificates and CRLs in the Directory. Records stored in the CA database use a message authentication code (MAC) for protection. Verification of these records is a Trust Authority option that is called integrity checking.

- It stores its own keys and certificates in the Trust Authority KeyStore.

- It generates audit records for security-relevant events and transmits them to the Audit server.

- It supports cross-certification and CA hierarchy.

- It supports generating and validating certificates that contain user-defined and user-chosen certificate extensions.

### CA Hierarchies

A CA hierarchy is a trust structure whereby one CA is located at the top of the structure and up to four layers of subordinate CAs are located below. When users or servers are registered with a CA, they receive a certificate signed by that CA and inherit the certification hierarchy of the layers above. In Trust Authority, a CA can be configured to be part of either an internal, upward, or downward hierarchy. The definition of each hierarchy type is as follows:

**Internal Hierarchy**

An internal hierarchy is a hierarchy comprised solely of Trust Authority CAs. CAs in this type of hierarchy use the PKIX Certificate Messaging Protocol (CMP).

**Upward Hierarchy**

An upward hierarchy is defined as a combination of one or more Trust Authority CAs and one or more third party CAs where the Trust Authority CA resides below the third party CAs in the trust chain. Thus the hierarchy is considered to progress upward from the Trust Authority CA. CAs in this type hierarchy use the PKCS #7 and PKCS #10 protocols.

**Downward Hierarchy**

A downward hierarchy is defined as a combination of one or more Trust Authority CAs and one or more third party CAs where the Trust Authority CA resides above the third party CAs in the trust chain. Thus the hierarchy is considered to progress downward from the Trust Authority CA. CAs in this type of hierarchy use the PKCS #7 and PKCS #10 protocols.

CAs can technically be in the middle of a hierarchy where there are third party CAs both above and below the Trust Authority CA. The Trust Authority CA can also be configured to be a self-signed CA. In this case, the CA does not participate in an hierarchy.

Trust Authority supports hierarchical certification between Trust Authority CAs that adhere to the PKIX CMP, PKCS #7, and PKCS #10 protocols. Trust Authority also supports hierarchy with third-party CAs the adhere to the PKCS #7 and PKCS #10 protocols.
Certificate extensions

X.509v3 certificate extensions provide methods for associating additional attributes with users or public keys and for managing the CA hierarchy. The X.509v3 format also allows user-communities to define private or common extensions to carry information unique to those communities.

Each extension in a certificate can be designated as critical or non-critical. A system using X.509v3 format certificates must reject a certificate if it encounters a critical extension it does not recognize; however, a non-critical extension can be ignored if it is not recognized.

There are three types of certificate extensions. They are:
- Standard extensions
- Common extensions
- Private extensions

Standard extensions

A standard certificate extension is one whose meaning and syntax are defined in ITU standard RFC 2459. All but one of these are also defined in X.509v3. A number of them are only defined within CA certificates. Trust Authority allows most standard extensions to be added to a certificate, under the rules set forth in Table 11:

Table 11. Certificate extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Usage</td>
<td>This extension indicates the purpose for which the certified public key is used. Its settings are usually defined by certificate templates.</td>
</tr>
<tr>
<td>Subject Alternative Name</td>
<td>This extension contains one or more alternative names, using a variety of name forms, for the entity that is bound by the CA to the certified public key. Certificate templates usually define which of these forms may be used in certificates using that template.</td>
</tr>
<tr>
<td>Subject Key Identifier</td>
<td>This extension identifies which public key is certified by a particular certificate. Its primary use is to distinguish keys when multiple keys are or have been certified for the same entity. Its value, in Trust Authority, is always set by the CA.</td>
</tr>
<tr>
<td>Authority Key Identifier</td>
<td>This extension identifies which public key was used by the issuer of a certificate when signing it. Its primary use is to distinguish keys when multiple keys are certified for the same issuer. Its value is always set by the CA.</td>
</tr>
<tr>
<td>Private Key Usage Period</td>
<td>This extension restricts the use of the private key that corresponds to a part of the certificate’s validity period. The current PKIX certificate profile, RFC 2459, specifies that its use is no longer recommended.</td>
</tr>
<tr>
<td>Certificate Policies extension</td>
<td>This extension contains a series of policy indicators. A policy indicator may consist only of an object identifier whose meaning must be published. Alternatively, it may consist of an object identifier along with a statement of the policy intended. The policy statement may be given by a URL from which it can be retrieved, or by a brief text statement included in the certificate.</td>
</tr>
<tr>
<td>Extension</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Issuer Alternative Name</td>
<td>This extension contains one or more alternative names, using any of a variety of name forms, for the issuer of the certificate. Its value is always set by the CA.</td>
</tr>
<tr>
<td>Subject Directory Attributes</td>
<td>This extension contains a series of extra directory attributes pertaining to the subject which are not part of the Distinguished Name. It must be non-critical.</td>
</tr>
<tr>
<td>Extended Key Usage</td>
<td>This extension contains a series of object identifiers that indicate the purpose for which the certified public key is used. Its settings are usually defined by certificate templates. Values for this extension may be defined by a user community, or may be derived from RFC 2459.</td>
</tr>
<tr>
<td>Basic Constraints</td>
<td>This extension is useful only for a CA certificate, and is present in all CA certificates generated by Trust Authority. It is always either absent or empty for any other certificate, and RFC 2459 recommends that it be absent rather than empty. In addition to indicating that a certificate is a CA certificate, it may contain a maximum certification length path, specifying how many more levels of CAs can be certified by this one. This extension must be critical.</td>
</tr>
<tr>
<td>Name Constraints</td>
<td>This extension is used only in a CA certificate. It specifies a name space within which all subject names and subject alternate names in certificates issued by the CA with this certificate, or by CAs certified by that CA, must be located. The purpose of this extension is to restrict names which may be used by certificates in this certificate’s path. Restrictions are defined in terms of permitted or excluded name subtrees. Any name matching a restriction in the excluded subtree list is invalid, regardless of information appearing in the permitted subtree list. This extension must be critical.</td>
</tr>
<tr>
<td>Policy Mappings</td>
<td>This extension is used only in CA certificates. Trust Authority does not set this extension in a certificate for which it generated the original request.</td>
</tr>
<tr>
<td>Policy Constraints</td>
<td>This extension, which is used only in a CA certificate, can be used for either of two purposes. It can prohibit policy mapping in certificates in the certification path, or it can require specific policies in such certificates.</td>
</tr>
<tr>
<td>CRL Distribution Points</td>
<td>This extension indicates where a partial CRL, containing revocation information about this certificate, can be found. It is set when a valid non-zero value is specified for the certPerDP parameter in the jonahca.ini file.</td>
</tr>
<tr>
<td>Authority Information Access</td>
<td>This extension indicates where and how certain information can be accessed about the issuer of the certificate in which the extension appears. It is not currently set in certificates created by Trust Authority.</td>
</tr>
</tbody>
</table>
**Common extensions**

Trust Authority defines a single non-standard extension as a common extension, which may be used by any organization that runs Trust Authority. It is the Host Identity Mapping extension. This extension associates the subject of the certificate with a corresponding identity on a host system.

**Private extensions**

Any client entity application written to use Trust Authority may define an extension whose identity and syntax is either private to that application or shared within a user community. Such extensions must be non-critical. If the extension is being defined for the first time, it must be assigned an object identifier and registered according to the provisions of ITU standard X.660 and ISO standard 9834-1, which are the same, and its syntax must also be registered.

**Extension request flow**

Extensions can be requested by a user, but must be validated by the CA or an RA acting on behalf of a CA. The process flows as follows:

1. The user requests an extension and supplies information for the extension, including a unique extension identifier and value, and specifies whether or not the extension is critical.
2. The extension request and information become part of a certificate request which is sent to the RA.
3. The RA or CA, while processing the certificate request, validates the extension against the organization’s certificate policy, and changes or overrides the extension request. If the extension is validated, the CA certifies it.

**Certificate revocation lists**

A certificate revocation list (CRL) is a digitally signed, time-stamped list of certificates that have been revoked by a CA. The certificates in this list should be considered unacceptable.

Certificates may be revoked when their validity period is over, or if they are suspected of being compromised. The status of the certificate is changed in the ICL. At the scheduled time, the CA creates the CRL containing the serial number and issuing CA DN of the certificate that has been revoked. Although a certificate and its information are changed in the ICL when the certificate is revoked, no revocation is final until the CRL is issued and published to the Directory.

In Trust Authority, the lifetime of a published CRL and the time period between CRL publications is set and can be modified in the CA configuration file, jonahca.ini.

You can also take advantage of CRL distribution points, thus enabling you to define and partition the number of CRLs distributed for subsequent processing. Instead of accumulating an increasingly large number of revoked certificates in one location or under one single entry in the Directory server, you can divide the list of revoked certificates over multiple distribution points. Enabling this feature is done by setting the certPerDp parameter in jonahca.ini configuration file.

**Cross-certification**

Cross-certification is a trust model whereby a certificate, that contains a public CA key associated with a corresponding private CA signature key that is used for issuing certificates, is issued by one CA to another CA. Typically, a cross-certificate is used to allow client...
systems or end entities in one registration domain to communicate securely with client
systems or end entities in another registration domain.

Although cross-certification between CAs can be in both directions, in Trust Authority only
one-way cross-certification requests are supported. Two-way cross-certification can be
performed by having each CA obtain a cross-certificate from the other. Trust Authority
supports cross-certification only between CAs that adhere to the PKIX CMP protocol.

Certification

Certification is the creation of a digital certificate for an entity or person. For Trust
Authority, certification occurs only after evaluation and approval of a certificate request by
the RA. As the result of registration, the CA issues the certificates.

Digital certificates

A digital certificate is an electronic credential issued by a trusted third party to a person or
entity. Each certificate is signed with the private key of the CA. It vouches for an individual,
business, or organizational identity.

Depending on the role of the CA, the certificate can attest to the authority of the bearer to
conduct e-business over the Internet. In a sense, a digital certificate performs a role similar
to that a driver’s license or a medical diploma. It certifies that the bearer of the
corresponding private key is authorized to conduct certain e-business activities.

A certificate contains information about the entity it certifies, whether person, machine, or
computer program. It includes the certified public key of that entity.

For Trust Authority, the type of certificate that is issued is consistent with the business
policies of your organization.

Distinguished names

A distinguished name (DN) is the unique name of a data entry stored in the Directory. The
DN uniquely identifies the position of an entry in the hierarchical structure of the Directory
called the Directory Information Tree (DIT).

This structure has a single root and an unlimited number of nodes branching off from the
root. Each node corresponds to a Directory entry identified by attributes. Expressing the DN
of an entry for a Directory depends on the syntax requirements of the Directory client and
the Directory server’s access protocol.

In Trust Authority, which uses the IBM SecureWay Directory, a DN for a Directory entry
might look like this:
/C=US/O=IBM/OU=Trust Authority/MAIL=cjsmith@vnet.ibm.com,CN=Chris Smith

where US is the country (C), IBM is the organization (O), Trust Authority is the division
within the organization (OU), cjsmith@vnet.ibm.com is the e-mail address (MAIL), and
Chris Smith is the common name (CN).

Issued certificate lists

An issued certificate list (ICL) is a complete list of the certificates that have been issued by
a CA and their current status. Certificates are indexed by serial number and state. The ICL is
maintained by the CA and stored in the CA database. This list is used to determine which certificates should be published on a certificate revocation list (CRL). The Trust Authority ICL has the following features:

- Open database connectivity (ODBC) implementation (which, in Trust Authority, is via DB2)
- Optional use of MACs on database records for integrity (ODBC-only)
- Runtime obfuscation of encryption key and MAC key
- Support for multiple connections (per thread), limited by a default maximum
- Access to the Audit logging facility

**Signing and signature validation**

To sign is to use a private digital key to generate a signature. To verify is to use the corresponding public key to validate the signature.

Trust Authority uses a PKIX-based cryptographic engine for signing and signature validation. This product enables applications to encrypt and decrypt information. Using this product, applications can verify a digital signature, retrieve a certificate from a Directory, and determine whether a certificate can be trusted. This cryptographic engine also provides seamless support for hardware-based encryption. It supports the IBM 4758 PCI Cryptographic Coprocessor or any hardware token that supports the PKCS #11 interface.

All Trust Authority software that performs cryptographic and KeyStore operations is signed by this cryptographic engine.

**Registration Authorities**

A Registration Authority (RA) is a server process that handles the administrative aspects of digital certification. In Trust Authority, an RA can approve or reject requests and initiate the revocation of certificates. The RA ensures that its organization’s business and certificate policies are applied.

**Registrars**

Registrars use the Trust Authority RA Desktop to perform administrative tasks on registration requests. For a registrar to be able to perform these tasks, you must enroll him or her as a registrar. Refer to "Add a registrar" on page 32 for instructions on doing this.

Using the RA Desktop, registrars can query the status of certificate requests, which include the following statuses: received, pending, approved, rejected, completed. Other query fields in the RA Desktop include first and last name, updated date, creation date, and certificate expiration date. When the RA Desktop displays the retrieved records of a query in a table, the registrar can take action on the results, such as approving or rejecting a registration request or revoking or changing the renewal status of a certificate.

**Registration domains**

A registration domain is set of resources, policies, and configuration options related to specific certificate registration processes. The domain name is a subset of the URL used to invoke registration operations and uniquely identifies the application. Trust Authority supports a single registration domain per Trust Authority installation.
**Multiple Registration Authorities**

Trust Authority supports Multiple Registration Authorities (MRA). When you install Trust Authority you install both an initial CA and an initial RA. MRA support enables you to install more than one RA under one CA. This enables you to set up different RA servers for different purposes. For example, you may have a parent company that needs to issue different kinds of certificates for the different subsidiary companies under the parent company. This is possible by setting up multiple RAs. Each RA uses the same CA, but each RA can issue certificates based on a different policy. Refer to "Installing Multiple RAs" on page 37 for instructions for setting up MRAs.

**Trust Chain Delivery**

Trust chain delivery is the concept of delivering the entire trust hierarchy of a CA for each certificate that is issued through an RA. Without trust chain delivery, new certificates were signed by a CA, however, the certificate contained no information about that CA’s hierarchy. Thus, when the certificate was authorized, only the fact that the certificate was signed by that CA could be determined. With trust chain delivery, each issued certificate receives the entire trust hierarchy of the signing CA. This allows the certificate to be authorized all the way up the CA’s hierarchy to the root of roots CA if it exists. Trust chain delivery can be turned off or on according to the administrator’s needs. See "Enabling or disabling Trust Chain Delivery” on page 37 for more information about enabling trust chain delivery.

When trust chain delivery is enabled, trust chains are delivered to the certificate requester when a certificate is issued. The trust chain has to be downloaded only once for each browser or server.

**4758 Cryptographic Coprocessor**

The IBM 4758 Cryptographic Coprocessor is a programmable, tamper-responding cryptographic PCI-bus card that helps to ensure the privacy and integrity of your organization’s signature key. It uses the IBM Common Cryptographic Architecture (CCA) API to provide a comprehensive set of cryptographic services including DES and RSA encryption. DES and RSA are the most widely used algorithms in commercial cryptographic systems. However, because they are so strong, the key management methodology that your organization uses becomes the more vulnerable portion of your system. When a key is compromised, the data encrypted with that key may be completely exposed. The IBM 4758 helps extend comprehensive protection to those keys by including:

- Triple-encryption of keys, using a special key stored within the dedicated hardware.
- Protection of end-to-end data communications
- Programmatically set roles and profiles which can be changed
- Use of a hardware-based random-number generator to help ensure the creation of unpredictable keys

The cryptographic processes are performed within a secure enclosure on the card. The card is designed to meet the stringent requirements of the FIPS PUB 140-1 level 4 standard. Software can run within the secure enclosure.

In Trust Authority, the 4758 provides CA signing key generation. Keys generated by 4758 are protected by the card by encrypting the key using the 4758 master key. The CA keys can be stored either in the KeyStore or the 4758.
The 4758 is an optional but recommended Trust Authority component that is available only on the AIX platform.

**Smart cards**

Smart cards are portable cryptographic devices, typically the size of a credit card. They are used for storing certificates and keys as well as for performing cryptographic operations, especially signing, without releasing the private key from the card. Not all users have access to smart card hardware, so Trust Authority provides a virtual smart card that acts as if a physical smart card is being used.

Trust Authority client users can store certificates on virtual and physical smart cards. When a user of the Trust Authority client application submits a certificate request, a private key is stored on the user’s virtual or physical smart card. When the certificate is approved, it is returned to the user. Any certificates that are stored on the smart card can be associated with a private key by using the same key identifier as the private key stored there.

The CA and RA each store a private key and corresponding self-signed certificate on their smart cards. This allows the RA to sign messages and the CA to sign CRLs and certificates without exposing their private keys outside the smart card.

Trust Authority implements the PKCS #11 interface to smart card storage.

**Auditing**

The Trust Authority Audit subsystem provides support for logging security-relevant actions, based on recommendations described in the financial industry standard X9.57. It allows you to archive and manage audit logs and to perform integrity checks of audit records.

The Audit subsystem consists of a client library and an audit server. The audit server receives audit events from authorized clients and writes the events to an integrity-protected audit log. All audit records are stored in a DB2 database. Audit events are also written to a file. Audit querying and reporting tools are provided by the DB2 UDB.

**Audit records**

Audit records are stored in an audit log DB2 database. The audit log contains one record per audit event. The audit log database is designed to aid in tamper detection, as required by the X9.57 financial industry standard. Each audit record is uniquely identified by a serial number.

**Audit events**

Trust Authority audit events are records indicating an occurrence of significance to a security-related task. The Audit server receives audit events from clients and writes them to an integrity-protected audit log.

Trust Authority audit events fall into the following categories:

- **Key management events**
  These are events related to the secure administration of keys, so that they are provided to users where and when they are needed.

- **Certificate management events**
These are events resulting from the management of digital certificates and the maintenance of information about certificates and CRLs in the Directory.

- **Security sensitive events**
  These are events resulting from the performance of security-sensitive tasks such as integrity checking, authentication, and certificate validation.

