

CA AuthMinder™

Installation and Deployment Guide for UNIX Platforms

r7.1



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © [set copyright date variable] CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Chapter 1: Understanding the Basics

With the exponential increase in cases of Internet-based fraud over the last few years, relying just on user name-password for authentication is no longer sufficient. The need for stronger authentication can either be to protect end users or to comply with government-mandated security requirements, internal policies, or best practices. However, adding stronger authentication often creates conflict between compliance requirements and user convenience. In other words, organizations must increase the security of their authentication processes by reducing the complexity, and reduce the risk of financial losses or brand damage by increasing the customer and partner access to applications and data.

CA AuthMinder (later referred to as AuthMinder) is a trusted strong authentication service that enables your application to verify and protect the identity of your end users by:

- Not transmitting passwords (either in clear or encrypted form) over the network.
- Enabling you to choose the authentication method that best suits the security and convenience of different types of users.
- Using ArcotID PKI® and ArcotID OTP, which are based on the patented **Cryptographic Camouflage** technique to protect keys.

In *Cryptographic Camouflage*, the keys are not encrypted with a password that is too long for exhaustive attacks. Instead, keys are encrypted such that only one password will decrypt it correctly, but many passwords can decrypt it to produce a key that looks valid enough to fool an attacker. As a result, this method protects a user's private key against dictionary attacks and Man-in-the-Middle (MITM) attacks, as a smartcard does, but entirely in the software format.

See "[How Cryptographic Camouflage Works](#)" (see page 14) for more information.

This guide provides information for planning the deployment of CA AuthMinder based on different solution requirements. Each solution consists of multiple components that interact with each other and other systems in an enterprise or multiple-network systems.

The biggest advantage of AuthMinder is that it enables you to upgrade security from simple user name-password authentication without changing the user login experience or critical business processes by using an ArcotID PKI for authentication.

This chapter introduces you to AuthMinder and covers the following topics:

- [AuthMinder as a Versatile Authentication Server](#) (see page 6)
- [Plugging Into AuthMinder](#) (see page 7)
- [AuthMinder Architecture](#) (see page 8)
- [ArcotID PKI Key Concepts](#) (see page 11)
- [User Authentication in AuthMinder](#) (see page 17)
- [What's New in this Release](#) (see page 19)

Note: CA AuthMinder still contains the terms Arcot and WebFort in some of its code objects and other artifacts. Therefore, you will find occurrences of Arcot and WebFort in all CA AuthMinder documentation. In addition, some of the topics in this guide do not follow the standard formatting guidelines. These inconsistencies will be fixed in a future release.

AuthMinder as a Versatile Authentication Server

AuthMinder is a *Versatile Authentication Server (VAS)* because it supports the implementation of a wide range of proprietary and open authentication mechanisms. In addition to supporting authentication by using Public Key Infrastructure (PKI) and one-time password (OTP), it is also designed to plug in any existing authentication methods, if required, thereby enabling your organization to handle changes to critical systems and partner applications seamlessly.

As a result, the VAS functionality of AuthMinder provides your organization the complete flexibility to choose the authentication method that best suits the needs of your end users. You can choose to:

- Integrate with a variety of standard authentication interfaces.
- Implement standards-based hardware or software authentication methods.
- Add new authentication methods, such as ArcotID PKI, while continuing to support legacy technology, such as OTP tokens.
- Extend AuthMinder VAS through Plug-ins to do proprietary authentication.

Plugging Into AuthMinder

AuthMinder provides the following authentication methods out-of-the-box:

- **ArcotID**

ArcotID PKI is a CA-proprietary secure software credential that provides two-factor authentication. An ArcotID PKI is a small data file that by itself can be used for strong authentication to a variety of clients such as, Web or Virtual Private Networks (VPNs).

See "[ArcotID PKI Key Concepts](#)" (see page 11) for more information on ArcotID PKI.

- **Password**

A regular credential, where the user is issued a username and a password to log in to the system.

- **One-Time Password**

One-time password is another credential generated by AuthMinder Server. An OTP can be numeric or an alpha-numeric string. It is also possible to configure the number of times it can be used.

- **OATH-Compliant One-Time Password**

One-time passwords that are complaint to Open Authentication (OATH) standards. AuthMinder supports both counter-based OATH OTPs (HOTPs) and time-based OATH OTPs (TOTPs).

- **Question and Answer**

Question and Answer (also known as *QnA*) is a challenge-response authentication mechanism. Users authenticate to AuthMinder Server by providing correct answers for the questions they are asked. These Questions and Answers are set by the users themselves during registration.

- **ArcotOTP**

ArcotID OTP is compliant to the OATH, Europay, MasterCard, and VISA (EMV) standards. If your application is integrated with ArcotID OTP, then it accepts the user's password as an input and generates passwords (also known as *passcodes*) on the users' device. The user, then, submits this generated passcode to authenticate to your Web application. Based on the authentication result, the user is granted access to the protected application or denied access.

Passcode generation is an offline process, which means that your application need not be connected to AuthMinder for generating passcodes.

- **LDAP Username-Password**

AuthMinder supports LDAP authentication, where the user credentials in the directory service are used to authenticate users.

You can issue one or more of these credentials to your users. You can also issue multiple credentials of the same type. For example, you can issue two password credentials, an ArcotID PKI credential, and a QnA credential for a single user.

If you want to extend the default authentication mechanisms, then AuthMinder provides you the flexibility to do so by writing [Plug-Ins](#) (see page 8).

Plug-Ins

Plug-in is a custom server-side component, written in C or C++, that enables you to extend the functionality of AuthMinder VAS. Plug-in is implemented as a custom event handler library within the context of the AuthMinder Server.

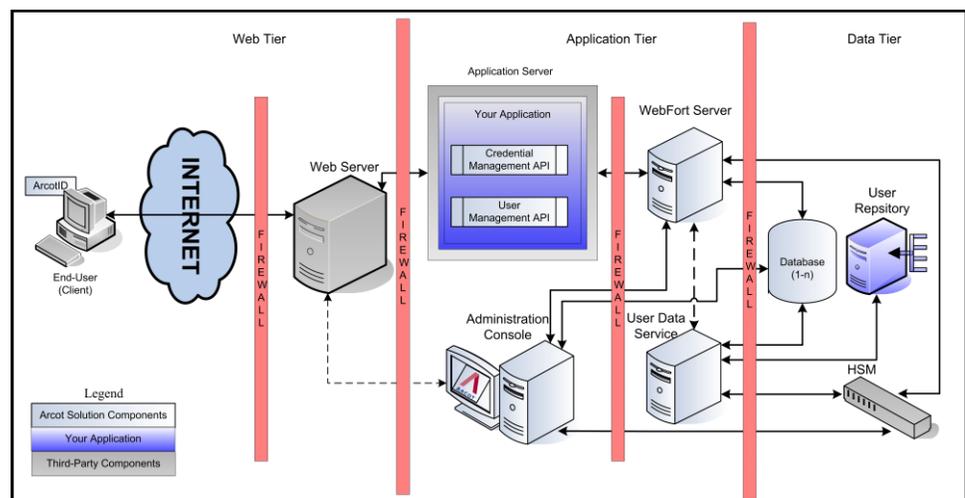
You must register your plug-in (by using the Administration Console) to a published set of events, so that the plug-in is invoked when the specified event occurs.

You can configure multiple plug-ins for one organization, and you can also configure the same plug-in for multiple organizations.

AuthMinder Architecture

You can install AuthMinder on a single system or you can distribute its components across multiple systems. However, to ensure maximum security of transactions, it is recommended that you implement the architecture shown in the following figure:

- [Web Tier](#) (see page 9)
- [Application Tier](#) (see page 10)
- [Data Tier](#) (see page 11)



The following subsections discuss the AuthMinder components in each of these layers.

Web Tier

This layer comprises the static (HTML) content and interacts directly with the user over a network or the Internet.

This layer serves the ArcotID PKI Client (Java, Flash, or Native) to the end user's browser. ArcotID PKI Client interacts with AuthMinder Server for user authentication. It collects the ArcotID PKI password, signs the challenge, and then sends the signed challenge to the AuthMinder Server for verification.

Note: See the *CA ArcotID PKI Client Reference Guide* for details on ArcotID PKI Client.

Application Tier

This layer constitutes AuthMinder Server, your application that uses AuthMinder SDKs, and the application servers where the Administration Console and the User Data Service (UDS) reside.

Note: All components in this layer can be installed on one system or can be distributed across multiple systems.

- **AuthMinder Server**

Server component that processes issuance and authentication requests from your application through AuthMinder SDKs.

- **Administration Console**

Web-based console for configuring server instances, communication mode between AuthMinder components, authentication policies, credential profiles, managing credentials, and for managing organizations, administrators, and users.

- **User Data Service**

The abstraction layer that provides access to user- and organization-related data from different types of user repositories, such as Relational Database Management Systems (RDBMSs) and directory servers (LDAPs).

- **Authentication API**

Java APIs that can be invoked by your application to forward authentication requests to AuthMinder Server.

- **Credential Management API**

Java APIs that can be invoked by your application to forward issuance requests to AuthMinder Server for creating and managing user credentials in AuthMinder.

- **User Management API**

The Web Services client that can be invoked by your application to forward issuance requests to User Data Service for creating and managing users in AuthMinder.

- **Sample Application**

Sample Application demonstrates the usage of AuthMinder Java APIs and how your application can be integrated with AuthMinder. The Sample Application can also be used to verify if AuthMinder was installed successfully, and if it is able to perform issuance and authentication operations.

Data Tier

This tier comprises the RDBMSs that AuthMinder uses to store configurations, credential information, and user data if other user repositories are not configured.

If Hardware Security Module (HSM) is used to encrypt the user sensitive data, then it will be a part of this tier.

ArcotID PKI Key Concepts

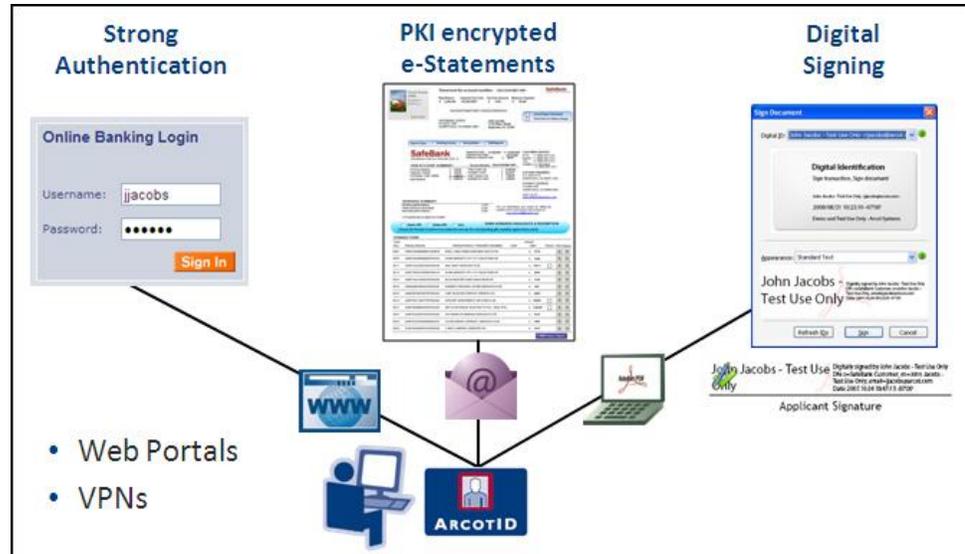
This section introduces the key concepts of ArcotID PKI, which is a prime credential that AuthMinder supports.

- [Introduction to ArcotID PKI](#) (see page 12)
- [ArcotID PKI File Structure](#) (see page 14)
- [How Cryptographic Camouflage Works](#) (see page 14)
- [Support Roaming Download](#) (see page 15)
- [ArcotID PKI as a Secure Container \(Key Authority\)](#) (see page 16)
- [ArcotID PKI Client](#) (see page 17)

Introduction to ArcotID PKI

An ArcotID PKI offers the same capabilities as a physical smartcard for authentication, digital signing, encryption, and decryption for PKI-enabled applications, without requiring any end-user hardware. The ArcotID PKI can authenticate to any Web application, even if that application does *not* support PKI-based authentication.

The following figure illustrates the use cases for ArcotID PKI.



An ArcotID PKI is a data file that is saved on an end user's computer, USB drive, or downloaded remotely for secure on-demand authentication. Unlike the simple password, an ArcotID PKI is *not* vulnerable to brute force password attacks. Additionally, the ArcotID PKI is not vulnerable to man-in-middle attacks, which, in turn, protects users from phishing attacks.

An ArcotID PKI can be used for strong authentication with a variety of applications, such as the Web or Virtual Private Networks (VPNs).

The ArcotID PKI is a configurable solution that bridges the gap between simple-but-insecure username-password-authentication and expensive-difficult-to-deploy, but very secure smartcard and USB token solutions.

The ArcotID PKI is based on industry standards and CA's patented Cryptographic Camouflage technology to provide software-only, strong authentication that is protected against brute force attacks.

Although an ArcotID PKI is protected by a password, it supports the following features to provide strong authentication:

- Only the correct ArcotID PKI password can access an ArcotID PKI.
- A plausible response is generated for every ArcotID PKI password entered, even if it is incorrect. As a result, preventing offline identification of the ArcotID PKI password is not easy.
- The ArcotID PKI authentication is a challenge-response authentication protocol, which ensures the user's password is only used locally and never transmitted or verified at the server end.
- Repeated incorrect ArcotID PKI password entries locks out ArcotID PKI depending on the maximum authentication attempts configured.
- Is valid only in the domain that issued the ArcotID PKI.
- ArcotID PKIs can only be used online, which means the user *must* connect to AuthMinder Server to validate their ArcotID PKI password.

ArcotID PKI File Structure

An ArcotID PKI contains the following main components:

1. The standard X.509v3 digital certificate with CA-specific extension.
2. A second pair of public and private keys that is generated for authenticating to AuthMinder Server. It is not used for general signing or encryption.

The public key is stored in the encrypted format. It is encrypted using the *Domain Key*, which is used to create and authenticate ArcotID PKIs. You can configure a domain key at the global-level or at the organization-level. An ArcotID PKI issued with the organization-specific domain key *cannot* be used across organizations.

The private key is cryptographically camouflaged by using the ArcotID PKI password.

3. A section to store the user's Open PKI keys and certificates, which they can use for signing, encrypting, and decrypting. See "[ArcotID PKI as a Secure Container \(Key Authority\)](#)" (see page 16) for more information.

How Cryptographic Camouflage Works

With the advent of support for public key cryptography in Web browsers, the use of public key cryptographic signatures and authentication protocols is becoming more common. The security of the private key, however, remains a problem. The most basic threat is the theft of a private key that is stored on a disk. Usually such a key is stored in a software key container, a file, wherein the keys are encrypted by using a password. An attacker who steals the container can try to guess the password using a dictionary attack.

To overcome such problems, CA proposes a method for secure storage of private keys in software, using cryptographic camouflage, where attacks on the key container are inherently supervised. The key container embeds the user's private key among spurious private keys. An attacker who tries to crack the key container will recover many plausible private keys, but will not be able to distinguish the correct private key from the spurious decoys until they use the keys to sign the challenge and send it to AuthMinder Server. In such cases, AuthMinder Server notices the multiple authentication failures and suspends the user's access.

Support Roaming Download

ArcotID PKI can be downloaded by using any device while the user is traveling. This feature is known as *Roaming*. AuthMinder Server offers roaming capabilities to enable the user to securely download the ArcotID PKI and authenticate from any system when the need arises. As a result, this provides instant access to the critical data and services whenever required, while keeping the data safe from unauthorized access.

A roaming user can be authenticated using QnA, OTP, or any customized third-party solutions.

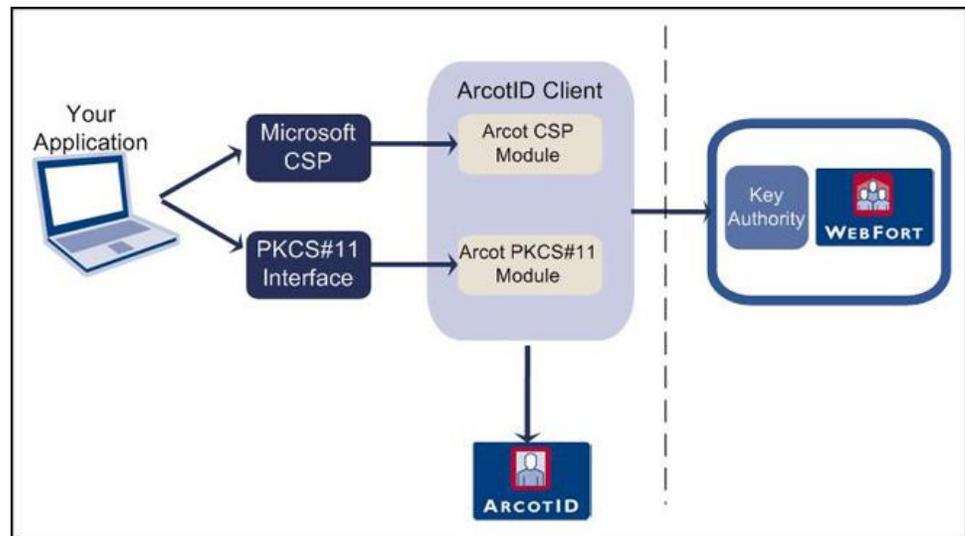
ArcotID PKI as a Secure Container (Key Authority)

In addition to providing strong authentication, ArcotID PKI can also be used as a secure container to store digital certificates and private keys that can be used for different applications or operations such as, email signing (S/MIME), document signing, certificate-based authentication (open PKI). This process of managing private key storage in the ArcotID PKI is performed by *Key Authority* (KA).

An unsigned attribute is created in the ArcotID PKI to store these credentials and this attribute is referred to as *Key Bag* or *Key Vault*. The digital certificates are stored in an unencrypted format in the Key Bag, but the private keys are encrypted using a key called *Key Authority* key, which is stored in the AuthMinder database.

To use the private keys that are stored in a Key Bag, the ArcotID PKI Client (see "[ArcotID PKI Client](#)" (see page 17)) makes a request for the KA key to AuthMinder Server by signing the request with the user's private key. The AuthMinder Server authenticates the incoming request and sends the KA key to the client, which then uses this key to open the Key Bag and access the private keys.

The following figure illustrates how to use ArcotID PKI as an open PKI container.



ArcotID PKI Client

The ArcotID PKI Client software is used in conjunction with AuthMinder Server. The ArcotID PKI Client enables an end user to use an ArcotID PKI in a Web browser to authenticate to a Web site, VPN, or other online resources.

To support a wide variety of application environments (operating systems, browsers, JVMs), the ArcotID PKI Client is available in a variety of flavors, such as:

- Native Client
- Flash Client
- Java Signed Applet
- Java Unsigned Applet
- JavaScript Client

Note: Refer to the *CA ArcotID PKI Client Reference Guide* for more information on these client types.

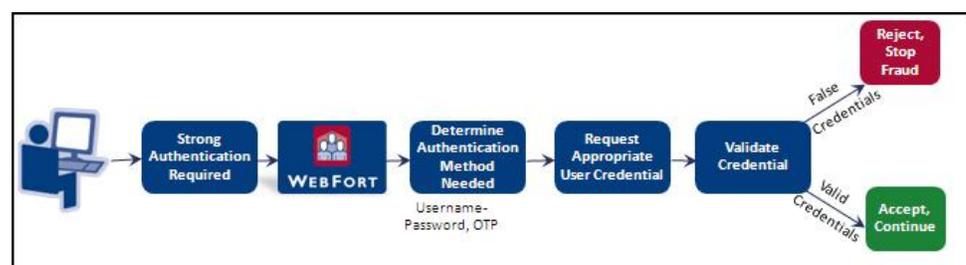
User Authentication in AuthMinder

A user trying to access the Web application protected by AuthMinder can be authenticated using any of the out-of-the-box credentials supported by AuthMinder.

In all the authentication mechanisms, the client is provided with an authentication token after every successful authentication. The authentication token is further used to prove that the client is already authenticated to the server. The authentication token is valid only for a certain interval, after which the client has to re-authenticate to the server.

All password type credentials namely, password, OTP, ArcotID OTP, and OATH OTP are based on the single-step authentication model, which means the credentials are passed by the client to the user and the server verifies the user credentials.

The following figure illustrates the typical authentication workflow.



However, ArcotID PKI and QnA are based on the challenge-response authentication model. These authentication mechanisms include multiple steps to authenticate users.

ArcotID PKI Authentication Workflow

Authentication using ArcotID PKI is a PKI-based challenge-response mechanism. The client obtains an authentication token by proving the private key of the user. The client-server interactions during authentication are as follows:

1. Get User Credentials

Your application or the resource which is protected by AuthMinder obtains the user credentials. For example, if the user's ArcotID PKI is not available on the system or the USB.

2. Get Appropriate Challenge

Your application requests for a challenge used to authenticate the user.

AuthMinder Server prepares a unique challenge and sends it to your application.

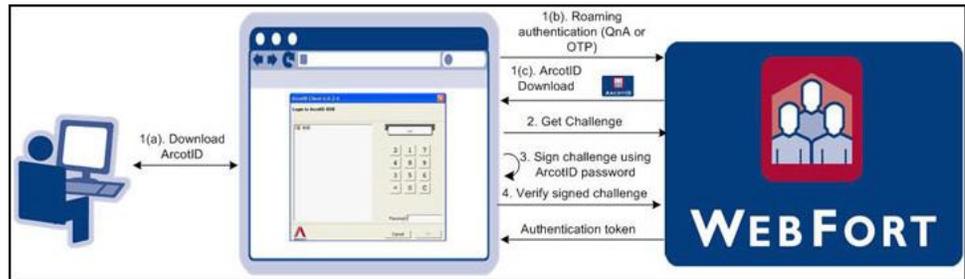
3. Generate Signature

The user enters the correct ArcotID PKI password to uncover the ArcotID PKI. The client signs this challenge with the user's private key that is available as a result of uncover. The challenge can either be pre-loaded on the client machine or can be downloaded from the server.

4. Verify Signed Challenge

The signed challenge is sent to the AuthMinder Server for verification. If the signature is verified successfully, the user can login or access your protected resource. For every successful transaction, AuthMinder also returns an authentication token for a user.

The following figure illustrates the ArcotID PKI authentication flow.



What's New in this Release

See the release notes for information about the key features and enhancements that have been introduced in release 7.1.

Chapter 2: Planning the Deployment

This chapter will help you to select a deployment model, and determine which AuthMinder components and prerequisite software to install on each system. Architecture diagrams for each deployment model are also provided to assist you with planning.

Note: In this guide, *system* refers to a physical device and *server* refers to the software that is run on the system.

The chapter covers the following topics:

- [Deployment Overview for Fresh Install](#) (see page 22)
- [Deployment Overview for Upgrade](#) (see page 24)
- [Choosing a Deployment Model](#) (see page 24)

Deployment Overview for Fresh Install

This section briefly outlines the steps for deploying AuthMinder:

1. Choose a deployment model that suits your business needs. AuthMinder can be installed on a single system or across multiple systems.
See ["Choosing a Deployment Model"](#) (see page 24) for more information.
2. Ensure that the system where you plan to install AuthMinder and its components meets all hardware requirements.
See ["Hardware Requirements"](#) (see page 34) for more information.
3. Install the prerequisite software.
See [Software Requirement](#) (see page 35)s for more information.
4. Configure the database server where you plan to seed the AuthMinder schema:
 - See ["Configuring Microsoft SQL Server"](#) (see page 43) for more information on configuring the MS SQL Server database.
 - See ["Configuring Oracle Database"](#) (see page 45) for more information on configuring the Oracle Database.
 - See ["Configuring IBM DB2 Universal Database"](#) (see page 47) for more information on configuring the IBM DB2 database.
 - See ["Configuring MySQL"](#) (see page 48) for more information on configuring the MySQL database.
5. Install AuthMinder:
 - See ["Deploying AuthMinder on a Single System"](#) (see page 59) for single-system deployment.
 - See ["Deploying AuthMinder on Distributed Systems"](#) (see page 97) for distributed-system deployment.
6. Run SQL scripts in the database to create the AuthMinder schema and set initial configuration values:
 - See ["Running Database Scripts"](#) (see page 70) for single-system deployment.
 - See ["Running Database Scripts"](#) (see page 109) for distributed-system deployment.
7. Copy required files and JARs on your application server. The Administration Console and the User Data Service use these files for proper functioning:
 - See ["Preparing Your Application Server"](#) (see page 72) for more information on copying the files for single-system deployments.
 - See ["Preparing Your Application Server"](#) (see page 110) for more information on copying the files for distributed deployments.
8. Deploy the Administration Console:

- See "[Deploying Administration Console](#)" (see page 79) for more information on deploying Administration Console for single-system deployments.
 - See "[Deploying Administration Console](#)" (see page 117) for more information on deploying Administration Console for distributed-system deployments.
9. **(Optional)** Prepare your Administration Console for localization, if you want to use languages other than English.
- See appendix, "[Preparing for Localization](#)" (see page 203) for more information.
10. Log in to Administration Console as Master Administrator and initialize AuthMinder:
- See "[Logging In to Administration Console](#)" (see page 81) and "[Bootstrapping the System](#)" (see page 82) for more information on initializing the Administration Console for single-system deployments.
 - See "[Logging In to Administration Console](#)" (see page 119) and "[Bootstrapping the System](#)" (see page 82) for more information on initializing the Administration Console for distributed-system deployments.
11. Start the AuthMinder Server and verify that the service has coming up correctly:
- See "[Starting AuthMinder Server](#)" (see page 84) and "[Verifying the Installation](#)" (see page 85) for more information on starting the AuthMinder Server for single-system deployments.
 - See "[Starting AuthMinder Server](#)" (see page 84) and "[Verifying the Installation](#)" (see page 85) for more information on starting the AuthMinder Server for distributed-system deployments.
12. Deploy and run the Sample Application to test AuthMinder installation:
- See "[Deploying Sample Application](#)" (see page 89) and "[Using Sample Application](#)" (see page 89) for more information on doing this in a single-system environments.
 - See "[Deploying Sample Application](#)" (see page 128), "[Configuring Sample Application for Communication with AuthMinder Server](#)" (see page 128), and "[Using Sample Application](#)" (see page 129) for more information on doing this in a distributed environment.
13. **(Optional)** Deploy User Data Service (UDS). You need to do so *only* if you want to use directory service as user repository:
- See "[Deploying User Data Service](#)" (see page 87) for more information on deploying and starting UDS for single-system deployments.
 - See "[Deploying User Data Service](#)" (see page 125) for more information on deploying and starting UDS for distributed deployments.

Deployment Overview for Upgrade

This section briefly outlines the steps for upgrading AuthMinder:

1. Check for the latest hardware and software support:
 - See "[Hardware Requirements](#)" (see page 34) for more information.
 - See "[Software Requirements](#)" (see page 35) for more information.
2. Upgrade your setup.

See chapter, "[Upgrading to AuthMinder 7.1](#)" (see page 143) for more information on upgrade instructions.

Choosing a Deployment Model

As a part of AuthMinder deployment, AuthMinder Server is the primary component that you must install. This is because, it authenticates users. Your applications that need to use AuthMinder Server can integrate with it by using Java SDKs or Web Services shipped along with it.

AuthMinder also requires an SQL database for storing server configuration data, user-specific preferences, and usage data.

Typically, all AuthMinder components are installed on a single system for development and simple testing. However, in production deployments and staging environments, AuthMinder Server should be installed on its own system. The shipped SDKs or Web Services are installed on a different system or systems that contain the application that users log in to.

AuthMinder is also shipped with Sample Application, which can be used to verify that AuthMinder is installed properly and is able to perform user authentication. Sample Application also serves as a code sample for integrating AuthMinder with your existing application(s).

The high-level deployment types supported by AuthMinder are:

- **Single-System Deployment** - For development or testing
See "[Deploying on a Single System](#)" (see page 25) for more information.
- **Distributed-System Deployment** - For production or staging environments
See "[Deploying on Distributed Systems](#)" (see page 27) for more information.
- **High-Availability Deployment** - For high availability and scalability, production, or staging environments
See "[Deploying in High-Availability Environment](#)" (see page 30) for more information.

Deploying on a Single System

In a single-system deployment, all components of AuthMinder and the Web applications are installed on a single system. The database can be on the same system where AuthMinder is installed, or on a different system. This deployment model is typically used for development, proof of concept, or initial testing.

It is possible to use both Java SDKs and Web services in a single-system deployment, refer to "Software Requirements" for the prerequisite software for these components.

To deploy AuthMinder on a single system, you must choose the *Complete* option during AuthMinder installation. See chapter, "[Deploying AuthMinder on a Single System](#)" (see page 59) for more information on the installation and post-installation steps.