- **Audit administrator action events**
  These are events resulting from actions related to the role of Audit administrator. The role of the Audit administrator is to implement the security policies used by your organization.

- **RA events**
  Events resulting from actions performed by an RA.

Refer to [“Audit event fields” on page 102](#) for detailed reference information on Audit events.

**Audit event masks**

In Trust Authority, audit event masks provide support for controlling the audit events that are actually sent to the Audit server. Note that there is a mandatory subset of audit events that are unaffected by mask specifications.

Refer to [“Specify an audit mask” on page 42](#) for instructions on specifying an audit mask.

**Mandatory versus optional audit events**

There is a subset of audit events that is mandatory. This set is unaffected by mask specifications in the audit client configuration file. Refer to [Table 18 on page 102](#) to determine which audit events are mandatory and which are optional.

**Integrity checking**

Trust Authority provides a tool for verifying that audit records have not sustained unauthorized modification. This tool is called the Audit Integrity Check tool. Integrity checking is different from protecting the confidentiality of data, which prevents unauthorized disclosure.

The Audit server does not sign each record. Instead it computes a message authentication code (MAC) over each record and maintains a MAC for the entire database.

**Integrity sealing**

In Trust Authority, integrity sealing is the signing of archived audit log database files. Signing of these files is performed using the Trust Authority Audit Archive and Sign command line tool.

**Audit log archiving**

Trust Authority supports the archiving of the current audit log to an archive file using the Audit Archive and Sign command line tool.

**DB2 databases**

IBM DB2 Universal Database (DB2 UDB) is a relational database management system that is Web-enabled with Java support. In Trust Authority, it performs the following tasks:

- Manages information about registration processes for digital certificates.
Web servers

Web servers are server programs that respond to requests for information resources from browser programs. Trust Authority uses the IBM WebSphere set of software products to provide a trusted base for these network transactions. The WebSphere Application Server and the IBM HTTP Server help provide the infrastructure for Web server functionality in Trust Authority.

In a Trust Authority system, the Web server software resides on the same machine as the RA. It provides a secure boundary between protected programs and the users that try to access them. Using Hypertext Transfer Protocols (HTTP and HTTPS) and Secure Sockets Layer (SSL) technology, the Web server product can encrypt communications between client and server. It can also perform client authentication to prevent unauthorized access or data tampering.

IBM WebSphere Application Server

The IBM WebSphere Application Server (WAS) is a Java application server designed to facilitate the management and deployment of Web applications. WAS must be installed on a host Web server that handles HTTP requests from browsers and delivers HTML back to them using the HTTP protocol. When WebSphere is installed, it modifies the configuration of its host Web server to redirect certain requests to WebSphere for processing rather than letting the Web server handle them. WAS makes use of a Java development and run-time environment on the host machine. This Java environment allows WebSphere to execute the Java programs used by the Trust Authority registration facility.

IBM HTTP Server

The Trust Authority server uses a three-server, three-port model to process client requests. Your organization may have chosen to install a single instance of the IBM HTTP Server and configure different virtual host names and ports to handle different types of requests.

Using this model, Trust Authority handles the following types of requests:

- Requests that do not require encryption or authentication.
- Requests that require encryption and server authentication.
- Requests that require encryption, server authentication, and client authentication.

Table 12 summarizes these configuration alternatives.

Table 12. Three-server, three-port model for IBM HTTP Servers

<table>
<thead>
<tr>
<th>Protocol</th>
<th>SSL</th>
<th>Server Authentication</th>
<th>Client Authentication</th>
<th>Sample port number for a single IP</th>
<th>Sample port number for a multiple IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>80</td>
<td>80</td>
</tr>
</tbody>
</table>
### Table 12. Three-server, three-port model for IBM HTTP Servers (continued)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>SSL</th>
<th>Server Authentication</th>
<th>Client Authentication</th>
<th>Sample port number for a single IP</th>
<th>Sample port number for a multiple IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>443</td>
<td>443</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>1443</td>
<td>443</td>
</tr>
</tbody>
</table>

### IP aliasing

In Internet technology, an alias is a name assigned to a server that makes the server independent of the name of its host machine. The alias must be defined in your system’s Domain Name System (DNS).

### Directory servers

Trust Authority uses the IBM SecureWay Directory for storing X.509 digital certificates, certificate revocation lists (CRLs), CA policies, and other information about registered servers and users. It can be used to find a public key certificate for a particular person or server by searching the Directory for that person’s or server’s unique distinguished name (DN) or other relevant information.

This server supports Lightweight Directory Access Protocol (LDAP) standards and uses DB2 UDB as its base. It runs as a stand-alone daemon and uses a client/server model to provide Trust Authority access to the server. It uses a Web-based interface to set up and maintain the Directory or to view data in it. The Directory can be an existing server or one that is installed and configured specifically for Trust Authority. Refer to Using the SecureWay Directory With Trust Authority located on the IBM SecureWay Trust Authority Web site: http://www.tivoli.com/support

### Object identifiers

An object identifier (OID) is a value, distinguishable from all other such values, which is associated with an object. Each ASN.1-defined OID forms the node of a tree (similar to the DIT). An object identifier tree is a tree whose root corresponds to that assigned by the ITU recommendation X.680. Its vertices correspond to administrative authorities responsible for allocating arcs (a segment that links two vertices) from that vertex (a point that may be the end of the arc or the intersection of arcs).

Each arc of the tree is labelled by an object identifier component which is a numeric value. Each object to be identified is allocated precisely one vertex (normally a leaf), and no other object (of the same or a different type) is allocated to that same vertex. Thus, an object is uniquely and unambiguously identified by the sequence of numeric values (object identifier components) labelling the arcs in a path from the root to the vertex allocated to the object.

Immediately below the root, three values are currently defined. They are 0, 1, and 2:

- 0 is allocated for CCITT recommendations.
- 1 is allocated for ISO recommendations.
- 2 is allocated for both organizations jointly.
For example, all OIDs reserved for the Directory Standard fall beneath the joint CCITT/ISO value of 2 and have been allocated the object identifier component of 5. Therefore all OIDs for Directory Standard objects start with the prefix 2.5.
This chapter provides reference information for the following:

- **Configuration files**
  Information includes the parameter names, descriptions, and format. It specifies whether the parameters are required or optional and whether they are reconfigurable.

- **Commands**
  Information includes the syntax and parameter descriptions for the Trust Authority command-line utilities.

- **Audit event fields**
  This section lists and describes the information that is contained in Trust Authority audit events.

- **Audit events**
  This section lists and describes the audit events, and specifies whether the event is mandatory or optional.

- **Audit database data**
  This section documents how the Audit event records are stored in the relational database tables. It gives the field names in the record (or column name in the table) where the data is stored, a description of the fields, and the format of the data.

- **Troubleshooting**
  This section provides information on basic troubleshooting as well as debug-level messaging.

### Configuration files

This section provides parameter descriptions for the following configuration files:

- **CA server configuration file, jonahca.ini**
- **RA server configuration file, jonahra.ini**
- **Audit server configuration file, AuditServer.ini**
- **Audit client configuration file, AuditClient.ini**

### File description

A configuration file is divided into sections. Each section begins with a header that is enclosed in brackets, for example, `[section]`. Within a section, there can be one or more statements that use the name and value pair format, `parameter=value`. 
The tables shown here identify all the sections and parameters that comprise the Trust Authority configuration files to be edited for system administration. The columns in the tables are as follows:

- Parameter
- Description
- Value after default configuration
  
  This column gives the default value, if there is one. A parameter is optional if there is a default value; otherwise, it is required.

- Safe to change after default configuration?
  
  This column provides guidance as to whether or not to change the specified value.

This system ignores blank spaces. Any unit specifiers that are present must occur in the correct sequence.

**CA server configuration file**

Table 13 on page 73 describes the CA server configuration file, jonahca.ini, parameters.
Table 13. CA server configuration file

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value after default configuration</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[OIDs] Standards-based mechanism for uniquely identifying items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C=</td>
<td>OID for country</td>
<td>2.5.4.6</td>
<td>No</td>
</tr>
<tr>
<td>O=</td>
<td>OID for organization</td>
<td>2.5.4.10</td>
<td>No</td>
</tr>
<tr>
<td>OU=</td>
<td>OID for organization</td>
<td>2.5.4.11</td>
<td>No</td>
</tr>
<tr>
<td>CN=</td>
<td>OID for common name</td>
<td>2.5.4.3</td>
<td>No</td>
</tr>
<tr>
<td>L=</td>
<td>OID for locality</td>
<td>2.5.4.7</td>
<td>No</td>
</tr>
<tr>
<td>ST=</td>
<td>OID for state</td>
<td>2.5.4.8</td>
<td>No</td>
</tr>
<tr>
<td>T=</td>
<td>OID for title</td>
<td>2.5.4.12</td>
<td>No</td>
</tr>
<tr>
<td>DC=</td>
<td>OID for Domain Component</td>
<td>0.9.2342.19200300.100.1.25</td>
<td>No</td>
</tr>
<tr>
<td>id-dsa=</td>
<td>OID for DSA</td>
<td>1.2.840.10040.4.1</td>
<td>No</td>
</tr>
<tr>
<td>id-dsa-with-sha1=</td>
<td>OID for DSA with SHA-1</td>
<td>1.2.840.10040.4.3</td>
<td>No</td>
</tr>
<tr>
<td>rsaEncryption=</td>
<td>OID for RSA encryption</td>
<td>1.2.840.113549.1.1.1</td>
<td>No</td>
</tr>
<tr>
<td>sha1WithRSAEncryption=</td>
<td>OID for SHA-1 with RSA encryption</td>
<td>1.2.840.113549.1.1.5</td>
<td>No</td>
</tr>
<tr>
<td>sha1=</td>
<td>OID for SHA-1</td>
<td>1.3.14.3.2.26</td>
<td>No</td>
</tr>
<tr>
<td>hmac-sha1=</td>
<td>OID for SHA-1 Hashed Message Authentication Code</td>
<td>1.3.6.1.5.5.8.1.2</td>
<td>No</td>
</tr>
<tr>
<td>pkcs7-data=</td>
<td>OID for PKCS #7</td>
<td>1.2.840.113549.1.7.1</td>
<td>No</td>
</tr>
<tr>
<td>pkcs12-certbag=</td>
<td>OID for certificate bag</td>
<td>1.2.840.113549.1.12.10.13</td>
<td>No</td>
</tr>
<tr>
<td>pkcs12-keybag=</td>
<td>OID for key bag</td>
<td>1.2.840.113549.1.12.10.1.1</td>
<td>No</td>
</tr>
<tr>
<td>X509-Certificate=</td>
<td>OID for X.509 certificate</td>
<td>1.2.840.113549.19.22.1</td>
<td>No</td>
</tr>
<tr>
<td>PasswordBasedMAC=</td>
<td>OID for password based MAC</td>
<td>1.2.840.113533.7.66.13</td>
<td>No</td>
</tr>
<tr>
<td>MyPolicy=</td>
<td>Example of an OID entry for PolicyName1 in CertPolicy section</td>
<td>1.34.67.7</td>
<td>No</td>
</tr>
<tr>
<td>MyLitePolicy=</td>
<td>Example of an OID entry for PolicyName2 in CertPolicy section</td>
<td>2.4.1.0</td>
<td>No</td>
</tr>
<tr>
<td>[AsymmetricKeyAlgs]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value after default configuration</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>DSA=</td>
<td>Definition of the DSA key algorithm</td>
<td>id-dsa</td>
<td>No</td>
</tr>
<tr>
<td>RSA=</td>
<td>Definition of the RSA key algorithm</td>
<td>rsaEncryption</td>
<td>No</td>
</tr>
<tr>
<td>[AsymmetricEncAlgs]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSA=</td>
<td>Definition of the DSA encryption algorithm</td>
<td>id-dsa</td>
<td>No</td>
</tr>
<tr>
<td>[AsymmetricSigAlgs]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSAwithSHA1=</td>
<td>Definition of the DSA with SHA-1 signature algorithm. Used for display and algorithm selection by the GUI or server.</td>
<td>id-dsa-with-sha1</td>
<td>No</td>
</tr>
<tr>
<td>RSAwithSHA1=</td>
<td>Definition of the RSA with SHA-1 signature algorithm. Used for display and algorithm selection by the GUI or server.</td>
<td>sha-1WithRSAEncryption</td>
<td>No</td>
</tr>
<tr>
<td>RSAwithMD5=</td>
<td>Definition of the RSA with MD5 signature algorithm. Used for display and algorithm selection by the GUI or server.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ObjectStore]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name=</td>
<td>File stem (without extension) to be used for CA data files</td>
<td>caObjectStore</td>
<td>No</td>
</tr>
<tr>
<td>Path=</td>
<td>Absolute path where Trust Authority CA files reside</td>
<td>For Windows NT: c:\Program Files\IBM\Trust Authority\etc\TrustAuthority\</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For AIX: /usr/lpp/iau/etc/TrustAuthority/</td>
<td></td>
</tr>
<tr>
<td>[CertPolicy]</td>
<td>Each signature algorithm must have its OID declared in the OIDs section. Each policy name must have a corresponding OID in the OIDs section.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SigAlg1=</td>
<td>Definition of the first signature algorithm. Must have a corresponding entry in the OIDs section.</td>
<td>sha-1WithRSAEncryption</td>
<td>No</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value after default configuration</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>StartTimeSpecifiable=</td>
<td>Whether the requestor, (client entity to RA or RA to CA) can specify the start time.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>MaxLifetime=</td>
<td>The maximum lifetime of a certificate (in hours).</td>
<td>17544h</td>
<td>Yes</td>
</tr>
<tr>
<td>LifeTimeDef=</td>
<td>Default certificate lifetime (in days).</td>
<td>180d</td>
<td>No</td>
</tr>
<tr>
<td>KeySpecifiable=</td>
<td>Whether the requestor (client entity or RA) can specify the public key of the subject.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeyUsageSupported=</td>
<td>Whether the key usage extension is supported.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeyUsageRequired=</td>
<td>Whether the key usage extension is required.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyCritical=</td>
<td>Whether the policy should be critical.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyRequired=</td>
<td>Whether the policy is required.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyName1=</td>
<td>The name of the primary policy. Must have corresponding OID in OIDs section.</td>
<td>MyPolicy</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Org=</td>
<td>The name of the organization that requires this policy</td>
<td>Your Organization</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Notice1=</td>
<td>Notice1 associated with Policy1</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Notice2=</td>
<td>Notice2 associated with Policy1</td>
<td>17</td>
<td>No</td>
</tr>
<tr>
<td>UserNoticeText1=</td>
<td>A legal statement or disclaimer for the relying party to read, and on which to base decisions</td>
<td>The legal statement or disclaimer.</td>
<td>No</td>
</tr>
<tr>
<td>CPS1=</td>
<td>URL where the statement of Policy1 can be read.</td>
<td><a href="http://localhost/index.html">http://localhost/index.html</a></td>
<td>No</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value after default configuration</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>PolicyName2=</td>
<td>Secondary policy name. If present, it must have a corresponding OID in the OIDs section.</td>
<td>My Lite Policy</td>
<td>No</td>
</tr>
<tr>
<td>CPS2=</td>
<td>URL where the statement of Policy2 can be read.</td>
<td><a href="http://localhost/index.html">http://localhost/index.html</a></td>
<td>No</td>
</tr>
<tr>
<td>TimeBetweenCRLs=</td>
<td>Default time between scheduled CRL publication.</td>
<td>1d</td>
<td>Yes</td>
</tr>
<tr>
<td>CRLDuration=</td>
<td>CRL lifetime.</td>
<td>2d</td>
<td>Yes</td>
</tr>
<tr>
<td>[CrossCertPolicy]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SigAlg1=</td>
<td>Definition of the first signature algorithm. Must have a corresponding entry in the OIDs section.</td>
<td>sha-1WithRSAEncryption</td>
<td>No</td>
</tr>
<tr>
<td>StartTimeSpecifiable=</td>
<td>Whether the requestor, (client entity or RA) can specify the start time.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>MaxLifeTime</td>
<td>The maximum lifetime of a certificate.</td>
<td>17544h</td>
<td>Yes</td>
</tr>
<tr>
<td>LifeTimeDef</td>
<td>The default lifetime of a certificate.</td>
<td>180d</td>
<td>No</td>
</tr>
<tr>
<td>KeySpecifiable=</td>
<td>Whether the requestor (client entity or RA) can specify the public key of the subject.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeyUsageSupported=</td>
<td>Whether the key usage extension is supported.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeyUsageRequired=</td>
<td>Whether the key usage extension is required.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyCritical=</td>
<td>Whether the policy should be critical</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyRequired=</td>
<td>Whether the policy is required.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyName1=</td>
<td>The name of the primary policy. Must have corresponding OID in OIDs section.</td>
<td>MyPolicy</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Org=</td>
<td>The name of the organization that requires this policy.</td>
<td>Your Organization</td>
<td>No</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value after default configuration</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------</td>
<td>------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>PolicyNotice1</td>
<td>Notice1 associated with Policy1</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>PolicyNotice2</td>
<td>Notice2 associated with Policy1</td>
<td>17</td>
<td>No</td>
</tr>
<tr>
<td>PolicyName2</td>
<td>Secondary policy name.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPS1</td>
<td>URL where the statement of Policy1 can be read</td>
<td><a href="http://localhost/index.html">http://localhost/index.html</a></td>
<td>No</td>
</tr>
<tr>
<td>PolicyName2</td>
<td>Secondary policy name.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPS2</td>
<td>URL where the Policy2 statement can be read</td>
<td><a href="http://localhost/index.html">http://localhost/index.html</a></td>
<td>No</td>
</tr>
<tr>
<td>TimeBetweenCRLs</td>
<td>Default time between scheduled CRL publication.</td>
<td>1d</td>
<td>Yes</td>
</tr>
<tr>
<td>CRLDuration</td>
<td>CRL lifetime</td>
<td>2d</td>
<td>Yes</td>
</tr>
<tr>
<td>[General]</td>
<td>MyName</td>
<td>/C=US/O=Your Organization/OU=Trust Authority/CN=Trust Authority CA</td>
<td>No</td>
</tr>
<tr>
<td>DefaultRA</td>
<td>The default RA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PreferredCryptoProvider</td>
<td>The GUID (globally unique identifier) for the cryptographic provider.</td>
<td>dda0c1e0-7b73-11d0-8e0c-0004ac602b18</td>
<td>No</td>
</tr>
<tr>
<td>CertsDP</td>
<td>Certificate distribution point.</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>CRLDISTName</td>
<td>Name of the distribution point to be inserted in the certificate.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TempPath</td>
<td>Path for storage of temporary files.</td>
<td>MyCRLDISTName%d</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 13. CA server configuration file (continued)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value after default configuration</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>PathToDLLs=</td>
<td>Path where PKIX libraries are stored.</td>
<td>For Windows NT: c:\pkix\ For AIX: /usr/pkix/</td>
<td>No</td>
</tr>
<tr>
<td>RA1=</td>
<td>DN of the RA for this CA.</td>
<td>/C=US/O=Your Organization/OU=Trust Authority/CN=Trust Authority RA</td>
<td>No</td>
</tr>
<tr>
<td>[Transport]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCPPort=</td>
<td>TCP port where the CA listens.</td>
<td>1830</td>
<td>Yes</td>
</tr>
<tr>
<td>TCPPHost=</td>
<td>Hostname of the machine where the CA resides.</td>
<td>Your hostname</td>
<td>No</td>
</tr>
<tr>
<td>PollInterval=</td>
<td>Polling interval.</td>
<td>10s</td>
<td>Yes</td>
</tr>
<tr>
<td>RetryInterval=</td>
<td>Retry Interval</td>
<td>10m</td>
<td>Yes</td>
</tr>
<tr>
<td>[KeyStore]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CurKeyStore=</td>
<td>KeyStore in use.</td>
<td>VSC</td>
<td>No</td>
</tr>
<tr>
<td>[VSC] This is a required section if the CurKeyStore parameter has a value of VSC.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model=</td>
<td>Type of storage used.</td>
<td>PKCS11_STORAGE_MODEL</td>
<td>No</td>
</tr>
<tr>
<td>Guid=</td>
<td>Global unique identifier.</td>
<td>7F529C80-C942-11D1-8FB0-0004AC61389A</td>
<td>No</td>
</tr>
<tr>
<td>InitialSOpw=</td>
<td>Initial password for the Audit Administrator</td>
<td>SOPIN</td>
<td>No</td>
</tr>
<tr>
<td>TokenDir=</td>
<td>Absolute path and file name for the virtual smart card.</td>
<td>For Windows NT: c:\Program Files\IBM\Trust Authority\etc\TrustAuthority\caKS.fil For AIX: /usr/lpp/iau/etc/TrustAuthority/caKS.fil</td>
<td>No</td>
</tr>
<tr>
<td>4758GUID=</td>
<td>GUID for 4758 card.</td>
<td>474d0880-b44c-11d1-b1cf-002035680b00</td>
<td>No</td>
</tr>
<tr>
<td>Use4758=</td>
<td>Indicator for whether CA uses the 4758.</td>
<td>false</td>
<td>No</td>
</tr>
<tr>
<td>4758ProfileUserId=</td>
<td>The user ID of the 4758 profile.</td>
<td>IBMCA001</td>
<td>No</td>
</tr>
<tr>
<td>4758ProfilePassPhrase=</td>
<td>The password for the 4758 profile.</td>
<td>Secure99</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 13. CA server configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value after default configuration</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>4758Retain=</td>
<td>Whether the CA uses the 4758Retain option. Available only if Use4758=true.</td>
<td>false (true only for RETAIN option, which is not recommended)</td>
<td>No</td>
</tr>
<tr>
<td>4758KeyLength=</td>
<td>The length of the 4758 key. The possible values are: 512,1024, and 2048. Available only if Use4758=true.</td>
<td>512</td>
<td>No</td>
</tr>
<tr>
<td>[TrustPolicy]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UseCRLs=</td>
<td>Whether CRLs and ARLs should be used as part of the validation process.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>AllowExpiredCRLs=</td>
<td>Whether expired CRLs are valid.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>AllowFutureCRLs=</td>
<td>Whether CRLs with future dates are valid.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>AllowExpiredCertificates=</td>
<td>Whether expired certificates are valid.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>AllowFutureCertificates=</td>
<td>Whether future certificates are valid.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>AllowCRLSearchToFail=</td>
<td>Whether it is an error to find no CRLs or ARLs for a given issuer.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>MaximumChainSearchDepth=</td>
<td>Maximum chain depth allowed during recursive breadth-first chain construction.</td>
<td>15</td>
<td>No</td>
</tr>
<tr>
<td>[RemoteServer]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxSessions=</td>
<td>Tuning parameter.</td>
<td>16</td>
<td>No</td>
</tr>
<tr>
<td>EncryptionPolicy=</td>
<td>Whether an encryption policy is used between a background server and its remote administrator if both are on the same system.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>NumAdmins=</td>
<td>Number of administrators certified.</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>
### Table 13. CA server configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value after default configuration</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin1DN=</td>
<td>DN of Administrator1.</td>
<td>Not used</td>
<td>No</td>
</tr>
<tr>
<td>IclOdbcProvider=</td>
<td>Type of provider for ODBC.</td>
<td>UDB</td>
<td>No</td>
</tr>
<tr>
<td>IclProtectionPolicy</td>
<td>ICL protection policy. The values are:</td>
<td>ContinueWithMessage</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Ignore</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ContinueWithMessage</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TerminateTransaction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IclOdbcDriverConnect=</td>
<td>Name of database for ICL.</td>
<td>DSN=ibmdb</td>
<td>No</td>
</tr>
<tr>
<td>MACLabel=</td>
<td>Label for MAC keys in KeyStore.</td>
<td>CA_MAC_Key</td>
<td>No</td>
</tr>
<tr>
<td>/C%EQ%US/O%EQ%Your Organization/OU%EQ%Trust Authority/CN%EQ%Trust Authority RA=pkix://localhost:829</td>
<td>This entry is set during configuration with the URL and DN for the RA</td>
<td>RA DN</td>
<td>Yes</td>
</tr>
</tbody>
</table>
RA server configuration file