Component Diagrams

The component diagrams depict a few of the possible deployment options for prerequisite software and AuthMinder components. If you perform a Complete install, then both Java SDKs and Web services will be installed on the system. You can use any of these methods for integrating AuthMinder with your Web application.

- Deploying Java SDKs
- Deploying Web Services

If you plan to perform a single-system deployment, then you must make the following decisions:

Decision Points:

- Install a database server on the system which has AuthMinder Server, or use an existing database on a separate system.
- Use Sample Application or write my own Web application.

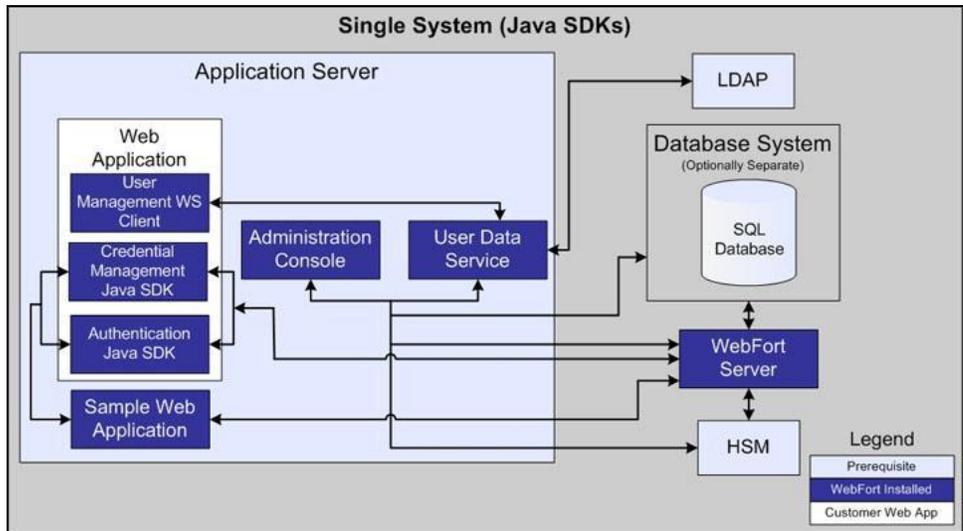
Important! Sample Application must *not* be used in production deployments. It is strongly recommended that you build your own Web application by using Sample Application as a code-reference.

- Use Java SDKs or Web services to integrate with your own Web application.

The following sections will help you to achieve your deployment decision.

Deploying Java SDKs

The following figure illustrates deployment of AuthMinder Server and Java SDKs on a single system:

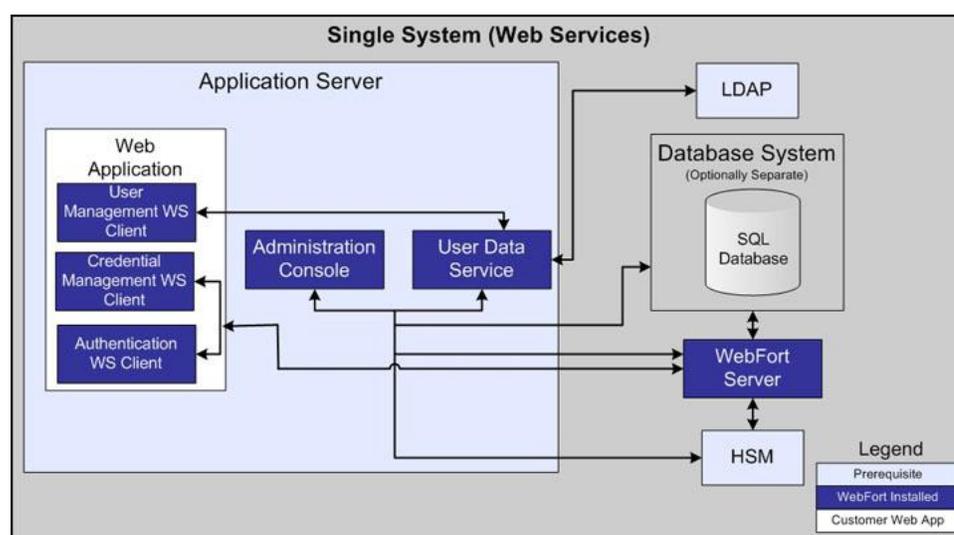


Note: The use of a Web server to deliver HTML pages for the application server is optional and is transparent to AuthMinder. In production deployments, this approach is generally used to improve Application Server performance and security. Refer to the documentation of your Application Server for detailed information.

Deploying Web Services

If you plan to deploy Web services, then the following figure illustrates AuthMinder Server and Web Services on a single system.

Note: Because all Web Services are now built into the AuthMinder Server module itself, you just need to install the AuthMinder Server on the target system and generate the requisite client stubs. No further configuration is required.



Deploying on Distributed Systems

In a distributed-system deployment, AuthMinder components are installed on different systems. This type of deployment increases the security and performance. This model is typically used for production deployments or staging environments.

The most common deployment is to install AuthMinder Server on one system and one or more Web applications on additional systems. To deploy AuthMinder on distributed systems, you must choose the *Custom* option during AuthMinder installation. See chapter, "[Deploying AuthMinder on Distributed Systems](#)" (see page 97) for more information on the installation and post-installation steps.

Component Diagrams

The diagrams in this section depict several possible options, where prerequisites and AuthMinder components can be installed on multiple systems:

- Deploying Single Application with Java SDKs
- Deploying Multiple Applications with Java SDKs
- Deploying Single Application with Web Services

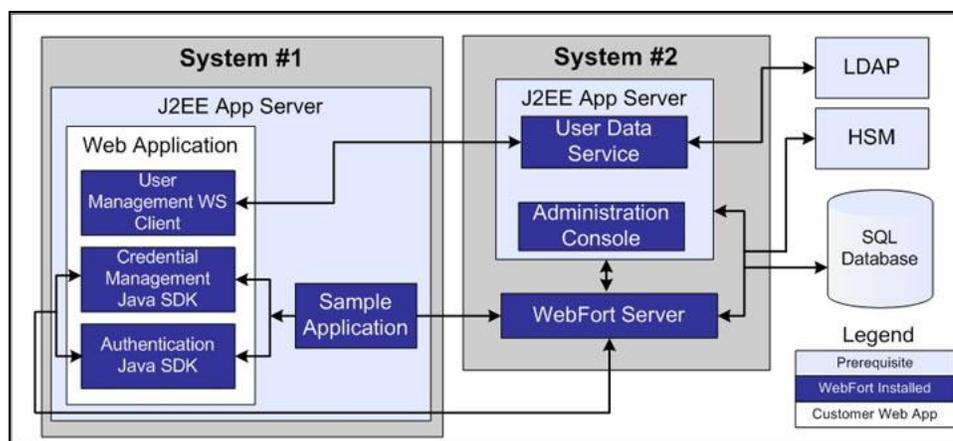
Decision Point

- Which AuthMinder components will be installed on each system?

The following sections will help you to achieve your deployment decision.

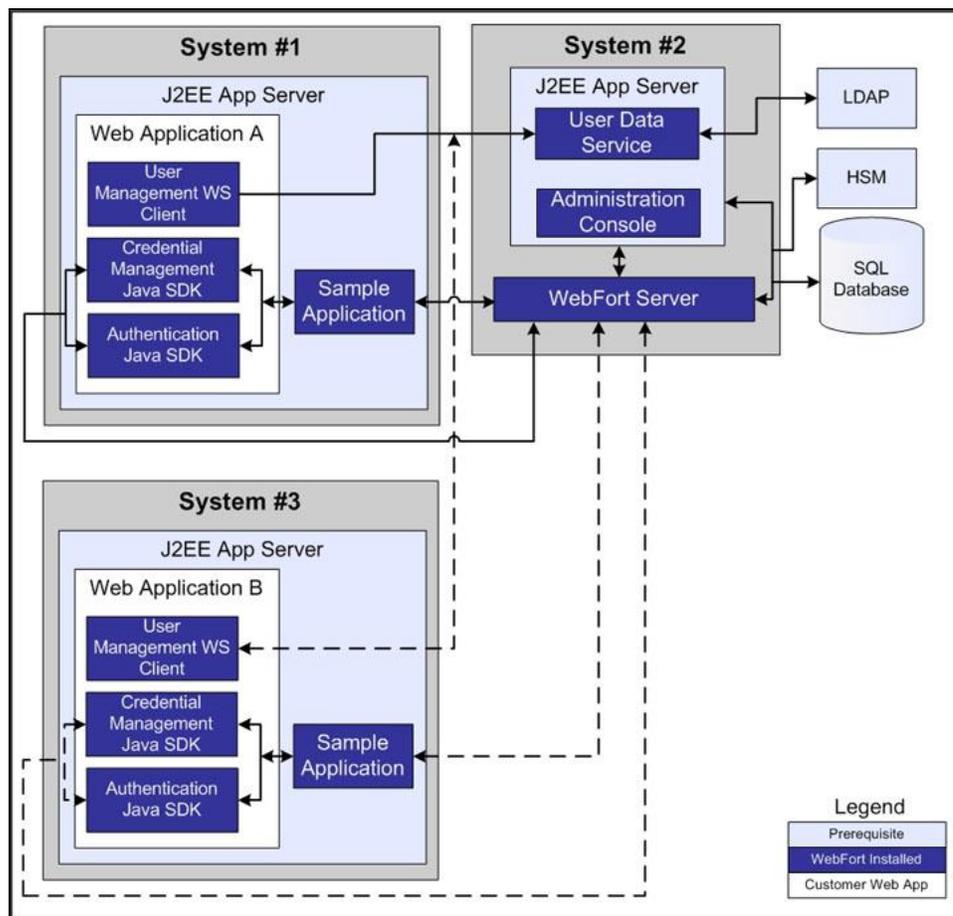
Deploying Single Application with Java SDKs

The following figure illustrates deployment of AuthMinder using Java SDKs on a single application:



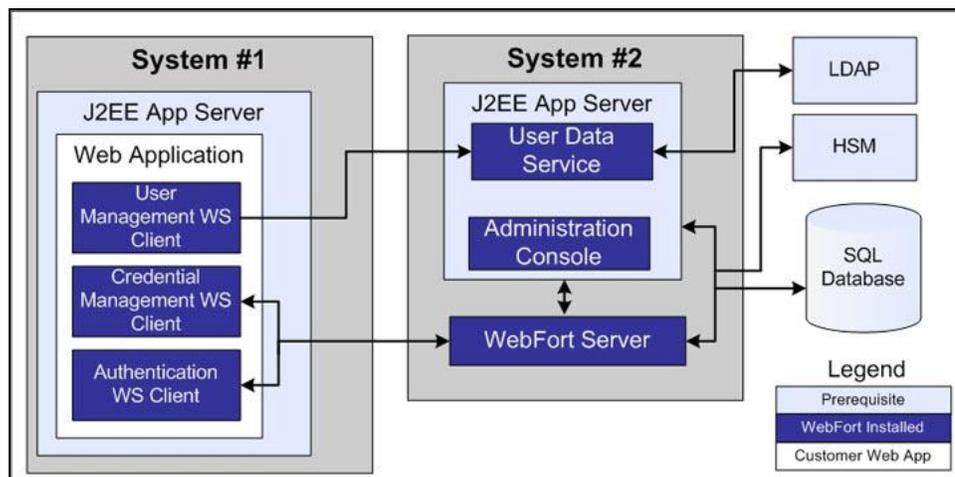
Deploying Multiple Applications with Java SDKs

The following figure illustrates AuthMinder deployment using Java SDK with multiple applications:



Deploying Single Application with Web Services

The following figure illustrates AuthMinder deployment using Web Services on a single application:



Deploying in High-Availability Environment

In a high-availability deployment, AuthMinder components are installed on more than one servers to provide high availability and scalability.

Component Diagrams

The diagrams in this section depict several possible options for which prerequisites and AuthMinder components can be installed on multiple systems for a high-availability deployment.

Decision Points

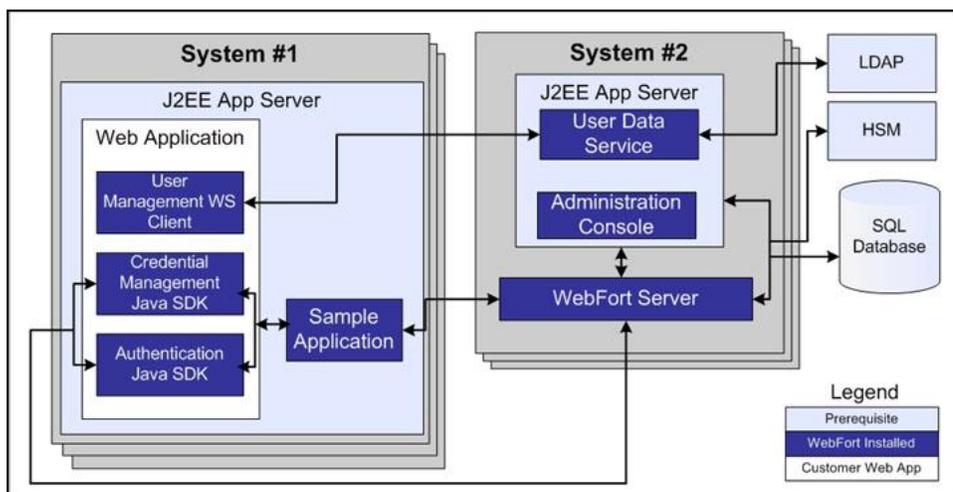
- When do I add a new server instance?
Typically, when your transaction rate exceeds the allowable threshold (as decided by your organizational policies), then you need to add a new server instance.
- How many AuthMinder Server, Administration Console, UDS, and SDK instances can I have?
 - **AuthMinder Servers:** Multiple instances are supported. The number depends on the transaction rate you want to achieve.
 - **Administration Consoles:** Multiple instances are supported. The number depends on the number of administrators in the system who log in to the Console at the same time.
 - **UDS Servers:** Currently, *only* one is supported. But if you need multiple instances of UDS, then you need to put them behind a Load Balancer. However, UDS failover is not supported.
 - **SDKs:** Multiple instances are supported. This number depends on the number of your application instances that you plan to support.

The following sections will help you to achieve your deployment decision.

- High-Availability Deployment Using Java SDK
- High-Availability Deployment Using Web Services

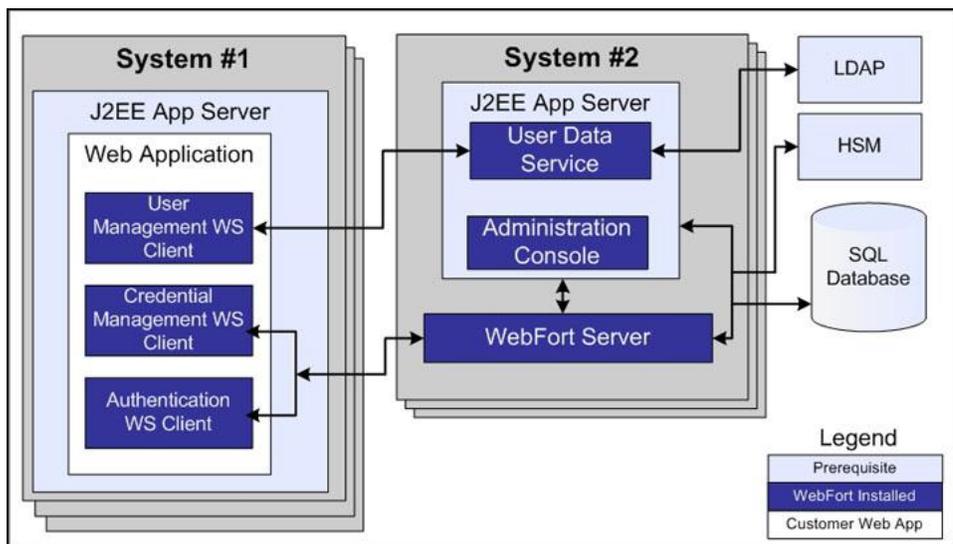
High-Availability Deployment Using Java SDK

The following figure illustrates multiple-instance deployment of AuthMinder using Java SDK.



High-Availability Deployment Using Web Services

The following figure illustrates multiple-instance deployment of AuthMinder using Web Services.



Chapter 3: Preparing for Installation

Before installing AuthMinder Server and its components, ensure that your computer meets the requirements described in this chapter. The chapter also provides configuration and planning-related information.

This chapter contains the following sections:

- [Hardware and Software Requirements](#) (see page 34)
- [Hardware Security Module \(HSM\) Requirements](#) (see page 35)
- [Software Requirements](#) (see page 35)
- [AuthMinder Component-Specific Prerequisites](#) (see page 42)
- [Configuring Database Server](#) (see page 43)
- [Getting Ready for Installation](#) (see page 50)
- [Pre-Installation Checklist](#) (see page 54)

Hardware Requirements

The minimum hardware requirements for installing AuthMinder include:

- Requirements for AuthMinder with database on single system:
 - RAM: 2 GB
 - Hard Drive Space: 10 GB
 - Processor: 2.4 GHz
- Requirements for AuthMinder with database on a different system:
 - RAM: 1 GB
 - Hard Drive Space: 300 MB
 - Processor: 2.4 GHz

The following 32-bit binaries are required for AuthMinder to function on an RHEL 6.x 64-bit computer. If you are planning to install AuthMinder, you can use the yum utility to install these binaries.

- `ibgfortran-4.4.5-6.el6.i686.rpm`
- `libstdc++-4.4.5-6.el6.i686.rpm`
- `libgomp-4.4.5-6.el6.i686.rpm`
- `libstdc++-devel-4.4.5-6.el6.i686.rpm`
- `compat-libtermcap-2.0.8-49.el6.i686.rpm`
- `ncurses-libs-5.7-3.20090208.el6.i686.rpm`
- `ncurses-5.7-3.20090208.el6.i686.rpm`

Note: Hardware resource requirements vary substantially for different applications and usage patterns. It is recommended that you load-test your site to determine the optimal memory required for the installation. While load-testing, you must keep in mind that some operating system utilities for monitoring memory can overstate memory usage (partially because of the representation of shared memory.) The preferred method for determining memory requirements is by monitoring the improvement in performance after adding more RAM/physical memory in the load test. Refer to your platform vendor documentation for information about how to configure memory and processor resources for testing purposes.

Hardware Security Module (HSM) Requirements

You can now store sensitive keys either in the database or in an HSM. Currently, you can store the various encryption keys and the AuthMinder Server listener SSL key in the HSM. The following table lists the requirements for the supported HSM modules.

HSM Module	Java Cryptography Extension (JCE)	PKCS #11
Thales nCipher netHSM (or nCipher netHSM)	JCE framework provided with the 32-bit versions of JDK 5.0, JDK 6.0, and JDK 7.0	pkcs11v2.01
SafeNet High Availability HSM (or Luna HSM)		

Note: The decision to use and configure an HSM, if required, must be made while you are still in the planning and preparation stages. Otherwise, you will need to re-initialize the database later, because all your current encryption would use keys in software.

Software Requirements

The following information for software requirement is provided in this section:

- [Minimum Software Requirements](#) (see page 35)
- [AuthMinder Component-Specific Prerequisites](#) (see page 42)

Minimum Software Requirements

This section lists the supported software for the following versions of Solaris and Linux. The following information for software requirement is provided in this section:

- [For Solaris SPARC](#) (see page 35)
- [For Red Hat Enterprise Linux](#) (see page 39)

For Solaris SPARC

The following table lists the minimal software required to install AuthMinder.

Note: For all third-party software mentioned in the following table, it is assumed that the higher versions are compatible with the specified supported version.

Software Type	Version
Operating System	Solaris 10 (SPARC)

Software Type	Version
Patches	<p>Latest patches</p> <p>Access latest patches at http://sunsolve.sun.com. Click the Patches and Updates link, click the Patch Cluster & Patch Bundle Downloads link, and under Solaris Patch Clusters expand the Recommended Patch Clusters to display the Solaris 10 SPARC 05/08 Patch Bundle entries.</p>
Database Server	<ul style="list-style-type: none"> ■ Microsoft SQL Server 2005, Standard Edition (SP2) or higher ■ Microsoft SQL Server 2008 Enterprise Edition
	<ul style="list-style-type: none"> ■ Oracle 10g ■ Oracle 11g or higher
	<ul style="list-style-type: none"> ■ IBM DB2 UDB 9.5 ■ IBM DB2 UDB 9.7
	<ul style="list-style-type: none"> ■ MySQL Enterprise Edition 5.1
JDBC Drivers (JARs) Important! It is strongly recommended that the JDBC JAR version is same as or higher than your database server version.	The JDBC driver version that is compatible with your database.
Directory Server	<p>The following Directory Servers are supported:</p> <ul style="list-style-type: none"> ■ SunOne Directory Server 5.3 ■ SunOne Directory Server 6.1 ■ SunOne Directory Server 6.3 ■ Oracle Directory Server 11g ■ CA Directory Server 12.0 Service Pack 10 ■ Microsoft Windows Active Directory 2003 ■ Microsoft Windows Active Directory 2008

Software Type	Version
Application Server	<p>The following Application Servers are supported:</p> <ul style="list-style-type: none">■ Apache Tomcat 5.5.x (x can be 31 or higher)■ Apache Tomcat 6.x■ Apache Tomcat 7.x■ Oracle WebLogic 10.1.x■ Oracle WebLogic 11g (WebLogic Server 10.3)■ IBM WebSphere 6.1.x <p style="text-align: right;">Important! If you are planning to use WebSphere 6.1, then ensure that you apply the 6.1.0.41: WebSphere Application Server V6.1 Fix Pack 41 and 6.1.0.41: Java SDK 1.5 SR12 FP5 Cumulative Fix for WebSphere Application Server.</p> <ul style="list-style-type: none">■ IBM WebSphere 7.0■ JBoss 5.1.x

Software Type	Version
	<p>The JVM Memory Settings (Heap Size) for the application server must be a minimum of 512 MB or higher to support User Data Service (UDS) deployment.</p> <p>Note: If you plan to create organizations in LDAP repository with large user base (for example, 100,000 users), then it is recommended to increase the heap size to 1 GB or higher.</p> <p>To set the heap size to 512 MB, use the <code>-Xmx512M</code> JVM memory setting. To set the heap size to 1 GB, use the <code>-Xmx1024M</code> JVM memory setting.</p> <p><i>Do not</i> use the <code>-Xms</code> parameter when you set the JVM memory setting.</p>
<p>JDK</p> <p>Note: If you perform a fresh installation of JDK, then you must include the new path in the <code>JAVA_HOME</code> environment variable, and ensure that the application server uses the same <code>JAVA_HOME</code>. If you fail to do so, then the Administration Console and other JDK-dependant components might fail to start.</p>	<p>The JDK version that is best compatible with the Application Server that you are using:</p> <ul style="list-style-type: none"> ■ IBM JDK 1.5 or higher ■ IBM JDK 1.6 or higher ■ Oracle JDK 5.0 ■ Oracle JDK 6.0 ■ Oracle JDK 7.0 ■ Oracle JRockit 5.0 or higher ■ Oracle JRockit 6.0 or higher <p style="color: red;">Important! <code><JROCKIT_HOME>/jre/bin /</code> must be included in <code>PATH</code> environment variable. In addition, This change in the <code>PATH</code> variable must be effective before you start the WebLogic application server.</p>
<p>Web Service Clients</p>	<p>The following clients are supported:</p> <ul style="list-style-type: none"> ■ Axis2 1.5 or higher ■ .NET Framework 4 or higher

Software Type	Version
Browsers	<p>The following Web browsers are supported:</p> <ul style="list-style-type: none"> ■ Mozilla Firefox 3.0 or higher ■ Apple Safari 5.0 or higher ■ Google Chrome 12 or higher ■ Internet Explorer 7.0 or higher

For Red Hat Enterprise Linux

The following table lists the minimal software required to install AuthMinder.

Note: For all third-party software mentioned in the following table, it is assumed that the higher versions are compatible with the specified supported version.

Software Type	Version
Operating System	<ul style="list-style-type: none"> ■ Red Hat Enterprise Linux 4.x (x86) (32-bit) ■ Red Hat Enterprise Linux 5.x (x86) (32-bit) ■ Red Hat Enterprise Linux 5.x (x86) (64-bit) ■ Red Hat Enterprise Linux 6.1 (32-bit and 64-bit) ■ Red Hat Enterprise Linux 6.2 (32-bit and 64-bit)
Update	Update 1 and higher
Patches	<p>Latest patches</p> <p>Access the latest patches at http://www.redhat.com. Log in to your account, download the latest updates and patches, and apply them as needed.</p>
Database Server	<ul style="list-style-type: none"> ■ Microsoft SQL Server 2005, Standard Edition (SP2) or higher ■ Microsoft SQL Server 2008 Enterprise Edition ■ Oracle 10g or higher ■ Oracle 11g Release 2 ■ IBM DB2 UDB 9.5 ■ IBM DB2 UDB 9.7 ■ MySQL Enterprise Edition 5.1

Software Type	Version
<p>JDBC Drivers (JARs)</p> <p>Important! It is strongly recommended that the JDBC JAR version is same as or higher than your database server version.</p>	<p>The JDBC driver version that is compatible with your database.</p>
<p>Directory Server</p>	<p>The following directory servers are supported:</p> <ul style="list-style-type: none"> ■ SunOne Directory Server 5.3 ■ SunOne Directory Server 6.1 ■ SunOne Directory Server 6.3 ■ Oracle Directory Server 11g ■ CA Directory Server 12.0 Service Pack 10 ■ Microsoft Windows Active Directory 2003 ■ Microsoft Windows Active Directory 2008
<p>Application Server</p>	<p>The following Application Servers are supported:</p> <ul style="list-style-type: none"> ■ Apache Tomcat 5.5.x (x can be 31 or higher) ■ Apache Tomcat 6.x ■ Apache Tomcat 7.x ■ Oracle WebLogic 10.1.x ■ IBM WebSphere 6.1.x <p style="text-align: right;">Important! If you are planning to use WebSphere 6.1, then ensure that you apply the 6.1.0.41: WebSphere Application Server V6.1 Fix Pack 41 and 6.1.0.41: Java SDK 1.5 SR12 FP5 Cumulative Fix for WebSphere Application Server.</p> <ul style="list-style-type: none"> ■ IBM WebSphere 7.0 ■ JBoss 5.1.x ■ Oracle WebLogic 11g (WebLogic Server 10.3)

Software Type	Version
	<p>The JVM Memory Settings (Heap Size) for the application server must be a minimum of 512 MB or higher to support User Data Service (UDS) deployment.</p> <p>Note: If you plan to create organizations in LDAP repository with large user base (for example, 100,000 users), then it is recommended to increase the heap size to 1 GB or higher.</p> <p>To set the heap size to 512 MB, use the -Xmx512M JVM memory setting. To set the heap size to 1 GB, use the -Xmx1024M JVM memory setting.</p> <p><i>Do not</i> use the -Xms parameter when you set the JVM memory setting.</p>
<p>JDK</p> <p>Note: If you perform a fresh installation of JDK, then you must include the new path in the JAVA_HOME environment variable, and ensure that the application server uses the same JAVA_HOME. If you fail to do so, then the Administration Console and other JDK-dependant components might fail to start.</p>	<p>The JDK version that is best compatible with the Application Server that you are using:</p> <ul style="list-style-type: none"> ■ IBM JDK 1.5 or higher ■ IBM JDK 1.6 or higher ■ Oracle JDK 5.0 or higher ■ Oracle JDK 6.0 or higher ■ Oracle JDK 7.0 ■ Oracle JRockit 5.0 or higher ■ Oracle JRockit 6.0 or higher <p>Important! If you are using JRockit, then you must ensure that <code><JROCKIT_HOME>/jre/bin/</code> must be included in PATH environment variable. In addition, This change in the PATH variable must be effective before you start the WebLogic application server.</p>
<p>Web Service Clients</p>	<p>The following clients are supported:</p> <ul style="list-style-type: none"> ■ Axis2 1.5 or higher ■ .NET Framework 4 or higher

Software Type	Version
Browsers	<p>The following Web browsers are supported:</p> <ul style="list-style-type: none"> ■ Mozilla Firefox 3.0 or higher ■ Apple Safari 5.0 or higher ■ Google Chrome 12 or higher ■ Internet Explorer 7.0 or higher

AuthMinder Component-Specific Prerequisites

The prerequisite software is determined by the AuthMinder components that will be installed on a system. See [Planning the Deployment](#) (see page 21) to determine which AuthMinder components to install for each deployment type.

The following table indicates the prerequisite software required by each AuthMinder component:

Component	Prerequisite		
	Database Server	JDK	Application Server
AuthMinder Server	✓		
Administration Console	✓	✓*	✓
User Data Service	✓	✓*	✓
Authentication Java SDK		✓*	✓
Credential Management Java SDK		✓*	✓
User Management Web Service		✓*	✓
Authentication Web Service		✓*	✓
Credential Management Web Service		✓*	✓
Sample Application		✓*	✓

* The JDK depends on the application server you are using.