Table 14 on page 82 provides information on the RA server configuration file, jonahra.ini, parameters.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value after default configuration</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[OIDs]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C=</td>
<td>OID for country.</td>
<td>2.5.4.6</td>
<td>No</td>
</tr>
<tr>
<td>O=</td>
<td>OID for organization.</td>
<td>2.5.4.10</td>
<td>No</td>
</tr>
<tr>
<td>OU=</td>
<td>OID for organizational unit.</td>
<td>2.5.4.11</td>
<td>No</td>
</tr>
<tr>
<td>CN=</td>
<td>OID for common name.</td>
<td>2.5.4.3</td>
<td>No</td>
</tr>
<tr>
<td>L=</td>
<td>OID for locality.</td>
<td>2.5.4.7</td>
<td>No</td>
</tr>
<tr>
<td>ST=</td>
<td>OID for state.</td>
<td>2.5.4.8</td>
<td>No</td>
</tr>
<tr>
<td>T=</td>
<td>OID for title.</td>
<td>2.5.4.12</td>
<td>No</td>
</tr>
<tr>
<td>DC=</td>
<td>OID for Domain Component</td>
<td>0.9.2342.19200300.100.1.25</td>
<td>No</td>
</tr>
<tr>
<td>id-dsa=</td>
<td>OID for DSA.</td>
<td>1.2.840.10040.4.1</td>
<td>No</td>
</tr>
<tr>
<td>id-dsa-with-sha1=</td>
<td>OID for DSA with SHA-1.</td>
<td>1.2.840.10040.4.3</td>
<td>No</td>
</tr>
<tr>
<td>rsaEncryption=</td>
<td>OID for RSA encryption.</td>
<td>1.2.840.113549.1.1.1</td>
<td>No</td>
</tr>
<tr>
<td>sha1WithRSAEncryption=</td>
<td>OI D for SHA-1 with RSA encryption.</td>
<td>1.2.840.113549.1.1.5</td>
<td>No</td>
</tr>
<tr>
<td>sha1=</td>
<td>OID for SHA-1.</td>
<td>1.3.14.3.2.26</td>
<td>No</td>
</tr>
<tr>
<td>hmac-sha1=</td>
<td>OID for SHA-1 Hashed Message</td>
<td>1.3.6.1.5.5.8.1.2</td>
<td>No</td>
</tr>
<tr>
<td>Authentication Code.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pkcs7-data=</td>
<td>OID for PKCS #7.</td>
<td>1.2.840.113549.17.1</td>
<td>No</td>
</tr>
<tr>
<td>pkcs12-certbag=</td>
<td>OID for PKCS #12 certificate bag.</td>
<td>1.2.840.113549.112.10.1.3</td>
<td>No</td>
</tr>
<tr>
<td>pkcs12-keybag=</td>
<td>OID for PKCS #12 key bag.</td>
<td>1.2.840.113549.112.10.1.1</td>
<td>No</td>
</tr>
<tr>
<td>X509-Certificate=</td>
<td>OID for X.509 certificate.</td>
<td>1.2.840.113549.19.22.1</td>
<td>No</td>
</tr>
<tr>
<td>PasswordBasedMAC=</td>
<td>OID for password-based MAC.</td>
<td>1.2.840.113533.7.66.13</td>
<td>No</td>
</tr>
<tr>
<td>MyPolicy=</td>
<td>Example of an OID entry for PolicyName1 in the CertPolicy section.</td>
<td>1.34.67.7</td>
<td>No</td>
</tr>
<tr>
<td>My Lite Policy=</td>
<td>Example of an OID entry for PolicyName2 in the CertPolicy section.</td>
<td>2.4.1.0</td>
<td>No</td>
</tr>
<tr>
<td>[AsymmetricKeyAlgs]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value after default configuration</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-----------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td><strong>DSA=</strong></td>
<td>Definition of the DSA key algorithm.</td>
<td>id-dsa</td>
<td>No</td>
</tr>
<tr>
<td><strong>RSA=</strong></td>
<td>Definition of the RSA algorithm.</td>
<td>rsaEncryption</td>
<td>No</td>
</tr>
<tr>
<td><strong>[AsymmetricEncAlgs]</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DSA=</strong></td>
<td>Definition of the DSA encryption algorithm.</td>
<td>id-dsa</td>
<td>No</td>
</tr>
<tr>
<td><strong>[AsymmetricSigAlgs]</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DSWithSHA1=</strong></td>
<td>Definition of the DSA with SHA-1 signature algorithm. Used for display and algorithm selection by the GUI or server.</td>
<td>id-dsa-with-sha1</td>
<td>No</td>
</tr>
<tr>
<td><strong>RSAWithSHA1=</strong></td>
<td>Definition of the RSA with SHA-1 signature algorithm. Used for display and algorithm selection by the GUI or server.</td>
<td>sha-1WithRSAEncryption</td>
<td>No</td>
</tr>
<tr>
<td><strong>RSAWithMD5=</strong></td>
<td>Definition of the RSA with MD5 signature algorithm. Used for display and algorithm selection by the GUI or server.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>[ObjectStore]</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Name=</strong></td>
<td>File name without extension to be used for the RA data files.</td>
<td>raObjectStore</td>
<td>No</td>
</tr>
<tr>
<td><strong>Path=</strong></td>
<td>Absolute path where the Trust Authority RA data files are located.</td>
<td>For Windows NT: c:\Program Files\IBM\Trust Authority\pkrf\Domains\YourDomain\etc\</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For AIX: /usr/lpp/iau/pkrf Domains/YourDomain/etc/</td>
<td></td>
</tr>
</tbody>
</table>

[IssuerCertPolicy] Each signature algorithm must have its OID declared in the OIDs section. Each policy name must have a corresponding OID in the OIDs section.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value after default configuration</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>SigAlg1=</td>
<td>Definition of the signature algorithm. Must have a corresponding entry in the OIDs section.</td>
<td>sha-1WithRSAEncryption</td>
<td>No</td>
</tr>
<tr>
<td>StartTimeSpecifiable=</td>
<td>Whether the requestor, (client entity to RA, or RA to CA) can specify the certificate start time.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>MaxLifeTime=</td>
<td>The maximum lifetime of a certificate.</td>
<td>8760h</td>
<td>No</td>
</tr>
<tr>
<td>LifeTimeDef=</td>
<td>Default certificate lifetime.</td>
<td>180d</td>
<td>No</td>
</tr>
<tr>
<td>KeySpecifiable=</td>
<td>Whether the requestor (client entity or RA) can specify the public key of the subject.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeyUsageSupported=</td>
<td>Whether the key usage extension is supported.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeyUsageRequired=</td>
<td>Whether the key usage extension is required.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyCritical=</td>
<td>Whether the policy should be critical.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyRequired=</td>
<td>Whether the policy is required.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyName1=</td>
<td>The name of the primary policy.</td>
<td>MyPolicy</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Org=</td>
<td>The name of the organization that requires the primary policy.</td>
<td>Your Organization</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Notice1=</td>
<td>Notice1 associated with Policy1.</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Notice2=</td>
<td>Notice2 associated with Policy1.</td>
<td>17</td>
<td>No</td>
</tr>
<tr>
<td>UserNoticeText1=</td>
<td>A legal statement or disclaimer for the relying party to read, and on which to base decisions.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value after default configuration</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>CPS1</td>
<td>URL where the statement of Policy1 can be read.</td>
<td><a href="http://localhost/index.html">http://localhost/index.html</a></td>
<td>No</td>
</tr>
<tr>
<td>PolicyName2=</td>
<td>Secondary policy name. If present, it must have a corresponding OID in the OIDs section.</td>
<td>My Lite Policy</td>
<td>No</td>
</tr>
<tr>
<td>CPS2=</td>
<td>URL where the Policy2 statement can be read.</td>
<td><a href="http://localhost/index.html">http://localhost/index.html</a></td>
<td>No</td>
</tr>
<tr>
<td>EERevokeRequests=</td>
<td>Whether a client entity can request the revocation of a certificate. ANY means the client entity can request the revocation of a certificate it requested. NONE means the client entity cannot request the revocation of certificates.</td>
<td>Any</td>
<td>No</td>
</tr>
<tr>
<td>[IssuerCrossCertPolicy]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SigAlg1=</td>
<td>Name of the first signature algorithm. Must have a corresponding entry in the OIDs section.</td>
<td>sha-1WithRSAEncryption</td>
<td>No</td>
</tr>
<tr>
<td>StartTimeSpecifiable=</td>
<td>Whether the requestor, (client entity or RA) can specify the start time.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeySpecifiable=</td>
<td>Whether the requestor (client entity or RA) can specify the public key of the subject.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeyUsageSupported=</td>
<td>Whether the key usage extension is supported.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>KeyUsageRequired=</td>
<td>Whether the key usage extension is required.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value after default configuration</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>PolicyCritical</td>
<td>Whether the policy should be critical.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyRequired</td>
<td>Whether the policy is required.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>PolicyName1</td>
<td>The name of the primary policy. Must have corresponding OID in OIDs section.</td>
<td>MyPolicy</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Org</td>
<td>The name of the organization that requires the primary policy.</td>
<td>Your Organization</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Notice1</td>
<td>Notice1 associated with Policy1.</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>Policy1Notice2</td>
<td>Notice2 associated with Policy1.</td>
<td>17</td>
<td>No</td>
</tr>
<tr>
<td>UserNoticeText1</td>
<td>A legal statement or disclaimer for the relying party to read, and on which to base decisions.</td>
<td>The legal statement or disclaimer</td>
<td>No</td>
</tr>
<tr>
<td>CSP1</td>
<td>URL where the Policy1 statement can be read.</td>
<td><a href="http://localhost/index.html">http://localhost/index.html</a></td>
<td>No</td>
</tr>
<tr>
<td>PolicyName2</td>
<td>Secondary policy name.</td>
<td>My Lite Policy</td>
<td>No</td>
</tr>
<tr>
<td>CPS2</td>
<td>URL where the Policy2 statement can be read.</td>
<td><a href="http://localhost/index.html">http://localhost/index.html</a></td>
<td>No</td>
</tr>
<tr>
<td>EERevokeRequests</td>
<td>Whether a client entity can request the revocation of a certificate. ANY means that the client entity can request the revocation of any certificate. SELF means the client entity can request the revocation of a certificate it requested. NONE means that the client entity cannot request the revocation of certificates.</td>
<td>ANY</td>
<td>No</td>
</tr>
<tr>
<td>MyName</td>
<td>Distinguished name of the entity. /C=US/O=Your Organization/OU=Trust Authority/CN=Trust Authority RA</td>
<td></td>
<td>No</td>
</tr>
</tbody>
</table>
Table 14. RA server configuration file  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value after default configuration</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer1</td>
<td>Distinguished name for this RA’s CA.</td>
<td>/C=US/O=Your Organization/OU=Trust Authority/CN=Trust Authority CA</td>
<td>No</td>
</tr>
<tr>
<td>Issuer1URL1</td>
<td>URL for this RA’s CA.</td>
<td>pkix://servername:1830</td>
<td>Yes</td>
</tr>
<tr>
<td>TempPath=</td>
<td>Path for storage of temporary files.</td>
<td>For Windows NT: c:\Program Files\IBM\Trust Authority\pkrf\Domains\YourDomain\etc\ For AIX: /usr/lpp/iau/pkrf/Domains\YourDomain/etc/</td>
<td>No</td>
</tr>
<tr>
<td>PathToDLLs=</td>
<td>Path where PKIX libraries are installed.</td>
<td>For Windows NT: c:\pkix\ For AIX: /usr/pkix/</td>
<td>No</td>
</tr>
<tr>
<td>TrustChainDelivery=</td>
<td>0 = Disable trust chain delivery 1 = Enable trust chain delivery</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>[Transport]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCPPort=</td>
<td>TCP port where the RA listens.</td>
<td>829</td>
<td>Yes</td>
</tr>
<tr>
<td>TCPIPHost=</td>
<td>TCP/IP host name of the machine where the RA resides.</td>
<td>Your hostname</td>
<td>No</td>
</tr>
<tr>
<td>PollInterval=</td>
<td>Polling interval.</td>
<td>30s</td>
<td>Yes</td>
</tr>
<tr>
<td>RetryInterval=</td>
<td>The amount of time between interrogations of the CA by the RA in case the poll time sent to the RA from the CA is earlier than the RA’s current clock time.</td>
<td>1m</td>
<td>Yes</td>
</tr>
<tr>
<td>[KeyStore]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CurKeyStore=</td>
<td>KeyStore in use.</td>
<td>VSC</td>
<td>No</td>
</tr>
<tr>
<td>[VSC] This is a required section if the CurKeyStore parameter has a value of VSC.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model=</td>
<td>Type of storage used.</td>
<td>PKCS11_STORAGE_MODE</td>
<td>No</td>
</tr>
<tr>
<td>GUID=</td>
<td>Global unique identifier.</td>
<td>7F529C80-C942-11D1-8FB0-0004AC61389A</td>
<td>No</td>
</tr>
<tr>
<td>InitialSOpw=</td>
<td>Initial password for the Audit Administrator.</td>
<td>SOPIN</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 14. RA server configuration file (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value after default configuration</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>TokenDir=</td>
<td>Absolute path and file name for the virtual smart card.</td>
<td>For Windows NT: c:\Public Files\IBM\Trust Authority\pkrf\Domains\YourDomain\etc\raKS.fil</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For AIX: /usr/lpp/iau/pkrf.Domains/YourDomain/etc/raKS.fil</td>
<td></td>
</tr>
<tr>
<td>[TrustPolicy]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UseCRLs=</td>
<td>Whether CRLs and ARLs should be used as part of the validation process.</td>
<td>T</td>
<td>No</td>
</tr>
<tr>
<td>AllowExpiredCRLs=</td>
<td>Whether expired CRLs are valid.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>AllowFutureCRLs=</td>
<td>Whether CRLs with future dates are valid.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>AllowExpiredCertificates=</td>
<td>Whether expired certificates are valid.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>AllowFutureCertificates=</td>
<td>Whether future certificates are valid.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>ApplyNameConstraintsToEEOnly=</td>
<td>Whether to apply name constraints to every certificate in the chain, or only the last.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>AllowCRLSearchToFail=</td>
<td>Whether it is an error to find no CRLs or ARLs for a given issuer.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>MaximumChainSearchDepth=</td>
<td>Maximum chain depth allowed during recursive breadth-first chain construction.</td>
<td>15</td>
<td>No</td>
</tr>
<tr>
<td>[LDAP]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NumServers=</td>
<td>Number of LDAP servers.</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Server1=</td>
<td>Hostname and port number for the LDAP server.</td>
<td>hostname:port</td>
<td>Yes</td>
</tr>
<tr>
<td>AuthName1</td>
<td>DN of the Directory Administrator</td>
<td>/C=US/O=Your Organization/OU=Trust Authority/CN=DirAdmin</td>
<td>No</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value after default configuration</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>AuthPwd1</td>
<td>Password of the Directory Administrator, (Change using the Trust Authority Change Password utility only)</td>
<td>Secure99</td>
<td>Yes</td>
</tr>
<tr>
<td>PostInterval=</td>
<td>Interval between checks to see if there is information to be posted to the Directory.</td>
<td>5m</td>
<td>Yes</td>
</tr>
<tr>
<td>[RemoteServer]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxSessions=</td>
<td>Tuning parameter.</td>
<td>16</td>
<td>No</td>
</tr>
<tr>
<td>EncryptionPolicy=</td>
<td>Whether an encryption policy is used between a background server and its remote administrator if both are on the same system.</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>NumAdmins=</td>
<td>Number of administrators certified.</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Admin1DN=</td>
<td>DN of Administrator1.</td>
<td>Not used</td>
<td>No</td>
</tr>
<tr>
<td>CurrentAdminPort</td>
<td>Current administration port.</td>
<td>Not used</td>
<td>No</td>
</tr>
<tr>
<td>[ICL]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IclOdbcProvider</td>
<td>Type of ODBC provider.</td>
<td>UDB</td>
<td>No</td>
</tr>
<tr>
<td>IclOdbcDriverConnect</td>
<td>Name of database for ICL.</td>
<td>DSN=pkrfdb</td>
<td>No</td>
</tr>
<tr>
<td>[URLs]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/C%EQ%US/O%EQ%Your Organization/OU%EQ%Trust Authority/CN%EQ%Trust Authority CA=pkix://localhost:1830</td>
<td>This entry is set during configuration with the URL and DN for the CA.</td>
<td>CA DN</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Audit server configuration file