Configuring Database Server

Before installing AuthMinder, you must set up a database that is used for storing user information, server configuration data, audit log data, and other information.

AuthMinder supports a primary database as well as a backup database that can be used during failover and failback in high-availability deployments. Database connectivity can be configured in either of the following ways:

- Automatically during AuthMinder installation, when the Installer edits the [arcotcommon.ini](#) (see page 181) file with the database information you supply.
- By manually:
 - Creating the Data Source Name (DSN)
 - Editing the [arcotcommon.ini](#) (see page 181) file.
 - Updating securestore.enc using the dbutil tool

There are specific configuration requirements for each supported database. Use the following information to set up the database server yourself, or provide this information to your database administrator (DBA) when you request a database account.

Important! To protect the database, it is recommended that you ensure that the database server is protected with firewall or other access control mechanisms and is set to the same time-zone as all Arcot products.

- [Configuring Microsoft SQL Server](#) (see page 43)
- [Configuring Oracle Database](#) (see page 45)
- [Configuring IBM DB2 Universal Database](#) (see page 47)
- [Configuring MySQL](#) (see page 48)

Configuring Microsoft SQL Server

This section provides the following configuration information for MS SQL Server:

Note: Refer to MS SQL Server documentation for detailed information on performing the tasks listed in this section.

1. [Verifying Authentication Mode](#) (see page 44)
2. [Creating a New Database](#) (see page 44)
3. [Creating a Database User](#) (see page 44)

Verifying Authentication Mode

Verify that MS SQL Server is configured to use the "**SQL Server Authentication**" authentication method. AuthMinder will not be able to connect to the database if SQL Server is configured to "**Windows Only**" authentication mode.

Creating a New Database

Use the following criteria to create a database:

1. The recommended name is arcotdb.
2. The database size must be configured to automatically grow.

Creating a Database User

Use the following steps to create a database user:

Note: MS SQL Server refers to user as a Login.

1. In the SQL Server Management Studio, go to `<SQL_Server_Name>`, expand the **Security** folder, and then click **Logins**.

Note: The `<SQL_Server_Name>` refers to the host name or IP address of the SQL Server where you created your database.

2. Right-click the **Logins** folder, and click **New Login**.
3. Enter the Login name. The recommended name is arcotuser.
4. Set the following parameters:
 - a. Authentication to **SQL Server Authentication**.
 - b. Specify **Password** and **Confirm password** for the login.
Ensure that you specify other password settings on this page according your organization's password policies.
 - c. **Default database** to the database (arcotdb) you created.
 - d. **User Mapping** for the login (in the **Users mapped to this login section**).
 - e. **User Mapping (SQL 2005)** for the default database to db_owner (in the **Database role membership for: <db_name>** section).

Configuring Oracle Database

This section provides the configuration information for Oracle database and AuthMinder Server.

Note: Refer to the Oracle database documentation for details on performing the tasks listed in the following sections.

Required Tablespaces

Running AuthMinder on Oracle requires two tablespaces:

- The first tablespace is used for configuration data, audit logs, and user information. This tablespace can be the default user tablespace in the AuthMinder database.
See "[Creating a New Database](#)" (see page 46) for creating a database.
- The second tablespace is used to run reports. For high performance, it is recommended that you use a separate tablespace.

Database Configuration Script

The database configuration script, `arcot-db-config-for-common-2.0.sql`, automatically creates the tablespace for reports if the database user running the script has sufficient permissions to create a tablespace. If the user does not have the required permissions, a DBA will need to manually create this tablespace and delete the section in this script which creates the reports tablespace.

If a tablespace with that name already exists, then it is removed and re-created.

Important! The parameters for creating the reports tablespace in the `arcot-db-config-for-common-2.0.sql` database script can be changed as per the DBA's preferences. However, the tablespace name must be `ARReports` to generate reports successfully.

Perform the following steps to set up the Oracle database:

1. [Creating a New Database](#) (see page 46)
2. [Creating a Database User](#) (see page 46)

Creating a New Database

Create a new database that stores information in the UTF-8 character set. This allows AuthMinder to use international characters including double-byte languages.

Perform the following steps to enable UTF-8 support:

1. Log in to the Oracle database server as SYS or SYSTEM.
2. Run the following command:
Update `sys.props$ set value$='UTF8'` where
name='NLS_NCHAR_CHARACTERSET' Or name = 'NLS_CHARACTERSET';
3. Restart the database and check whether the character set is configured to UTF-8.

Creating a Database User

Create a user with the following criteria:

1. Create a user (the recommended name is arcotuser) in the new database arcotdb.
2. Set the quota of user to *at least* 5 to 10GB for a development or test deployment, which is primarily used for audit logs.

Note: If the deployment is for production, staging, or other intensive testing, refer to appendix, "Database Reference" to determine the quota required for an user.

3. Grant the user the following privileges:
 - CREATE TABLE
 - CREATE INDEX
 - CREATE SEQUENCE
 - CREATE PROCEDURE
 - CREATE SESSION
 - DML PRIVILEGES
 - RESOURCE PRIVILEGES
 - CONNECT PRIVILEGES
 - ALTER TABLE
 - Additional privileges for upgrade only:
 - ALTER EXTENT PARAMETERS
 - CREATE TABLESPACE
 - Additional privileges for working with reports:
 - UNLIMITED TABLESPACE
 - (Optional) DROP TABLESPACE

Configuring IBM DB2 Universal Database

This section provides the following configuration information for IBM DB2 Universal Database (UDB):

Note: Refer to IBM DB2 UDB documentation for detailed information on performing the tasks listed in this section.

- [Creating a New Database](#) (see page 47)
- [Creating a Database User](#) (see page 48)

Creating a New Database

Create a new database (the recommended name is `arcotdb`) that stores information in the UTF-8 character set. This allows AuthMinder to use international characters including double-byte languages.

Perform the following steps to configure the IBM DB2 UDB:

1. Log in to the IBM DB2 UDB database server.
2. Run the following command to enable UTF-8 support:
`create db <DB-NAME> using codeset utf-8 territory us;`
3. Set the tablespace page size to **16K**. By default, the tablespace page size is 4K.
Refer to the vendor documentation for more information on changing the tablespace page size.
4. If your data volume is high, then the default size of transaction log file might be insufficient. Therefore, it is recommended to increase the log file size.
Refer to the vendor documentation for more information on changing the log file size.
5. Ensure that the configuration changes have been applied.

If you plan to use alternate schema, then refer to appendix, "[Configuring Alternate Schema for IBM DB2 Universal Database](#)" (see page 227) for more information.

Creating a Database User

Create a user with the following criteria:

1. Create a user (recommended name is `arcotuser`), with a schema in the new database `arcotdb`.
2. Grant the user the following privileges:
 - CREATE TABLE
 - CREATE INDEX
 - CREATE SEQUENCE
 - CREATE PROCEDURE
 - CREATE SESSION
 - DML PRIVILEGES
 - CONNECT PRIVILEGES
 - ALTER TABLE
 - Additional privileges for upgrade only:
 - CREATE TABLESPACE (with `AUTORESIZE = yes`)
 - Additional privileges for working with reports:
 - DROP TABLESPACE

Configuring MySQL

This section provides the following configuration information for MySQL:

Note: Refer to MySQL documentation for detailed information on performing the tasks listed in this section.

- [Enabling Support for the InnoDB Transaction Engine](#) (see page 49)
- [Creating a New Database](#) (see page 49)
- [Creating a Database User](#) (see page 50)

Enabling Support for the InnoDB Transaction Engine

AuthMinder uses the InnoDB storage engine of MySQL. To check whether this storage engine is supported by your MySQL installation, use the SHOW ENGINES command. If the output of this command shows that InnoDB is not supported, enable support for InnoDB.

Note: For information about the procedure to enable support for InnoDB, see MySQL documentation.

Creating a New Database

To create a database:

1. Open a MySQL command window.
2. Run the following command to create the database schema:
`CREATE SCHEMA '<schema-name>' DEFAULT CHARACTER SET utf8;`
3. Run the following command to create the database user:
`CREATE USER '<user-name>' identified by '<user-password>';`

Creating a Database User

Create a user with the following criteria:

1. Create a user (recommended name is arcotuser) in the new database arcotdb.
2. Grant the user the following privileges:
 - Object rights:
 - SELECT
 - INSERT
 - UPDATE
 - DELETE
 - EXECUTE
 - DDL rights:
 - CREATE
 - ALTER
 - CREATE ROUTINE
 - ALTER ROUTINE
 - DROP
 - Other rights:
 - GRANT OPTION

Getting Ready for Installation

Before you proceed with AuthMinder installation, you must set up the AuthMinder datastore, the Database Client, and gather the required Database information for use during the installation later. You must also ensure that the prerequisite JDK version and application server required by AuthMinder components are installed.

This section discusses the following topics:

- [Configure UTF-8 Support on Client Systems](#) (see page 51)
- [Database Information that You Need for Installing AuthMinder](#) (see page 51)
- [Requirements for Java-Dependent Components](#) (see page 53)
- [\(Optional, Only If You are Using HSMs\) Requirements for HSM](#) (see page 54)

Configure UTF-8 Support on Client Systems

You need to enable the UTF-8 support on the systems where you plan to install AuthMinder components that communicate with the database server. For example, AuthMinder Server, Administration Console, and User Data Service. This section walk you through the steps to do so.

To enable UTF-8 support on your UNIX platforms, set the following environment variables:

- NLS_LANG=en_US.UTF-8
- LC_CTYPE=en_US.UTF-8

Database Information that You Need for Installing AuthMinder

Perform the tasks described in this section on the system where you will install AuthMinder or the system that uses AuthMinder components.

MS SQL Server

Get the following database information from the DBA. You will need this information when you install AuthMinder:

1. Name of the DSN to be created by the installer
2. Server
3. Database
4. User Name
5. Password
6. Port Number

For more information about these parameters, see the table for MS SQL Server in [Performing Complete Installation](#) (see page 62).

IBM DB2 UDB

Get the following database information from the DBA. You will need this information when you install AuthMinder:

1. Name of the DSN to be created by the installer
2. Server
3. Database
4. Host Name
5. Port Number
6. User Name
7. Password

For more information about these parameters, see the table for IBM DB2 UDB in [Performing Complete Installation](#) (see page 62).

Oracle Database

Get the following database information from the DBA. You will need this information when you install AuthMinder:

1. Name of the DSN to be created by the installer
2. Service ID (Instance identifier of the Oracle database)
3. Host Name
4. Port Number
5. User Name
6. Password

For more information about these parameters, see the table for Oracle Database in [Performing Complete Installation](#) (see page 62).

MySQL

Get the following database information from the DBA. You will need this information when you install AuthMinder:

1. Name of the DSN to be created by the installer
2. Server
3. Database
4. User Name
5. Password
6. Port Number

For more information about these parameters, see the table for MySQL in [Performing Complete Installation](#) (see page 62).

Requirements for Java-Dependent Components

Install the following components required by Administration Console, AuthMinder Java SDKs, and Web Services:

- JDK

Note: If you perform a fresh installation of JDK, then you *must* set the JAVA_HOME environment variable. The path variable must point to \$JAVA_HOME/**bin**/. And ensure that the application server uses the same JAVA_HOME. If you fail to do so, then the Administration Console and other JDK-dependant components might fail to start.

- Application Server
- UDS

(Optional, Only If You are Using HSMs) Requirements for HSM

If you are planning to use HSM to store encryption keys, then set up the following before you proceed:

1. HSM Server
2. HSM Client
3. At least one 3DES key created in HSM

Important! Ensure that you have securely written down the labels of these 3DES keys. You will need them later for encrypting information in the database.

Refer to your platform vendor documentation for detailed information about how to install and configure your HSM Server and Client components and generate the required keys.

Pre-Installation Checklist

It is recommended that you complete this checklist before you proceed with the installation and setup of AuthMinder.

Note: The items and values in the following checklist are samples. Before you begin installation, modify this checklist so that it meets the requirements of your operating environment.

Your Information	Example Entry	Your Entry
HARDWARE		
Processor	SPARC	
RAM	2 GB	
Disk Space	20 GB	
SOFTWARE		
Operating System	Solaris 10	
Distribution	Enterprise Edition	
Service Pack (or Patch)	SP3	
DATABASE		
Type	Oracle	
Database Name (<i>MS SQL and DB2 Only</i>)	arcotdb	
DSN Name	arcotdsn	
Host Name (or Server IP Address)	51.100.25.24	

Your Information	Example Entry	Your Entry
Port	1521	
Service ID (<i>Oracle Databases Only</i>)	oradb1	
Database User	arcotuser	
Database Login Password	password1234!	
Configured Privileges: Note: For all CREATE privileges, the corresponding DROP privilege is implied.		
Oracle Database		
CREATE TABLE		
CREATE INDEX		
CREATE SEQUENCE		
CREATE PROCEDURE		
CREATE SESSION		
DML PRIVILEGES		
RESOURCE PRIVILEGES		
CONNECT PRIVILEGES		
ALTER TABLE (<i>For Upgrade Only</i>)		
ALTER EXTENT PARAMETERS		
CREATE TABLESPACE (<i>For Reports</i>)		
UNLIMITED TABLESPACE (<i>For Reports, Optional</i>)		
DROP TABLESPACE		
MS SQL Server Note: The user performing these actions must belong to the ddladmin role.		
CREATE TABLE		
CREATE INDEX		
CREATE PROCEDURE		

Your Information	Example Entry	Your Entry
REFERENCES		
DML PRIVILEGES		
CONNECT PRIVILEGES		
ALTER <i>(For Upgrade Only)</i>		
IBM DB2 UDB		
CREATE TABLE		
CREATE INDEX		
CREATE SEQUENCE		
CREATE PROCEDURE		
CREATE SESSION		
DML PRIVILEGES		
CONNECT PRIVILEGES		
ALTER TABLE <i>(For Upgrade Only)</i>		
CREATE TABLESPACE (with AUTORESIZE = yes) <i>(For Reports)</i>		
DROP TABLESPACE		
MySQL		
SELECT		
INSERT		
UPDATE		
DELETE		
EXECUTE		
CREATE		
ALTER		
CREATE ROUTINE		
ALTER ROUTINE		
DROP		
GRANT OPTION		

Your Information	Example Entry	Your Entry
APPLICATION SERVER		
Type	Apache Tomcat 5.5	
Host Name	localhost	
Port	8080	
JDK	1.5.0_10	
DIRECTORY SERVICE (OPTIONAL)		
Host Name	ds.myldap.com	
Port	389	
Schema Name	inetorgperson or user	
Base Distinguished Name	dc=myldap,dc=com	
User Name	cn=admin,cn=Administrators,cn=dsc	
Password	mypassword1234!	
WEB SERVER (OPTIONAL)		
Type	IIS 6	
Host Name	mywebserver.com	
Port	443	

Chapter 4: Deploying AuthMinder on a Single System

Use the **Arcot WebFort 7.1 InstallAnywhere Wizard** to install AuthMinder components. This Wizard supports *Complete* and *Custom* installation types. However, to install and configure AuthMinder on a single computer, you must use the **Complete** option when you run the installer.

The following list provides a quick overview of the process:

1. Execute the AuthMinder installer to add AuthMinder components to your file-system and configure them to access your SQL database.
See "[Performing Complete Installation](#)" (see page 62) for install instructions.
2. Execute the database scripts to create schema and database tables. Also ensure that the database setup was successful.
See "[Running Database Scripts](#)" (see page 70) and "[Verifying the Database Setup](#)" (see page 71) for more information.
3. Prepare the application server to copy the files that Web components use.
See "[Preparing Your Application Server](#)" (see page 72) for more information.
4. Deploy Administration Console on the application server and verify the deployment.
See "[Deploying Administration Console](#)" (see page 79) and "[Verifying the Administration Console](#)" (see page 81) for more information.
5. (Optional) Prepare your Administration Console for localization, if you want to use languages other than English.
See appendix, "[Preparing for Localization](#)" (see page 203) for more information.

6. Log in to Administration Console as the Master Administrator to initialize AuthMinder.
See "[Logging In to Administration Console](#)" (see page 81) and "[Bootstrapping the System](#)" (see page 82) for more information.
7. Start the AuthMinder Server and verify if the service start successfully.
See "[Starting AuthMinder Server](#)" (see page 84) and "[Verifying the Installation](#)" (see page 85) for more information.
8. Deploy User Data Service on the application server and verify the deployment.
See "[Deploying User Data Service](#)" (see page 87) for more information.
9. Deploy and use Sample Application to test AuthMinder configuration.
Note: Sample Application is automatically installed as a part of Complete installation.
See "[Deploying Sample Application](#)" (see page 89) and "[Using Sample Application](#)" (see page 89) for more information.
10. (Optional) To ensure secure communication between AuthMinder components, you can configure them to support SSL (Secure Socket Layer) transport mode.
See the topic titled "Configuring SSL" in *CA AuthMinder Administration Guide* for more information.
11. Complete the installation checklist.
See "[Post-Installation Checklist](#)" (see page 95) for more information.

Important Notes Related to the Installation

You must keep the following points in mind while installing AuthMinder either on a single system or in a distributed environment:

- You must ensure that the *<install_location>* *must not contain* any special characters (such as ~ ! @ # \$ % ^ & * () _ + = { } [] ' ").
- Currently, you cannot modify or repair AuthMinder components by using the installer. You *must* uninstall the component and then re-install it.
- Do not close the installer window, if the installation is in progress. If at any time during installation (*especially during the last stages*), if you click the **Cancel** button to abort the installation, then the installer might not remove *all* the directories it has created so far. You will manually need to clean up the installation directory, *<install_location>/arcot/* and its subdirectories.
- If you run the installer on a system that already contains an instance of an existing \$ARCOT_HOME, then:
 - You will *not* be prompted for an installation directory.

- You will *not* be prompted for the database set up. The Installer will use the existing database.
- You will also *not* be prompted to set up encryption.

The AuthMinder installer supports the following types of installation. You must use the *Complete* installation for single-system deployment.

1. **Complete** - Installs all AuthMinder components on a single system.
2. **Custom** - Installs the AuthMinder components that you select.

Performing Complete Installation

To install all AuthMinder component on a single system, you must use the *Complete* option. *Custom* installation allows you to install only the selected components from the package. This option is recommended for advanced users.

Note: Before proceeding with the installation, ensure that all the prerequisite software are installed and the database is set up, as described in chapter, "[Preparing for Installation](#)" (see page 33).

Follow these steps:

1. Log in and navigate to the directory where you untarred the installer.
2. Ensure that you have the permission to run the installer. If not, you must run the following command:

```
chmod a=rx Arcot-WebFort-7.1-<platform_name>-Installer.bin
```
3. Run the installer as follows:
prompt> sh Arcot-WebFort-7.1-<platform_name>-Installer.bin
The Welcome message appears.
4. Press Enter to continue with the installation.
The License Agreement for AuthMinder appears.
5. On the License Agreement page:
 - a. Read the text carefully and press Enter to display the next page of the license text. You might have to press Enter multiple times, until the entire text for License Agreement is displayed.

At the end of the license agreement, you will be prompted for acceptance of the terms of license agreement (DO YOU ACCEPT THE TERMS OF LICENSE AGREEMENT?).
 - b. Enter y to accept the acceptance of License Agreement and to continue with the installation.

Note: If you press n, then a warning message will be displayed and the installation will be aborted.

The Choose Installation Location options appear.
6. Perform one of the following steps:
 - If the installer detects an existing Arcot home directory, then it displays the path to that directory. Press Enter to accept this directory path.
 - If the installer does not detect an existing Arcot home directory, it displays the default directory path and prompts you to either accept the default path or specify a new path. Press Enter if you want to accept the default path. Alternatively, enter the absolute path of the directory where you want to install AuthMinder and press Enter to continue.

Note: The installation directory name that you specify must not contain spaces. Otherwise, some AuthMinder scripts and tools might not function as intended.

The installation types (Complete and Custom) supported by AuthMinder appear.

7. Enter 1 to select the default (Complete) option and install all components of AuthMinder, and press Enter to continue.

The Database Type options appear.

8. Enter the number corresponding to the database, and press Enter to continue:
 - 1 - Microsoft SQL Server
 - 2 - IBM DB2 (UDB)
 - 3 - Oracle Database
 - 4 - MySQL

The Primary Database Access Configuration options appear.

9. Depending on the database that you are using:
 - Specify the information listed in the following table if you specified 1 (Microsoft SQL Server) in the preceding step.

Parameter	Description
Primary ODBC DSN	The installer creates an ODBC connection that AuthMinder uses to connect to the database. The recommended value to enter is arcotdsn.
User Name	The database user name for AuthMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) Note: It is recommended that you use different user names for the primary and backup DSNs.
Password	The password associated with the User Name you specified in the previous field and which is used by AuthMinder to access the database. This password is specified by the database administrator.
Server Name	The host name or IP address of the AuthMinder datastore. <ul style="list-style-type: none"> ■ Default Instance Syntax: <server_name> Example: demodatabase ■ Named Instance Syntax: <server_name>\<instance_name> Example: demodatabase\instance1

Parameter	Description
Port Number	The port on which the database server listens to the incoming requests. Note: Press Enter, if you want to accept the default port.
Database	The name of the MS SQL database instance.

- Specify the information listed in the following table if you specified **2** (IBM DB2 (UDB)) in the preceding step:

Parameter	Description
Primary ODBC DSN	The installer creates an ODBC connection that AuthMinder uses to connect to the database. The recommended value to enter is arcotdsn.
User Name	The database user name for AuthMinder to access the database. This name is specified by the database administrator. Note: The User Name for the Primary and Backup DSNs <i>must</i> be different.
Password	The password associated with the User Name you specified in the previous field and which is used by AuthMinder to access the database. This password is specified by the database administrator.
Host Name	The host name or IP address of the AuthMinder datastore. <ul style="list-style-type: none"> ■ Default Instance Syntax: <server_name> Example: demodatabase ■ Named Instance Syntax: <server_name>\<instance_name> Example: demodatabase\instance1
Port Number	The port at which the Database listens to the incoming requests. Note: Press Enter, if you want to accept the default port.
Database	The name of the IBM DB2 database instance.

- Specify the information listed in the following table if you specified 3 (Oracle Database Server) in the preceding step:

Parameter	Description
Primary ODBC DSN	The installer creates an ODBC connection that AuthMinder uses to connect to the database. The recommended value to enter is arcotdsn.
User Name	The database user name for AuthMinder to access the database. This name is specified by the database administrator. Note: The User Name for the Primary and Backup DSNs <i>must</i> be different.
Password	The password associated with the User Name you specified in the previous field and which is used by AuthMinder to access the database. This password is specified by the database administrator.
Service ID	The Oracle System Identifier (SID) that refers to the instance of the Oracle database running on the server.
Port Number	The port at which the Database listens to the incoming requests. Note: Press Enter, if you want to accept the default port.
Host Name	The host name or IP address of the AuthMinder datastore. Syntax: <server_name> Example: demodatabase

- Specify the information listed in the following table if you specified 4 (MySQL) in the preceding step.

Parameter	Description
Primary ODBC DSN	The installer creates an ODBC connection that AuthMinder uses to connect to the database. The recommended value to enter is arcotdsn.
User Name	The database user name for AuthMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) Note: The User Name for the Primary and Backup DSNs <i>must</i> be different.

Parameter	Description
Password	The password associated with the User Name you specified in the previous field and which is used by AuthMinder to access the database. This password is specified by the database administrator.
Server Name	The host name or IP address of the AuthMinder datastore. <ul style="list-style-type: none">■ Default Instance Syntax: <server_name> Example: demodatabase■ Named Instance Syntax: <server_name>\<instance_name> Example: demodatabase\instance1
Port Number	The port on which the database server listens to the incoming requests. Note: Press Enter, if you want to accept the default port.
Database	The name of the MS SQL database instance.

The Backup Database Access Configuration options appear.

1. Perform one of the following steps:
 - Type N to skip the configuration of the secondary DSN, when prompted, and press Enter to continue.
 - Type Y to configure the secondary DSN, when prompted, and press Enter to continue.

For information about the parameters, see the table in the preceding step for the database that you are using.

The Encryption Configuration options appear. Use these options to select the encryption mode and set the information used for encryption.

2. Specify the following information:
 - **Master Key:** Enter the master key, which will be used to encrypt the data stored in the database. By default, the value of master key is set to MasterKey. This key is stored in the securestore.enc file, which is located at `<install_location>/arcot/conf`.

If you want to change the value of the master key after the installation, then you must regenerate the securestore.enc file with a new master key value. For more information, see the *CA AuthMinder Administration Guide*.

- Enter y if you want to use a Hardware Security Module (HSM) to encrypt the sensitive data. Enter n if you want to use the software encryption, in this case you do not have to provide the following HSM information.

Note: The following options appear *only* if you choose to use HSM.

- Enter 1 if you want to use Luna HSM or 2 if you want to use nCipher netHSM.
- **HSM PIN:** Enter the password that is used to connect to the HSM.
- **Shared Library:** The absolute path to the PKCS#11 shared library corresponding to the HSM.

Note: You must enter the absolute path to the libcknfast.so file.

- **Storage Slot Number:** The HSM slot where the 3DES keys used for encrypting the data are present. The default value for Luna is 0, and for nCipher netHSM the default value is 1.

Note: The HSM parameter values are recorded in the arcotcommon.ini file located at `<install_location>/arcot/conf`. To change these values after installation, you must edit the arcotcommon.ini file, as discussed in appendix, "[Configuration Files and Options](#)" (see page 181).

3. Press Enter to continue.

The Pre-Installation Summary appears. This summary lists the product name, installation folder, type of installation, and components that are selected.

4. Review the product details displayed carefully and press Enter to proceed with the installation. If you would like to change a configuration option, type back until you reach the option, make the required changes, and press Enter to proceed.

The Installing message appears. This might take several minutes, because the installer now:

- Puts all the components and their related binaries in the installation directory.
- Stores database settings in the [arcotcommon.ini](#) (see page 181) file and the password in the `securestore.enc` file.
- Writes to the required INI files.
- Sets the environment variables such as, `ARCOT_HOME`, `ODBCHOME`, `ODBCINI`, `LD_LIBRARY_PATH`, `LC_MESSAGES`, `NLS_LANG`, and `LC_CTYPE` in the `arwfenv` file.
- Creates or overwrites, as specified earlier, the Primary DSN and Backup DSN (if selected and configured) using the selected ODBC driver in the `odbc.ini` file.

After the preceding tasks are completed successfully, the Installation Complete message appears.

5. Press Enter to exit the installer.

You might have to wait for a few minutes (for the installer to clean up temporary files) until the prompt re-appears.

6. Ensure that UTF-8 support is enabled:
 - a. Navigate to the `<install_location>/arcot/odbc32v70wf/odbc.ini` file.
 - b. Locate the [ODBC] section.
 - c. Ensure that the `IANAAppCodePage=106` entry is present in the section.
 - d. If you do not find this entry, then add it.
 - e. Save and close the file.

Installation Logs

After installation, you can access the installation log file (`Arcot_WebFort_Install_<mm_dd_yyyy_hh_mm_ss>.log`) in the `<install_location>` directory. For example, if you had specified the `/opt` directory as the installation directory, then the installation log file is created in the `/opt` directory.

If the installation fails for some reason, then error messages are recorded in this log file.

Performing Post-Installation Tasks

This section guides you through the post-installation tasks that you must perform after installing AuthMinder. These steps are required for configuring AuthMinder correctly and must be *performed in the following order*:

1. [Running Database Scripts](#) (see page 70)
2. [Verifying the Database Setup](#) (see page 71)
3. [Preparing Your Application Server](#) (see page 72)
4. [Deploying Administration Console](#) (see page 79)
5. [Verifying the Administration Console](#) (see page 81)
6. [Logging In to Administration Console](#) (see page 81)
7. [Bootstrapping the System](#) (see page 82)
8. [Starting AuthMinder Server](#) (see page 84)
9. [Verifying the Installation](#) (see page 85)
10. [Deploying User Data Service](#) (see page 87)
11. [Deploying Sample Application](#) (see page 89)
12. [Using Sample Application](#) (see page 89)

Note: After completing these post-installation tasks, perform the Java SDKs and Web services configuration as discussed in chapter, "[Configuring AuthMinder Java SDKs and Web Services](#)" (see page 137).

Running Database Scripts

AuthMinder ships with SQL scripts that create its schema and set initial configuration values in the AuthMinder database.

To configure the database used by AuthMinder:

1. Locate the folder with the scripts for your database type. The default location is:
 - (for MS SQL) `<install_location>/arcot/dbscripts/mssql`
 - (for Oracle) `<install_location>/arcot/dbscripts/oracle`
 - (for IBM DB2 UDB) `<install_location>/arcot/dbscripts/db2`
 - (for MySQL) `<install_location>/arcot/dbscripts/mysql`
2. Run the *scripts in the following order* by using the database vendor tools:
 - a. `arcot-db-config-for-common-2.0.sql`

Important! If you have installed CA RiskMinder 3.1, you need not run `arcot-db-config-for-common-2.0.sql` because you have already run it while installing CA RiskMinder 3.1.
 - b. `arcot-db-config-for-webfort-7.1.sql`

Note: If you encounter any error while executing the scripts, then check with your database administrator whether you have the required privileges.