The Audit server configuration file, AuditServer.ini, specifies the configuration variables for the Audit server. These variables configure the basic operational characteristics of the server and specify how to log debug and error messages. Variables also control which events are logged. The following sections provide supplementary information about the entries in Table 15 on page 91.

- Generic Service handler parameters
- Event severity levels
- Trace levels

Generic Service handler parameters
The following service handler parameters can be used for each service:

- service.count - The number of available services
- service.x.name - The name of a service
- service.x.classname - The classname of a service
- service.x.dpolicy - The classname of a delivery policy
- service.default.count - The number of default services
- service.default.x.name - The name of a default service

Trace levels
Trace levels are as follows:

- All - Logs error, warning, informational, and event trace messages
- Error - Logs error messages
- Warning - Logs error messages and warning messages
- Eventinfo - Logs error messages and event trace messages
- None - Logs no messages
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server Port</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>acceptor.classname=</td>
<td>The name of the acceptor class.</td>
<td>com.ibm.irg.sysmgmt.daemon.acceptors.SMSocketAcceptor</td>
<td>No</td>
</tr>
<tr>
<td>acceptor.arg=</td>
<td>The port at which the Audit Server listens</td>
<td>7222</td>
<td>Yes</td>
</tr>
<tr>
<td>acceptor.init.retries=</td>
<td>The number of times to attempt a bind to the server socket</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>acceptor.init.delay=</td>
<td>The number of seconds to wait between binding attempts. (The program assumes seconds.)</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Service Handler</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>service.count=</td>
<td>The number of available services</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td><strong>Log Service</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>service.1.name=</td>
<td>The name of the log service</td>
<td>log</td>
<td>No</td>
</tr>
<tr>
<td>service.1.classname=</td>
<td>The classname of the log service</td>
<td>com.ibm.irg.sysmgmt.daemon.services.log.SMLogService</td>
<td>No</td>
</tr>
<tr>
<td>service.1.dpolicy=</td>
<td>The classname of the log service delivery policy</td>
<td>com.ibm.irg.sysmgmt.daemon.services.log.SMLogDeliveryPolicy</td>
<td>No</td>
</tr>
<tr>
<td><strong>Audit Service</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>service.2.name=</td>
<td>The name of the audit service</td>
<td>audit</td>
<td>No</td>
</tr>
<tr>
<td>service.2.classname=</td>
<td>The classname of the audit service</td>
<td>com.ibm.irg.sysmgmt.daemon.services.audit.SMAuditService</td>
<td>No</td>
</tr>
<tr>
<td>service.2.dpolicy=</td>
<td>The classname of the audit service delivery policy</td>
<td>com.ibm.irg.sysmgmt.daemon.services.audit.SMAuditDeliveryPolicy</td>
<td>No</td>
</tr>
<tr>
<td><strong>Default Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>service.default.count=</td>
<td>The number of available default services</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>service.default.1.name=</td>
<td>The name of the default service</td>
<td>log</td>
<td>No</td>
</tr>
<tr>
<td>service.default.2.name=</td>
<td>The name of the default service</td>
<td>audit</td>
<td>No</td>
</tr>
<tr>
<td><strong>Event Configuration</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Default</td>
<td>Safe to change after configuration?</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------</td>
<td>-------------------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>irgsmd.event.config=</td>
<td>The event configuration file</td>
<td>smevents.conf</td>
<td>No</td>
</tr>
<tr>
<td>Log Service Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| event.log.filename=     | The file for capturing events                        | For Windows NT: c:\Program Files\IBM\Trust Authority\logs\smevents.log  
For AIX: /usr/lpp/iau/logs/smevents.log | Yes                                |
| event.log.append=       | Flag indicating whether or not to append or overwrite the event log | true                                             | Yes                                |
| event.log.severity.min= | Event log minimum severity                           | 1                                                | Yes                                |
| event.log.severity.max= | Event log maximum severity                           | 7                                                | Yes                                |
| Service Configuration   |                                                      |                                                 |                                     |
| audit.ksfile=           | The Audit KeyStore file. This file is used to store Audit cryptographic keys. |                                                      | No                                 |
| audit.catalog           | The filename of the NLS catalog file.                 | For Windows NT: c:\Program Files\IBM\Trust Authority\catalog\IRGVDS.DLL  
For AIX: /usr/lpp/iau/catalog/IRGVDS.cat   | No                                 |
<p>| audit.db.instance=      | The name of the Audit DB2 instance.                   |                                                 | No                                 |
| audit.db.name=          | The name of the Audit database.                      |                                                 | No                                 |
| audit.db.connection=    | The algorithm for managing the database connection.   | single                                           | No                                 |
| audit.log.connect.retries= | The number of retries allowed for making the connection to the database/log file. | 3                                                | Yes                                |
| audit.log.update.retries= | The number of retries allowed for updating the audit log. | 3                                                | Yes                                |
| audit.log.timeout       | Timeout value allowed for updating the audit log.    | 60s                                              | Yes                                |
| audit.log.integrity=    | Enables or disables integrity checking                | true                                             | No                                 |
| Trace                   |                                                      |                                                 |                                     |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>trace.enable=</td>
<td>A flag to enable or disable tracing</td>
<td>false</td>
<td>Yes</td>
</tr>
<tr>
<td>trace.level.name=</td>
<td>The name representing a trace level</td>
<td>all</td>
<td>Yes</td>
</tr>
<tr>
<td>trace.event.enable=</td>
<td>Enables or disables event tracing</td>
<td>false</td>
<td>No</td>
</tr>
<tr>
<td>trace.log.filename=</td>
<td>The trace log file name</td>
<td>For Windows NT: c:\Program Files\IBM\Trust Authority\logs\iausmd.log For AIX: /usr/lpp/iau/logs/iausmd.log</td>
<td>Yes</td>
</tr>
<tr>
<td>trace.log.append=</td>
<td>Flag indicating whether or not to append to the existing trace file</td>
<td>true</td>
<td>Yes</td>
</tr>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>error.log.filename=</td>
<td>The name of the error file.</td>
<td>For Windows NT: c:\Program Files\IBM\Trust Authority\logs\iausmd.err For AIX: /usr/lpp/iau/logs/iausmd.err</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Audit client configuration file, AuditClient.ini

The Audit client configuration file, AuditClient.ini, specifies the configuration variables for the Audit client.
### Table 16. Audit client configuration file

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Safe to change after configuration?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection Settings</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EnableAudit=</td>
<td>Enables or disables audit events.</td>
<td>true</td>
<td>No</td>
</tr>
<tr>
<td>HostName=</td>
<td>Audit server host.</td>
<td>your hostname</td>
<td>No</td>
</tr>
<tr>
<td>Port=</td>
<td>Audit server port. Must match the port specified</td>
<td>59998</td>
<td>Yes</td>
</tr>
<tr>
<td>AuthType=</td>
<td>Authentication mechanism. This parameter is not supported</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>UseSSL=</td>
<td>Should SSL be used for the client/server session? For future use. This parameter is not supported in this release of Trust Authority.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SSLKeyDB=</td>
<td>SSL key database. This parameter is not supported in this release of Trust Authority.</td>
<td>For AIX: /usr/lpp/iau/audit/client/ssl.kdb For Windows NT: c:\Program Files\IBM\Trust Authority\audit\client\ssl.kdb</td>
<td>No</td>
</tr>
<tr>
<td>SSLv2Token=</td>
<td>SSL key database token. This parameter is not supported in this release of Trust Authority.</td>
<td>token</td>
<td>No</td>
</tr>
<tr>
<td>Retries=</td>
<td>Number of tries to send event</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>Timer</td>
<td>The amount of time to wait, in milliseconds, between retry attempts to connect to the server.</td>
<td>1000</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Component Mask Sections</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExcludedEvents=</td>
<td>CA events that will not be sent</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>ExcludedEvents=</td>
<td>RA events that will not be sent</td>
<td>ReceiptOfCertRequest</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Command-line utilities

This section provides the syntax and the parameter descriptions for the following command-line utilities:
- CA Certification utility
- Add RA User utility
- Enable RA Encryption utility
- Audit Archive and Sign utility
- Audit Integrity Check utility

CA Certification utility

The CA Certification utility allows you to request certification from another CA on behalf of the Trust Authority certificate authority (CA) using either the cross-certification model or the hierarchy trust model.

Note: If you are planning to download the hierarchical CA certificates into Netscape Communicator as part of your overall hierarchy solution, then you should use the "-m .." flag with the CA Certification utility in order to exclude the Name Constraints extension from the resulting CA certificate. You should not use any of the Name Constraints command line options other than "-m .." (in other words, not -i/I, -d/D, -u/U, -n/N, or -m/M) with the CA Certification utility. For example, an instance of such a command might look like this:

```bash
CaCertRq -m .. -h -r "c:\Program Files\IBM\Trust Authority\ccprereg.reg" -P 1835 -W Secure99
```

Syntax

The syntax for this utility is:

```bash
CaCertRq
    [-i ipAddressMask [-I ipAddressMask]]
    [-d dns [-D dns]]
    [-m emailAddress [-M emailAddress]]
    [-u uri [-U uri]]
    [-n directoryName [-N directoryName]]
    [-p inhibitPolMap]
    [-h [-m ..]]
    [-H filename -e | -E]
    [-C filename [-B filename]]
    [-S filenames]
    -r preregistrationpath
    -P 1835
    -W password
```

Parameters

```
[-i ipAddressMask [-I ipAddressMask]]
The Internet protocol (IP) address mask specified in CDIF format. The lowercase parameter (-i) adds the specified IP address mask to the permitted subtree list. The uppercase parameter (-I) is the same as the lowercase parameter except that it adds the specified IP address mask to the excluded subtree list.

For example, Organization X’s mask is 9.0.0.0/255.0.0.0, while Organization X’s Department Y’s current mask is 9.210.134.0/255.255.254.0.

[-d dns [-D dns]]
Domain Name Server (DNS) address. The lowercase parameter (-d) adds the specified
```
DNS address to the permitted subtree list. The uppercase parameter (-D) is the same as the lowercase parameter except that it adds the specified DNS address to the excluded subtree list.

If the address starts with a period, all hosts ending with that substring (including the "." ) are intended, but if it does not, only the host matching that string is intended.

For example, the constraint ".orga.com" matches us.orga.com, vneto.orga.com, and w3.software.orga.com, but not orga.com (nor kidorga.com) itself. The constraint orga.com matches orga.com but not us.orga.com or the others. This suggests that a permitted subtree without a leading "." indicates only a single possible node.

\[-m emailAddress [ -M emailAddress]\]
E-mail addresses. The lowercase parameter (-m) adds the specified e-mail address to the permitted subtree list. The uppercase parameter (-M) is the same as the lowercase parameter except that it adds the specified e-mail address to the excluded subtree list.

An e-mail address can be in standard format (no wildcards) or can be just the DNS address. If you specify a DNS address, the rules are the same as those listed under the -d option. This suggests that a permitted subtree containing a standard e-mail address indicates only a single possible user.

\[-u uri [ -U uri]\]
URI. The lowercase parameter (-u) adds the specified URI to the permitted subtree list. The uppercase parameter (-U) is the same as the lowercase parameter except that it adds the specified URI to the excluded subtree list.

The node portion of the Uniform Resource Identifier (URI) is subject to the same rules as those listed under the -d option, unless it contains an IP address. In such a case, it is treated as an exact match.

\[-n directoryName [ -N directoryName]\]
The Directory name. The lowercase parameter (-n) adds the specified Directory name to the permitted subtree list. The uppercase parameter (-N) is the same as the lowercase parameter except that it adds the Directory name to the excluded subtree list.

These names should be in standard LDAP format. Names matching each of the supplied relative distinguished names (RDNs) in a constraint will be considered to match the constraint, whatever other RDNs may be present.

**Note:** When you specify a directory path that contains embedded spaces as the parameter of a command, you must enclose the path in double quotation marks (" ").

\[-h \]
Specifies that the certificate request use the hierarchy trust model.

\[-H filename\]
creates a response file in a PKCS #7 format for a third party CA. This creates a downward hierarchy.

\[-m ..\]
Specifies that the Name Constraints extension be excluded from the certificate.

\[-C filename\]
Specifies the filename to which the binary-encoded PKCS #10 request is written. The -C and -B parameters are mutually exclusive.
[-B filename]
Specifies the filename to which the base 64 binary-encoded PKCS #10 request is written. The -C and -B parameters are mutually exclusive.

[-S filename]
Specifies the filename that contains the PKCS #7 response.

[-e | -E]
Used in conjunction with the -H flag to designate what format the response file is created in, either binary or base64. The -e flag creates the response file in binary, and the -E flag creates the response file in base64.

[-E]
-r preregistrationpath
The path name of a preregistration file generated by the issuing system.

-P 1835
The administrative port number of the CA, which is 1835.

-p inhibitPolMap
The value of the inhibitPolicyMapping extension field.
If this option is not present, this field defaults to 1.

-W password
The password entered when the preregistration file was generated.

Add RA User utility
The Add RA User utility lets you add an RA administrative user (or registrar) to a specified Trust Authority registration domain.

Note: When you specify a directory path that contains embedded spaces as the parameter of a command, you must enclose the path in double quotation marks (" ").

Syntax
add_rauser domain_configuration_file domain_name ID [access_profile]

Parameters
domain_configuration_file
The absolute path to the domain configuration file (domain.cfg). This file contains a section for each registration domain configured in the system. It is used to locate the virtual root directory of the specified registration domain.

domain_name
The name of the registration domain that the added registrar will be administering.

ID Either the request ID returned by the browser at the time the user requested the certificate, or the credential UUID you obtained by using the RA Desktop to view the user’s certificate record.