Verifying the Database Setup

After running the required database scripts, you must verify whether the AuthMinder schemas were seeded correctly. To do so, you need to use the arwfutil tool.

Perform the following steps to check the database setup:

1. Open a command prompt window.
2. Navigate to the following location:
`<install_location>/arcot/sbin`
3. At the command prompt enter the following command:
`./arwfutil vdb`

The preceding command creates the `arcotwebfort-vdb-<dd>-<mmm>-<yy>.txt` file in the `<install_location>/arcot/logs` directory.

4. Open the `arcotwebfort-vdb-<dd>-<mmm>-<yy>.txt` file in a text editor, and check for entries of the following type:
`ARWF* FOUND`

These lines indicate that your database setup has been successful.

5. Close the `arcotwebfort-vdb-<dd>-<mmm>-<yy>.txt` file.

Preparing Your Application Server

Two components of AuthMinder, User Data Service (UDS) and Administration Console, are Web-based and can be deployed on any of the following supported application servers:

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic
- JBoss

Before you deploy the WAR files for these Web applications on the application server of your choice, you must copy the Arcot-proprietary files and JDBC JAR file required by UDS and Administration Console to the appropriate location on your application server. This section walks you through the steps to copy the required crypto-files to your application server:

- [Step 1: Setting Java Home](#) (see page 72)
- [Step 2: Copying Arcot Files to Your Application Server](#) (see page 73)
- [Step 3: Copying JDBC JARs to Your Application Server](#) (see page 76)
- [Step 4: \(Mandatory for Oracle WebLogic 10.1\) Creating Enterprise Archive File](#) (see page 78)

Step 1: Setting Java Home

Before you deploy the WAR files for UDS and Administration Console on the application server of your choice, ensure that you set the JAVA_HOME environment variable. For Apache Tomcat, set JAVA_HOME to the Java home directory corresponding to the JDK that you are using.

In addition, include \$JAVA_HOME/bin in the PATH environment variable. If you fail to do so, then Administration Console and other JDK-dependent components may fail to start.

Step 2: Copying Database Access Files to Your Application Server

UDS and Administration Console use the following files to access the AuthMinder database securely:

- libArcotAccessKeyProvider.so available at:
`<install_location>/arcot/native/<platform name>/<32bit-or-64bit>/`
- arcot-crypto-util.jar available at:
`<install_location>/arcot/java/lib/`

As a result, these files must be copied to the appropriate location on the application server where you have deployed these AuthMinder components. The following subsections provide information about copying these files for:

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic
- JBoss Application Server

Apache Tomcat

To copy the database access files on Apache Tomcat:

1. Copy the libArcotAccessKeyProvider.so file to the following directory:
 - **For Solaris:** `<JAVA_HOME used by Apache Tomcat>/jre/bin`
 - **For RHEL:** `<JAVA_HOME used by Apache Tomcat>/jre/bin`
2. Copy the arcot-crypto-util.jar file to the following directory:
`<JAVA_HOME used by Apache Tomcat>/jre/lib/ext`
3. Set and export the LD_LIBRARY_PATH to the directory where the libArcotAccessKeyProvider.so file is copied.
4. Restart the application server.

Note: An application server restart is required as part of some of the remaining installation tasks. To minimize the number of times you restart the application server, restart it once after performing the last task that requires a restart.

IBM WebSphere

Note: Depending on the WebSphere version that you are using, the steps that you follow may be different.

To copy the database access files on IBM WebSphere:

1. Log in to IBM WebSphere Administration Console.
2. Click **Environment**, and then click **Shared Libraries**.

- a. From the **Scope** drop-down, select a valid visibility scope. The scope must include the target server/node on which the application is deployed.
 - b. Click **New**.
 - c. Enter the **Name**, for example, **ArcotJNI**.
 - d. Specify the **Classpath**. This path must point to the location where the arcot-crypto-util.jar file is present and must also include the file name. For example, *<install_location>/arcot/java/lib/arcot-crypto-util.jar*.
 - e. Enter the JNI library path. This path must point to the location where the libArcotAccessKeyProvider.so file is present. For example, *<install_location>/arcot/java/native/linux/<32bit-or-64bit>*.
 - f. Click **Apply**.
3. Configure server-level class loaders.
 - a. Click **Servers**, and then click **Application Servers**.
 - b. Under **Application Servers** access the settings page of the server for which the configuration has been performed.
 - c. Click **Java and Process Management**, and then click **Class Loader**.
 - d. Click **New**. Select default **Classes loaded with parent class loader first** and click **OK**.
 - e. Click the auto-generated **Class Loader ID**.
 - f. In the class loader **Configuration** page, click **Shared Library References**.
 - g. Click **Add**, select the shared library that you created earlier in this procedure (for example, ArcotJNI), and then click **Apply**.
 - h. Save the changes.
 4. Copy libArcotAccessKeyProvider.so file to the following directory:
 - **For Solaris:** *<JAVA_HOME used by IBM WebSphere>/jre/bin*
 - **For RHEL:** *<JAVA_HOME used by IBM WebSphere>/jre/bin*
 5. Restart the application server.

Note: An application server restart is required as part of some of the remaining installation tasks. To minimize the number of times you restart the application server, restart it once after performing the last task that requires a restart.

Oracle WebLogic

To copy the database access files on Oracle WebLogic:

1. Copy libArcotAccessKeyProvider.so to the following directory:
 - **For Solaris:**
 - For Sun JDK: *<JAVA_HOME used by Oracle WebLogic instance>/jre/bin*

- For all other versions of JDK: *<JAVA_HOME used by Oracle WebLogic instance>/jre/bin*
 - **For RHEL:** *<JAVA_HOME used by Oracle WebLogic instance>/jre/bin*
2. Set and export the LD_LIBRARY_PATH to the directory where the libArcotAccessKeyProvider.so file is copied.
 3. Copy arcot-crypto-util.jar to the *<JAVA_HOME used by Oracle WebLogic instance>/jre/lib/ext* directory.
 4. Log in to WebLogic Administration Console.
 5. Navigate to **Deployments**.
 6. Enable the **Lock and Edit** option.
 7. Click **Install** and navigate to the directory that contains the arcot-crypto-util.jar file.
 8. Click **Next**.
The Application Installation Assistant screen appears.
 9. Click **Next**.
The Summary page appears.
 10. Click **Finish**.
 11. Activate the changes.
 12. Restart the application server.

Note: An application server restart is required as part of some of the remaining installation tasks. To minimize the number of times you restart the application server, restart it once after performing the last task that requires a restart.

JBoss Application Server

To copy the database access files on JBoss Application Server:

1. Copy libArcotAccessKeyProvider.so to the following directory:
 - **For Solaris:** *<JAVA_HOME used by JBoss Application Server instance>/jre/bin*
 - **For RHEL:** *<JAVA_HOME used by JBoss Application Server instance>/jre/bin*
2. Copy arcot-crypto-util.jar to the *<JAVA_HOME used by JBoss Application Server instance>/jre/lib/ext* directory.
3. Restart the application server.

Note: An application server restart is required as part of some of the remaining installation tasks. To minimize the number of times you restart the application server, restart it once after performing the last task that requires a restart.

Step 3: Copying JDBC JARs to Your Application Server

Administration Console, User Data Service (UDS), and Sample Application, which are the Java-dependant components of AuthMinder need Java Database Connectivity (JDBC) Java ARchive (JAR) files to connect to the database. You must copy these files to the application server.

Note: Before proceeding with the steps mentioned in the following subsections, ensure that you have downloaded the JDBC JARs, see chapter, "[Preparing for Installation](#)" (see page 33) for more information on supported JDBC JARs.

The following sub-sections walk you through the steps for copying the JDBC JAR required for your database to one of the following application servers:

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic
- JBoss Application Server

Apache Tomcat

To copy the JDBC JAR file to the Apache Tomcat installation directory:

1. Navigate to the location where you have downloaded the JDBC JAR file.
2. Copy the JDBC JAR file and paste it in the following directory:
 - **For Apache Tomcat 5.5.x:** <TOMCAT-HOME>\common\lib
 - **For Apache Tomcat 6.x and 7.x:** <TOMCAT-HOME>\lib

Alternatively, add the path that contains the JDBC JAR file to the **Classpath** environment variable.

3. Restart Apache Tomcat.

IBM WebSphere

To copy the JDBC JAR file to IBM WebSphere:

1. Log in to IBM WebSphere Administration Console.
2. Click **Environment**, and then click **Shared Libraries**.
 - a. From the **Scope** drop-down, select a valid visibility scope. The scope must include the target server/node on which the application is deployed.
 - b. Click **New**.
 - c. Enter the **Name**, for example, **JDBCJAR**.
 - d. Specify the **Classpath**. This path must point to the location where the JDBC JAR file is present and must also include the file name.

- e. Click **Apply** to save the changes made.
3. Configure server-level class loaders.
Note: You can either create a class loader or use the one that you created while performing the procedure described in [Step 2: Copying Database Access Files to Your Application Server](#) (see page 73).
 - a. Click **Servers**, and then click **Application Servers**.
 - b. Under **Application Servers** access the settings page of the server for which the configuration are performed.
 - c. Click **Java and Process Management**, and then click **Class Loader**.
 - d. Click **New**. Select default **Classes loaded with parent class loader first** and click **OK**.
 - e. Click the auto-generated **Class Loader ID**.
 - f. In the class loader **Configuration** page, click **Shared Library References**.
 - g. Click **Add**, select **JDBCJAR** and then click **Apply**.
 - h. Save the changes made.
4. Restart IBM WebSphere.

Oracle WebLogic

To copy the JDBC JAR file to Oracle WebLogic:

Note: If you are using Oracle Database, then you need *not* perform the configurations mentioned in this section, because WebLogic supports Oracle database by default.

1. Copy the JDBC JAR file to the following directory:
`<JAVA_HOME used by Oracle WebLogic instance>/jre/lib/ext`
2. Login to WebLogic Administration Console.
3. Navigate to **Deployments**.
4. Enable the **Lock and Edit** option.
5. Click **Install** and navigate to the directory that contains the JDBC JAR file.
6. Click **Next**.
The Application Installation Assistant screen appears.
7. Click **Next**.
The Summary page appears.
8. Click **Finish**.
9. Activate the changes.
10. Restart the Oracle WebLogic server.

JBoss Application Server

To copy the JDBC JAR file on JBoss Application Server:

1. Copy the JDBC JAR file to the following location on the JBOSS installation directory:
<JAVA_HOME Used by JBOSS>/server/default/lib
2. Restart the application server.

Step 4: (Mandatory for Oracle WebLogic 10.1) Creating Enterprise Archive File

Most application servers (such as WebSphere and WebLogic) enable you to bundle related JAR or WAR files from one vendor into a single enterprise application (or archive). As a result, all the related JARs or WARs can be deployed together, and can be loaded by a class loader. This archive also contains an application.xml file, which is generated automatically and describes how to deploy each bundled module.

By default, AuthMinder provides WAR files to deploy Administration Console and User Data Service. If required, you can also change the format of these files to EAR and then deploy the EAR files.

Follow these steps:

1. Open the command prompt window.
2. Navigate to the <install_location>/arcot/tools/common/bundlemanager directory.
3. Run the bundlemanager tool to create the EAR file by using the following command:

```
java -jar bundle-manager.jar -ear <filename.ear> -warList  
<war_file_name>
```

Note: In the preceding command, replace the <war_file_name> with arcotadmin.war for generating EAR file for Administration Console, and arcotuds.war for generating EAR file for User Data Service.

The above command generates the EAR file and it is available at <install_location>/arcot/java/webapps.

Deploying Administration Console

Note: If you plan to deploy Administration Console on IBM WebSphere 7.0, then refer to the instructions provided in appendix, "[Deploying Administration Console on IBM WebSphere 7.0](#)" (see page 247).

You need the file `arcotadmin.war` to deploy the AuthMinder Administration Console. This file is available at:

```
<install_location>/arcot/java/webapps/
```

Note: To manage AuthMinder Server by using Administration Console, you must ensure that Administration Console can access the system where AuthMinder Server is installed by its hostname.

To deploy Administration Console:

1. Change the working directory to:
`<install_location>/arcot/sbin`
2. Type `source arwfenv` and press **Enter** to set the `$ARCOT_HOME` environment variable.
3. Restart the application server for the changes to take effect.
4. Deploy `arcotadmin.war` in the appropriate directory on the application server.

Note: The deployment procedure depends on the application server that you are using. Refer to your application server vendor documentation for detailed instructions.

For example, in case of Apache Tomcat, you must deploy the WAR file at `<APP_SERVER_HOME>/webapps/`.

5. **(For JBoss Only)** Perform the following steps if you have deployed Administration Console on JBoss application server:
 - a. Copy the Bouncy Castle JAR file (`bcprov-jdk15-146.jar`) from `<install_location>/arcot/java/lib/` to the following location:
`<JBOSS_HOME>/common/lib`
 - b. Navigate to the following location:
`<JBOSS_HOME>/server/default/conf/`
 - c. Open `jboss-log4j.xml` file in a text editor.
 - d. Add the following log configuration in the `<log4j:configuration>` section:


```
<appender name="arcotadminlog"
class="org.apache.log4j.RollingFileAppender">
  <errorHandler
class="org.jboss.logging.util.OnlyOnceErrorHandler"></errorHandler>
  <param name="Threshold" value="INFO"/>
  <param name="MaxFileSize" value="10MB"/>
  <param name="MaxBackupIndex" value="100"/>
  <param name="Encoding" value="UTF-8"/>
  <param name="Append" value="true"/>
```

```
<param name="File" value="\${arcot.home}/logs/arcotadmin.log"/>
<layout class="org.apache.log4j.PatternLayout">
<param name="ConversionPattern" value="%d{yyyy-MM-dd hh:mm:ss,SSS z} : [%t]
: %-5p : %-5c{3} : %m%n"/>
</layout>
<filter class="org.jboss.logging.filter.TCLMCFilter">
<param name="AcceptOnMatch" value="true"/>
<param name="DeployURL" value="arcotadmin.war"/>
</filter>
<!-- end the filter chain here -->
<filter class="org.apache.log4j.varia.DenyAllFilter"></filter>
</appender>
```

- e. Add the following log category:

```
<category name="com.arcot">
<priority value="INFO" />
<appender-ref ref="arcotadminlog"></appender-ref>
</category>
```

- f. Add the following category for cryptographic operations:

```
<category name="com.arcot.crypto.impl.NCipherCrypter">
<priority value="FATAL" />
<appender-ref ref="arcotadminlog"></appender-ref>
</category>
```

- g. Save and close the file.

- h. Take the backup of existing JBoss logging libraries. These library files are available at:

```
<JBOSS_HOME>/lib
```

- i. Upgrade the JBoss logging libraries available at <JBOSS_HOME>/lib to version 2.1.1. The following table lists the JAR file names and the location from where you can download the files.

File Name	Location
jboss-logging-jdk-2.1.1.GA.jar	http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-jdk/2.1.1.GA/
jboss-logging-spi-2.1.1.GA.jar	http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-spi/2.1.1.GA/
jboss-logging-log4j-2.1.1.GA.jar	http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-log4j/2.1.1.GA/

- 6. Ensure that the application is restarted.

Verifying the Administration Console

The arcotadmin.log file is used for logging the Administration Console information.

To verify if the Administration Console was deployed successfully:

1. Navigate to the following location:
`<install_location>/arcot/logs`
2. Open the arcotadmin.log file in any editor and locate the following lines:
Arcot Administration Console Configured Successfully.

This line indicates that your Administration Console was deployed successfully.

Note: Check and resolve if there are any FATAL and ERROR messages, and review all the WARNING messages for unexpected conditions.

Important! If you want to localize your Administration Console, then you must prepare your Administration Console for the same, as discussed in appendix, "[Preparing for Localization](#)" (see page 203).

After you are done with the steps for localization, then resume performing the post-installation tasks that follow.

Logging In to Administration Console

When logging in to Administration Console for the first time, you *must* use the Master Administrator credentials that are created automatically in the database during installation.

To log in to Administration Console:

1. Start Administration Console in a Web browser window, by using the following URL:

`http://<host>:<app_server_port>/arcotadmin/masteradminlogin.htm`

Note: The host and port information that you specify in the preceding URL must be of the application server where Administration Console is deployed.

2. Log in to Administration Console as a Master Administrator with the default Master Administrator account credentials. The credentials are:
 - **User Name:** masteradmin
 - **Password:** master1234!

Bootstrapping the System

Before you can start using the Administration Console to manage AuthMinder, you must first perform the following mandatory steps to initialize the system:

- Change the default Master Administrator password
- Specify the Global Key label
- Specify the authentication mechanism for the Default organization

Bootstrapping is a wizard-driven process that walks you through these setup tasks. Other administrative links are enabled after you perform these tasks.

Before you proceed with [Performing the Bootstrapping Tasks](#) (see page 83), you must understand the related concept of Default Organization.

Default Organization

When you deploy the Administration Console, an organization is created by default. This organization is referred to as *Default Organization* (**DEFAULTORG**). As a single-organization system, the Default Organization itself can be used without creating any new organizations.

Performing the Bootstrapping Tasks

When you first log in to the Administration Console as the Master Administrator (MA), the Summary screen for the Bootstrap wizard screen appears.

To bootstrap the system using the wizard:

1. Click **Begin** to start the process.

The Change Password screen appears.

2. Specify the **Old Password**, **New Password**, **Confirm Password**, and click **Next**.

The Configure Global Key Label screen appears.

3. Specify the **Global Key Label**, and click **Next**.

AuthMinder enables you to use hardware- or software-based encryption for your sensitive data. Irrespective of hardware or software encryption, all Arcot products use *Global Key Label* for encrypting user and organization data.

If you are using hardware encryption, then this label serves only as a reference (or pointer) to the actual 3DES key stored in the HSM device, and therefore the key label *must* match the HSM key label. However, in the case of software-based encryption, this label acts as the reference to actual software key in database.

Important! After you complete the bootstrapping process, you can *not* update this key label.

The **Storage Type** field indicates whether the encryption key is stored in the database (**Software**) or the HSM (**Hardware**).

The Configure Default Organization screen appears.

4. Under the **Default Organization Configuration** section, specify the following parameters for the Default Organization:
 - **Display Name:** The descriptive name of the organization. This name appears on all other Administration Console pages and reports.
 - **Administrator Authentication Mechanism:** The mechanism that is used to authenticate administrators belonging to the Default Organization. Administration Console supports three types of authentication methods for the administrators to log in:
 - **Basic**

If you choose this option, then the inbuilt authentication method provided by the Administration Console is used for authenticating the administrators.
 - **LDAP User Password**

If you choose this option, then the administrators are authenticated using their credentials that are stored in the directory service.

Note: If this mechanism is used for authenticating administrators, then you must deploy UDS, as discussed in the "[Deploying User Data Service](#)" (see page 87) section.

■ **WebFort User Password**

If you select the **WebFort User Password** option here, then the credentials are issued and authenticated by the AuthMinder Server.

Note: Refer to *CA AuthMinder Administration Guide* for more information on how to do this.

5. Under the **Key Label Configuration** section of the Configure Default Organization screen, specify the following:
 - **Use Global Key:** This option is selected by default. Deselect this option if you want to override the Global Key Label you specified in the preceding step and specify a new encryption label.
 - **Key Label:** If you deselected the **Use Global Key** option, then specify the new key label that you want to use for the Default Organization.
 - **Storage Type:** This field indicates whether the encryption key is stored in the database (**Software**) or the HSM (**Hardware**).
6. Click **Finish** to complete the bootstrapping process.

The Administration Console initialization is completed, as indicated in the Finish screen.
7. Click **Continue** to proceed with other configurations using the Administration Console.

Note: If you want to localize your Administration Console, then you must prepare your Administration Console for the same, as discussed in appendix, "[Preparing for Localization](#)" (see page 203) for more information. After you are done with the steps for localization, then resume performing the post-installation tasks that follow.

Starting AuthMinder Server

Perform the following steps to start AuthMinder Server:

1. Navigate to the following directory:
`<install_location>/arcot/bin/`
2. Run the following command:
`./webfortserver start`

Note: If you want to stop the server, then execute `./webfortserver stop` command.

Verifying the Installation

You can verify whether the AuthMinder Server and the Web applications have started successfully by:

- [Using Log files](#) (see page 85)
- [Using webfortserver Utility](#) (see page 85)
- [Checking the Ports](#) (see page 86)

Using Log files

Perform the following steps to verify if AuthMinder Server started correctly:

1. Navigate to the following location:
`<install_location>/arcot/logs`
2. Open the `arcotwebfortstartup.log` file in any editor and locate the following lines:
INSTANCE_VER.....: [7.1]
Arcot WebFort Authentication Service READY

These lines indicate that AuthMinder Server is installed successfully.

Note: Ensure that the log files do not contain any FATAL and WARNING messages.

Using webfortserver Utility

The `webfortserver` tool enables you to check the release of AuthMinder you have installed. See the *CA AuthMinder Administration Guide* for more information about this tool.

Follow these steps:

1. Navigate to the following location:
`<install_location>/arcot/bin`
2. Run the following command to start the tool in interactive mode:
`./webfortserver -i`
3. Enter version at the prompt.
The `webfort-ver-<dd>-<mmm>-<yy>.txt` file is created in the `<install_location>/arcot/logs` folder.
4. Open this file and check for the following lines to ensure that you are using the current version:
 - The version of AuthMinder library files in the bin section is 7.1.
 - The version of the UDS library file (`arwfuds.dll`) in the bin section is 2.0.3.
5. Close the file.

Checking the Ports

Perform the following steps to verify if the AuthMinder Server is listening to different protocols on the default ports:

1. Navigate to the following location:
`<install_location>/arcot/logs`
2. Open the `arcotwebfortstartup.log` file in any editor and search for the protocol names to verify if they are listening on the correct port, as shown in the following snippet.

```
PROTOCOLNAME : [Administration-WS]
PORTNO : 9745
PROTOCOLID : [Transaction-Native]
PORTNO : 9742
PROTOCOLID : [ServerManagement-WS]
PORTNO : 9743
PROTOCOLID : [Transaction-WS]
PORTNO : 9744
```

Note: Refer to appendix, ["Default Port Numbers and URLs"](#) (see page 231) for information on default ports and protocols.

Deploying User Data Service

AuthMinder can access user data either from a relational database (RDBMS) or directly from an LDAP server:

You need the file `arcotuds.war` to deploy the User Data Service (UDS). This file is available at:

`<install_location>/arcot/java/webapps/`

To deploy User Data Service:

1. Install `arcotuds.war` in the appropriate directory on the application server.

Note: The deployment procedure depends on the application server that you are using. Refer to your application server vendor documentation for detailed instructions.

For example, in case of Apache Tomcat, you must deploy the WAR file at `<APP_SERVER_HOME>/webapps/`.

2. **(For WebSphere Only)** Configure to reload the UDS class when the application files are updated.
 - a. Navigate to **Application > Enterprise Applications** and access the UDS settings page.
 - b. Under **Class loader order**, select the **Classes loaded with local class loader first (parent last)** option.
 - c. Under **WAR class loader policy**, select the **Single class loader for application**.
 - d. Click **Apply**.
3. **(For JBoss Only)** Perform the following steps if you have deployed UDS on JBoss application server:
 - a. Copy the Bouncy Castle JAR file (`bcprov-jdk15-146.jar`) from `<install_location>/arcot/java/lib/` to the following location: `<JBoss_HOME>/common/lib`
 - b. Navigate to the following location: `<JBoss_HOME>/server/default/conf/`
 - c. Open `jboss-log4j.xml` file in a text editor.
 - d. Add the following log configuration in the `<log4j:configuration>` section:


```
<appender name="arcotudslog" class="org.apache.log4j.RollingFileAppender">
  <errorHandler
    class="org.jboss.logging.util.OnlyOnceErrorHandler"></errorHandler>
  <param name="Threshold" value="INFO"/>
  <param name="MaxFileSize" value="10MB"/>
  <param name="MaxBackupIndex" value="100"/>
  <param name="Encoding" value="UTF-8"/>
  <param name="Append" value="true"/>
  <param name="File" value="${arcot.home}/logs/arcotuds.log"/>
</layout class="org.apache.log4j.PatternLayout">
```

```
<param name="ConversionPattern" value="%d{yyyy-MM-dd hh:mm:ss,SSS z} : [%t]
: %-5p : %-5c{3}(%L) : %m%n"/>
</layout>
<filter class="org.jboss.logging.filter.TCLMCFilter">
<param name="AcceptOnMatch" value="true"/>
<param name="DeployURL" value="arcotuds.war"/>
</filter>
<!-- end the filter chain here -->
<filter class="org.apache.log4j.varia.DenyAllFilter"></filter>
</appender>
```

- e. Add the following line in the com.arcot category that you created in [Deploying Administration Console](#) (see page 79):
<appender-ref ref="arcotudslog"></appender-ref>
 - f. Add the following line in the for cryptographic category that you created in [Deploying Administration Console](#) (see page 79):
<appender-ref ref="arcotudslog"></appender-ref>
 - g. Save and close the file.
4. Restart the application server.
 5. Perform the following steps to verify if UDS started correctly:
 - a. Navigate to the following location:
<install_location>/arcot/logs
 - b. Open the arcotuds.log file in any editor and locate the following lines:
User Data Service (Version: 2.0.3) initialized successfully.
This line indicates that UDS was deployed successfully

Note: Check and resolve if there are any FATAL and ERROR messages, and review all the WARNING messages for unexpected conditions.

Deploying Sample Application

Important! Sample Application must *not* be used in production deployments. It is recommended that you build your own web application by using Sample Application as a code-reference.

Sample Application can be used to verify if AuthMinder was installed and configured properly. In addition, it demonstrates:

- The typical AuthMinder workflows
- The tasks that you can perform using the AuthMinder Java APIs
- Integration of your application with AuthMinder

Sample Application is automatically installed as a part of Complete installation of AuthMinder. To deploy the Sample Application:

1. Deploy the `webfort-7.1-sample-application.war` file from the following location:
`<install_location>/arcot/samples/java`
2. Start Sample Application.
3. Access the Sample Application using the following URL:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`

Using Sample Application

Sample Application enables you to issue and authenticate the credentials that AuthMinder Supports. You can use the Sample Application to perform these operations to test whether the AuthMinder installation has been successful.

This section covers the following tasks:

- [Creating Users](#) (see page 90)
- [Setting Up ArcotID PKI Client](#) (see page 91)
- [Creating ArcotID PKI Credential](#) (see page 92)
- [Downloading ArcotID PKI](#) (see page 93)
- [Authenticating Using ArcotID PKI](#) (see page 94)

Creating Users

Note: User creation must be performed either using Administration Console or Web services.

To create users using Administration Console:

1. Log in to the Administration Console as a Global Administrator (GA) or an Organization Administrator (OA). The URL for the purpose is:
http://<host>:<app_server_port>/arcotadmin/adminlogin.htm
2. If already not activated, activate the **Manage Users and Administrators** sub-tab under the **Users and Administrators** tab.
3. In the left-hand pane, under **Manage Users and Administrators**, click **Create User** to open the Create User page.
4. On the Create User page:
 - a. Enter a unique user name, their organization name, and optionally, other user information in the **User Details** section
 - b. If required, specify other user information in the corresponding fields on the page.
 - c. Select the required **User Status**.
 - d. Click **Create User**.

The "Successfully created the user" message appears if the specified user was successfully added to the database.

5. Return to the AuthMinder Sample Application page.

Setting Up ArcotID PKI Client

To set up the ArcotID PKI Client to communicate with the AuthMinder Server for authenticating users with their ArcotID PKI, you need to perform the following steps:

1. Access the Sample Application using the following URL:
http://<host>:<app_server_port>/webfort-7.1-sample-application/
2. In the left-hand pane, click **Setup** -> **ArcotID Client** to open ArcotID Client Settings page.
3. In the **Choose ArcotID Client** section, select the type of client that has to be used to authenticate the ArcotID PKI.
4. In the **Choose ArcotID Download Type** section, select the location where you want to store the ArcotID PKI.
5. In the **Choose Where & When to Obtain the ArcotID Challenge** section, select the mode of obtaining the ArcotID PKI challenge.
6. Click **Select** to save the settings.

The "The operation was successful" message appears if the ArcotID PKI Client configuration was performed successfully.

Creating ArcotID PKI Credential

To create ArcotID PKI credential for users:

1. Access the Sample Application using the following URL:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`
2. In the left-hand pane, click **ArcotID** -> **Issuance** -> **Create** to open Create ArcotID page.
3. Specify the name of the user you created in the **User Name** field.
4. If required, specify the user's organization in the **Organization** field.
5. Specify the password to be used for authentication in the **ArcotID Password** field.
6. If required, specify the profile that has to be used to issue ArcotID in the **Profile Name** field.
7. If required, specify the name-value pairs of the **Unsigned Attributes**. The attributes are set in the unsigned portion of the ArcotID PKI.
8. If required, specify the **Custom Attributes** to be used for creating the ArcotID PKI.
9. If required, specify the **Additional Input** that you want to pass to the AuthMinder Server.
10. If required, pass the following **Transaction Logging Parameters**:
 - In the **Log Level** field, choose the logging level.
Note: For more information on the log levels, see the topic titled "AuthMinder Logging" in *CA AuthMinder Administration Guide*.
 - Select **Enable Trace Logging** if you want to capture flow details.
 - Select **Enable DB Logging** if you want to log the database activities.
 - Select **Enable Sensitive Data Logging** if you want to log the sensitive data.
11. Click **Create** to create the credential.

The "The operation was successful" message appears if the ArcotID PKI was created successfully for the user.

Downloading ArcotID PKI

To download the ArcotID PKI:

1. Access the Sample Application using the following URL:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`
2. In the left-hand pane, click **ArcotID** -> **Issuance** -> **Download** to open the Download ArcotID page.
3. Specify the name of the user you created in the **User Name** field.
4. If required, specify the user's organization in the **Organization** field.
5. If required, specify the profile that has been used to issue ArcotID PKI in the **Profile Name** field.
6. If required, specify the **Additional Input** that you want to pass to the AuthMinder Server.
7. If required, pass the following **Transaction Logging Parameters**:
 - In the **Log Level** field, choose the logging level.
Note: See the topic titled "AuthMinder Logging" in *CA AuthMinder Administration Guide* for more information on the log levels.
 - Select **Enable Trace Logging** if you want to capture flow details.
 - Select **Enable DB Logging** if you want to log the database activities.
 - Select **Enable Sensitive Data Logging** if you want to log the sensitive data.
8. Click **Download** to download the user's ArcotID PKI.

Authenticating Using ArcotID PKI

To authenticate using the ArcotID PKI:

1. Access the Sample Application using the following URL:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`
2. In the left-hand pane, click **ArcotID** -> **Authentication** -> **Authenticate** to open the ArcotID Authentication page.
3. Specify the name of the user you created in the **User Name** field.
4. Specify the user's organization in the **Organization** field.
5. Specify the user's ArcotID PKI password in the **ArcotID Password** field.
6. If you are using aliases to identify the users, then specify the **Application Context** based on the alias of the user that you want to authenticate.
7. If required, select the **Token Type** that has to be returned to the user after successful authentication.
Note: Refer to the chapter, "Authenticating Users" in the *CA AuthMinder Developer's Guide* for more information on the token types.
8. If required, specify the **Authentication Policy Name** that has to be used for authenticating users.
9. If you have selected SAML as the token type, then specify the **SAML Policy Name** to be used.
10. If required, specify the **Additional Input** that you want to pass to the AuthMinder Server.
11. If required, pass the following **Transaction Logging Parameters**:
 - In the **Log Level** field, choose the logging level.
Note: See the topic titled, "AuthMinder Logging" in *CA AuthMinder Administration Guide* for more information on the log levels.
 - Select **Enable Trace Logging** if you want to capture flow details.
 - Select **Enable DB Logging** if you want to log the database activities.
 - Select **Enable Sensitive Data Logging** if you want to log the sensitive data.
12. Click **Authenticate** to verify the user's ArcotID PKI.

Post-Installation Checklist

It is recommended that you fill this checklist with the installation and setup information for AuthMinder. You will need this information for various administrative tasks that you will perform later.

Your Information	Example Entry	Your Entry
ARCOT_HOME	/var/opt/arcot/	
SYSTEM INFORMATION		
Host Name	my-bank	
User Name	administrator	
Password	password1234!	
Configured Components	WebFort Server Administration Console User Data Service	
ADMINISTRATION CONSOLE INFORMATION		
Host Name	localhost	
Port	8080	
Master Administrator Password	mypassword1234!	
USER DATA SERVICE INFORMATION		
Host Name	localhost	
Port	8080	
Application Context Root	arcotuds	

Chapter 5: Deploying AuthMinder on Distributed Systems

Use the **Arcot WebFort 7.1 InstallAnywhere Wizard** to install AuthMinder components. This Wizard supports *Complete* and *Custom* installation types. However, to install and configure AuthMinder in a distributed environment, you must use the **Custom** option when you run the installer.

The following steps provide a quick overview of the process:

1. Run the AuthMinder installer to install AuthMinder Server and Administration Console and to configure them to access your SQL database. You can also choose to install the Web Services on the same system.

See ["Installing on the First System"](#) (see page 99) for installation instructions.

2. Execute the database scripts to create AuthMinder schema and database tables. Also ensure that the database setup was successful.

See ["Running Database Scripts"](#) (see page 109) and ["Verifying the Database Setup"](#) (see page 71) for more information.

3. Prepare the application server to copy the files that Web components use.

See ["Preparing Your Application Server"](#) (see page 110) for more information.

4. Deploy Administration Console on the application server and verify the deployments.

See ["Deploying Administration Console"](#) (see page 117) and ["Verifying the Administration Console"](#) (see page 81) for more information.

5. (Optional) Prepare your Administration Console for localization, if you want to use languages other than English.

See appendix, ["Preparing for Localization"](#) (see page 203) for more information.

6. Log in to Administration Console with the Master Administrator credentials to initialize AuthMinder.

See "[Logging In to Administration Console](#)" (see page 81) and "[Bootstrapping the System](#)" (see page 82) for more information.

7. Start the AuthMinder Server and verify if the service start successfully.

See "[Starting AuthMinder Server](#)" (see page 84) and "[Verifying the Installation](#)" (see page 85) for more information.

8. Deploy User Data Service on the application server and verify the deployment.

See "[Deploying User Data Service](#)" (see page 125) for more information.

9. Install the Java SDKs and Web Services on one or more systems.

See "[Installing on the Second System](#)" (see page 70) for more information.

10. Deploy, configure, and use Sample Application to test AuthMinder configuration.

Note: To install the Sample Application *only*, you must ensure that you select only the **SDKs and Sample Application** option and proceed with the installation.

See "[Deploying Sample Application](#)" (see page 128), "[Configuring Sample Application for Communication with AuthMinder Server](#)" (see page 128), and "[Using Sample Application](#)" (see page 129) for more information.

11. (Optional) To ensure secure communication between AuthMinder components, you can configure them to support SSL (Secure Socket Layer) transport mode.

See the topic titled "Configuring SSL" in *CA AuthMinder Administration Guide* for more information.

12. Complete the installation checklist.

See "[Post-Installation Checklist](#)" (see page 135) for more information.

Important Notes Related to Installation

You must keep the following points in mind while installing AuthMinder either on a single system or in a distributed environment:

- You must ensure that the `<install_location>` *must not contain* any special characters (such as `~ ! @ # $ % ^ & * () _ + = { } [] ' "`).
- Currently, you cannot modify or repair AuthMinder components by using the installer. You *must* uninstall the component and then re-install it.
- Do not close the installer window, if the installation is in progress. If at any time during installation (*especially during the last stages*), if you click the **Cancel** button to abort the installation, then the installer might not remove *all* the directories it has created so far. You will manually need to clean up the installation directory, `<install_location>/arcot/` and its subdirectories.

- If you run the installer on a system that already contains an instance of an existing \$ARCOT_HOME, then:
 - You will *not* be prompted for an installation directory.
 - You will *not* be prompted for the database set up. The Installer will use the existing database.
 - You will also *not* be prompted to set up encryption.

Installing on the First System

In a distributed scenario, you can distribute AuthMinder, Administration Console, Java SDKs, and Web Services on different systems, typically you install AuthMinder Server on the first system. *Custom installation* allows you to install only the selected components from the package. This option is recommended for advanced users.

Note: Before proceeding with the installation, ensure that all the prerequisite software are installed and the database is set up, as described in chapter, "[Preparing for Installation](#)" (see page 33).

Perform the following steps to install AuthMinder and related components:

1. Log in and navigate to the directory where you untarred the installer.
2. Ensure that you have the permission to run the installer. If not, you must run the following command:
`chmod a=rx Arcot-WebFort-7.1-<platform_name>-Installer.bin`
3. Run the installer as follows:
`prompt> sh Arcot-WebFort-7.1-<platform_name>-Installer.bin`

If you are running the installer with root login, then a warning message appears. Enter **Y** to continue, or **N** to quit the installation. If you exit the installer, then you must run the installer again.

The Welcome message appears.

4. Press **Enter** to continue with the installation.
The License Agreement for AuthMinder appears.

5. On the License Agreement page:
 - a. Read the text carefully and press **Enter** to display the next page of the license text. You might have to press **Enter** multiple times, until the entire text for License Agreement is displayed.

At the end of the license agreement, you will be prompted for acceptance of the terms of license agreement (**DO YOU ACCEPT THE TERMS OF LICENSE AGREEMENT?**).

- b. Enter **y** to accept the acceptance of License Agreement and to continue with the installation.

Note: If you press **n**, then a warning message will be displayed and the installation will be aborted.

The Choose Installation Location options appear.

6. Perform one of the following steps:
 - If the installer detects an existing Arcot home directory, then it displays the path to that directory. Press **Enter** to accept this directory path.
 - If the installer does not detect an existing Arcot home directory, it displays the default directory path and prompts you to either accept the default path or specify a new path. Press **Enter** if you want to accept the default path. Alternatively, enter the absolute path of the directory where you want to install AuthMinder and press **Enter** to continue.

Note: The installation directory name that you specify must not contain spaces. Otherwise, some AuthMinder scripts and tools might not function as intended.

The installation types (Complete and Custom) supported by AuthMinder appear.

7. Select the required option and press **Enter** to continue with the installation.

Note: If you have selected option **1**, then a new folder **arcot** will be created in the specified location.

The installation types (**Complete** and **Custom**) supported by AuthMinder appear.

8. Type **2** and press **Enter** to accept the **Custom** installation option and to continue with the installation.

The Choose Product Features options appear. Use these options to select the specific components that you want to install on the system.

9. Specify a comma-separated list (*without any space between the comma and the number*) of numbers representing the AuthMinder components you would like to install.

On the first system, you will install the following components:

- a. **Arcot WebFort Authentication Server**
- b. **Arcot Administration Console**
- c. **Arcot User Data Service**

The following table describes all components that are installed by the AuthMinder installer and the numbers that you must enter to install them.

Option	Component	Description
1	Arcot WebFort Authentication Server	<p>This option installs the core Processing engine (WebFort Server) that serves the following requests from SDKs, Administration Console, and web services:</p> <ul style="list-style-type: none"> ■ Credential Issuance Configurations ■ Credential Authentication Configurations ■ Server Configurations <p>In addition, this component also enables you to access the following Web services:</p> <ul style="list-style-type: none"> ■ Authentication and Authorization Web service - Provides the programming interface for authenticating and authorizing users. ■ Issuance SDK and Web Services - Provides the programming interface for creating, reading, and updating user credential information in the AuthMinder database. ■ Authentication Web Service - Provides the programming interface for authenticating users. ■ Credential Management Web Service - Provides the programming interface for creation and management of user credentials. ■ Administration Web Service - Provides the programming interface used by the AuthMinder Administration Console. ■ Bulk Operations Web Service: Provides the programming interface for uploading and fetching OATH tokens.

Option	Component	Description
2	Arcot WebFort Java SDK and WS	<p>This option provides programming interfaces (in form of APIs and Web Services) that can be invoked by your application to forward authentication and user credential issuance requests to the AuthMinder Server. This package comprises the following sub-components:</p> <ul style="list-style-type: none"> ■ Authentication Java SDK and Web Services- Provides the programming interface for authentication with AuthMinder Server. ■ Credential Management Java SDK and Web Services - Provides the programming interface for creation and management of user credentials. ■ Administration Web Service - Provides the programming interface for creating configurations. ■ Bulk Operations Web Service: Provides the programming interface for uploading and fetching OATH tokens. <p>Refer to chapter, "Configuring AuthMinder Java SDKs and Web Services" (see page 137) for more information on configuring these components.</p>
3	Arcot WebFort Sample Application	This option provides Web-based interface for demonstrating the usage of AuthMinder Java APIs. In addition, it can also be used to verify if AuthMinder was installed successfully, and if it is able to perform credential management and authentication requests.
4	Arcot Administration Console	This option provides the Web-based interface for managing AuthMinder Server and authentication-related configurations.
5	Arcot User Data Service	This option installs UDS that acts as an abstraction layer for accessing different types of user repositories, such as relational databases (RDBMSs) and directory servers (LDAPs.)

1. After you have entered the numbers corresponding to the desired components for installation, press **Enter** to continue.

The database types supported by AuthMinder appear.

2. Specify the number corresponding to the database, and press **Enter** to continue:
 - **1** - Microsoft SQL Server
 - **2** - IBM DB2 (UDB)
 - **3** - Oracle Database
 - **4** - MySQL

The Primary Database Access Configuration options appear.

3. Depending on the database that you are using:
 - Specify the information listed in the following table if you specified **1** (Microsoft SQL Server) in the preceding step.

Parameter	Description
Primary ODBC DSN	The installer creates an ODBC connection that AuthMinder uses to connect to the database. The recommended value to enter is arcotdsn.
User Name	The database user name for AuthMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) Note: The User Name for the Primary and Backup DSNs <i>must</i> be different.
Password	The password associated with the User Name you specified in the previous field and which is used by AuthMinder to access the database. This password is specified by the database administrator.
Server Name	The host name or IP address of the AuthMinder datastore. <ul style="list-style-type: none"> ■ Default Instance Syntax: <server_name> Example: demodatabase ■ Named Instance Syntax: <server_name>\<instance_name> Example: demodatabase\instance1
Port Number	The port on which the database server listens to the incoming requests. Note: Press Enter , if you want to accept the default port.

Parameter	Description
Database	The name of the MS SQL database instance.

- Specify the information listed in the following table if you specified **2** (IBM DB2 (UDB)) in the preceding step:

Parameter	Description
Primary ODBC DSN	The installer creates an ODBC connection that AuthMinder uses to connect to the database. The recommended value to enter is arcotdsn.
User Name	The database user name for AuthMinder to access the database. This name is specified by the database administrator. Note: The User Name for the Primary and Backup DSNs <i>must</i> be different.
Password	The password associated with the User Name you specified in the previous field and which is used by AuthMinder to access the database. This password is specified by the database administrator.
Host Name	The host name or IP address of the AuthMinder datastore. <ul style="list-style-type: none"> ■ Default Instance Syntax: <server_name> Example: demodatabase ■ Named Instance Syntax: <server_name>\<instance_name> Example: demodatabase\instance1
Port Number	The port at which the Database listens to the incoming requests. Note: Press Enter , if you want to accept the default port.
Database	The name of the database that AuthMinder will access.

- Specify the information listed in the following table if you specified **3** (Oracle Database Server) in the preceding step.

Parameter	Description
Primary ODBC DSN	The installer creates an ODBC connection that AuthMinder uses to connect to the database. The recommended value to enter is arcotdsn.

Parameter	Description
User Name	The database user name for AuthMinder to access the database. This name is specified by the database administrator. Note: The User Name for the Primary and Backup DSNs <i>must</i> be different.
Password	The password associated with the User Name you specified in the previous field and which is used by AuthMinder to access the database. This password is specified by the database administrator.
Service ID	The Oracle System Identifier (SID) that refers to the instance of the Oracle database running on the server.
Port Number	The port at which the Database listens to the incoming requests. Note: Press Enter , if you want to accept the default port.
Host Name	The host name or IP address of the AuthMinder datastore. Syntax: <server_name> Example: demodatabase

- Specify the information listed in the following table if you specified **4** (MySQL) in the preceding step.

Parameter	Description
Primary ODBC DSN	The installer creates an ODBC connection that AuthMinder uses to connect to the database. The recommended value to enter is arcotdsn.
User Name	The database user name for AuthMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) Note: The User Name for the Primary and Backup DSNs <i>must</i> be different.
Password	The password associated with the User Name you specified in the previous field and which is used by AuthMinder to access the database. This password is specified by the database administrator.

Parameter	Description
Server Name	<p>The host name or IP address of the AuthMinder datastore.</p> <ul style="list-style-type: none"> ■ Default Instance Syntax: <server_name> Example: demodatabase ■ Named Instance Syntax: <server_name>\<instance_name> Example: demodatabase\instance1
Port Number	<p>The port on which the database server listens to the incoming requests.</p> <p>Note: Press Enter, if you want to accept the default port.</p>
Database	The name of the MS SQL database instance.

The Backup Database Access Configuration options appear.

1. Perform one of the following steps:
 - Type **N** to skip the configuration of the secondary DSN, when prompted, and press **Enter** to continue.
 - Type **Y** to configure the secondary DSN, when prompted, and press **Enter** to continue.

For information on the tasks to be performed, see the table in the preceding step.

The Encryption Configuration options appear. Use these options to select the encryption mode and set the information used for encryption.

2. Specify the following information:
 - **Master Key:** Enter the master key, which will be used to encrypt the data stored in the database. By default, the value of master key is set to MasterKey. This key is stored in the securestore.enc file, which is located at <install_location>/arcot/conf.

If you want to change the value of the master key after the installation, then you must regenerate the securestore.enc file with a new master key value. For more information, see the *CA AuthMinder Administration Guide*.
 - Enter **y** if you want to use a Hardware Security Module (HSM) to encrypt the sensitive data. Enter **n** if you want to use the software encryption, in this case you do not have to provide the following HSM information.

- Enter **1** if you want to use Luna HSM or **2** if you want to use nCipher netHSM.
- **HSM PIN:** Enter the password that is used to connect to the HSM.
- **Shared Library:** The absolute path to the PKCS#11 shared library corresponding to the HSM.

For Luna (libCryptoki2.so) and for nCipher netHSM (libcknfast.so), enter the absolute path and full name of the file.

- **Storage Slot Number:** The HSM slot where the 3DES keys used for encrypting the data are present. The default value for Luna is 0, and for nCipher netHSM the default value is 1.

Note: The HSM parameter values are recorded in the `arcotcommon.ini` file located at `<install_location>/arcot/conf`. To change these values after installation, you must edit the `arcotcommon.ini` file, as discussed in appendix, "[Configuration Files and Options](#)" (see page 181).

3. Press **Enter** to continue.

The Pre-Installation Summary appears. This summary lists the product details, installation directory, type of installation, and components that are to be installed.

4. Review the product details displayed carefully and press **Enter** to proceed. If you would like to change a configuration option, type **back** until you reach the option, make the required changes, and press **Enter**.

The Installing message appears. This might take several minutes, because the installer now:

- Puts all the components and their related binaries in the installation directory.
- Stores database settings in the [arcotcommon.ini](#) (see page 181) file and the password in the `securestore.enc` file.
- Writes to the required INI files.
- Sets the environment variables such as, `ARCOT_HOME`, `ODBCHOME`, `ODBCINI`, `LD_LIBRARY_PATH`, `LC_MESSAGES`, `NLS_LANG`, and `LC_CTYPE` in the `arwfenv` file.
- Creates or overwrites, as specified earlier, the Primary DSN and Backup DSN (if selected and configured) using the selected ODBC driver in the `odbc.ini` file.

After the preceding tasks are completed successfully, the Installation Complete message appears.

5. Press **Enter** to exit the installer.
You might have to wait for a few minutes (for the installer to clean up temporary files) until the prompt re-appears.
6. Ensure that UTF-8 support is enabled:
 - a. Navigate to the `<install_location>/arcot/odbc32v70wf/odbc.ini` file.
 - b. Locate the [ODBC] section.
 - c. Ensure that the `IANAAppCodePage=106` entry is present in the section.
 - d. If you do not find this entry, then add it.
 - e. Save and close the file.

Installation Logs

After installation, you can access the installation log file (Arcot_WebFort_Install_<mm_dd_yyyy_hh_mm_ss>.log) in the `<install_location>` directory. For example, if you had specified the `/opt` directory as the installation directory, then the installation log file is created in the `/opt` directory.

If the installation fails for some reason, then error messages are recorded in this log file.

Performing Post-Installation Tasks on the First System

The following post-installation steps are discussed in this section:

1. [Running Database Scripts](#) (see page 109)
2. [Verifying the Database Setup](#) (see page 71)
3. [Preparing Your Application Server](#) (see page 110)
4. [Deploying Administration Console](#) (see page 117)
5. [Verifying the Administration Console](#) (see page 81)
6. [Logging In to Administration Console](#) (see page 81)
7. [Bootstrapping the System](#) (see page 82)
8. [Starting AuthMinder Server](#) (see page 122)
9. [Verifying the Installation](#) (see page 85)
10. [Deploying User Data Service](#) (see page 125)

Note: After completing these post-installation tasks, perform the Java SDKs and Web services configuration as discussed in chapter, "[Configuring AuthMinder Java SDKs and Web Services](#)" (see page 137).

Running Database Scripts

AuthMinder ships with SQL scripts that create its schema and set initial configuration values in the AuthMinder database.

To configure the database used by AuthMinder:

1. Locate the folder with the scripts for your database type. The default location is:

(for MS SQL) `<install_location>/arcot/dbscripts/mssql`

(for Oracle) `<install_location>/arcot/dbscripts/oracle`

(for IBM DB2 UDB) `<install_location>/arcot/dbscripts/db2`

(for MySQL) `<install_location>/arcot/dbscripts/mysql`

2. Run the *scripts in the following order* by using the database vendor tools:

- a. `arcot-db-config-for-common-2.0.sql`

Important! If you have installed CA RiskMinder 3.1, you need not run `arcot-db-config-for-common-2.0.sql` because you have already run it while installing CA RiskMinder 3.1.

- b. `arcot-db-config-for-webfort-7.1.sql`

Note: If you encounter any error while executing the scripts, then check with your database administrator whether you have the required privileges.

Verifying the Database Setup

After running the required database scripts, you must verify whether the AuthMinder schemas were seeded correctly. To do so, you need to use the `arwfutil` tool.

Perform the following steps to check the database setup:

1. Open a command prompt window.
2. Navigate to the following location:
`<install_location>/arcot/sbin`
3. At the command prompt enter the following command:
`./arwfutil vdb`

The preceding command creates the `arcotwebfort-vdb-<dd>-<mmm>-<yy>.txt` file in the `<install_location>/arcot/logs` directory.

4. Open the `arcotwebfort-vdb-<dd>-<mmm>-<yy>.txt` file in a text editor, and check for entries of the following type:

```
ARWF*      FOUND
```

These lines indicate that your database setup has been successful.

5. Close the `arcotwebfort-vdb-<dd>-<mmm>-<yy>.txt` file.

Preparing Your Application Server

Two components of AuthMinder, User Data Service (UDS) and Administration Console, are web based and can be deployed on any of the following supported application servers:

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic
- JBoss

Before you deploy the WAR files for these web applications on the application server of your choice, copy the AuthMinder files and JDBC JARs that are required by UDS and Administration Console to the appropriate location on the application server. This section walks you through the steps to copy the required crypto files to your application server:

- [Step 1: Setting Java Home](#) (see page 110)
- [Step 2: Copying Database Access Files to Your Application Server](#) (see page 111)
- Step 3: Copying JDBC JARs to Your Application Server
- [Step 4: \(Mandatory for Oracle WebLogic 10.1\) Creating Enterprise Archive File](#) (see page 116)

Step 1: Setting Java Home

Before you deploy the WAR files for UDS and Administration Console on the application server of your choice, ensure that you set the JAVA_HOME environment variable. For Apache Tomcat, set JAVA_HOME to the Java home directory corresponding to the JDK that you are using.

In addition, include \$JAVA_HOME/bin in the PATH environment variable. If you fail to do so, then Administration Console and other JDK-dependent components may fail to start.

Step 2: Copying Database Access Files to Your Application Server

UDS and Administration Console use the following files to access the AuthMinder database securely:

- libArcotAccessKeyProvider.so available at:
`<install_location>/arcot/native/<platform name>/<32bit-or-64bit>/`
- arcot-crypto-util.jar available at:
`<install_location>/arcot/java/lib/`

As a result, these files must be copied to the appropriate location on the application server where you have deployed these AuthMinder components. The following subsections provide information about copying these files for:

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic
- JBoss Application Server

Apache Tomcat

To copy the database access files on Apache Tomcat:

1. Copy the libArcotAccessKeyProvider.so file to the following directory:
 - **For Solaris:** `<JAVA_HOME used by Apache Tomcat>/jre/bin`
 - **For RHEL:** `<JAVA_HOME used by Apache Tomcat>/jre/bin`
2. Copy the arcot-crypto-util.jar file to the `<JAVA_HOME used by Apache Tomcat>/jre/lib/ext` directory.
3. Set and export the LD_LIBRARY_PATH to the directory where the libArcotAccessKeyProvider.so file is copied.
4. Restart the application server.

Note: An application server restart is required as part of some of the remaining installation tasks. To minimize the number of times you restart the application server, restart it once after performing the last task that requires a restart.

IBM WebSphere

Note: Depending on the WebSphere version that you are using, the steps that you follow may be different.

To copy the database access files on WebSphere 6.1:

1. Log in to WebSphere Administration Console.
2. Click Environment, and then click Shared Libraries.

- a. From the Scope drop-down, select a valid visibility scope. The scope must include the target server/node on which the application is deployed.
 - b. Click New.
 - c. Enter the Name, for example, ArcotJNI.
 - d. Specify the Classpath. This path must point to the location where the arcot-crypto-util.jar file is present and must also include the file name. For example, <install_location>/arcot/java/lib/arcot-crypto-util.jar.
 - e. Enter the JNI library path. This path must point to the location where the libArcotAccessKeyProvider.so file is present. For example, the <install_location>/arcot/java/native/linux/<32bit-or-64bit> directory.
 - f. Click Apply.
3. Configure server-level class loaders.
 - a. Click Servers, and then click Application Servers.
 - b. Under Application Servers access the settings page of the server for which the configuration has been performed.
 - c. Click Java and Process Management, and then click Class Loader.
 - d. Click New. Select default Classes loaded with parent class loader first and click OK.
 - e. Click the auto-generated Class Loader ID.
 - f. In the class loader Configuration page, click Shared Library References.
 - g. Click Add, select the shared library that you created earlier in this procedure (for example, ArcotJNI), and then click Apply.
 - h. Save the changes.
 4. Copy libArcotAccessKeyProvider.so file to the following directory:
 - **For Solaris:** <JAVA_HOME used by IBM WebSphere>/jre/bin
 - **For RHEL:** <JAVA_HOME used by IBM WebSphere>/jre/bin
 5. Restart the application server.

Note: An application server restart is required as part of some of the remaining installation tasks. To minimize the number of times you restart the application server, restart it once after performing the last task that requires a restart.

Oracle WebLogic

To copy the database access files on the Oracle WebLogic server:

1. Copy libArcotAccessKeyProvider.so to the following directory:
 - **For Solaris:** <JAVA_HOME used by Oracle WebLogic instance>/jre/bin
 - **For RHEL:** <JAVA_HOME used by Oracle WebLogic instance>/jre/bin

2. Set and export the LD_LIBRARY_PATH to the directory where the libArcotAccessKeyProvider.so file is copied.
3. Copy arcot-crypto-util.jar to the <JAVA_HOME used by Oracle WebLogic instance>/jre/lib/ext directory.
4. Log in to WebLogic Administration Console.
5. Navigate to **Deployments**.
6. Enable the **Lock and Edit** option.
7. Click **Install** and navigate to the directory that contains the arcot-crypto-util.jar file.
8. Click **Next**.
The Application Installation Assistant screen appears.
9. Click **Next**.
The Summary page appears.
10. Click **Finish**.
11. Activate the changes.
12. Restart the application server.

Note: An application server restart is required as part of some of the remaining installation tasks. To minimize the number of times you restart the application server, restart it once after performing the last task that requires a restart.

JBoss Application Server

To copy the database access files on JBoss Application Server:

1. Copy libArcotAccessKeyProvider.so to the following directory:
 - **For Solaris:** <JAVA_HOME used by JBoss Application Server instance>/jre/bin
 - **For RHEL:** <JAVA_HOME used by JBoss Application Server instance>/jre/bin
2. Copy arcot-crypto-util.jar to the <JAVA_HOME used by JBoss Application Server instance>/jre/lib/ext directory.
3. Restart the application server.

Note: An application server restart is required as part of some of the remaining installation tasks. To minimize the number of times you restart the application server, restart it once after performing the last task that requires a restart.

Step 3: Copying JDBC JARs to Your Application Server

Administration Console, User Data Service (UDS), and Sample Application, which are the Java-dependant components of AuthMinder need Java Database Connectivity (JDBC) Java ARchive (JAR) files to connect to the database. You must copy these files to the application server.