When you specify the ID parameter, the program first assumes that the ID is a request ID. It then uses this value to obtain the credential UUID (for example, sp0ApHvpzvCicr1T8sSkW==). In summary, the certificate must have been previously issued to the user who is requesting authorization as a registrar.

If a matching record is not found in the registration database, the program assumes that the ID is a credential UUID. It then attempts to find a corresponding request record. If
there is no match, the program issues a warning message and attempts to add an entry to the registrar’s table (rausers) in the database.

*access_profile*

An optional parameter that specifies the access profile associated with the registrar that you are adding to the system. If you do not specify a value, the program defaults to the value, RAUser.

This parameter must be a valid profile specified in one of the Trust Authority registration administration configuration files. The profile lists the types of queries, actions, and so forth, that the associated registrars are allowed to perform.

**Enable RA Database Encryption utility**

By default, information stored in the registration facility database is not encrypted. After configuration, you can enable database encryption by running the Enable RA Database Encryption (iauEnableRADBSec) program.

**Syntax**

\[
\text{iauEnableRADBSec} \ -d \ \text{domain\_name} \ -r \ \text{install\_directory} \ [i]
\]

**Parameters**

- `-d domain_name`
  The name of your registration domain. This value must match the value specified for Registration domain name in the Setup Wizard.

- `-r install_directory`
  The full path to where the Trust Authority product is installed.

- `-i install_directory`
  Optional parameter that directs the system to display debugging messages (currently in English only).

**Audit Archive and Sign utility**

The Audit Archive and Sign utility lets you archive and sign audit log files.

**Note:** When you specify a directory path that contains embedded spaces as the parameter of a command, you must enclose the path in double quotation marks (" ").

**Syntax**

\[
\text{AuditArchiveAndSign} \ [-c \ ConfigFile\ Path] \ [-n] \ ArchiveFileName
\]

**Parameters**

- `-c ConfigFilePath`
  The absolute path name of the Audit server configuration file.

  - The default path on AIX is: /usr/lpp/iau/etc/TrustAuthority/AuditServer.ini
  - The default path on Windows NT is: c:\Program Files\IBM\Trust Authority\etc\TrustAuthority\AuditServer.ini

  **Note:** This parameter is optional. If you do not specify a value, the default Audit server configuration file is used.

- `-n` Specifies that audit records should not be deleted from the database.
The prefix of the filename to which the audit logs are to be written. The utility adds the extensions .ixf and .sig to the archive and signature files, respectively.

**Audit Integrity Check utility**

The Audit Integrity Check utility performs integrity checking on audit records that have been archived as well as on the Audit server database. This tool is effective only if integrity checking is enabled for your system. The syntax for this command takes the following forms:

- **Form1**
  
  This form checks the integrity of the Audit server database, and prompts you for the Audit Administrator password.

- **Form2**
  
  This form checks the integrity of one or more Audit server archive files and prompts you for the Audit Administrator password.

- **Form3**
  
  This form checks the integrity of all the archive files under the specified directory and prompts you for the Audit Administrator password.

**Note:** When you specify a directory path that contains embedded spaces as the parameter of a command, you must enclose the path in double quotation marks (" ").

**Syntax**

**Form1:**

AuditIntegrityCheck [-c ConfigFilePath] -d

**Form2:**

AuditIntegrityCheck [-c ConfigFilePath] -a ArchiveFileName1 ArchiveFileName2 ArchiveFileName3

**Form3:**

AuditIntegrityCheck [-c ConfigFilePath] -A ArchiveFileDirectory

**Parameters**

- **-c ConfigFilePath**
  
  The absolute path name of the configuration file for the Audit server.

  - The default path on AIX is: /usr/lpp/iau/etc/TrustAuthority/AuditServer.ini
  
  - The default path on Windows NT is: c:\Program Files\IBM\TrustAuthority\etc\TrustAuthority\AuditServer.ini

  **Note:** This parameter is optional. If you do not specify a value, the default Audit server configuration file is used.

- **-a ArchiveFileName1 ArchiveFileName2 ArchiveFileName3**
  
  The pathname prefix of the archive files that are the targets of the integrity check. The prefix represents a set of files that are associated with the archive.

  For example, the command -a /local/archive/archive1-1067 would process multiple files with the pathname prefix /local/archive/archive1-1067, as follows:
ArchiveFileDirectory

The pathname of the directory that contains archive files that are the targets of the integrity check. All files with _audit_log.ixf and _audit_log.sig are processed.

Create New RA utility

The Create New RA utility is used when setting up multiple RAs (MRA). The Create New RA utility creates a data file that is used by the Configuration Wizard to set up the MRA. The utility creates the data file based on a template file called mraTemplate.ini. This utility can be run on either the CA server or the primary RA server.

Syntax
createNewRA [-f filename]

Parameters

-f filename

The name of the data file is filename.

Note: This parameter is optional. If you do not specify a value, the name of the data file will be mraConfig.dat.

MRA Enrollment Authorization utility

The MRA Enrollment Authorization utility is used when setting up multiple RAs (MRA). The MRA Enrollment Authorization utility reads the pre-enrollment file created during MRA setup, and authorizes the new MRA. This allows the CA to sign any certificates issued by the MRA once setup is complete. This utility is run on the CA server.

Syntax
authRAEnrollment -f filename

Parameters

-f filename

This tells the authRAEnrollment command the name of the file to use to enroll the MRA.

Note: This parameter is optional. If you do not specify a value, the name of the data file will be mraConfig.dat.
Audit event fields

Table 17 lists and describes the information that is contained in Trust Authority audit events.

Table 17. Audit event fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Name</td>
<td>The event identifier, which is specified as a token by an audit client.</td>
</tr>
<tr>
<td>Affected Entity</td>
<td>The entity affected by the action for which an audit event is being sent. This is specified by an audit client.</td>
</tr>
<tr>
<td>Affected Entity Type</td>
<td>The type of the affected entity. This is specified by an audit client.</td>
</tr>
<tr>
<td>Authorized Entity</td>
<td>The entity that has authorized the operation. This is specified by an audit client.</td>
</tr>
<tr>
<td>Authorized Entity Role</td>
<td>The role of the authorized entity.</td>
</tr>
<tr>
<td>Storage Media</td>
<td>If applicable, the storage media referenced in the operation. This is specified by an audit client.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity of the event, which can be either information or alert.</td>
</tr>
<tr>
<td>Reporting Component Type</td>
<td>The Trust Authority component type that is reporting the event.</td>
</tr>
<tr>
<td>Client Source</td>
<td>Identifying information for the Trust Authority component that is sending the event.</td>
</tr>
<tr>
<td>Client Timestamp</td>
<td>The time the audit event was sent from a Trust Authority component.</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>The time the audit event was written to the database or file.</td>
</tr>
<tr>
<td>Extra Info</td>
<td>Any additional information that is relevant to the operation. This is specified by an audit client.</td>
</tr>
</tbody>
</table>

Audit events

Table 18 lists and describes the Trust Authority audit events and indicates whether they are mandatory or optional.

Table 18. Audit events

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
<th>Mandatory or optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeyGeneration</td>
<td>Indicates that a cryptographic key has been generated.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>KeyImport</td>
<td>Indicates that a cryptographic key has been imported into Trust Authority.</td>
<td>Optional</td>
</tr>
<tr>
<td>KeyExport</td>
<td>Indicates that a cryptographic key has been exported from Trust Authority.</td>
<td>Optional</td>
</tr>
<tr>
<td>KeyStorage</td>
<td>Indicates that a cryptographic key has been stored.</td>
<td>Optional</td>
</tr>
<tr>
<td>Event</td>
<td>Description</td>
<td>Mandatory or optional</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>KeyRollover</td>
<td>Indicates that a cryptographic key has been replaced with a new key that is used for the same purpose as the key it replaced.</td>
<td>Optional</td>
</tr>
<tr>
<td>KeyCompromise</td>
<td>Indicates that a cryptographic key has been compromised.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CAKeyDistribution</td>
<td>Indicates that the public key of the CA has been written to the Directory.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CertGeneration</td>
<td>Indicates that a certificate has been generated.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CertRevocation</td>
<td>Indicates that a certificate has been revoked.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CertRenewal</td>
<td>Indicates that a certificate has been renewed.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CertSuspension</td>
<td>Indicates that a certificate has been temporarily suspended.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CertResumption</td>
<td>Indicates that a previously suspended certificate has been made active.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CRLQuery</td>
<td>Indicates that the Certificate Revocation List (CRL) has been read.</td>
<td>Optional</td>
</tr>
<tr>
<td>CRLUpdate</td>
<td>Indicates that the Certificate Revocation List (CRL) has been updated.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>SubmitCertRequest</td>
<td>Indicates that a request for a certificate has been submitted.</td>
<td>Optional</td>
</tr>
<tr>
<td>Enrollment</td>
<td>Indicates that an RA has requested enrollment with a CA.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Unenrollment</td>
<td>Indicates that an RA has requested to unenroll from a CA.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CertDeletion</td>
<td>Indicates that the CA has deleted a certificate from its certificate database.</td>
<td>Optional</td>
</tr>
<tr>
<td>SuccessfulAuthWithPassword</td>
<td>Indicates that an authentication attempt using a password has succeeded.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>UnsuccessfulAuthWithPassword</td>
<td>Indicates that an authentication attempt using a password has failed.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>SuccessfulAuthWithCert</td>
<td>Indicates that an authentication attempt using a certificate has succeeded.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>UnsuccessfulAuthWithCert</td>
<td>Indicates that an authentication attempt using a certificate has failed.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>SuccessfulCertValidation</td>
<td>Indicates that the entire certificate chain up to the certificate of the Root CA has been verified.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>UnsuccessfulCertValidation</td>
<td>Indicates that the entire certificate chain up to the certificate of the Root CA has been invalidated.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>PasswordChange</td>
<td>Indicates that an attempt to change a password has succeeded.</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>
### Table 18. Audit events (continued)

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
<th>Mandatory or optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLUpdate</td>
<td>Indicates that a user or entity has been added or removed from an Access Control List (ACL).</td>
<td>Mandatory</td>
</tr>
<tr>
<td>SuccessfulIntegrityCheck</td>
<td>Indicates that an attempt to check the integrity of an audit record resulting from a transaction with a non-Trust Authority component has succeeded.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>UnsuccessfulIntegrityCheck</td>
<td>Indicates that an attempt to check the integrity of an audit record resulting from a transaction with a non-Trust Authority component has failed.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>SuccessfulAcquirePrivilege</td>
<td>Indicates that an attempt to acquire some level of access to a Trust Authority KeyStore has succeeded.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>UnsuccessfulAcquirePrivilege</td>
<td>Indicates that an attempt to acquire some level of access to a Trust Authority KeyStore has failed.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>SOAdd</td>
<td>Indicates that an Audit Administrator has been added to the system.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>SODelete</td>
<td>Indicates that an Audit Administrator has been deleted from the system.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>AudEventMaskChange</td>
<td>Indicates that the mask that defines the set of Audit events that a client can send to the server has changed.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CACrossCertRequest</td>
<td>Indicates that a cross-certified CA certificate has been requested.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>CAHierarchicalCertRequest</td>
<td>Indicates that a hierarchically certified CA certificate has been requested.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>SubmitRevocationRequest</td>
<td>Indicates that a registrar (RA administrator) has submitted a revocation request.</td>
<td>Optional</td>
</tr>
<tr>
<td>RequestSetRenewable</td>
<td>Indicates that a registrar has requested that a certificate be set to renewable status.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>RequestSetNonRenewable</td>
<td>Indicates that a registrar has requested that a certificate be set to non-renewable status.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>RequestPend</td>
<td>Indicates that a registrar has requested that a credential request be set to pending status.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>ReceiptOfCertRequest</td>
<td>Indicates that a certificate request has been received by an RA.</td>
<td>Optional</td>
</tr>
<tr>
<td>ReceiptOfRevocationRequest</td>
<td>Indicates that a request to revoke a certificate has been received by an RA.</td>
<td>Optional</td>
</tr>
<tr>
<td>ReceiptOfRenewalRequest</td>
<td>Indicates that a request to renew a certificate has been received by an RA.</td>
<td>Optional</td>
</tr>
<tr>
<td>RequestApproval</td>
<td>Indicates that an RA has approved a certificate request.</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>
### Table 18. Audit events (continued)

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
<th>Mandatory or optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestRejection</td>
<td>Indicates that an RA has rejected a certificate request.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>RequestCompletion</td>
<td>Indicates that a request for a certificate has completed.</td>
<td>Optional</td>
</tr>
<tr>
<td>Preregistration</td>
<td>Indicates that a registrar has received a preregistration request.</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

### Audit database data

The Trust Authority Audit database uses a schema that is based on the recommendations that are described in the Public Key Cryptography for the Financial Services Industry standard, X9.57. This section describes the following database tables.

**Note:** In this section, the field names for the audit records correspond to the column names in the database tables.

- **Keys**
  This is a control table for private/secret keys used for signing, encrypting, and generating message authentication codes (MACs).

- **Event Severities (event_severities)**
  This table describes all the event severities.

- **Event Control (event_ctl)**
  This table describes all the events that can be audited.

- **Sources**
  This table holds a list of all event sources.

- **Authorized Entities (auth_entities)**
  This table holds a list of all authorized entities.

- **Affected Entity Types (afctd_entity_types)**
  This table holds a list of all authorized roles.

- **Authorized Roles (auth_roles)**
  This table holds a list of all roles for authorized entities.

- **Component Types (component_types)**
  This table holds a list of all component types.

- **Audit Log (audit_log)**
  This is the main table for audit log records.

- **System**
  This table contains information that applies to the entire audit system. It has only one row.
Keys table

The Keys table maintains information about all that cryptographic keys that are used by the Audit subsystem.

Table 19. Keys table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>key_id</td>
<td>The unique internal identifier of the key.</td>
<td>smallint</td>
</tr>
<tr>
<td>alg_oid</td>
<td>The algorithm associated with the key.</td>
<td>varchar</td>
</tr>
<tr>
<td>label</td>
<td>The KeyStore label or some token used to locate the real key.</td>
<td>varchar</td>
</tr>
<tr>
<td>integrity</td>
<td>This field is used to maintain the integrity of the record.</td>
<td>varchar for bit data</td>
</tr>
</tbody>
</table>

Event Severities table

This table maintains information about event severities. It is a read-only table that is loaded during installation and configuration.

Table 20. Event Severities table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>severity_id</td>
<td>The unique internal identifier of the event severity.</td>
<td>smallint</td>
</tr>
<tr>
<td>severity_desc</td>
<td>The national language support (NLS) string describing the severity.</td>
<td>varchar</td>
</tr>
</tbody>
</table>

Event Control table

This table maintains information about all the events that an audit client can send to the audit server. It is a read-only table that is loaded during installation and configuration.

Table 21. Event Control table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>event_id</td>
<td>The unique identifier of the event.</td>
<td>smallint</td>
</tr>
<tr>
<td>event_desc</td>
<td>A viewable description of the event. This field may be used for display purposes.</td>
<td>varchar</td>
</tr>
<tr>
<td>event_key</td>
<td>A short unique string that identifies and describes the event. This is used by the Java resource bundles.</td>
<td>varchar</td>
</tr>
<tr>
<td>event_severity_id</td>
<td>The severity ID of this event. This is the foreign key that references the event_severities table.</td>
<td>smallint</td>
</tr>
</tbody>
</table>

Sources table

This table maintains the list of all the audit clients. An audit client is a Trust Authority component that generates audit events.
Table 22. Sources table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>source_id</td>
<td>The unique identifier of the audit client.</td>
<td>smallint</td>
</tr>
<tr>
<td>source</td>
<td>The identifier of the audit client that can be used for display purposes. In most cases, this is the DN of the audit client.</td>
<td>varchar</td>
</tr>
<tr>
<td>integrity</td>
<td>This field is used to maintain the integrity of the record.</td>
<td>varchar for bit data</td>
</tr>
</tbody>
</table>

Authorized Entities table

This table maintains a list of all the authorized entities. An authorized entity is the entity that authorized the operation for which an audit event was generated.

Table 23. Authorized Entities table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_entity_id</td>
<td>The unique internal identifier of the authorized entity.</td>
<td>smallint</td>
</tr>
<tr>
<td>auth_entity_desc</td>
<td>The identifier of the authorized entity that can be used for display purposes. In most cases, this is the DN of the authorized entity.</td>
<td>varchar</td>
</tr>
<tr>
<td>integrity</td>
<td>This field is used to maintain the integrity of the record.</td>
<td>varchar for bit data</td>
</tr>
</tbody>
</table>

Authorized Roles table

This table maintains information about the roles of authorized entities. It is a read-only table that is loaded during installation and configuration.

Table 24. Authorized Roles table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_role_id</td>
<td>The unique internal identifier of the authorized role.</td>
<td>smallint</td>
</tr>
<tr>
<td>auth_role_desc</td>
<td>The NLS string describing the authorized role.</td>
<td>varchar</td>
</tr>
</tbody>
</table>

Affected Entity Types table

This table maintains information about the various types of affected entities. An affected entity is the entity that is affected by the operation for which an audit event is generated. The read-only table is loaded during installation and configuration.

Table 25. Affected Entity Types table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>afctd_entity_id</td>
<td>The unique internal identifier of the affected entity type.</td>
<td>smallint</td>
</tr>
</tbody>
</table>
Table 25. Affected Entity Types table fields (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>afctd_entity_desc</td>
<td>The NLS string describing the affected entity type.</td>
<td>varchar</td>
</tr>
</tbody>
</table>

Component Types table

This table maintains information about the component types of the audit clients. It is a read-only table that is loaded during installation and configuration.

Table 26. Component Types table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>component_type_id</td>
<td>The unique internal identifier of the component type.</td>
<td>smallint</td>
</tr>
<tr>
<td>component_desc</td>
<td>The NLS string describing the component type.</td>
<td>varchar</td>
</tr>
</tbody>
</table>

Audit Log table

This table contains the audit records.

Table 27. Audit Log table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>serial_num</td>
<td>The unique serial number of the audit record.</td>
<td>smallint</td>
</tr>
<tr>
<td>src_date_time</td>
<td>The timestamp specifying when the event was generated by the source (audit client).</td>
<td>timestamp</td>
</tr>
<tr>
<td>cr_date_time</td>
<td>The timestamp when the audit record was created by the audit server.</td>
<td>timestamp</td>
</tr>
<tr>
<td>event_id</td>
<td>The internal identifier of the event. This is a foreign key that references the event_ctl table.</td>
<td>smallint</td>
</tr>
<tr>
<td>source_id</td>
<td>The internal identifier of the source that generated this event. This is a foreign key that references the sources table.</td>
<td>smallint</td>
</tr>
<tr>
<td>component_type_id</td>
<td>The internal identifier of the component type of the source that generated this event. This is a foreign key that references the component_types table.</td>
<td>smallint</td>
</tr>
<tr>
<td>auth_entity_id</td>
<td>The internal identifier of the entity that authorized the event. This is a foreign key that references the auth_entities table.</td>
<td>smallint</td>
</tr>
</tbody>
</table>
### Table 27. Audit Log table fields (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_role_id</td>
<td>The internal identifier of the role of the entity that authorized the event. This is a foreign key that references the auth_entities table.</td>
<td>smallint</td>
</tr>
<tr>
<td>afctd_entity</td>
<td>The name or DN of the type of the entity affected by the event.</td>
<td>varchar</td>
</tr>
<tr>
<td>afctd_entity_id</td>
<td>The internal identifier of the type of the entity affected by the event.</td>
<td>smallint</td>
</tr>
<tr>
<td>storage_media</td>
<td>The storage media associated with the audit event.</td>
<td>varchar</td>
</tr>
<tr>
<td>extra_info</td>
<td>Additional information associated with the audit event.</td>
<td>varchar</td>
</tr>
<tr>
<td>sig_key_id</td>
<td>The internal identifier of the key used to generate the integrity field. This is a foreign key that references the keys table.</td>
<td>smallint</td>
</tr>
<tr>
<td>enc_key_id</td>
<td>The internal identifier of the key used to encrypt selected fields in this record. This is a foreign key that references the keys table. In the current release of Trust Authority, none of the fields are encrypted.</td>
<td>smallint</td>
</tr>
<tr>
<td>integrity</td>
<td>This field is used to maintain the integrity of the record.</td>
<td>varchar for bit data</td>
</tr>
</tbody>
</table>

### System table

This table maintains state information about the Audit database.

### Table 28. System table fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>first_sn</td>
<td>The serial number of the first audit record in the audit_log.</td>
<td>integer</td>
</tr>
<tr>
<td>next_sn</td>
<td>The serial number of the next audit record in the audit_log.</td>
<td>integer</td>
</tr>
<tr>
<td>audit_int</td>
<td>Used to maintain the integrity of the audit_log table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>archive_int</td>
<td>Used to maintain the integrity of the archive_ctl table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>events_int</td>
<td>Used to maintain the integrity of the events_ctl table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>auth_ent_int</td>
<td>Used to maintain the integrity of the auth_entities table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>auth_role_int</td>
<td>Used to maintain the integrity of the auth_roles table.</td>
<td>varchar for bit data</td>
</tr>
</tbody>
</table>
Table 28. System table fields (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>sources_int</td>
<td>Used to maintain the integrity of the sources table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>afctd_ent_type_int</td>
<td>Used to maintain the integrity of the afctd_entities table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>keys_int</td>
<td>Used to maintain the integrity of the Keys table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>event_sevs_int</td>
<td>Used to maintain the integrity of the Event Severities table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>comp_types_int</td>
<td>Used to maintain the integrity of the Component Types table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>system_int</td>
<td>Used to maintain the integrity of the System table.</td>
<td>varchar for bit data</td>
</tr>
<tr>
<td>sig_key_id</td>
<td>The internal identifier of the key used to generate the integrity fields of this record. This is a foreign key that references the keys table.</td>
<td>smallint</td>
</tr>
</tbody>
</table>

Troubleshooting

The default installation of Trust Authority does not generate log files for the verbose or debug level of error messages. This section provides basic steps for troubleshooting problems with debug-level messaging disabled. Then, if you need more information, you can enable debug messaging using the procedure in Troubleshooting with debug-level messaging enabled.

Basic troubleshooting

To troubleshoot the problems that you encounter while administering Trust Authority, follow these basic steps:

1. Stop the Trust Authority system.
2. Reboot the machine.
3. Start the Trust Authority system.
4. Check the status of each component to make sure it started completely and is running.
5. Reproduce the problem and check the logs of each component to locate the error.

Troubleshooting with debug-level messaging enabled

If you encounter a problem and need detailed assistance to resolve it, enable the debug-level environment variable. You can adjust this value as needed to obtain lesser or greater message detail. The logging level specified in the debug-level variable determines the amount of logging performed. The value is a positive integer with a useful range of 0 to 1000. The higher the level the greater quantity of logging. Following are the recommended Trust Authority debug levels:

- DebugLevel=25 - minimal logging, but all errors are logged.

Use this level for steady operations where no problems are actively being diagnosed, but you want to be able to view the logs in case something does go wrong.
- **DebugLevel=75** - includes additional logging from lower software layers. Use this level when there are problems between the CA and RA servers. The volume of output is approximately five times greater than at debug level 25.

- **DebugLevel=101** - includes extensive and detailed logging of higher-level RA server processing including certificate profile processing and application framework (AFW) interactions. Use this when you are trying to diagnose problems in the RA server’s certificate processing layers. The output volume is approximately ten times greater than at debug level 25.

- **DebugLevel=201** - extensive tracing of heavily used critical internal functions. Use this level only when you are trying to diagnose problems like abnormal terminations of the RA server. The volume of output is approximately 100 times greater then at debug level 25.

**Note:** Because the logging volume is so great at level 100, you should not use this setting in a production environment except under special circumstances. At this debugging level or higher, it is possible to exhaust the amount of free space available in the file system thereby causing numerous failures throughout the system.

To troubleshoot with debug-level messaging enabled, follow these steps (this procedure uses 100 as a sample debug-level value):

1. Stop the Trust Authority system.
2. Modify the irgAutoCa.ini file as follows:
   - In AIX, set the debug parameter between 50 and 100 in /usr/lpp/iau/etc/TrustAuthority/irgAutoCa.ini
   - In Windows NT, set the debug parameter between 50 and 100 in C:\Program Files\IBM\Trust Authority\etc\TrustAuthority\irgAutoCa.ini
3. Enable the debug-level environment variable, as follows:
   - In AIX, enter the following command on the command line of the machine where the RA server resides:
     ```bash
     export DebugLevel=
     ```
   - In Windows NT:
     a. Invoke System Properties by right-clicking on the **My Computer** icon of Windows NT Explorer and selecting **Properties**.
     b. Click on the **Environment** tab.
     c. Enter **DebugLevel** in the **Variable** field and **100** in the **Value** field.
     d. Click on **Set** and then click on **Apply**.
     e. Click **OK** to close the System Properties window.
4. Start the Trust Authority system.
5. Check the status of each component to make sure it started completely and is running.
6. Reproduce the problem and check the logs of each component to locate the error. Log files named irgrasvr.log.nnnn are created, where nnnn is a randomly generated number.
On AIX, the default location of the log files is
/usr/lpp/iau/etc/TrustAuthority/caSS.log.xxxxx

On Windows NT, the default location is c:\Program Files\IBM\Trust
Authority\etc\TrustAuthority\caSS.log.xxxxx
Notices

This information was developed for products and services offered in the U.S.A. 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 in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (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 information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information 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
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

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 document 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM’s suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

**Trademarks and service marks**

The following terms are trademarks of International Business Machines Corporation or Tivoli Systems, Inc. in the United States, or other countries, or both:

IBM
AIX
AIX/6000
DB2
DB2 Universal Database
RISC System/6000
RS/6000
SecureWay
Tivoli
WebSphere
The Trust Authority program ("the Program") includes portions of the IBM WebSphere Application Server and the IBM HTTP Web Server ("IBM Servers"). You are not authorized to install or use the IBM Servers other than in connection with your licensed use of the Program. The IBM Servers must reside on the same machine as the Program, and you are not authorized to install or use the IBM Servers separate from the Program.

The Program includes portions of DB2 Universal Database. You are authorized to install and use these components only in association with your licensed use of the Program and IBM WebSphere Application Server for the storage and management of data used or generated by the Program and IBM WebSphere Application Server, and not for other data management purposes. For example, this license does not include inbound connections to the database from other applications for queries or report generation. You are authorized to install and use these components only with and on the same machine as the Program.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

Pentium is a trademark of Intel Corporation in the United States, other countries, or both.

This program contains security software from RSA Data Security, Inc. Copyright © 1994 RSA Data Security, Inc. All rights reserved.

This program contains Standard Template Library (STL) software from Hewlett-Packard Company. Copyright (c) 1994.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

This program contains Standard Template Library (STL) software from Silicon Graphics Computer Systems, Inc. Copyright (c) 1996–1999.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Other company, product, and service names may be trademarks or service marks of others.
Related information

The Trust Authority product documentation is available in Portable Document Format (PDF) and HTML format on the IBM SecureWay Trust Authority Documentation CD-ROM. HTML versions of some publications are installed with the product and are accessible from the user interfaces.

Be aware that the product may have changed since the publications were produced. For the latest product information, and for information about accessing a publication in the language and format of your choice, see the Readme file. The latest version of the Readme file is available on the IBM SecureWay Trust Authority Web site: http://www.tivoli.com/support

The Trust Authority library includes the following documentation:

Up and Running
This book provides an overview of the product. It lists the product requirements, includes installation procedures, and provides information about how to access the online help available for each product component. This book is printed and distributed with the product.

System Administration Guide
This book contains general information about administering the Trust Authority system. It includes procedures for starting and stopping the servers, changing passwords, administering the server components, performing audits, and running data integrity checks.

Configuration Guide
This book contains information about how to use the Setup Wizard to configure a Trust Authority system. You can access the HTML version of this guide while viewing online help for the Wizard.

Registration Authority Desktop Guide
This book contains information about how to use the RA Desktop to administer certificates throughout the certificate life cycle. You can access the HTML version of this guide while viewing online help for the Desktop.

User’s Guide
This book contains information about how to obtain and manage certificates. It provides procedures for using the Trust Authority browser enrollment forms to request, renew, and revoke certificates. It also discusses how to preregister for PKIX-compliant certificates, and how to use the Trust Authority Client to manage these certificates. You can access the HTML version of this guide while viewing online help for the Client.

Customization Guide
This book shows you how to customize the Trust Authority registration facility to support the registration and certification goals of your business policies. For example, you can learn how to customize HTML and Java Server pages, notification letters, certificate profiles, and policy exits.

Public Key Infrastructure System Administration Guide 117
The Trust Authority Web site includes other documents that may help you install, administer, and use Trust Authority. For example, you can find supplemental guidelines on the Directory schema and learn how to integrate Trust Authority with the IBM SecureWay 4758 PCI Coprocessor.
Glossary

This glossary defines the terms and abbreviations in this book that may be new or unfamiliar and terms that may be of interest. It includes terms and definitions from:


Numbers

4758 PCI Cryptographic Coprocessor
A programmable, tamper-responding cryptographic PCI-bus card offering high performance DES and RSA cryptographic processing. The cryptographic processes occur within a secure enclosure on the card. The card meets the stringent requirements of the FIPS PUB 140-1 level 4 standard. Software can run within the secure enclosure. For example, credit card transaction processing can use the SET™ standard.

A

Abstract Syntax Notation One (ASN.1)
An ITU notation that is used to define the syntax of information data. It defines a number of simple data types and specifies a notation for identifying these types and for specifying values of these types. These notations can be applied whenever it is necessary to define the abstract syntax of information without curbing how the information is encoded for transmission.

access control list (ACL)
A mechanism for limiting the use of a specific resource to authorized users.

ACL
Access control list.

action history
Accumulated events in the life cycle of a credential.

American National Standard Code for Information Interchange (ASCII)
The standard code that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set uses a coded character set that consists of 7-bit coded characters (8 bits including a bit for parity checking). The character set consists of control characters and graphic characters.

American National Standards Institute (ANSI)
An organization that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. It consists of producers, consumers, and general interest groups.

ANSI
American National Standards Institute.

applet
A computer program that is written in Java® and runs inside a Java-compatible Web browser. Also known as a Java applet.

ASCII

ASN.1
Abstract Syntax Notation One.
asymmetric cryptography
Cryptography that uses different, asymmetric keys for encryption and decryption. Each user receives a pair of keys: a public key accessible to all, and a private key known only to the user. A secure transaction can occur when the public key and the corresponding private key match, enabling the decryption of the transaction. This is also known as key pair cryptography. Contrast with symmetric cryptography.

asynchronous communication
A mode of communication that does not require the sender and recipient to be present simultaneously.

audit client
Any client in the system that sends audit events to the Trust Authority Audit server. Before an audit client sends an event to the Audit server, it establishes a connection with the Audit server. After the connection is established, the client uses the audit subsystem client library to deliver events to the Audit server.

audit log
In Trust Authority, a table in a database that stores one record per audit event.

Audit server
A Trust Authority server that receives audit events from audit clients and writes them to an audit log.

audit subsystem
In Trust Authority, a subsystem that provides the support for logging security-relevant actions. It conforms to recommendations in standard X9.57, of the standards set forth in Public Key Cryptography for the Financial Services Industry.

audit trail
Data, in the form of a logical path, that links a sequence of events. An audit trail enables tracing of transactions or the history of a given activity.

authentication
The process of reliably determining the identity of a communicating party.

authorization
Permission to access a resource.

base64 encoding
A common means of conveying binary data with MIME.

Basic Encoding Rules (BER)
The rules specified in ISO 8825 for encoding data units described in abstract syntax notation 1 (ASN.1). The rules specify the encoding technique, not the abstract syntax.

BER
Basic Encoding Rules.

browser
See Web browser.

browser certificate
A digital certificate is also known as a client-side certificate. It is issued by a CA through an SSL-enabled Web server. Keys in an encrypted file enable the holder of the certificate to encrypt, decrypt, and sign data. Typically, the Web browser stores these keys. Some applications permit storage of the keys on smart cards or other media. See also digital certificate.

business process objects
A set of code used to accomplish a specific registration operation, such as checking the status of an enrollment request or verifying that a public key was sent.

business process template
A set of business process objects that are run in a specified order.
bytecode
Machine-independent code that is generated by the Java compiler and run by the Java interpreter.

CA
Certificate authority.

CA certificate
A certificate your Web browser accepts, at your request, from a CA it does not recognize. The browser can then use this certificate to authenticate communications with servers that hold certificates issued by that CA.

CA hierarchy
In Trust Authority, a trust structure whereby one CA is located at the top of the structure and up to four layers of subordinate CAs are located below. When users or servers are registered with a CA, they receive a certificate signed that is by that CA, and they inherit the certification hierarchy of the layers above.

CA server
The server for the Trust Authority Certificate Authority (CA) component.

CAST-64
A block cipher algorithm that uses a 64-bit block size and a 6-bit key. It was designed by Carlisle Adams and Stafford Tavares.

CCA
IBM Common Cryptographic Architecture.

CDSA
Common Data Security Architecture.

certificate authority (CA)
The software responsible for following an organization’s security policies and assigning secure electronic identities in the form of certificates. The CA processes requests from RAs to issue, renew, and revoke certificates. The CA interacts with the RA to publish certificates and CRLs in the Directory. See also digital certificate.

certificate extension
An optional feature of the X.509v3 certificate format that provides for the inclusion of additional fields in the certificate. There are standard extensions and user-defined extensions. Standard extensions exist for various purposes, including key and policy information, subject and issuer attributes, and certification path constraints.

certificate policy
A named set of rules that indicates the applicability of a certificate to a particular class of applications that have common security requirements. For example, a certificate policy might indicate whether a particular certification type allows a user to conduct transactions for goods within a given price range.

certificate profile
A set of characteristics that define the type of certificate wanted (such as SSL certificates or IPSec certificates). The profile aids in managing certificate specification and registration. The issuer can change the names of the profiles and specify characteristics of the desired certificate, such as the validity period, key usage, DN constraints, and so forth.

certificate revocation list (CRL)
A digitally signed, time-stamped list of certificates that the certificate authority has revoked. The certificates in this list should be considered unacceptable. See also digital certificate.

certification
The process during which a trusted third party issues an electronic credential that vouches for an individual, business, or organizational identity.

CGI
Common Gateway Interface.
chain validation
The validation of all CA signatures in the trust hierarchy through which a given certificate was issued. For example, if a CA was issued its signing certificate by another CA, both signatures are validated during validation of the certificate that the user presents.

class
In object-oriented design or programming, a group of objects that share a common definition and therefore share common properties, operations, and behavior.

cleartext
Data that is not encrypted. Synonym for plaintext.

client
(1) A functional unit that receives shared services from a server. (2) A computer or program that requests a service of another computer or program.

client/server
A model in distributed processing in which a program at one site sends a request to a program at another site and waits for a response. The requesting program is called a client; the answering one is called a server.

code signing
A technique for signing executable programs with digital signatures. Code signing is designed to improve the reliability of software that is distributed over the Internet.

Common Cryptographic Architecture (CCA)
IBM software that enables a consistent approach to cryptography on major IBM computing platforms. It supports application software that is written in a variety of programming languages. Application software can call on CCA services to perform a broad range of cryptographic functions, including DES and RSA encryption.

Common Data Security Architecture (CDSA)
An initiative to define a comprehensive approach to security service and security management for computer-based security applications. It was designed by Intel, to make computer platforms more secure for applications.

Common Gateway Interface (CGI)
Standard method of transmitting information between Web pages and Web servers.

confidentiality
The property of not being divulged to unauthorized parties.

credential
Confidential information used to prove one’s identity in an authentication exchange. In environments for network computing, the most common type of credential is a certificate that a CA has created and signed.

CRL
Certificate revocation list.