Note: Before proceeding with the steps mentioned in the following subsections, ensure that you have downloaded the JDBC JARs, see chapter, "[Preparing for Installation](#)" (see page 33) for more information on supported JDBC JARs.

The following sub-sections walk you through the steps for copying the JDBC JAR required for your database to one of the following application servers:

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic
- JBoss Application Server

Apache Tomcat

To copy the JDBC JAR file to the Apache Tomcat installation directory:

1. Navigate to the location where you have downloaded the JDBC JAR file.
2. Copy the JDBC JAR file and paste it in the following directory:
 - **For Apache Tomcat 5.5.x:** <TOMCAT-HOME>\common\lib
 - **For Apache Tomcat 6.x and 7.x:** <TOMCAT-HOME>\lib

Alternatively, add the path that contains the JDBC JAR file to the **Classpath** environment variable.

3. Restart Apache Tomcat.

IBM WebSphere

To copy the JDBC JAR file to the IBM WebSphere:

1. Log in to WebSphere Administration Console.
2. Click **Environment**, and then click **Shared Libraries**.
 - a. From the **Scope** drop-down, select a valid visibility scope. The scope must include the target server/node on which the application is deployed.
 - b. Click **New**.
 - c. Enter the **Name**, for example, **JDBCJAR**.
 - d. Specify the **Classpath**. This path must point to the location where the JDBC JAR file is present and must also include the file name.

- e. Click **Apply** to save the changes made.
3. Configure server-level class loaders.
Note: You can either create a class loader or use the one that you created while performing the procedure described in [Step 2: Copying Database Access Files to Your Application Server](#) (see page 111).
 - a. Click **Servers**, and then click **Application Servers**.
 - b. Under **Application Servers** access the settings page of the server for which the configuration are performed.
 - c. Click **Java and Process Management**, and then click **Class Loader**.
 - d. Click **New**. Select default **Classes loaded with parent class loader first** and click **OK**.
 - e. Click the auto-generated **Class Loader ID**.
 - f. In the class loader **Configuration** page, click **Shared Library References**.
 - g. Click **Add**, select **JDBCJAR** and then click **Apply**.
 - h. Save the changes made.
4. Restart IBM WebSphere.

Oracle WebLogic

To copy the JDBC JAR file to Oracle WebLogic:

Note: If you are using Oracle database, then you need *not* perform the configurations mentioned in this section, because WebLogic supports Oracle database by default.

1. Copy the JDBC JAR file to the following directory:
`<JAVA_HOME used by Oracle WebLogic instance>/jre/lib/ext`
directory.
2. Login to WebLogic Administration Console.
3. Navigate to **Deployments**.
4. Enable the **Lock and Edit** option.
5. Click **Install** and navigate to the directory that contains the JDBC JAR file.
6. Click **Next**.

The Application Installation Assistant screen appears.

7. Click **Next**.
The Summary page appears.
8. Click **Finish**.
9. Activate the changes.
10. Restart Oracle WebLogic Server.

JBoss Application Server

To copy the JDBC JAR file on JBoss Application Server:

1. Copy the JDBC JAR file to the following location on the JBOSS installation directory:
`<JBOSS_HOME>/server/default/lib`
2. Restart JBoss Application Server.

Step 4: (Mandatory for Oracle WebLogic 10.1) Creating Enterprise Archive File

Most application servers (such as WebSphere and WebLogic) enable you to bundle related JAR files or WAR files from the same vendor into a single enterprise application (or archive). As a result, all such related JARs or WARs can be deployed together and loaded by a class loader. Such an archive also contains an application.xml file, which is generated automatically and describes how to deploy each bundled module.

By default, AuthMinder provides WAR files to deploy Administration Console and User Data Service. If required, you can change the format of these files to EAR and then deploy the EAR files.

Perform the following steps to create an EAR file:

1. Open the command prompt window.
2. Navigate to the `<install_location>/arcot/tools/common/bundlemanager` directory.
3. Run the bundlemanager tool to create the EAR file by using the following command:

```
java -jar bundle-manager.jar -ear <filename.ear> -warList  
<war_file_name>
```

Note: In this command, replace `<war_file_name>` with `arcotadmin.war` to generate the EAR file for Administration Console. Similarly, replace `<war_file_name>` with `arcotuds.war` to generate the EAR file for UDS.

This command generates the EAR file at `<install_location>/arcot/java/webapps`.

Deploying Administration Console

Note: If you plan to deploy Administration Console on WebSphere 7.0, then refer to the instructions provided in appendix, "[Deploying Administration Console on IBM WebSphere 7.0](#)" (see page 247).

You need the file `arcotadmin.war` to deploy the AuthMinder Administration Console. This file is available at:

`<install_location>/arcot/java/webapps/`

Note: To manage AuthMinder Server by using Administration Console, you must ensure that Administration Console can access the system where AuthMinder Server is installed by its hostname.

To deploy Administration Console:

1. Change the working directory to:
`<install_location>/arcot/sbin`
2. Type `source arwfenv` and press **Enter** to set the `$ARCOT_HOME` environment variable.
3. Restart the application server for the changes to take effect.
4. Deploy `arcotadmin.war` in the appropriate directory on the application server.

Note: The deployment procedure depends on the application server that you are using. Refer to your application server vendor documentation for detailed instructions.

For example, in the case of Apache Tomcat, you must deploy the WAR file at `<APP_SERVER_HOME>/webapps/`.

5. **(For JBoss Only)** Perform the following steps if you have deployed Administration Console on JBoss application server:
 - a. Copy the Bouncy Castle JAR file (`bcprov-jdk15-146.jar`) from `<install_location>/arcot/java/lib/` to the following location:
`<JBOSS_HOME>/common/lib`
 - b. Navigate to the following location:
`<JBOSS_HOME>/server/default/conf/`
 - c. Open `jboss-log4j.xml` file in a text editor.
 - d. Add the following log configuration in the `<log4j:configuration>` section:


```
<appender name="arcotadminlog"
class="org.apache.log4j.RollingFileAppender">
  <errorHandler
class="org.jboss.logging.util.OnlyOnceErrorHandler"></errorHandler>
  <param name="Threshold" value="INFO"/>
  <param name="MaxFileSize" value="10MB"/>
  <param name="MaxBackupIndex" value="100"/>
  <param name="Encoding" value="UTF-8"/>
  <param name="Append" value="true"/>
```

```
<param name="File" value="\${arcot.home}/logs/arcotadmin.log"/>
<layout class="org.apache.log4j.PatternLayout">
<param name="ConversionPattern" value="%d{yyyy-MM-dd hh:mm:ss,SSS z} : [%t]
: %-5p : %-5c{3} : %m%n"/>
</layout>
<filter class="org.jboss.logging.filter.TCLMCFilter">
<param name="AcceptOnMatch" value="true"/>
<param name="DeployURL" value="arcotadmin.war"/>
</filter>
<!-- end the filter chain here -->
<filter class="org.apache.log4j.varia.DenyAllFilter"></filter>
</appender>
```

- e. Add the following log category:

```
<category name="com.arcot">
<priority value="INFO" />
<appender-ref ref="arcotadminlog"></appender-ref>
</category>
```
- f. Add the following category for cryptographic operations:

```
<category name="com.arcot.crypto.impl.NCipherCrypter">
<priority value="FATAL" />
<appender-ref ref="arcotadminlog"></appender-ref>
</category>
```
- g. Save and close the file.
- h. Take the backup of existing JBoss logging libraries. These library files are available at:

```
<JBOSS_HOME>/lib
```
- i. Upgrade the JBoss logging libraries available at <JBOSS_HOME>/lib to version 2.1.1. The following table lists the JAR file names and the location from where you can download the files:

File Name	Location
jboss-logging-jdk-2.1.1.GA.jar	http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-jdk/2.1.1.GA/
jboss-logging-spi-2.1.1.GA.jar	http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-spi/2.1.1.GA/
jboss-logging-log4j-2.1.1.GA.jar	http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-log4j/2.1.1.GA/

- 6. Ensure that the application is restarted.

Verifying the Administration Console

The arcotadmin.log file is used for logging the Administration Console information.

To verify if the Administration Console was deployed successfully:

1. Navigate to the following location:
`<install_location>/arcot/logs`
2. Open the arcotadmin.log file in any editor and locate the following lines:
Arcot Administration Console Configured Successfully.

This line indicates that your Administration Console was deployed successfully.

Note: Check and resolve if there any FATAL and ERROR messages, and review all the WARNING messages for unexpected conditions.

Important! If you want to localize your Administration Console, then you must prepare your Administration Console for the same, as discussed in appendix, "[Preparing for Localization](#)" (see page 203).

After you are done with the steps for localization, then resume performing the post-installation tasks that follow.

Logging In to Administration Console

When logging in to Administration Console for the first time, you *must* use the Master Administrator credentials that are created automatically in the database during installation.

To log in to Administration Console:

1. Start Administration Console in a Web browser window, by using the following URL:

`http://<host>:<app_server_port>/arcotadmin/masteradminlogin.htm`

Note: The host and port information that you specify in the preceding URL must be of the application server where Administration Console is deployed.

2. Log in to Administration Console as a Master Administrator with the default Master Administrator account credentials. The credentials are:
 - **User Name:** masteradmin
 - **Password:** master1234!

Bootstrapping the System

Before you can start using the Administration Console to manage AuthMinder, you must first perform the following mandatory steps to initialize the system:

- Change the default Master Administrator password
- Specify the Global Key label
- Specify the authentication mechanism for the Default organization

Bootstrapping is a wizard-driven process that walks you through these setup tasks. Other administrative links are enabled after you perform these tasks.

Before you proceed with [Performing the Bootstrapping Tasks](#) (see page 83), you must understand the related concept of Default Organization.

Default Organization

When you deploy the Administration Console, an organization is created by default. This organization is referred to as *Default Organization* (**DEFAULTORG**). As a single-organization system, the Default Organization itself can be used without creating any new organizations.

Performing the Bootstrapping Tasks

When you first log in to the Administration Console as the Master Administrator (MA), the Summary screen for the Bootstrap wizard screen appears.

To bootstrap the system using the wizard:

1. Click **Begin** to start the process.

The Change Password screen appears.

2. Specify the **Old Password**, **New Password**, **Confirm Password**, and click **Next**.

The Configure Global Key Label screen appears.

3. Specify the **Global Key Label**, and click **Next**.

AuthMinder enables you to use hardware- or software-based encryption of your sensitive data. Irrespective of hardware or software encryption, all Arcot products use *Global Key Label* for encrypting user and organization data.

If you are using hardware encryption, then this label serves only as a reference (or pointer) to the actual 3DES key stored in the HSM device, and therefore the key label *must* match the HSM key label. However, in the case of software-based encryption, this label acts as the reference to the actual software key in the database.

Caution: After you complete the bootstrapping process, you can *not* update this key label.

The **Storage Type** field indicates whether the encryption key is stored in the database (**Software**) or the HSM (**Hardware**).

The Configure Default Organization screen appears.

4. Under the **Default Organization Configuration** section, specify the following parameters for the Default Organization:
 - **Display Name:** The descriptive name of the organization. This name appears on all other Administration Console pages and reports.
 - **Administrator Authentication Mechanism:** The mechanism that is used to authenticate administrators belonging to the Default Organization. Administration Console supports three types of authentication methods for the administrators to log in:
 - **Basic**

If you choose this option, then the inbuilt authentication method provided by the Administration Console is used for authenticating the administrators.
 - **LDAP User Password**

If you choose this option, then the administrators are authenticated using their credentials that are stored in the directory service.

Note: If this mechanism is used for authenticating administrators, then you must deploy UDS, as discussed in the "[Deploying User Data Service](#)" (see page 125) section.

■ **WebFort User Password**

If you select the **WebFort User Password** option here, then the credentials are issued and authenticated by the AuthMinder Server.

Note: Refer to *CA AuthMinder Administration Guide* for more information on how to do this.

5. Under the **Key Label Configuration** section of the Configure Default Organization screen, specify the following:
 - **Use Global Key:** This option is selected by default. Deselect this option if you want to override the Global Key Label you specified in the preceding step and specify a new encryption label.
 - **Key Label:** If you deselected the **Use Global Key** option, then specify the new key label that you want to use for the Default Organization.
 - **Storage Type:** This field indicates whether the encryption key is stored in the database (**Software**) or the HSM (**Hardware**).
6. Click **Finish** to complete the bootstrapping process.

The Administration Console initialization is completed, as indicated in the Finish screen.
7. Click **Continue** to proceed with other configurations using the Administration Console.

Note: If you want to localize your Administration Console, then you must prepare your Administration Console for the same, as discussed in appendix, "[Preparing for Localization](#)" (see page 203) for more information. After you are done with the steps for localization, then resume performing the post-installation tasks that follow.

Starting AuthMinder Server

Perform the following steps to start AuthMinder Server:

1. Navigate to the following directory:
`<install_location>/arcot/bin/`
2. Run the following command:
`./webfortserver start`

Note: If you want to stop the server, then execute `./webfortserver stop` command.

Verifying the Installation

You can verify whether the AuthMinder Server and the web applications have started successfully by applying the instructions that are described in the following sections:

- [Using Log files](#) (see page 85)
- [Using webfortserver Utility](#) (see page 85)
- [Checking the Ports](#) (see page 86)

Using Log files

Perform the following steps to verify that AuthMinder Server started correctly:

1. Navigate to the following location:
`<install_location>/arcot/logs`
2. Open the `arcotwebfortstartup.log` file in any editor. This log file is created in the application server home directory.
3. Look for the following lines in the log file:
`INSTANCE_VER.....: [7.1]`
`Arcot WebFort Authentication Service READY`

These lines indicate that AuthMinder Server is installed successfully.

Note: Ensure that the log files do not contain any FATAL and WARNING messages.

Using webfortserver Utility

The webfortserver tool enables you to check the release of AuthMinder that you have installed.

Follow these steps:

1. Navigate to the following location:
`<install_location>/arcot/bin`
2. Run the following command to start the webfortserver tool in interactive mode:
`./webfortserver -i`
3. Enter the release number at the prompt.
The webfort-ver-<dd>-<mmm>-<yy>.txt file is created in the
`<install_location>/arcot/logs` folder.
4. Open this file and check for the following lines to ensure that you are using the current release:
 - The version of AuthMinder library files in the bin section is 7.1.
 - The version of the UDS library file (arwfuds.dll) in the bin section is 2.0.3.
5. Close the file.

Checking the Ports

Perform the following steps to verify if the AuthMinder Server is listening to different protocols on the default ports:

1. Navigate to the following location:
`<install_location>/arcot/logs`
2. Open the arcotwebfortstartup.log file in any editor and search for the protocol names to verify if they are listening on the correct port, as shown in the following snippet.
PROTOCOLNAME : [Administration-WS]
PORTNO : 9745
PROTOCOLID : [Transaction-Native]
PORTNO : 9742
PROTOCOLID : [ServerManagement-WS]
PORTNO : 9743
PROTOCOLID : [Transaction-WS]
PORTNO : 9744

Note: Refer to appendix, "[Default Port Numbers and URLs](#)" (see page 231) for information on default ports and protocols.

Deploying User Data Service

AuthMinder can access user data either from a relational database (RDBMS) or directly from an LDAP server:

You need the file `arcotuds.war` to deploy the User Data Service (UDS). This file is available at:

`<install_location>/arcot/java/webapps/`

To deploy User Data Service:

1. Deploy `arcotuds.war` in the appropriate directory on the application server.

Note: The deployment procedure depends on the application server that you are using. Refer to your application server vendor documentation for detailed instructions.

For example, in case of Apache Tomcat, you must deploy the WAR file at `<APP_SERVER_HOME>/webapps/`.

2. **(For WebSphere Only)** Configure to reload the UDS class when the application files are updated.
 - a. Navigate to **Application > Enterprise Applications** and access the UDS settings page.
 - b. Under **Class loader order**, select the **Classes loaded with local class loader first (parent last)** option.
 - c. Under **WAR class loader policy**, select the **Single class loader for application**.
 - d. Click **Apply**.
3. **(For JBoss Only)** Perform the following steps if you have deployed UDS on JBoss application server:
 - a. Copy the Bouncy Castle JAR file (`bcprov-jdk15-146.jar`) from `<install_location>/arcot/java/lib/` to the following location: `<JBOSS_HOME>/common/lib`
 - b. Navigate to the following location: `<JBOSS_HOME>/server/default/conf/`
 - c. Open `jboss-log4j.xml` file in a text editor.
 - d. Add the following log configuration in the `<log4j:configuration>` section:


```
<appender name="arcotudslog" class="org.apache.log4j.RollingFileAppender">
  <errorHandler
    class="org.jboss.logging.util.OnlyOnceErrorHandler"></errorHandler>
  <param name="Threshold" value="INFO"/>
  <param name="MaxFileSize" value="10MB"/>
  <param name="MaxBackupIndex" value="100"/>
  <param name="Encoding" value="UTF-8"/>
  <param name="Append" value="true"/>
  <param name="File" value="${arcot.home}/logs/arcotuds.log"/>
</layout class="org.apache.log4j.PatternLayout">
```

```
<param name="ConversionPattern" value="%d{yyyy-MM-dd hh:mm:ss,SSS z} : [%t]
: %-5p : %-5c{3}(%L) : %m%n"/>
</layout>
<filter class="org.jboss.logging.filter.TCLMCFilter">
<param name="AcceptOnMatch" value="true"/>
<param name="DeployURL" value="arcotuds.war"/>
</filter>
<!-- end the filter chain here -->
<filter class="org.apache.log4j.varia.DenyAllFilter"></filter>
</appender>
```

- e. Add the following line in the com.arcot category that you created in [Deploying Administration Console](#) (see page 117):
<appender-ref ref="arcotudslog"></appender-ref>
 - f. Add the following line in the for cryptographic category that you created in [Deploying Administration Console](#) (see page 117):
<appender-ref ref="arcotudslog"></appender-ref>
 - g. Save and close the file.
4. Restart the application server.
 5. Perform the following steps to verify if UDS started correctly:
 - a. Navigate to the following location:
<install_location>/arcot/logs
 - b. Open the arcotuds.log file in any editor and locate the following lines:
User Data Service (Version: 2.0.3) initialized successfully.
This line indicates that UDS was deployed successfully.

Note: Check and resolve if there any FATAL and ERROR messages, and review all the WARNING messages for unexpected conditions.

Installing on the Second System

After installing AuthMinder Server and Administration Console you must install the other components on the second system in this distributed-system deployment. The specific components to install must have been determined when you performed your planning in chapter, "[Planning the Deployment](#)" (see page 21).

Before proceeding with the installation, ensure that all the prerequisite software components are installed on this system as described in chapter, "[Preparing for Installation](#)" (see page 33).

Perform the following steps to install AuthMinder components:

1. Locate the AuthMinder installer Arcot-WebFort-7.1-<platform name>-Installer.bin file.
2. Run the installer using the following command:

```
prompt> sh Arcot-WebFort-7.1-<platform name>-Installer.bin
```

The installer starts preparing for the installation.
3. Perform the steps described in [Installing on the First System](#) (see page 99), until you reach the the Choose Product Features screen.
4. When the Choose Product Features options appear, select the components that you want to install. Typically, you will be installing the SDKs and Sample Application.
5. After you have selected the components, follow the steps from Step 12 through Step 19 to complete the installation.

Performing Post-Installation Tasks on the Second System

Perform the following post-installation tasks on the second system where you have installed Java SDKs, Web services, and Sample Application:

- [Deploying Sample Application](#) (see page 128)
- [Configuring Sample Application for Communication with AuthMinder Server](#) (see page 128)
- [Using Sample Application](#) (see page 129)

Deploying Sample Application

Important! Do not use Sample Application in production deployments. It is recommended that you build your own web application by using Sample Application as a code reference.

Sample Application can be used to verify that AuthMinder is installed and configured correctly. In addition, Sample Application demonstrates the following features:

- The typical AuthMinder workflows
- The tasks that you can perform using the AuthMinder Java APIs
- Integration of your application with AuthMinder

Follow these steps:

1. Deploy the webfort-7.1-sample-application.war file from the following location:
`<install_location>/arcot/samples/java`
2. Start Sample Application.
3. Access Sample Application by using the following URL:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`

Configuring Sample Application for Communication with AuthMinder Server

If Sample Application and AuthMinder Server are installed on different systems, then configure Sample Application to communicate with the AuthMinder Server.

Follow these steps:

1. Access the Sample Application in a Web browser window. The default URL for Sample Application is as follows:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`
The WebFort 7.1 Sample Application page appears.
2. In the navigation pane, click **Setup -> Server Connectivity** to open the WebFort Server Connectivity page.
3. Specify the values for the connection parameters that are listed in the following table:

Note: The configurations that are made using these parameters are valid for the current session. Set these values again if you restart Sample Application or the application server.

Field	Default Value	Description
IP Address	localhost	The host name or the IP address where the AuthMinder Server is available.

Field	Default Value	Description
Port	9742	The port on which the Authentication and the Issuance service is available.
Maximum Active Connections	64	The maximum number of database connections maintained by the Sample Application.

4. Click **Set Up** to save the connection.

To configure the Sample Application to communicate with another AuthMinder Server instance:

1. Click the **[+]** sign preceding **Additional Server Configurations**.
2. Specify the **IP Address** and **Port** connection parameters.
3. Click **Set Up** to configure the connection.

Using Sample Application

Sample Application enables you to issue and authenticate the credentials that AuthMinder Supports. You can use the Sample Application to perform these operations to test whether the AuthMinder installation has been successful.

This section covers the following tasks:

- [Creating Users](#) (see page 90)
- [Setting Up ArcotID PKI Client](#) (see page 91)
- [Creating ArcotID PKI Credential](#) (see page 92)
- [Downloading ArcotID PKI](#) (see page 93)
- [Authenticating Using ArcotID PKI](#) (see page 94)

Creating Users

Note: User creation must be performed either using Administration Console or Web services.

To create users using Administration Console:

1. Log in to the Administration Console as a Global Administrator (GA) or an Organization Administrator (OA). The URL for the purpose is:
http://<host>:<app_server_port>/arcotadmin/adminlogin.htm
2. If already not activated, activate the **Manage Users and Administrators** sub-tab under the **Users and Administrators** tab.
3. In the left-hand pane, under **Manage Users and Administrators**, click **Create User** to open the Create User page.
4. On the Create User page:
 - a. Enter a unique user name, their organization name, and optionally, other user information in the **User Details** section
 - b. If required, specify other user information in the corresponding fields on the page.
 - c. Select the required **User Status**.
 - d. Click **Create User**.

The "Successfully created the user" message appears if the specified user was successfully added to the database.

5. Return to the WebFort Sample Application page.

Setting Up ArcotID PKI Client

To set up the ArcotID PKI Client to communicate with the AuthMinder Server for authenticating users with their ArcotID PKI, you need to perform the following steps:

1. Access the Sample Application using the following URL:
http://<host>:<app_server_port>/webfort-7.1-sample-application/
2. In the left-hand pane, click **Setup** -> **ArcotID Client** to open ArcotID Client Settings page.
3. In the **Choose ArcotID Client** section, select the type of client that has to be used to authenticate the ArcotID PKI.
4. In the **Choose ArcotID Download Type** section, select the location where you want to store the ArcotID PKI.
5. In the **Choose Where & When to Obtain the ArcotID Challenge** section, select the mode of obtaining the ArcotID PKI challenge.
6. Click **Select** to save the settings.

The "The operation was successful" message appears if the ArcotID PKI Client configuration was performed successfully.

Creating ArcotID PKI Credential

To create ArcotID PKI credential for users:

1. Access the Sample Application using the following URL:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`
2. In the left-hand pane, click **ArcotID** -> **Issuance** -> **Create** to open Create ArcotID page.
3. Specify the name of the user you created in the **User Name** field.
4. If required, specify the user's organization in the **Organization** field.
5. Specify the password to be used for authentication in the **ArcotID Password** field.
6. If required, specify the profile that has to be used to issue ArcotID PKI in the **Profile Name** field.
7. If required, specify the name-value pairs of the **Unsigned Attributes**. The attributes are set in the unsigned portion of the ArcotID PKI.
8. If required, specify the **Custom Attributes** to be used for creating the ArcotID PKI.
9. If required, specify the **Additional Input** that you want to pass to the AuthMinder Server.
10. If required, pass the following **Transaction Logging Parameters**:
 - In the **Log Level** field, choose the logging level.
Note: See the topic titled "WebFort Logging" in *CA AuthMinder Administration Guide* for more information on the log levels.
 - Select **Enable Trace Logging** if you want to capture flow details.
 - Select **Enable DB Logging** if you want to log the database activities.
 - Select **Enable Sensitive Data Logging** if you want to log the sensitive data.
11. Click **Create** to create the credential.

The "The operation was successful" message appears if the ArcotID PKI was created successfully for the user.

Downloading ArcotID PKI

To download the ArcotID PKI:

1. Access the Sample Application using the following URL:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`
2. In the left-hand pane, click **ArcotID** -> **Issuance** -> **Download** to open the Download ArcotID page.
3. Specify the name of the user you created in the **User Name** field.
4. If required, specify the user's organization in the **Organization** field.
5. If required, specify the profile that has been used to issue ArcotID PKI in the **Profile Name** field.
6. If required, specify the **Additional Input** that you want to pass to the AuthMinder Server.
7. If required, pass the following **Transaction Logging Parameters**:
 - In the **Log Level** field, choose the logging level.
Note: Refer to the topic titled "AuthMinder Logging" in *CA AuthMinder Administration Guide* for more information on the log levels.
 - Select **Enable Trace Logging** if you want to capture flow details.
 - Select **Enable DB Logging** if you want to log the database activities.
 - Select **Enable Sensitive Data Logging** if you want to log the sensitive data.
8. Click **Download** to download the user's ArcotID PKI.

Authenticating Using ArcotID PKI

To authenticate using the ArcotID PKI:

1. Access the Sample Application using the following URL:
`http://<host>:<app_server_port>/webfort-7.1-sample-application/`
2. In left-hand pane, click **ArcotID** -> **Authentication** -> **Authenticate** to open the ArcotID Authentication page.
3. Specify the name of the user you created in the **User Name** field.
4. Specify the user's organization in the **Organization** field.
5. Specify the user's ArcotID PKI password in the **ArcotID Password** field.
6. If you are using aliases to identify the users, then specify the **Application Context** based on the alias of the user that you want to authenticate.
7. If required, select the **Token Type** that has to be returned to the user after successful authentication.

Note: Refer to the topic titled "Authenticating Users" in the *CA AuthMinder Developer's Guide* for more information on the token types.
8. If required, specify the **Authentication Policy Name** that has to be used for authenticating users.
9. If you have selected SAML as the token type, then specify the **SAML Policy Name** to be used.
10. If required, specify the **Additional Input** that you want to pass to the AuthMinder Server.
11. If required, pass the following **Transaction Logging Parameters**:
 - In the **Log Level** field, choose the logging level.

Note: Refer to the topic titled "AuthMinder Logging" in *CA AuthMinder Administration Guide* for more information on the log levels.
 - Select **Enable Trace Logging** if you want to capture flow details.
 - Select **Enable DB Logging** if you want to log the database activities.
 - Select **Enable Sensitive Data Logging** if you want to log the sensitive data.
12. Click **Authenticate** to verify the user's ArcotID PKI.

Post-Installation Checklist

It is recommended that you fill this checklist with the installation and setup information for AuthMinder. You will need this information for various administrative tasks that you will perform later.

Your Information	Example Entry	Your Entry
ARCOT_HOME	/var/opt/arcot/	
SYSTEM INFORMATION		
Host Name	my-bank	
User Name	administrator	
Password	password1234!	
Configured Components	WebFort Server Administration Console User Data Service	
ADMINISTRATION CONSOLE INFORMATION		
Host Name	localhost	
Port	8080	
Master Administrator Password	mypassword1234!	
USER DATA SERVICE INFORMATION		
Host Name	localhost	
Port	8080	
Application Context Root	arcotuds	

Chapter 6: Configuring AuthMinder Java SDKs and Web Services

This chapter describes the steps to configure the Java Software Development Kit (SDK) and Web services provided by AuthMinder.

The chapter covers the following topics:

- [AuthMinder APIs](#) (see page 138)
- [Configuring Java SDKs](#) (see page 138)
- [Working With AuthMinder Web Services](#) (see page 140)
- [Enabling SSL Communication](#) (see page 142)

AuthMinder APIs

AuthMinder is shipped with a set of Java APIs that are available as JAR files at the following location:

`<install_location>/arcot/sdk/client/java/lib/arcot/`

In this location you will find the following JAR files:

- **arcot-webfort-authentication.jar**

The JAR file that you need to implement the AuthMinder Authentication SDK. This file consists the Java package that contains the logic for user authentication. Operations that this package enables include:

- Authenticate users using credentials that AuthMinder supports
- Authenticate custom credentials

- **arcot-webfort-common.jar**

This file consists the Java package that contains the classes that are common for Authentication and Credential Management (Issuance) operations. This package is used for the following operations:

- Pass additional information to the AuthMinder Server
- Specify the One-Time Password type
- Specify the User and Credential Status

- **arcot-webfort-issuance.jar**

The JAR file that you need to implement the WebFort Credential Management SDK. This file consists the Java package that contains the logic for managing credentials. Operations that this package enables include:

- Create and manage credentials
- Manage ArcotID PKI key bag

Configuring Java SDKs

This section provides the procedure to configure the Authentication and Credential Management Java SDKs so that they can be integrated with your existing application.