CRL publication interval
Set in the CA configuration file, the interval of time between periodic publications of the CRL to the Directory.

cross-certification
A trust model whereby one CA issues to another CA a certificate that contains the public key associated with its private signature key. A cross-certified certificate allows client systems or end entities in one administrative domain to communicate securely with client systems or end entities in another domain.

cryptographic
Pertaining to the transformation of data to conceal its meaning.

cryptography
In computer security, the principles, means, and methods for encrypting plaintext and decrypting encrypted text.
daemon
A program that carries out tasks in the background. It is implicitly called when a condition occurs that requires its help. A user need not be aware of a daemon, because the system usually spaws it automatically. A daemon might live forever or the system might regenerate it at intervals. The term (pronounced demon) comes from mythology. Later, it was rationalized as the acronym DAEMON: Disk And Execution MONitor.

Data Encryption Standard (DES)
An encryption block cipher, defined and endorsed by the U.S. government in 1977 as an official standard. IBM developed it originally. DES has been extensively studied since its publication and is a well-known and widely used cryptographic system. DES is a symmetric cryptographic system. When it is used for communication, both the sender and receiver must know the same secret key. This key is used to encrypt and decrypt the message. DES can also be used for single-user encryption, such as to store files on a hard disk in encrypted form. DES has a 64-bit block size and uses a 56-bit key during encryption. It is was originally designed for implementation in hardware. NIST has recertified DES as an official U.S. government encryption standard every five years.

Data Storage Library (DL)
A module that provides access to persistent data stores of certificates, CRLs, keys, policies, and other security-related objects.

decrypt
To undo the encryption process.

DEK
Document encrypting key.

DER
Distinguished Encoding Rules.

DES
Data Encryption Standard.

Diffie-Hellman
A method of establishing a shared key over an insecure medium, named after the inventors (Diffie and Hellman).

digital certificate
An electronic credential that is issued by a trusted third party to a person or entity. Each certificate is signed with the private key of the CA. It vouches for an individual, business, or organizational identity. Depending on the role of the CA, the certificate can attest to the authority of the bearer to conduct e-business over the Internet. In a sense, a digital certificate performs a similar role to a driver’s license or a medical diploma. It certifies that the bearer of the corresponding private key has authority to conduct certain e-business activities. A certificate contains information about the entity it certifies, whether person, machine, or computer program. It includes the certified public key of that entity.

digital certification
See certification.

digital signature
A coded message added to a document or data that guarantees the identity of the sender. A digital signature can provide a greater level of security than a physical signature. The reason for this is that a digital signature is not an encrypted name or series of simple identification codes. Instead, it is an encrypted summary of the message that is being signed. Thus, affixing a digital signature to a message provides solid identification of the sender. (Only the sender’s key can create the signature.) It also fixes the content of the message that is being signed (the encrypted message summary must match the message content or the signature is not valid). Thus, a digital signature cannot be copied from one message and applied to another because the summary, or hash, would not match. Any alterations to the signed message would also invalidate the signature.
Digital Signature Algorithm (DSA)
A public key algorithm that is used as part of the Digital Signature Standard. It cannot be used for encryption, only for digital signatures.

Directory
A hierarchical structure intended as a global repository for information related to communications (such as e-mail or cryptographic exchanges). The Directory stores specific items that are essential to the PKI structure, including public keys, certificates, and certificate revocation lists.
Data in the Directory is organized hierarchically in the form of a tree, with the root at the top of the tree. Often, higher level organizations represent individual countries, governments, or companies. Users and devices are typically represented as leaves of each tree. These users, organizations, localities, countries, and devices each have their own entry. Each entry consists of typed attributes. These provide information about the object that the entry represents.
Each entry in the Directory is bound with an associated distinguished name (DN). This is unique when the entry includes an attribute that is known to be unique to the real world object. Consider the following example DN. In it, the country (C) is US, the organization (O) is IBM, the organizational unit (OU) is Trust, and the common name (CN) is CA1.

C=US/O=IBM/OU=Trust/CN=CA1

Directory server
In Trust Authority, the IBM SecureWay Directory. This Directory supports LDAP standards and uses DB2 as its base.

Distinguished Encoding Rules (DER)
Provides constraints on the BER. DER selects just one type of encoding from those that the encoding rules allow, eliminating all of the sender’s options.

distinguished name (DN)
The unique name of a data entry that is stored in the Directory. The DN uniquely identifies the position of an entry in the hierarchical structure of the Directory.

dl
Data Storage Library.
dn
Distinguished name.
document encrypting key (DEK)
Typically, a symmetric encryption/decryption key, such as DES.
domain
See security domain and registration domain.

DSA
Digital Signature Algorithm.

e-business
Business transactions over networks and through computers. It includes buying and selling goods and services. It also includes transferring funds through digital communications.

e-commerce
Business-to-business transactions. It includes buying and selling goods and services (with customers, suppliers, vendors, and others) on the Internet. It is a primary element of e-business.

end-entity
The subject of a certificate that is not a CA.

encrypt
To scramble information so that only someone who has the appropriate decryption code can obtain the original information through decryption.
encryption/decryption
Using the public key of the intended recipient to encipher data for that person, who then uses the private key of the pair to decipher the data.

enrollment
In Trust Authority, the process of obtaining credentials for use over the Internet. Enrollment encompasses the requesting, renewing, and revoking of certificates.

enrollment attribute
An enrollment variable that is contained in an enrollment form. Its value reflects the information that is captured during the enrollment. The value of the enrollment attribute remains the same throughout the lifetime of the credential.

enrollment variable
See enrollment attribute.

extranet
A derivative of the Internet that uses similar technology. Companies are beginning to apply Web publishing, electronic commerce, message transmission, and groupware to multiple communities of customers, partners, and internal staff.

F
File Transfer Protocol (FTP)
An Internet client/server protocol for use in transferring files between computers.

firewall
A gateway between networks that restricts the flow of information between networks. Typically, the purpose of a firewall is to protect internal networks from unauthorized use from the outside.

FTP
File Transfer Protocol.

G
gateway
A functional unit that allows incompatible networks or applications to communicate with each other.

H
hierarchy
The organization of Certificate Authorities (CA) in a trust chain, starting with the self-signed CA or root of roots at the top, and ending with the CA that issues certificates to end users.

HTML
Hypertext Markup Language.

HTTP
Hypertext Transaction Protocol.

HTTP server
A server that handles Web-based communications with browsers and other programs in a network.

hypertext
Text that contains words, phrases, or graphics that the reader can click with the mouse to retrieve and display another document. These words, phrases, or graphics are known as hyperlinks. Retrieving them is known as linking to them.

Hypertext Markup Language (HTML)
A markup language for coding Web pages. It is based on SGML.
Hypertext Transaction Protocol (HTTP)
An Internet client/server protocol for transferring hypertext files across the Web.

ICL
Issued certificate list.

IETF (Internet Engineering Task Force)
A group that focuses on engineering and developing protocols for the Internet. It represents an international community of network designers, operators, vendors, and researchers. The IETF is concerned with the development of the Internet architecture and the smooth use of the Internet.

IniEditor
In Trust Authority, a tool used to edit configuration files.

instance
In DB2, an instance is a logical database management environment for storing data and running applications. It allows definition of a common set of configuration parameters for multiple databases.

integrity
A system protects the integrity of data if it prevents unauthorized modification (as opposed to protecting the confidentiality of data, which prevents unauthorized disclosure).

integrity checking
The checking of audit records that result from transactions with external components.

internal structure
See schema.

International Standards Organization (ISO)
An international organization tasked with developing and publishing standards for everything from wine glasses to computer network protocols.

International Telecommunication Union (ITU)
An international organization within which governments and the private sector coordinate global telecommunication networks and services. It is the leading publisher of telecommunication technology, regulatory, and standards information.

Internet
A worldwide collection of networks that provide electronic connection between computers. This enables them to communicate with each other via software devices such as electronic mail or Web browsers. For example, some universities are on a network that in turn links with other similar networks to form the Internet.

intranet
A network within an enterprise that usually resides behind firewalls. It is a derivative of the Internet and uses similar technology. Technically, intranet is a mere extension of the Internet. HTML and HTTP are some of the commonalties.

IPSec
An Internet Protocol Security standard, developed by the IETF. IPSec is a network layer protocol, designed to provide cryptographic security services that flexibly support combinations of authentication, integrity, access control, and confidentiality. Because of its strong authentication features, it has been adopted by many VPN product vendors as the protocol for establishing secure point-to-point connections over the Internet.

ISO
International Standards Organization.

issued certificate list (ICL)
A complete list of the certificates that have been issued and their current status. Certificates are indexed by serial number and state. This list is maintained by the CA and stored in the CA database.
ITU
International Telecommunication Union.

J

Java
A set of network-aware, non-platform-specific computer technologies developed by Sun Microsystems, Incorporated. The Java environment consists of the Java OS, the virtual machines for various platforms, the object-oriented Java programming language, and several class libraries.

Java applet
See applet. Contrast with Java application.

Java application
A stand-alone program that is written in the Java language. It runs outside the context of a Web browser.

Java class
A unit of Java program code.

Java language
A programming language, developed by Sun Microsystems, designed specifically for use in applet and agent applications.

Java Virtual Machine (JVM)
The part of the Java run-time environment responsible for interpreting bytecodes.

K

key
A quantity used in cryptography to encipher or decipher information.

Key Backup and Recovery
This feature of Trust Authority enables you to backup and recover the end entity certificates and their corresponding public and private keys certified by Trust Authority. The certificate and keys are stored in a PKCS #12 file. This file is protected by a password. The password is set at the time the certificate and keys are backed up.

key pair
Corresponding keys that are used in asymmetric cryptography. One key is used to encrypt and the other to decrypt.

KeyStore
A DL for storing Trust Authority component credentials, such as keys and certificates, in an encrypted format.

L

LDAP

Lightweight Directory Access Protocol (LDAP )
A protocol used to access the Directory.

M

MAC
Message authentication code.

MD2
A 128-bit message-digest hash function, designed by Ron Rivest. It is used with MD5 in the PEM protocols.

MD4
A 128-bit message-digest hash function, designed by Ron Rivest. It is several times faster than MD2.
MD5
A one-way message-digest hash function, designed by Ron Rivest. It is an improved version of MD4. MD5 processes input text in 512-bit blocks, divided into 16 32-bit sub-blocks. The output of the algorithm is a set of four 32-bit blocks, which concatenate to form a single 128-bit hash value. It is also used along with MD2 in the PEM protocols.

message authentication code (MAC)
A secret key that is shared between the sender and the recipient. The sender authenticates, and the recipient verifies. In Trust Authority, MAC keys are stored in the KeyStores for the CA and Auditing components.

message digest
An irreversible function that takes an arbitrary-sized message and produces a fixed length quantity. MD5 is an example of a message digest algorithm.

MIME (Multipurpose Internet Mail Extensions)
A freely available set of specifications that allows the interchange of text in languages with different character sets. It also allows multimedia e-mail among many different computer systems that use Internet mail standards. For example, the e-mail messages may contain character sets other than US-ASCII, enriched text, images, and sounds.

modulus
In the RSA public key cryptographic system, the product (n) of two large primes: p and q. The best size for an RSA modulus depends on one’s security needs. The larger the modulus, the greater the security. The current RSA Laboratories–recommended key sizes depend on the planned use for the key: 768 bits for personal use, 1024 bits for corporate use, and 2048 bits for extremely valuable keys like the key pair of a CA. A 768-bit key is expected to be secure until at least the year 2004.

N

National Language Support (NLS)
Support within a product for differences in locales, including language, currency, date and time format, and numeric presentation.

National Security Agency (NSA)
The official security body of the U.S. government.

NIST
National Institute of Standards and Technology, formerly known as NBS (National Bureau of Standards). It promotes open standards and interoperability in computer-based industries.

NLS
National language support.

nonce
A string that is sent down from a server or application, requesting user authorization. The user that is asked for authentication signs the nonce with a private key. The user’s public key and the signed nonce are sent back to the server or application that requested authentication. The server then attempts to decipher the signed nonce with the user’s public key. If the deciphered nonce is the same as the original nonce that was sent, the user is authenticated.

non-repudiation
The use of a digital private key to prevent the signer of a document from falsely denying having signed it.

NSA
National Security Agency.

O

object
In object-oriented design or programming, an abstraction encapsulating data and the operations associated with that data. See also class.
object identifier (OID)
An administratively assigned data value of the type defined in abstract syntax notation 1 (ASN.1).

object type
The kind of object that can be stored in the Directory. For example, an organization, meeting room, device, person, program, or process.

ODBC
Open Database Connectivity.

Open Database Connectivity (ODBC)
A standard for accessing different database systems.

Open Systems Interconnect (OSI)
The name of the computer networking standards that the ISO approved.

OSI
Open Systems Interconnect.

PC card
Similar to a smart card, and sometimes called a PCMCIA card. This card is somewhat larger than a smart card and usually has a greater capacity.

PEM
Privacy-enhanced mail.

PKCS
Public Key Cryptography Standards.

PKCS #1
See Public Key Cryptography Standards.

PKCS #7
See Public Key Cryptography Standards.

PKCS #10
See Public Key Cryptography Standards.

PKCS #11
See Public Key Cryptography Standards.

PKCS #12
See Public Key Cryptography Standards.

PKI
Public key infrastructure.

PKIX
An X.509v3-based PKI.

PKIX certificate management protocol (CMP)
A protocol that enables connections with PKIX-compliant applications. PKIX CMP uses TCP/IP as its primary transport mechanism, but an abstraction layer over sockets exists. This enables support for additional polling transports.

PKIX CMP
PKIX certificate management protocol.

PKIX listener
The public HTTP server that a particular registration domain uses to listen for requests from the Trust Authority Client application.
plaintext
Unencrypted data. Synonym for cleartext.

policy exit
In a registration facility, an organization-defined program that is called by the registration application. The rules specified in a policy exit apply the organization’s business and security preferences to the enrollment process.

preregistration
In Trust Authority, a process that allows one user, typically an administrator, to enroll other users. If the request is approved, the RA provides information that allows the user to obtain the certificate at a later time using the Trust Authority Client application.

privacy
Protection from the unauthorized disclosure of data.

privacy-enhanced mail (PEM)
The Internet privacy-enhanced mail standard, that the Internet Architect Board (IAB) adopted to provide secure electronic mail over the Internet. The PEM protocols provide for encryption, authentication, message integrity, and key management.

private key
The key in a public/private key pair that is available only to its owner. It enables the owner to receive a private transaction or make a digital signature. Data signed with a private key can be verified only with the corresponding public key. Contrast with public key. See also public/private key pair.

protocol
An agreed-on convention for inter-computer communication.

proxy server
An intermediary between the computer that is requesting access (computer A) and the computer that is being accessed (computer B). Thus, if an end user makes a request for a resource from computer A, this request is directed to a proxy server. The proxy server makes the request, gets the response from computer B, and then forwards the response to the end user. Proxy servers are useful for accessing World Wide Web resources from inside a firewall.

public key
The key in a public/private key pair that is made available to others. It enables them to direct a transaction to the owner of the key or verify a digital signature. Data encrypted with the public key can be decrypted only with the corresponding private key. Contrast with private key. See also public/private key pair.

Public Key Cryptography Standards (PKCS)
Informal inter-vendor standards developed in 1991 by RSA Laboratories with representatives from various computer vendors. These standards cover RSA encryption, the Diffie-Hellman agreement, password-based encryption, extended-certificate syntax, cryptographic message syntax, private-key information syntax, and certification syntax.

PKCS #1 describes a method for encrypting data by using the RSA public key cryptosystem. Its intended use is in the construction of digital signatures and digital envelopes.

PKCS #7 specifies a general format for cryptographic messages.

PKCS #10 specifies a standard syntax for certification requests.

PKCS #11 defines a technology-independent programming interface for cryptographic devices such as smart cards.

PKCS #12 specifies a portable format for storing or transporting a user’s private keys, certificates, miscellaneous secrets, and so forth.

public key infrastructure (PKI)
A standard for security software that is based on public key cryptography. The PKI is a system of digital certificates, certificate authorities, registration authorities, certificate management services, and distributed directory services. It is used to verify the identity and authority of each party involved in any transaction over the Internet. These transactions might involve operations where identity verification is required. For example, they might confirm the origin of proposal bids, authors of e-mail messages, or financial transactions.
The PKI achieves this by making the public encryption keys and certificates of users available for authentication by a valid individual or organization. It provides on-line directories that contain the public encryption keys and certificates that are used in verifying digital certificates, credentials, and digital signatures.

The PKI provides a means for swift and efficient responses to verification queries and requests for public encryption keys. It also identifies potential security threats to the system and maintains resources to deal with security breaches. Lastly, the PKI provides a digital timestamping service for important business transactions.

**public/private key pair**
A public/private key pair is part of the concept of key pair cryptography (introduced in 1976 by Diffie and Hellman to solve the key management problem). In their concept, each person obtains a pair of keys, one called the public key and the other called the private key. Each person’s public key is made public while the private key is kept secret. The sender and receiver do not need to share secret information: all communications involve only public keys, and no private key is ever transmitted or shared. It is no longer necessary to trust some communications channel to be secure against eavesdropping or betrayal. The only requirement is that public keys must be associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory).

Anyone can send a confidential message by using public information. However, the message can be decrypted only with a private key, which is in the sole possession of the intended recipient. Furthermore, key pair cryptography can be used not only for privacy (encryption), but also for authentication (digital signatures).

**RA**
Registration authority.

**RA Desktop**
A Java applet that provides RAs with a graphical interface for processing requests for credentials and administering them throughout their lifetime.

**RA server**
The server for the Trust Authority Registration Authority component.

**RC2**
A variable key-size block cipher, designed by Ron Rivest for RSA Data Security. *RC* stands for *Ron’s Code* or *Rivest’s Cipher*. It is faster than DES and is designed as a drop-in replacement for DES. It can be made more secure or less secure against exhaustive key search than DES by using appropriate key sizes. It has a block size of 64 bits and is about two to three times faster than DES in software. RC2 can be used in the same modes as DES.

An agreement between the Software Publishers Association (SPA) and the United States government gives RC2 special status. This makes the export approval process simpler and quicker than the usual cryptographic export process. However, to qualify for quick export approval a product must limit the RC2 key size to 40 bits with some exceptions. An additional string can be used to thwart attackers who try to precompute a large look-up table of possible encryptions.