- [Configuring Authentication Java SDK](#) (see page 139)
- [Configuring Credential Management Java SDK](#) (see page 140)

Note: Before proceeding with the configuration steps in this section, ensure that the JAR file required for implementing Java APIs are installed in the `<install_location>/arcot/sdk/client/java/lib/` location.

Configuring Authentication Java SDK

To configure Authentication SDK for using in a J2EE Application:

1. Copy the listed JAR files from the following location:
`<install_location>/arcot`
to the appropriate location in your `<APP_SERVER_HOME>` directory. For example, on Apache Tomcat this location is `<Application_Home>/WEB-INF/lib` directory
 - `sdk/client/java/lib/arcot/arcot-webfort-authentication.jar`
 - `sdk/client/java/lib/arcot/arcot-webfort-common.jar`
 - `sdk/client/java/lib/external/bcprov-jdk15-146.jar`
 - `sdk/client/java/lib/external/commons-pool-1.5.5.jar`
2. Copy the `webfort.authentication.properties` configuration file containing the server connection parameters from:
`<install_location>/arcot/sdk/client/java/properties`
to the appropriate location in your `<APP_SERVER_HOME>` directory. For example, on Apache Tomcat this location is `<Application_Home>/WEB-INF/classes/properties`.

Note: To know more about APIs and their initialization, refer to *CA AuthMinder Java Developer's Guide* and the AuthMinder Javadocs at `<install_location>/arcot/docs/webfort/Arcot-WebFort-7.1-authentication-sdk-javadocs.zip`.

Configuring Credential Management Java SDK

To configure Credential Management SDK for using in a J2EE Application:

1. Copy the listed JAR files from the following location:

`<install_location>/arcot`

to the appropriate location in your `<APP_SERVER_HOME>` directory. For example, on Apache Tomcat this location is `<Application_Home>/WEB-INF/lib` directory

- `sdk/client/java/lib/arcot/arcot-webfort-common.jar`
- `sdk/client/java/lib/arcot/arcot-webfort-issuance.jar`
- `sdk/client/java/lib/external/bcprov-jdk15-146.jar`
- `sdk/client/java/lib/external/commons-pool-1.5.5.jar`

2. Copy the `webfort.issuance.properties` configuration file containing the server connection parameters from:

`<install_location>/arcot/sdk/client/java/properties`

to the appropriate location in your `<APP_SERVER_HOME>` directory. For example, on Apache Tomcat this location is `<Application_Home>/WEB-INF/classes/properties`.

Note: To know more about Java APIs and their initialization, refer to *CA AuthMinder Java Developer's Guide* and the AuthMinder Javadocs at `<install_location>/arcot/docs/webfort/Arcot-WebFort-7.1-issuance-sdk-javadocs.zip`.

Working With AuthMinder Web Services

This section covers the following topics:

- [Introduction to AuthMinder Web Services](#) (see page 141)
- [Generating Client Code](#) (see page 141)

Introduction to AuthMinder Web Services

AuthMinder provides Web Services for managing users, organizations, and user credentials, and authenticating users and performing bulk operations.

The following table lists the WSDL documents that you can use to generate the Web Services client code to communicate with the AuthMinder Server. These WSDLs are available at the following location:

`<install_location>/arcot/wsdl/`

WSDL File	Description
<code>uds/ArcotOrganizationManagementSvc.wsdl</code>	Used to create and manage organizations in your setup.
<code>uds/ArcotConfigManagementSvc.wsdl</code>	Used to create and manage user account types.
<code>uds/ArcotUserManagementSvc.wsdl</code>	Used to create and manage users and user accounts.
<code>webfort/ArcotWebFortAdminSvc.wsdl</code>	Used to define AuthMinder configurations.
<code>webfort/ArcotWebFortIssuanceSvc.wsdl</code>	Used to manage user credentials.
<code>webfort/ArcotWebFortAuthSvc.wsdl</code>	Used to authenticate users.
<code>webfort/ArcotWebFortBulkOperationsSvc.wsdl</code>	Used to perform bulk operations such as, creating credentials, assigning and fetching OATH tokens that are available to the organizations.

Generating Client Code

Perform the following steps to generate the client code:

1. Stop the application server.
2. Navigate to the following location:
`<install_location>/arcot/wsdl/<required_folder>`
3. Use the WSDL file to generate the client code.
4. Restart the application server.
5. In a browser window, access the end point URLs to verify if the client can access the Web Service.

Note: Refer to the *CA AuthMinder Web Services Developer's Guide* for more information on how to access the end point URLs.

Enabling SSL Communication

AuthMinder supports Secure Socket Layer (SSL) between the AuthMinder Server and Java SDKs. The topic titled "Configuring SSL" in *CA AuthMinder Administration Guide* describes how to set the transport mode as SSL between the AuthMinder Server and its clients.

Chapter 7: Upgrading to Release 7.1

This section describes the steps to upgrade your existing release of AuthMinder to release 7.1. It includes the following topics:

- [Upgrade Overview](#) (see page 143)
- [Database Privileges Required for Upgrade](#) (see page 143)
- [Upgrading to 7.1](#) (see page 145)
- [\(In Error Scenario Only\) Reverting to Your Initial Setup \(IG\)](#) (see page 165)
- [Troubleshooting Issues During Upgrade](#) (see page 161)

Notes on Backward Compatibility

- The older Java SDK client will continue to work with the new installation of AuthMinder Server. It will *not* require any client code modification.
- Older WSDLs will continue to work with the new installation of AuthMinder Server. They will *not* require any client code modification.

Upgrade Overview

You can upgrade to release 7.1 from any of the following releases:

- 5.3.x or 5.4.x
- 6.x or 7.0

Important! If you have a release of AuthMinder that is not listed here, then you must apply the required patches to upgrade to one of these releases, and then proceed with the upgrade. Refer to the corresponding Release Notes for the patch upgrade instructions.

If you are upgrading from release 5.3.x or 5.4.x, first upgrade to release 6.0 and then upgrade to release 7.1. However, if you are upgrading from release 6.x or 7.0, then directly upgrade to release 7.1.

Database Privileges Required for Upgrade

The following table lists the database privileges that you must have for performing the database procedures related to upgrading to release 7.1:

Database Type	Upgrade Privileges	Run time Privileges
Oracle	CREATE TABLE	CREATE TABLE

Database Type	Upgrade Privileges	Run time Privileges
	CREATE ANY INDEX	DML Privileges
	CREATE ANY SEQUENCE	
	CREATE TABLESPACE (<i>for Reports</i>)	
	CREATE PROCEDURE	
	UNLIMITED TABLESPACE (<i>for Reports, optional</i>)	
	DROP TABLESPACE	
	ALTER ANY TABLE	
	ALTER TABLESPACE	
	DML Privileges (including CREATE SESSION privilege)	
MS SQL Server Note: The UserID must also have the Database role of ddladmin.	CREATE TABLE	CREATE TABLE
	CREATE INDEX	DML Privileges
	CREATE PROCEDURE	
	EXECUTE PROCEDURE	
	REFERENCES	
	ALTER TABLE	
	DML Privileges	
IBM DB2	CREATE TABLE	CREATE TABLE
	CREATE INDEX	DML Privileges
	CREATE SEQUENCE	
	CREATE TABLESPACE	
	CREATE TABLESPACE with AUTORESIZE = yes (<i>for Reports, optional</i>)	
	CREATE PROCEDURE (SQL - native)	
	ALTER TABLE	
	DML Privileges	

Upgrading to 7.1

Upgrading from 5.3.x or 5.4.x to 7.1 is a two-stage procedure. You first upgrade to 6.0 and then upgrade from 6.0 to 7.1. In contrast, if you are upgrading from 6.x or 7.0, you directly upgrade to 7.1.

Perform the following steps to upgrade to 7.1:

1. [Performing Pre-Upgrade Tasks](#) (see page 146)
2. If you are migrating from 5.3.x or 5.4.x, perform the steps described in the following sections:
 - [Migrating the Database to Release 6.0 for Arcot Common Components](#) (see page 148)
 - [Migrating the Database to Release 6.0 for AuthMinder Components](#) (see page 150)
 - [Performing Post-Migration Tasks for Release 6.0](#) (see page 151)
3. [Updating the securestore.enc File and Setting the TrustStore Password \(shared\)](#) (see page 152)
4. [Migrating the Database to Release 7.1 for Arcot Common Components](#) (see page 153)
5. [Migrating the Database to Release 7.1 for AuthMinder Components](#) (see page 155)
6. [Uninstalling the Existing Release of AuthMinder](#) (see page 156)
7. Depending on whether the existing AuthMinder deployment is on a single system or distributed system, perform the steps described in one of the following sections:
 - [Reinstalling AuthMinder on a Single System](#) (see page 157)
 - [Reinstalling AuthMinder on a Distributed System](#) (see page 158)
8. During upgrade, if there are any warnings during AuthMinder Server startup and if your transactions fail, then perform the steps listed in [\(In Error Scenario Only\) Reverting to Your Initial Setup \(IG\)](#) (see page 165).
9. [Performing Post-Upgrade Tasks \(shared\)](#) (see page 159)

Performing Pre-Upgrade Tasks

Important! For an AuthMinder deployment on a distributed system, perform the upgrade procedure on the system where the AuthMinder Server is installed.

Perform the following pre-upgrade tasks before you begin the upgrade procedure:

- Shut down the following servers:
 - AuthMinder Server
 - Application server where Administration Console and User Data Service are deployed.
- Ensure that you have JDK 1.5 or later installed on the system where you plan to upgrade.
- Ensure that the database on which you are performing the upgrade is available throughout the upgrade process.
- Ensure that the database on which you are performing the upgrade is not set up for replication. Disable database replication before the upgrade.
- Copy the contents of your existing ARCOT_HOME directory to a new directory.

Here, ARCOT_HOME refers to the base directory that contains the entire directory structure created by the existing AuthMinder 5.3.x and 5.4.x installations. Typically, ARCOT_HOME refers to *<install_location>/arcot/*.

Copy the entire contents of \$ARCOT_HOME to a new directory. This directory is referred to as ARCOT_HOME_BACKUP in this document.

You will need this directory to revert to the initial setup if there are any errors during upgrade.

- Open the \$ARCOT_HOME/conf/arcotcommon.ini file in a text editor, and then perform the following steps:
 - a. Ensure that the primary database details are correct. The upgrade tool uses the database that is configured in this file for the upgrade.
 - b. If you have configured a backup database, then you must disable the backup database by commenting out the lines containing the following properties in the arcot/db/backupdb section of the arcotcommon.ini file:
 - URL.1
 - AppServerConnectionPoolName.1
 - Username.1
 - c. Include the following section in the arcotcommon.ini file:

```
[arcot/crypto/device]
HSMDevice=S/W
```
 - d. Save and close the arcotcommon.ini file.

- On IBM DB2, if you are upgrading from release 6.x, then set the SYSTEM TEMPORARY tablespace page size to a minimum of 16K. Refer to database vendor documentation for more information.
- Back up the database containing the AuthMinder schema.
- Consult your DBA to configure your database according to the database volume requirements for AuthMinder.
- Ensure that you have sufficient database privileges to upgrade AuthMinder. For the complete list of privileges, see Prerequisites for Upgrading to AuthMinder 7.1.
- If you have stored your user details in an LDAP repository in the previous release, ensure that the LDAP server is available throughout the upgrade process.
- Ensure that the \$ARCOT_HOME environment variable is set to the directory where AuthMinder is installed.
- If you have registered plug-ins with this AuthMinder installation, then save the AuthMinder startup log. This file is available in the \$ARCOT_HOME/logs directory, and it contains details of the plug-ins. After the upgrade, recompile the plug-ins.

Migrating the Database to Release 6.0 for Arcot Common Components

Note: Perform the procedure that is described in this section only if you are upgrading from 5.3.x or 5.4.x. If you are upgrading from 6.x or 7.0, you need not migrate the database to 6.0 for Arcot common components.

Important! If you have installed CA RiskMinder with CA AuthMinder and you have completed the RiskMinder upgrade, then do *not* migrate the database for Arcot common components because this step has already been performed during the RiskMinder upgrade.

Migrate the database for the Arcot common components that are used in release 6.0.

Follow these steps:

1. Copy the upgrade directory to a temporary location on the system where you plan to upgrade.

This folder contains the following zip files that are applicable for this migration path:

- arcot-common-upgrade-0.x-1.0.zip
- arcot-webfort-upgrade-5.x-6.0.zip

2. Copy the arcot-common-upgrade-0.x-1.0.zip file to the \$ARCOT_HOME directory:
3. Extract the contents of the arcot-common-upgrade-0.x-1.0.zip file in this directory.
4. Navigate to the following directory:
\$ARCOT_HOME/dbscripts/<db_type>/
5. Run the arcot-db-config-for-common-1.0.sql script.

Note: For Microsoft SQL server, if you run the database script from the command line using SQLCMD, then specify the -I option to set the QUOTED_IDENTIFIER connection option to ON.

6. Navigate to the following directory:
\$ARCOT_HOME/dbscripts/<db_type>/upgrade-scripts/
7. Run the arcot-upgrade-for-common-1.0.sql script.

Note: For Microsoft SQL server, if you run the database script from the command line using SQLCMD, then specify the -I option to set the QUOTED_IDENTIFIER connection option to ON and the -x option to disable variable substitution.

8. Copy the JDBC JAR that is compatible with the database that you are using to the following directory:
\$ARCOT_HOME/java/lib
9. Copy the libArcotAccessKeyProvider.so file to <JAVA_HOME used by App Server>/jre/bin. In AuthMinder release 6.x, the libArcotAccessKeyProvider.so file is available in java/ext/<platform name>/<32bit or 64 bit>.
10. Copy the \$ARCOT_HOME/java/lib/arcot-crypto-util.jar file

to
JAVA_HOME-used-by-App-Server/jre/lib/ext

11. Set and export the LD_LIBRARY_PATH to the directory where libArcotAccessKeyProvider.so is copied.
12. Change your working directory to:
\$ARCOT_HOME/tools/upgrade
13. Ensure that you have the permission to run the upgrade-common.sh upgrade tool.
14. Run the upgrade tool by entering the following command at the prompt:
upgrade-common.sh
15. Check the log file \$ARCOT_HOME/logs/upgrade-common.log to ensure that the common database migration was successful.

Important! While migrating the database, if you encounter errors due to a step that you have performed incorrectly, restore the database backup that you created earlier. After you verify that the database is correctly restored, retry the database migration procedure.

Migrating the Database to Release 6.0 for AuthMinder Components

Important! Perform the procedure that is described in this section only if you are upgrading from 5.3.x or 5.4.x. If you are upgrading from 6.x or 7.0, do not migrate the database to 6.0 for AuthMinder.

After you migrate the database for Arcot common components, migrate the database to the release 6.0 state for AuthMinder components.

Follow these steps:

1. Copy the `arcot-webfort-upgrade-5.x-6.0.zip` file to the `$ARCOT_HOME` directory.
2. Unzip the `arcot-webfort-upgrade-5.x-6.0.zip` file in this directory.
3. Navigate to the following directory:
`$ARCOT_HOME/arcot-webfort-upgrade-5.x-6.0/dbscripts/<db_type>`
4. Run the scripts in the *following* order:
 - a. `arcot-db-config-for-webfort-6.0.sql`
 - b. `upgrade-for-webfort.sql`
5. Open a command window.
6. Change your working directory to:
`$ARCOT_HOME/arcot-webfort-upgrade-5.x-6.0/tools/<platform_name>`
7. At the prompt, run the following command:
`wfupgrade -migrate`

The preceding command loads the OpenSSL CA and the domain key that is used for the ArcotID, and migrates the configuration data from older database tables to the release 6.0 tables.

The `wfupgrade` tool generates the `webfort-6.0-upgrade.log` file in the `$ARCOT_HOME/logs/` directory.

Important! Ensure that you run the `wfupgrade` upgrade tool from the system where the AuthMinder Server is installed.

8. Open the `webfort-6.0-upgrade.log` file in a text editor and ensure that it does not contain any FATAL and WARNING messages.

Performing Post-Migration Tasks for Release 6.0

After you migrate the database for AuthMinder, remove the earlier AuthMinder schema from the database and then perform some other post-migration steps.

Follow these steps:

1. Navigate to the following location:
`$ARCOT_HOME/dbscripts/<db_type>/upgrade-scripts`
2. Run the `arcot-post-upgrade-for-common-1.0.sql` script file.

Important! If you have installed both CA AuthMinder and CA RiskMinder, then run this script only after you have upgraded both CA AuthMinder and CA RiskMinder.

This script ensures that the following changes are applied:

- The user ID for Master Administrator is changed from MASTER_ADMIN to MASTERADMIN.
 - The password for the MASTERADMIN account is master1234!.
 - The MASTERADMIN account belongs to the MASTERADMIN organization. This is useful when you filter reports.
 - The Administrators group is configured with WebFort User Password authentication. Administrators belonging to this group must continue to use the same user name and password.
 - Group2 is the initial Default Organization.
3. Navigate to the following location:
`$ARCOT_HOME/dbscripts/<db_type>`
 4. Run the following AuthMinder-specific script available in this directory to drop the AuthMinder schema:
 - For AuthMinder 5.3.x: `ArcotWebFort.drop.sql`
 - For AuthMinder 5.4.x: `drop-webfort-<version_number>.sql`
 5. For the 5.3.x or 5.4.x Java SDKs to work with AuthMinder 6.0:
 - a. Remove the existing AuthMinder Java SDK files.
 - b. Update your applications with the following new files:
 - `$ARCOT_HOME/arcot-webfort-upgrade-5.x-6.0/sdk/java/lib/arcot/arcot_core.jar`
 - `$ARCOT_HOME/arcot-webfort-upgrade-5.x-6.0/sdk/java/lib/arcot/arwf_isuance_sdk.jar`
 - `$ARCOT_HOME/arcot-webfort-upgrade-5.x-6.0/sdk/java/lib/arcot/arwf_auth_sdk.jar`
 - All files available in `$ARCOT_HOME/arcot-webfort-upgrade-5.x-6.0/sdk/java/lib/external/`

- c. Migrate the existing configuration that is saved in the authentication.properties file to the following file:
\$ARCOT_HOME/arcot-webfort-upgrade-5.x-6.0/sdk/java/properties/webfort-5.4.1.authentication.properties
 - d. Migrate the existing configuration that is saved in the issuance.properties file to the following file:
\$ARCOT_HOME/arcot-webfort-upgrade-5.x-6.0/sdk/java/properties/webfort-5.4.1.issuance.properties
6. Perform the following steps to verify that the migration of the database to the Release 6.0 state was successful:
 - a. Navigate to the following location:
<install_location>/arcot/logs
 - b. Open the arcotwebfortstartup.log file in any editor, and locate the following lines in the file:
INSTANCE_VER.....: [6.0]
Arcot WebFort Authentication Service READY
 - c. Ensure that the log file does not contain any FATAL or WARNING messages.

Updating the securestore.enc File and Setting the TrustStore Password

Perform the steps that are described in this section if application server connection pooling was being used or if SSL has been configured for the connection with the database.

Follow these steps:

1. If application server connection pooling was being used in your existing AuthMinder deployment, then update the database details in the securestore.enc file as follows:
 - a. Change the working directory to the following location where DBUtil is available:
\$ARCOT_HOME/sbin
 - b. Run the following command for the primary database:
DBUtil -pi <DB_username> <DB_password>
2. If SSL has been configured for the connection with the database, then update the database details in the securestore.enc file as follows:
 - a. Change the working directory to the following location where DBUtil is available:
\$ARCOT_HOME/sbin
 - b. If the database is enabled for SSL communication, then set the truststore password, as follows:
DBUtil -pi TrustStorePath.1 <truststore-password>

Migrating the Database to Release 7.1 for Arcot Common Components

Important! If you have installed CA RiskMinder with AuthMinder and you have completed the RiskMinder upgrade, then do *not* migrate the database for Arcot common components because this step has already been performed during the RiskMinder upgrade.

Migrate the database for the Arcot common components that are used in release 7.1.

Follow these steps:

1. Copy the upgrade directory to a temporary location on the system where you plan to upgrade.

This folder contains the following zip files that are applicable for this migration path:

- arcot-common-upgrade-1.0.x-2.0.zip
 - arcot-webfort-upgrade-6.x-or-7.0-7.1.zip
2. Copy the arcot-common-upgrade-1.0.x-2.0.zip file to the following directory: \$ARCOT_HOME
 3. Extract the contents of the arcot-common-upgrade-1.0.x-2.0.zip file in this directory.
 4. Navigate to the \$ARCOT_HOME/tools/common/upgrade/ directory.
 5. Extract the contents of the arcot-common-db-upgrade.zip file in this directory.
 6. Copy the database jar file corresponding to your database to the \$ARCOT_HOME/tools/common/upgrade/lib directory with the same name, as follows:
 - Oracle: ojdbc.jar
 - Microsoft SQL Server: sqljdbc.jar
 - DB2: db2jcc.jar

Note: For Oracle and DB2, use the JDBC jar version that is applicable to your database. For Microsoft SQL Server, use sqljdbc4.0 (Microsoft JDBC Driver 4.0 for SQL Server).

7. Locate the JAVA_HOME used by the existing installation. When you run the upgrade tool, ensure that you use the same JAVA_HOME or the JAVA_HOME from a later, supported version.
8. Ensure that the libArcotAccessKeyProvider.so file is configured for your application server. This file is in the \$ARCOT_HOME/native/<platform-name>/32bit-or-64bit/ directory.
9. Set and export the LD_LIBRARY_PATH to the directory where the libArcotAccessKeyProvider.so file is copied.
10. Change your working directory to:

\$ARCOT_HOME/tools/common/upgrade/

11. Run the arcot-common-upgrade-framework.jar file by using the following command:

```
java [JVM_Options] -jar arcot-common-upgrade-framework.jar
[--log-file <log-file-name>] [--log-level
<log-level>][--commit-batch-size <batch_size>] [--product-name
common][--prompt][--mst]
```

The following table describes the options that are supported by this JAR file:

Option	Description
JVM-Options	<p>The following JVM options are required only if LDAP organizations are configured:</p> <ul style="list-style-type: none"> ■ -Xmx1024m: Sets the maximum heap size to 1 GB. If more than 1,00,000 users exist in the configured LDAP, then it is highly recommended that you increase the heap size to 2048m (2 GB). ■ -Dcom.arcot.ldap.migration.timeout=<duration>: The time duration, in minutes, after which the migration of the LDAP organization is marked as failed. The LDAP migration timeout for 1,00,000 users is approximately 240 minutes or 4 hours. However, this would depend on the type of hardware configuration being used. The default value of this parameter is 240 minutes.
log-file	<p>Specifies the path to the log file:</p> <ul style="list-style-type: none"> ■ If you do not provide any value, the arcot_common_upgrade.log file is created in the \$ARCOT_HOME/logs/ directory. ■ If you provide an absolute path, the log file is created at the given location. ■ If you provide a file name, the log file is created in \$ARCOT_HOME with the given file name.
-log-level	Specifies the log level. If you do not provide any value, the upgrade log level is set to INFO.
commit-batch-size	Specifies the number of transactions to be issued to the database before a COMMIT statement is issued.
product-name	Specifies the name of the product that has to be migrated. The default value is common.

Option	Description
prompt	Prompts whether to proceed further after each phase is completed successfully. You can choose to run the tool later to continue from where it stopped. If this option is not specified, the upgrade tool runs without any prompting until the upgrade process is completed.
mst	Refers to the Monitoring Sleep Time. If you specify this option, the upgrade tool prints diagnostic messages describing the progress made during upgrade after sleeping for the specified duration (in minutes.) The default value is two minutes.

1. Check the log file (default file is \$ARCOT_HOME/logs/arcot_common_upgrade.log) to make sure that the common database upgrade is successful.

The upgrade tool also prints verification information.

Important! While migrating the database, if you encounter errors due to a step that you have performed incorrectly, restore the database backup that you created earlier. After you verify that the database is correctly restored, retry the database migration procedure.

Migrating the Database to Release 7.1 for AuthMinder Components

After you upgrade the database for Arcot common components, upgrade the database to the release 7.1 state for AuthMinder.

Follow these steps:

1. Unzip the arcot-webfort-upgrade-6.x-or-7.0-7.1.zip in \$ARCOT_HOME directory.
2. Change your working directory to the following directory:
\$ARCOT_HOME/arcot-webfort-upgrade-6.x-or-7.0-7.1.zip/tools/<platform_name>

3. At the prompt, run the following command:
wfupgrade -migrate

This command migrates the 6.x or 7.0 configuration data from older database tables to the 7.1 tables.

The wfupgrade tool generates the webfort-7.1-upgrade.log file in the \$ARCOT_HOME/logs directory.

Important! Ensure that you run the wfupgrade upgrade tool from the system where the AuthMinder Server is installed.

4. Open the log file in a text editor and ensure that it does not contain any FATAL and WARNING messages.

Uninstalling the Existing Release of AuthMinder

Uninstall the existing release of AuthMinder, and undeploy the Administration Console and UDS that are installed on the application server.

Note: If the instructions given in this section do not match the uninstallation options available in your existing AuthMinder installation, follow the uninstallation instructions that are given in the installation guide for your existing release of AuthMinder.

Follow these steps:

1. Uninstall the existing release of AuthMinder as follows:
 - a. Ensure that the following components have been shut down gracefully:
 - AuthMinder Server
 - Any application servers where other AuthMinder components are deployed
 - b. Ensure that the Administration Console is not open.
 - c. Ensure that all INI and other files that are related to AuthMinder configuration are closed.
 - d. Navigate to the following directory:
`$ARCOT_HOME/arcot/Uninstall_Arcot WebFort`
 - e. Run the installer using the following command:
`sh Uninstall Arcot WebFort`
You are prompted to specify uninstallation options.
 - f. Enter 1 in the wizard window to specify that you want all features and components to be removed.
 - g. Press Enter to confirm and continue with the uninstallation.
 - h. Press Enter to exit the wizard and complete the uninstallation.
 - i. Delete any files that are left over in the ARCOT_HOME directory.
2. Undeploy the Administration Console and User Data Service Web applications from the application server and shut down the application server gracefully. Then, refresh the application server cache.
3. Ensure that the `.com.zerog.registry.xml` file has been deleted. This hidden file is created during the installation. The location of this file depends on the user account that was used to install AuthMinder:
 - If you had installed AuthMinder as the root user, then this file is located in the `/var` directory.
 - If you had installed AuthMinder as any other user, then this file is located in the user's HOME directory.

Reinstalling AuthMinder

Depending on whether you had earlier deployed AuthMinder on a single system or distributed system, perform the tasks described in one of the following sections:

- [Reinstalling AuthMinder on a Single System](#) (see page 157)
- [Reinstalling AuthMinder on a Distributed System](#) (see page 158)

Reinstalling AuthMinder on a Single System

Perform the tasks described in the following sections to reinstall AuthMinder on a single system:

Important! The information given in these sections apply to both a fresh installation of AuthMinder and an upgrade of an existing AuthMinder installation. Some of the steps mentioned in these sections do not apply in an upgrade scenario. For example, MySQL-related steps do not apply because MySQL is supported only from release 7.1 onward.

Important! Use the database that you had migrated earlier during the upgrade operation. In addition, install AuthMinder at the same location where the older release was installed. If you install in a different location, the AuthMinder Server will *not* start.

1. Performing Complete Installation
2. Verifying the Database Setup
3. Preparing Your Application Server
4. Deploying Administration Console
5. Verifying the Administration Console
6. Logging In to Administration Console
7. Starting WebFort Server
8. Verifying the Installation

Note: If there are any warnings during Server startup or while verifying the installation, and if your transactions fail, then the upgrade has not been performed successfully. You can revert to your initial setup by following the steps listed in "(In Error Scenario Only) Reverting to Your Initial Setup" in the CA AuthMinder Installation and Deployment Guide for UNIX Platforms.

9. Deploying User Data Service
10. [Deploying Sample Application](#) (see page 89)
11. Using Sample Application
12. Post-Installation Checklist

Reinstalling AuthMinder on a Distributed System

Perform the tasks described in the following sections to reinstall AuthMinder on a distributed system:

Important! The information given in these sections apply to both a fresh installation of AuthMinder and an upgrade of an existing AuthMinder installation. Some of the steps mentioned in these sections do not apply in an upgrade scenario. For example, MySQL-related steps do not apply because MySQL is supported only from release 7.1 onward.