**registrar**
A user who has been authorized to access the RA Desktop, to administer certificates and requests for certificates.

**registration authority (RA)**
The software that administers digital certificates to ensure that an organization’s business policies are applied from the initial receipt of an enrollment request through certificate revocation.

**registration database**
Contains information about certificate requests and issued certificates. The database stores enrollment data and all changes to the certificate data throughout its life cycle. The database can be updated by RA processes and policy exits, or by registrars.

**registration domain**
A set of resources, policies, and configuration options related to specific certificate registration processes. The domain name is a subset of the URL that is used to run the registration facility.

**registration facility**
A Trust Authority application framework that provides specialized means of enrolling entities (such as browsers, routers, e-mail, and secure client applications) and managing certificates throughout their life cycle.
registration process
In Trust Authority, the steps for validating a user, so that the user and the user’s public key can become certified
and participate in transactions. This process can be local or Web-based, and can be automated or administered by
human interaction.

repudiate
To reject as untrue; for example, to deny that you sent a specific message or submitted a specific request.

request ID
A 24- to 32-character ASCII value that uniquely identifies a certificate request to the RA. This value can be used
on the certificate request transaction to retrieve the status of the request or the certificate that is associated with
it.

RSA
A public key cryptographic algorithm that is named for its inventors (Rivest, Shamir, and Adelman). It is used for
encryption and digital signatures.

schema
As relates to the Directory, the internal structure that defines the relationships between different object types.

Secure Electronic Transaction (SET)
An industry standard that facilitates secure credit card or debit card payment over untrusted networks. The
standard incorporates authentication of cardholders, merchants, and card-issuing banks because it calls for the
issuance of certificates.

Secure Sockets Layer (SSL)
An IETF standard communications protocol with built-in security services that are as transparent as possible to
the end user. It provides a digitally secure communications channel.
An SSL-capable server usually accepts SSL connection requests on a different port than requests for standard
HTTP requests. SSL creates a session during which the exchange signals to set up communications between two
modems need to occur only once. After that, communication is encrypted. Message integrity checking continues
until the SSL session expires.

security domain
A group (a company, work group or team, educational or governmental) whose certificates have been certified by
the same CA. Users with certificates that are signed by a CA can trust the identity of another user that has a
certificate signed by the same CA.

server
(1) In a network, a data station that provides functions to other stations; for example, a file server. (2) In TCP/IP,
a system in a network that handles the requests of a system at another site, called a client/server.

server certificate
A digital certificate, issued by a CA to enable a Web server to conduct SSL-based transactions. When a browser
connects to the server by using the SSL protocol, the server sends the browser its public key. This enables
authentication of the identity of the server. It also enables encrypted information to be sent to the server. See also
CA certificate, digital certificate, and browser certificate.

servlet
A server-side program that gives Java-enabled servers additional functionality.

SET
Secure Electronic Transaction.

SGML
Standard Generalized Markup Language.

SHA-1 (Secure Hash Algorithm)
An algorithm that was designed by NIST and NSA for use with the Digital Signature Standard. The standard is
the Secure Hash Standard; SHA is the algorithm that the standard uses. SHA produces a 160-bit hash.
sign
To use your private key to generate a signature. The signature is a means of proving that you are responsible for and approve of the message you are signing.

signing/verifying
To sign is to use a private digital key to generate a signature. To verify is to use the corresponding public key to verify the signature.

Simple Mail Transfer Protocol (SMTP)
A protocol that transfers electronic mail over the Internet.

site certificate
Similar to a CA certificate, but valid only for a specific Web site. See also CA certificate.

smart card
A piece of hardware, typically the size of a credit card, for storing a user’s digital keys. A smart card can be password-protected.

S/MIME
A standard that supports the signing and encryption of e-mail transmitted across the Internet. See MIME.

SMTP
Simple Mail Transfer Protocol.

SSL
Secure Sockets Layer.

Standard Generalized Markup Language (SGML)
A standard for describing markup languages. HTML is based on SGML.

symmetric cryptography
Cryptography that uses the same key for both encryption and decryption. Its security rests in the key — revealing the key means that anyone could encipher and decipher messages. The communication remains secret only as long as the key remains secret. Contrast with asymmetric cryptography.

symmetric key
A key that can be used for both encryption and decryption. See also symmetric cryptography.

T

target
A designated or selected data source.

TCP/IP

top CA
The CA at the top of a PKI CA hierarchy.

TP
Trust Policy.

transaction ID
An identifier provided by the RA in response to a preregistration enrollment request. It enables a user running the Trust Authority Client application to obtain the pre-approved certificate.

Transmission Control Protocol/Internet Protocol (TCP/IP)
A set of communication protocols that support peer-to-peer connectivity functions for local and wide area networks.

triple DES
A symmetric algorithm that encrypts the plaintext three times. Although many ways exist to do this, the most secure form of multiple encryption is triple-DES with three distinct keys.
Trust Authority
An integrated IBM SecureWay security solution that supports the issuance, renewal, and revocation of digital certificates. These certificates can be used in a wide range of Internet applications, providing a means to authenticate users and ensure trusted communications.

trust chain
A set of certificates that consists of the trusted hierarchy from the user certificate to the root or self-signed certificate.

trust domain
A set of entities whose certificates have been certified by the same CA.

trusted computer base (TCB)
The software and hardware elements that collectively enforce an organization’s computer security policy. Any element or part of an element that can effect security policy enforcement is security-relevant and part of the TCB. The TCB is an object that is bounded by the security perimeter. The mechanisms that carry out the security policy must be non-circumventable, and must prevent programs from gaining access to system privileges to which they are not authorized.

trust model
A structuring convention that governs how certificate authorities certify other certificate authorities.

tunnel
In VPN technology, an on-demand virtual point-to-point connection made through the Internet. While connected, remote users can use the tunnel to exchange secure, encrypted, and encapsulated information with servers on the corporate private network.

type
See object type.

Unicode
A 16-bit character set that is defined by ISO 10646. The Unicode character encoding standard is an international character code for information processing. The Unicode standard encompasses the principal scripts of the world and provides the foundation for the internationalization and localization of software. All source code in the Java programming environment is written in Unicode.

Uniform Resource Locator (URL)
A scheme for addressing resources on the Internet. The URL specifies the protocol, host name or IP address. It also includes the port number, path, and resource details needed to access a resource from a particular machine.

URL
Uniform Resource Locator.

user authentication
The process of validating that the originator of a message is the identifiable and legitimate owner of the message. It also validates that you are communicating with the end user or system you expected to.

UTF-8
A transformation format. It enables information processing systems that handle only 8-bit character sets to convert 16-bit Unicode to an 8-bit equivalent and back again without loss of information.

Virtual Private Network (VPN)
A private data network that uses the Internet rather than phone lines to establish remote connections. Because users access corporate network resources through an Internet Service Provider (ISP) rather than a telephone company, organizations can significantly reduce remote access costs. A VPN also enhances the security of data exchanges. In traditional firewall technology, message content can be encrypted, but the source and destination addresses are not. In VPN technology, users can establish a tunnel connection in which the entire information packet (content and header) is encrypted and encapsulated.
VPN
Virtual Private Network.

W

Web browser
Client software that runs on a desktop PC and enables the user to browse the World Wide Web or local HTML pages. It is a retrieval tool that provides universal access to the large collection of hypermedia material available in the Web and Internet. Some browsers can display text and graphics, and some can display only text. Most browsers can handle the major forms of Internet communication, such as FTP transactions.

Web server
A server program that responds to requests for information resources from browser programs. See also server.

WebSphere Application Server
An IBM product that helps users develop and manage high-performance Web sites. It eases the transition from simple Web publishing to advanced e-business Web applications. The WebSphere Application Server consists of a Java-based servlet engine that is independent of both the Web server and its underlying operating system.

World Wide Web (WWW)
That part of the Internet where a network of connections is established between computers that contain hypermedia materials. These materials provide information and can provide links to other materials in the WWW and Internet. WWW resources are accessed through a Web browser program.

X

X.500
A standard for putting into effect a multipurpose, distributed and replicated directory service by interconnecting computer systems. Jointly defined by the International Telecommunications Union (ITU), formerly known as CCITT, and the International Organization for Standardization and International Electro-Chemical Commission (ISO/IEC).

X.509 certificate
A widely-accepted certificate standard designed to support secure management and distribution of digitally signed certificates across secure Internet networks. The X.509 certificate defines data structures that accommodate procedures for distributing public keys that are digitally signed by trusted third parties.

X.509 Version 3 certificate
The X.509v3 certificate has extended data structures for storing and retrieving certificate application information, certificate distribution information, certificate revocation information, policy information, and digital signatures. X.509v3 processes create time-stamped CRLs for all certificates. Each time a certificate is used, X.509v3 capabilities allow the application to check the validity of the certificate. It also allows the application to determine whether the certificate is on the CRL. X.509v3 CRLs can be constructed for a specific validity period. They can also be based on other circumstances that might invalidate a certificate. For example, if an employee leaves an organization, their certificate would be put on the CRL.
Index

Numerics
4758 CA Profile 6
4758 Cryptographic Coprocessor 55
administering 55
cloning 12
Cryptographic Coprocessor 64

A
access control list (ACL) 57
Activity Report 13
add registrars 31
add the first registrar 31
administering
4758 Cryptographic Coprocessor 55
Audit subsystem 39
CA server 18
DB2 UDB 51
Directory server 53
HTTP Server 16
RA server 30
Trust Authority 5
WebSphere Application Server 15
administrator, add the first RA 31
Affected Entity Types table 107
AIX
backing up 12
restoring 12
aliasing, IP 68
ASN.1 68
audit
change binding attempts 44
change hostname 42
change interval between binding 44
change interval between retries 43
change port 42, 43
database data 105
event fields 102
log 45
mask 42
records, search 41
reports, generate 47
Audit administrator action events 66
Audit Administrator password 6
audit events 102 (continued)
security sensitive 66
Audit Log table 108
Audit subsystem
about 65
administering 39
Audit Archive and Sign utility 99
Audit Integrity Check utility 100
change audit log settings 45
change audit mask 42
change error log settings 47
change event log settings 45
change hostname 43
change interval between retries 43
change port 43
change trace log settings 46
client configuration file 94
database views 40
event masks 66
events, about 65
log files, archive 47
log files, sign 47
logs 44, 51
mandatory events 66
optional events 66
records, about 65
server configuration file 90
status 50
AuditClient.ini 10, 94
AuditServer.ini 10, 90
authentication 1
Authority Information Access extension 59
Authority Key Identifier extension 59
Authorized Entities table 107
Authorized Roles table 107

B
backing up 12
AIX 12
Trust Authority databases 12
Windows NT 12
Basic Constraints extension 59
binding 44

C
CA (certificate authority)
cross-certification 58
hierarchies 58

Public Key Infrastructure System Administration Guide 137
CA (certificate authority) (continued)

downward 58
internal 58
upward 58

CA Certificates
granting to third party CAs 28

CA server
administering 18
change the CRL settings 19
change the polling interval 19
configuration file 72
database integrity checking 22
listener port 18
logs 28
status 29

CAIntegrityCheck utility 22
CARemac utility 21

CCITT 68
certificate
certificate revocation list (CRL) 1
extensions 58
issuing 1
renewing 1
revoction list (CRL) 61
revoking 1
tracking activity 13
tracking usage 12

Certificate Authority (CA) 57
certificate management events 65
Certificate Policies extension 59
certificates 62
certification, digital 1, 62
Change Password utility 5
checking the HTTP Server logs 18
cloning the 4758 12
command-line utilities
Add RA User 98
Audit Archive and Sign 47
Audit Integrity Check 48
CA Certification 22, 25, 28, 96
Create New RA 39, 101
Enable RA Database Encryption 99
MRA Enrollment Authorization 38, 101
commands, entering 5
common extensions 61
communications, trusted 1
Component Types table 108
components, Trust Authority
4758 Cryptographic Coprocessor 55
Audit server 39
CA server 18
IBM HTTP Server 16
IBM SecureWay Directory 53
RA server 30
starting server 7
stopping server 7
Websphere Application Server 15
confidentiality 63
configuration files 10, 71
Audit client 94
Audit Server 90

configuration files 10, 71 (continued)
AuditClient.ini 94
AuditServer.ini 90
CA server 72
jonahca.ini 72
jonahra.ini 81
modifying 9
RA server 81
Control Program password 5
creation interval, CRL 19
CRL (certificate revocation list) 61
creation interval 19
lifetime 20
settlings, change 19
CRL Distribution Points extension 59
cross-certification 22, 58, 61
Cryptographic Coprocessor, 4758 64
cryptographic engine 57

data, audit database 105
database data, audit 105
DB2 databases 66
administering 51
logs 53
status 51

DES 55
digital certificates 62
digital certification 1
Directory 53
Directory Administrator password 6
Directory servers
about 68
administering 53
change RA settings for 37
entires 24, 27
logs 55
server 36
status 54
distinguished name (DN) 62
DNS addresses 23, 26
domains, registration 63

e-mail addresses 24, 26
cryption, enable RA database 33
cryption key 63
error log 47
Event Control table 106
event log 45
Event Severities table 106
events, audit 42, 102
Extended Key Usage extension 59
extensions
certificate 58
extensions (continued)
common 61
Name Constraints 27
private 61
request flow 61
standard 59

F
fields, audit event 102
files, configuration 71

H
hierarchies, CA 58
hierarchy
making updates after changing CA hierarchy 27
hostname 36, 42
HTTP Server 16
administering 16
logs, checking 18
status 16

I
IBM HTTP Server 16, 67
ICL (issued certificate list) 62
protection key 21
protection policy 20
IniEditor utility
adding a parameter 11
adding a section 11
editing parameters 10
running 9
saving a file 11
using 10
integrity 63
integrity checking 22, 48, 66
integrity sealing 66
IP address mask 23, 26
IP addresses, changing 11
IP aliasing 68
ISO 68
issued certificate list (ICL) 62
Issuer Alternative Name extension 59
ITU standard 59, 68

J
jonahca.ini 10, 72
jonahra.ini 10, 81

K
key management events 65
Key Usage extension 59
keys
encryption 63
MAC 63
Keys table 106

L
LDAP, (Lightweight Directory Access Protocol) 53
lifetime, CRL 20
listener port 34
logs
audit 44, 45
Audit server 51
Audit subsystem 47
CA server 28
DB2 53
Directory server 55
error 47
event 45
HTTP Server 18
RA server 35
trace 46
WebSphere Application Server 16

M
MAC key 63
mask, audit 42
message authentication code (MAC) 58
messages, PKIX 57
modifying configuration files 9
Multiple RA
create new RA utility 39, 101
installation 37
MRA Enrollment Authorization utility 38, 101
overview 64

N
Name Constraints extension 27, 59

O
object identifiers (OIDs) 68
P

passwords
4758 CA Profile 6
Audit Administrator 6
changing 5
Control Program 5
Directory Administrator 6
PKIX messages 57
Policy Constraints extension 59
Policy Mappings extension 59
polling interval 19, 34
port 36, 42, 43
Audit server 42, 43
CA server 18
Directory server 36
RA server 34
post interval 37
private extensions 61
Private Key Usage Period extension 59
protection
key, ICL 21
policy, ICL 20
protection policy, ICL 20

R

RA (registration authority)
about 63
add an administrator 32
Add RA User utility 32, 98
add the first administrator 31
add the first registrar 31
administrators (registrars) 63
RA events 66
RA server 30
administering 30
change hostname of Directory 36
change port of Directory 36
change the listener port 34
change the polling interval 34
change the retry interval 35
configuration file 81
enable database encryption 33, 99
logs 35
status 35
registrars, add 31
registration domains 63
reports
Activity 13
Usage 12
reports, audit 47
restoring 12
AIX 12
Trust Authority databases 12
Windows NT 12
retries 43
retry interval 35
revocation list (CRL), certificate 1
RFC 2459 59

S

security, Trust Authority 57
security sensitive events 66
servers
Audit 39
CA server 18
Directory 36, 68
HTTP 16, 67
RA 30
starting components 7
stopping components 7
Web 67
WebSphere Application 15, 67
signature validation 63
signing 63
smart cards 65
Sources table 106
SQL command 31
standard extensions 59
standards
ASN.1 68
CCITT 68
ISO 68
ITU 59, 68
RFC 2459 59
X.509v3 59
X.680 68
starting server components 7
status
audit server 50
CA server 29
DB2 databases 51
Directory server 54
HTTP Server 16
RA server 35
WebSphere Application Server 15
stopping server components 7
Subject Alternative Name extension 59
Subject Directory Attributes extension 59
Subject Key Identifier extension 59
System table 109

T

tables, audit database
Affected Entity Types table 107
Audit Log table 108
Authorized Entities 107
Authorized Roles 107
Component Types table 108
Event Control 106
Event Severities 106
Keys 106
Sources 106
tables, audit database (continued)  
   System table 109  
   timer 43  
   trace log 46  
   troubleshooting 110  
Trust Authority  
   administering 5  
   backing up 12  
   Control utility 7  
   description 1  
   IP address, changing 11  
   reports 12, 13  
   restoring 12  
   security 57  
Trust Chain Delivery  
   disabling 37  
   enabling 37  
   overview 64  
trusted communications 1

U
Uniform Resource Identifiers (URIs) 24  
URIs 27  
Usage Report 12  
utilities  
   Activity Report 13  
   Add RA User 32  
   Audit Archive and Sign 47, 99  
   Audit Integrity Check 48  
   Audit Integrity Check utility 100  
   CA Certification 22, 25, 28, 96, 98  
   CAIntegrityCheck 22  
   CARemac 21  
   Change Password 5  
   Create New RA 39, 101  
   Enable RA Database Encryption 99  
   IniEditor 9  
   MRA Enrollment Authorization 38, 101  
   Trust Authority Control 7  
   Usage Report 12

V
validation, signature 63  
views, audit database 40

W
Web servers 67  
WebSphere Application Server 15, 67  
   administering 15  
   logs 16  
   status 15  
Windows NT  
   backing up 12  
   restoring 12

X
X.509v3 59  
X.680 68

Public Key Infrastructure System Administration Guide 141