Important! Use the database that you had migrated earlier during the upgrade operation. In addition, install AuthMinder at the same location where the older release was installed. If you install in a different location, the AuthMinder Server will *not* start.

1. Installing on the First System
2. Verifying the Database Setup
3. [Preparing Your Application Server](#) (see page 110)
4. Deploying Administration Console
5. Verifying the Administration Console
6. Logging In to Administration Console
7. Starting WebFort Server
8. [Verifying the Installation](#) (see page 123)
9. Deploying User Data Service
10. Installing on the Second System
11. [Deploying Sample Application](#) (see page 128)
12. [Configuring Sample Application for Communication with WebFort Server](#) (see page 128)
13. Using Sample Application
14. Post-Installation Checklist

Performing Post-Upgrade Tasks

Perform the following post-upgrade task:

- If you have disabled the backup database, then enable it by editing the `arcot/db/backupdb` section of the `$ARCOT_HOME/conf/arcotcommon.ini` file, and synchronize the backup database with the primary database. If you disable database replication before the upgrade, then enable replication after the upgrade.
- If you had registered plug-ins with this AuthMinder installation before the upgrade, then recompile the plug-ins. Ensure that the names of the recompiled files are the same as before. Use the AuthMinder startup log that you had saved before the upgrade to determine details of the plug-ins.
- If you require multi-byte character or internationalization support in AuthMinder and if your database does not currently support multi-byte data, then migrate the database to a character set that supports multi-byte data. For more information, see "Configuring Database Server" in the *CA AuthMinder Installation and Deployment Guide for UNIX Platforms*.

If your upgrade path is from 5.3.x or 5.4.x, then perform the following configurations using Administration Console.

- Create new authentication policies at the global-level for all the previous authentication configurations.

See the Configuring Policies and Profiles section in chapter, "Managing Global WebFort Configurations" of *CA AuthMinder Administration Guide* for more information.
- Re-create the RADIUS configuration.

See "Configuring AuthMinder for RADIUS" in the *CA AuthMinder Administration Guide* for more information.
- Set up the following AuthMinder Server instance configurations:
 - Database connectivity settings
 - Log file settings
See "Setting Up Server Instances" in the *CA AuthMinder Administration Guide* for more information.
- Set up thread configurations per protocol.

See "Configuring Communication Protocols" in the *CA AuthMinder Administration Guide* for more information.
- Create ASSP configurations.

See "Configuring ASSP" in the *CA AuthMinder Administration Guide* for more information.
- (Optional) Configure ArcotID grace period. To do so, set the **Allow Successful Authentication** field in the ArcotID policy.

See "Configuring Policies and Profiles" in the *CA AuthMinder Administration Guide* for more information.

- (Optional) Enable caller verification for QnA credential. To do so, set the **Enable Caller Verification** field in the QnA policy.

See "Configuring Policies and Profiles" in chapter, "Managing Global WebFort Configurations" of *CA AuthMinder Administration Guide* for more information.

Troubleshooting Issues During Upgrade

This section describes troubleshooting steps that you can apply to resolve errors that you might face while upgrading AuthMinder.

Problem:

The upgrade tool fails with the following error:
Error Occured: IO exception while parsing,
\$ARCOT_HOME/tools/common/upgrade/xml/arcot-common-upgrade-meta-data.xml

Cause:

The upgrade tool was not able to find the arcot-common-upgrade-meta-data.xml file.

Solution:

Check if the arcot-common-upgrade-meta-data.xml file exists in \$ARCOT_HOME/tools/common/upgrade/xml. This error can commonly occur when the arcot-common-db-upgrade.zip file is not extracted using the **Extract To Here** option.

Problem:

The upgrade tool fails with the following error:
Internal Error: Could not initialize upgrade tool. Error:: Cannot load JDBC driver class 'oracle.jdbc.driver.OracleDriver' Error Occured: Upgrade Initialization Error:oracle.jdbc.driver.OracleDriver

Cause:

The upgrade tool could not find the JDBC library to connect to the database.

Solution:

Check whether the JDBC library is copied to the \$ARCOT_HOME/tools/common/upgrade/lib directory.

If the JDBC library is already copied, then check whether the name of the JDBC jar file is correctly specified, as described in the procedure to migrate the database to release 6.0 for Arcot common components. Also, check if the JDBC jar file corresponds to the database configured in the arcotcommon.ini file against the DbType parameter.

Problem:

The upgrade tool fails with the following error:
FATAL: ARCOT_HOME Environment Variable Not Set

Cause:

\$ARCOT_HOME is not set.

Solution:

Set the \$ARCOT_HOME environment variable and run the upgrade tool.

Problem:

The upgrade tool fails with the following error:
Error Occured: Upgrade Initialization Error:Could not create DBService instance"

Solution:

Check if the user name and password are configured correctly in the arcotcommon.ini and securestore.enc files, respectively.

Problem:

The upgrade tool fails with the following error:
Error Occured: Upgrade Initialization Error:Io exception: The Network Adapter could not establish the connection"

Solution:

Check if the JDBC URL is correct and points to the correct database. Ensure that the database is up and running.

Problem:

The upgrade tool fails with the following error:
javax.crypto.BadPaddingException: Given final block not properly padded

Cause:

The key label used to encrypt the data is not the same as the key used for decryption.

Solution:

Ensure that the master key label used to encrypt data in the database is the same as the key label used by the upgrade tool to decrypt data. The master key label is stored in the `securestore.enc` file in the `$ARCOT_HOME/conf` folder.

Problem:

The upgrade tool fails with the following or similar error:
"java.sql.SQLException: ORA-20010: -1031-ORA-01031: insufficient privileges"

Cause:

The database user configured in the `arcotcommon.ini` file does not have sufficient database privileges to carry out the database upgrade.

Solution:

Ensure that the administrator performing the upgrade has the required database privileges. Installation time privileges are applicable to upgrade also.

Problem:

The upgrade tool fails with the following or similar error:
"ORA-01536: space quota exceeded for tablespace"

Cause:

The database user configured in the `arcotcommon.ini` file has exhausted the space quota in the tablespace.

Solution:

The DBA must increase the quota for the user. You must restart the upgrade tool after you reimport the pre-upgrade data.

Problem:

After the upgrade process, the Administration Console fails to start and returns the following error:

```
ERROR : taglib.tiles.InsertTag : ServletException in
'/WEB-INF/jsp/dynamic/navbar_GA.jsp': File
"/WEB-INF/jsp/dynamic/navbar_GA.jsp"
```

Cause:

The Work folder of the application server where the Administration Console is deployed still contains the cache of the earlier Administration Console version.

Solution:

Clear the Work folder of the application server where the Administration Console is deployed and restart the application server.

Problem:

After the upgrade process, an administrator belonging to the LDAP repository can no longer log in to the Administration Console.

Cause:

The administrator might be disabled in LDAP.

Solution:

Ensure that the administrator is not disabled in LDAP. Disabled administrators are not allowed to log in to the Administration Console.

Problem:

After the upgrade process, for the IBM DB2 database the AuthMinder Server-specific reports fail to generate with the following error:
DB2 SQL Error: SQLCODE=-1585, SQLSTATE=54048, SQLERRMC=null

Cause:

The SYSTEM TEMPORARY tablespace page size might be set to the default value of 4K, which is not sufficient for report queries that involve columns exceeding 4K size.

Solution:

Set the SYSTEM TEMPORARY tablespace page size to a minimum of 16K. Refer to database vendor documentation for more information.

(In Error Scenario Only) Reverting to Your Initial Setup

During upgrade, if there are any warnings during Server startup and if your transactions fail, then you might want to revert your setup to the initial stage.

To revert to the initial setup:

1. Uninstall AuthMinder 7.1.
Refer to chapter, "[Uninstalling AuthMinder](#)" (see page 167) for more information.
2. Install the version that you want to revert to. For example, 5.3.x, 5.4.x, or 6.x.
Refer to the *CA AuthMinder 7.1 Installation and Deployment Guide* that is shipped with the corresponding version.
3. Navigate to the location where ARCOT_HOME_BACKUP directory is available.
4. Copy the contents of ARCOT_HOME_BACKUP to your current \$ARCOT_HOME.
5. Deploy the Web components, such as Administration Console.
6. Start AuthMinder Server and the application server.
7. Test the installation.

Chapter 8: Uninstalling AuthMinder

This chapter guides you through the steps for uninstalling AuthMinder and related components. The chapter covers the following sections:

- [Uninstalling AuthMinder Schema](#) (see page 167)
- [Uninstalling AuthMinder](#) (see page 168)
- [Post-Uninstallation Steps](#) (see page 169)

Uninstalling AuthMinder Schema

To uninstall AuthMinder schema from the database:

1. Based on the database type you are using, navigate to the folder:
 - (for MS SQL) `<install_location>/arcot/dbscripts/mssql`
 - (for Oracle) `<install_location>/arcot/dbscripts/oracle`
 - (for DB2) `<install_location>/arcot/dbscripts/db2`
 - (for MySQL) `<install_location>/arcot/dbscripts/mysql`
2. Run the scripts *in the following order*:
 - a. `drop-webfort-7.1.sql`
 - b. `drop-arcot-common-2.0.sql`

This deletes all the database tables.

Uninstalling AuthMinder

Perform the following tasks to uninstall AuthMinder:

1. Stop AuthMinder Server.
2. Ensure that INI files are not open in any editor.
3. Navigate to the following directory:
prompt> `cd <install_directory>/arcot/Uninstall_Arcot WebFort`
4. Run the installer using the following command:
prompt> `sh Uninstall Arcot WebFort`
You are prompted to specify uninstallation options.
5. Select the type of uninstallation:
 - **1-Completely remove all features and components:** Select this option if you want to uninstall *all* components of the AuthMinder on the current system.
 - **2-Choose specific features that were installed by InstallAnywhere:** Select this option if you want to uninstall only *selected* components of the AuthMinder on the current system.
6. Press **Enter** to continue.

If you selected to uninstall all components, then proceed to Step 8.

If you selected to uninstall selected components, then the Choose Product Features screen appears.

The Choose Product Features screen appears.

7. **(For Uninstalling Specific Components Only)** This screen displays the AuthMinder components that are installed on the current system. Enter the component numbers (separated by comma) and press **Enter**.

Important! To Uninstall Specific Features, you must follow the reverse sequence in which you performed the installation. For example, if you have installed AuthMinder Authentication Server followed by Administration Console, then you have to first uninstall Administration Console and then AuthMinder Authentication Server.

The Uninstall Complete screen appears at the end of successful uninstallation.

8. Press **Enter** to exit the wizard.

Post-Uninstallation Steps

The following are the post-uninstallation steps:

1. Delete the <install_location>/arcot folder.
2. Uninstall the following Web applications from the application server:
 - arcotadmin - Administration Console
 - arcotuds - User Data Service
 - webfort-7.1-sample-application - Sample Application
3. Ensure that the .com.zerog.registry.xml file has been deleted. This hidden file is copied during the installation. The location of this file depends on the user account that was used to install AuthMinder:
 - If you had installed AuthMinder as the root user, then this file is located in the /var directory.
 - If you had installed AuthMinder as any other user, then this file is located in the user's HOME directory.

Note: If you have performed distributed-system deployment, then locate these files on the system where you have deployed the particular application.

Appendix A: AuthMinder File System Structure

This appendix provides information about the location of all the files that are installed by the AuthMinder installer.

Important! Do not delete any of the files that are installed by AuthMinder.

- [Issuance and Authentication AuthMinder Server Files](#) (see page 171)
- [Administration Console Files](#) (see page 173)
- [User Data Service Files](#) (see page 175)
- [Authentication Java SDK Files](#) (see page 177)
- [Issuance Java SDK Files](#) (see page 177)
- [Web Services Files](#) (see page 178)
- [Plug-In SDK](#) (see page 179)

Issuance and Authentication AuthMinder Server Files

The following table lists the folder location of the files that are used by AuthMinder Server:

Folder	File Description
<code><install_location>/</code>	Contains the arcotkey and wfkey files. These file are used by the installer to detect previously installed Arcot products. If you delete these file, then the installer will not be able to detect previously installed Arcot products, and will allow new installations to be performed in any location. As a result, the installer will not be able to ensure the same destination folder for multiple Arcot products and components, in which case, the products (or components) might not work, as expected. This file has no impact on patches and upgrade. The wfdbkey and wftpkey files are the reference key files for database and third-party JAR files. These keys files are used during upgrade.
<code><install_location>/arcot/bin</code>	Contains webfortserver script that calls the server binary in the sbin folder.

Folder	File Description
<install_location>/arcot/conf	Contains the following configuration files: <ul style="list-style-type: none"> ■ arcotcommon.ini (see page 181) ■ securestore.enc: File that contains the keys used for encrypting the sensitive data.
<install_location>/arcot/db/scripts	Contains the SQL scripts to create the AuthMinder schema. See " Running Database Scripts " (see page 70) for information on the database scripts.
<install_location>/arcot/logs	Contains the AuthMinder Server log file.
<install_location>/arcot/odbc32v70wf	Contains the branded DataDirect ODBC libraries for all the databases supported by AuthMinder.
<install_location>/arcot/sbin	Contains library files and following executables that are required by administrators: <ul style="list-style-type: none"> ■ arwfutil - It is used to shut down and refresh the AuthMinder Server. ■ arwfserver.real - Contains a symbolic link to the script that sets the configuration and executes arwfutil binary. ■ arfwwatchdog - This tool monitors the server health and also starts the server if it stops. ■ arwfenv- Script that is used to set the environment variables.
<install_location>/samples/xml/webfort	Contains the following files used by the ArcotWebFortBulkOperationsSvc.wsdl file: <ul style="list-style-type: none"> ■ oath-token-assign.xml ■ oath-token-upload.xml
<install_location>/arcot/sdk/server/plugin/c/docs	Contains the following plug-in interface file: <ul style="list-style-type: none"> ■ webfort-plugin-cpp-interface.html
<install_location>/arcot/sdk/server/plugin/c/include/webfort/vas	Contains the following SDK plug-in header files: <ul style="list-style-type: none"> ■ wf-common-interface.h ■ wf-common-interface.hpp ■ wf-plugin-interface.h
<install_location>/arcot/sdk/server/plugin/c/lib	Contains the arwfpluginsdk.so containing the plug-in libraries.

Folder	File Description
<install_location>/arcot/tools/<platform_name>	Contains the following file: <ul style="list-style-type: none"> ■ DBUtil: Tool for editing securestore.enc, which stores the database information needed to connect to the AuthMinder database in the encrypted format.
<install_location>/arcot/Uninstall_Arcot_WebFort	Contains the executable required to uninstall AuthMinder.

Administration Console Files

The following table lists the folder location of the files that are used by Administration Console:

Folder	File Description
<install_location>/	Contains the arcotkey and wfkey files. These files are used by the installer to detect previously installed Arcot products. If you delete these files, then the installer will not be able to detect previously installed Arcot products, and will allow new installations to be performed in any location. As a result, the installer will not be able to ensure the same destination folder for multiple Arcot products and components, in which case, the products (or components) might not work, as expected. This file has no impact on patches and upgrade. The wfdbkey and wftpkey files are the reference key files for database and third-party JAR files. These key files are used by the installer during upgrade.
<install_location>/arcot/conf	Contains the following configuration files: <ul style="list-style-type: none"> ■ arcotcommon.ini (see page 181) ■ adminserver.ini (see page 190) ■ securestore.enc: File that contains the keys used for encrypting the sensitive data.
<install_location>/arcot/conf/resourcebundles	Contains the following message properties files: <ul style="list-style-type: none"> ■ arcot-common-message_en_US.properties ■ arcot-uds-message_en_US.properties
<install_location>/arcot/dbscripts	Contains the SQL scripts to create the Administration Console schema. See " Running Database Scripts " (see page 70) for information on the database scripts.

Folder	File Description
<install_location>/arcot/java/lib	Contains the WAR and JAR files required by the Administration Console Framework: <ul style="list-style-type: none"> ■ adminframework.jar ■ adminframework.war ■ arcot-common.jar ■ arcot-crypto-util.jar ■ bcprov-jdk15-146.jar
<install_location>/arcot/java/lib/sdk	This is an empty directory. You have to include the JAR files that have to be used by the bundlemanager tool to create the arcotadmin.war and arcotuds.war files.
<install_location>/arcot/java/webapps	Contains the arcotadmin.war file required to deploy Administration Console.
<install_location>/arcot/logs	Contains the Administration Console log file.
<install_location>/arcot/java/native/<platform_name>/<32 or 64 bit>	Contains the libArcotAccessKeyProvider.so file that is used to read the contents of securestore.enc file.
<install_location>/arcot/odbc32/v70wf	Contains the branded DataDirect ODBC libraries for all the databases supported by AuthMinder.
<install_location>/arcot/resourcepacks	Contains the following AuthMinder and Administration Console packages: <ul style="list-style-type: none"> ■ bundle_webfort.zip ■ bundler_adminconsole.zip
<install_location>/arcot/resourcepacks/i18n//	Contains the following properties file to update the account status value: <ul style="list-style-type: none"> ■ framework-useraccount-status.properties
<install_location>/arcot/tools/common/	Contains the following subdirectories: <ul style="list-style-type: none"> ■ The arreporrtool subdirectory contains the report tool that enables you to export reports. ■ The bundlemanager subdirectory contains the files required by the Administration Console Resource pack.
<install_location>/arcot/tools/<platform_name>	Contains the following file: <ul style="list-style-type: none"> ■ DBUtil: Tool for editing securestore.enc, which stores the database information needed to connect to the AuthMinder database in the encrypted format.

Folder	File Description
<install_location>/arcot/Uninstall_Arcot WebFort	Contains the uninstallation-related files.

User Data Service Files

The following table lists the folder location of the files that are used by User Data Service:

Folder	File Description
<install_location>/	Contains the arcotkey file. This file is used by the installer to detect previously installed Arcot products. If you delete these file, then the installer will not be able to detect previously installed Arcot products, and will allow new installations to be performed in any location. As a result, the installer will not be able to ensure the same destination folder for multiple Arcot products and components, in which case, the products (or components) might not work, as expected. This file has no impact on patches and upgrade. The wfdbkey and wftkey files are the reference key files for database and third-party JAR files. These keys files are used during upgrade.
<install_location>/arcot/conf	Contains the following configuration files: <ul style="list-style-type: none"> ■ arcotcommon.ini (see page 181) ■ udsserver.ini (see page 192) ■ securestore.enc: File that contains the keys used for encrypting the sensitive data.
<install_location>/arcot/conf/resourcebundles	Contains the following message properties files: <ul style="list-style-type: none"> ■ arcot-common-message_en_US.properties ■ arcot-uds-message_en_US.properties
<install_location>/arcot/dbscripts	Contains the SQL scripts to create the Administration Console schema. See " Running Database Scripts " (see page 70) for information on the database scripts.
<install_location>/arcot/docs/uds	Contains the arcot-uds-2_0-wsdl-docs.zip file, which contains the WSDL documents for UDS.

Folder	File Description
<install_location>/arcot/java/lib	Contains the WAR and JAR files required by the User Data Service: <ul style="list-style-type: none"> ■ arcot-common.jar ■ arcot-crypto-util.jar ■ arcot-euds.jar ■ bcprov-jdk15-146.jar ■ udsframework.war
<install_location>/arcot/java/lib/sdk	This is an empty directory. You have to include the JAR files that have to be used by the bundlemanager tool to create the arcotadmin.war and arcotuds.war files.
<install_location>/arcot/java/webapps	Contains the arcotuds.war file required to deploy and User Data Service.
<install_location>/arcot/logs	Contains the UDS log file.
<install_location>/arcot/java/native/<platform_name>/<32 or 64 bit>	Contains the libArcotAccessKeyProvider.so file that is used to read the contents of securestore.enc file.
<install_location>/arcot/odbc32/v70wf	Contains the branded DataDirect ODBC libraries for all the databases supported by AuthMinder.
<install_location>/arcot/tools/common/	Contains the following subdirectory: <ul style="list-style-type: none"> ■ The uds-monitor subdirectory contains the tool that enables you to monitor UDS.
<install_location>/arcot/tools/<platform_name>	Contains the following file: <ul style="list-style-type: none"> ■ DBUtil: Tool for editing securestore.enc, which stores the database information needed to connect to the AuthMinder database in the encrypted format.
<install_location>/arcot/Uninstall_Arcot WebFort	Contains the uninstallation-related files.
<install_location>/arcot/wsdl/uds	Contains the following documents: <ul style="list-style-type: none"> ■ WSDL files that the Web services client uses to generate the code: <ul style="list-style-type: none"> ArcotConfigManagementSvc.wsdl ArcotOrganizationManagementSvc.wsdl ArcotUserManagementSvc.wsdl ■ XSD files that are used by Web services to perform bulk operations: <ul style="list-style-type: none"> ArcotUserSchema.xsd

Authentication Java SDK Files

The following table lists the folder location of the files that are used by Authentication Java SDK:

Folder	File Description
<install_location>/arcot/docs/webfort	Contains the Arcot-WebFort-7.1-authentication-sdk-javadocs.zip file, which contains the Javadocs for Authentication SDK.
<install_location>/arcot/samples/java	Contains the webfort-7.1-sample-application.war file to deploy Sample Application.
<install_location>/arcot/sdk/client/java/lib/arcot	Contains the following JAR files for AuthMinder Authentication Java SDK. <ul style="list-style-type: none"> ■ arcot-webfort-common.jar ■ arcot-webfort-authentication.jar
<install_location>/arcot/sdk/client/java/lib/external	Contains the third-party JAR files required by AuthMinder Authentication Java SDK. <ul style="list-style-type: none"> ■ bcprov-jdk15-146.jar ■ commons-pool-1.5.5.jar
<install_location>/arcot/sdk/client/java/properties	Contains the sample properties (webfort.authentication.properties) file. You can either use the parameters of this file for initializing the Java SDKs or use the init() function.

Issuance Java SDK Files

The following table lists the folder location of the files that are used by Issuance Java SDK:

Folder	File Description
<install_location>/arcot/docs/webfort	Contains the Arcot-WebFort-7.1-issuance-sdk-javadocs.zip file, which contains the Javadocs for Issuance SDK.
<install_location>/arcot/samples/java	Contains the webfort-7.1-sample-application.war file to deploy Sample Application.
<install_location>/arcot/sdk/client/java/lib/arcot	Contains the following JAR files for Issuance Java SDK. <ul style="list-style-type: none"> ■ arcot-webfort-common.jar ■ arcot-webfort-issuance.jar

Folder	File Description
<install_location>/arcot/sdk/client/java/lib/external	Contains the third-party JAR files required by AuthMinder Issuance Java SDK. <ul style="list-style-type: none"> ■ bcprov-jdk15-146.jar ■ commons-pool-1.5.5.jar
<install_location>/arcot/sdk/client/java/properties	Contains the sample properties (webfort.issuance.properties) file. You can either use the parameters of this file for initializing the Java SDKs or use the init() function.

Web Services Files

The following table lists the location of the files that are related to the UDS Web services:

Folder	File Description
<install_location>/arcot/docs/uds	Contains the arcot-uds-2_0-wsdl-docs.zip file, which contains the WSDL documents for UDS.
<install_location>/arcot/docs/webfort	Contains the following WSDL documents: <ul style="list-style-type: none"> ■ Arcot-WebFort-7.1-admin-wsdl docs.zip ■ Arcot-WebFort-7.1-authentication-wsdl docs.zip ■ Arcot-WebFort-7.1-issuance-wsdl docs.zip
<install_location>/arcot/wsdl/uds	Contains the following documents: <ul style="list-style-type: none"> ■ WSDL files that the Web services client uses to generate the code: <ul style="list-style-type: none"> ArcotConfigManagementSvc.wsdl ArcotOrganizationManagementSvc.wsdl ArcotUserManagementSvc.wsdl ■ XSD files that are used by Web services to perform bulk operations: <ul style="list-style-type: none"> ArcotUserSchema.xsd

Folder	File Description
<install_location>/arcot/wsdl/w ebfort	<p>Contains the following documents:</p> <ul style="list-style-type: none"> ■ WSDL files that the Web services client uses to generate the code: ArcotWebFortAdminSvc.wsdl ArcotWebFortAuthSvc.wsdl ArcotWebFortBulkOperationsSvc.wsdl ArcotWebFortIssuanceSvc.wsdl ■ XSD files that are used by Web services to perform bulk administration and credential operations: ArcotWebFortAdminMsgs.xsd ArcotWebFortAdminSchema.xsd ArcotWebFortAuthMsgs.xsd ArcotWebFortAuthSchema.xsd ArcotWebFortCommonSchema.xsd ArcotWebFortCredMgmt.xsd ArcotWebFortIssuanceMsgs.xsd ArcotWebFortIssuanceSchema.xsd ArcotWebFortTokenXchange.xsd

Plug-In SDK

The following table lists the folder location of the files that are used by plug-in SDK:

Folder	File Description
<install_location>/arcot/sdk/ser ver/plugin/c/lib	Contains the arwfpluginsdk.so containing the plug-in libraries.
<install_location>/arcot/sdk/ser ver/plugin/c/include/webfort/va s	<p>Contains the following SDK plug-in header files:</p> <ul style="list-style-type: none"> ■ wf-common-interface.h ■ wf-common-interface.hpp ■ wf-plugin-interface.h
<install_location>/arcot/sdk/ser ver/plugin/c/lib	Contains the arwfpluginsdk.so containing the plug-in libraries.

Appendix B: Configuration Files and Options

This topic discusses the configuration files that AuthMinder uses and the parameters that you must configure in these files.

The following AuthMinder configuration files are available in the `<install_location>/arcot/conf` location:

- [arcotcommon.ini](#) (see page 181)
- [adminserver.ini](#) (see page 190)
- [udsserver.ini](#) (see page 192)

The following properties files are available in the `<install_location>/arcot/sdk/client/java/properties/` location:

- [webfort.authentication.properties](#) (see page 194)
- [webfort.issuance.properties](#) (see page 196)

INI Files

arcotcommon.ini

The `arcotcommon.ini` file contains the parameters for database and instance settings for the AuthMinder Server and other components (Administration Console and User Data Service) of AuthMinder. This section discusses these parameters of `arcotcommon.ini` file:

- [Parameters Used by AuthMinder Server](#) (see page 181)
- [Parameters Used by Administration Console and User Data Service](#) (see page 185)

Parameters Used by AuthMinder Server

The following table lists the database and encryption settings used by AuthMinder Server. Additional database configurations for the AuthMinder Server must be performed using the Instance Management screen of the Administration Console.

Parameter	Default	Description
Common Database Parameters in[arcot/db/dbconfig] Section		

Parameter	Default	Description
DbType	No default	The type of the database applicable to all database connections. The supported values are: <ul style="list-style-type: none"> ■ mssqlserver ■ oracle ■ db2 ■ mysql
EnableBrandLicensing	0	Whether a branded ODBC driver is in use. This can be used when you are using the branded ODBC drivers from DataDirect.
BrandLicenseFile	No Default	The license file name when you use a branded ODBC driver.
StartWithAnyPool	1 (enabled)	Enables AuthMinder to start with the backup database when the primary database is unavailable.
Primary and Backup Database Connection Parameters in [arcot/db/primarydb] and [arcot/db/backupdb] Sections		
Datasource.N	No Default	The name of the ODBC System Data Source Name (DSN) pointing to the primary database hosting the server data.
Username.N	No Default	The User Name used by the server to access the database.
Encryption Mode Setting Parameter in [arcot/crypto/device] Section		
HSMDevice	s/w	The mode that sets whether the AuthMinder information must be encrypted with a key stored in the database or with one in a Hardware Security Module (HSM). Supported values are: <ul style="list-style-type: none"> ■ s/w: Indicates that the data will be encrypted with the key label stored in the database. ■ chrysalis: Indicates the Chrysalis (Luna) HSM will be used to encrypt the data. ■ nfast: Indicates nFast (nCIPHER nethSM) will be used to encrypt the data.
Chrysalis (Luna) HSM Configuration Parameters in [crypto/pkcs11modules/chrysalis] Section		

Parameter	Default	Description
sharedLibrary	No Default	The absolute path to the PKCS#11 shared library corresponding to the HSM. The default value for Chrysalis (Luna) is: /usr/lunasa/lib/libCryptoki2.so
storageSlot	0	The HSM slot where the 3DES keys used for encrypting the data are present.
accelSlot	0	The slot for internal use by Arcot.
sessionCount	20	The maximum number of sessions that can be established with the HSM device.
nFast (nCipher netHSM) HSM Configuration Parameters in [crypto/pkcs11modules/nfast] Section		
sharedLibrary	No Default	The absolute path to the PKCS#11 shared library corresponding to the HSM. The default value for nFast (nCipher netHSM) is: /opt/nfast/toolkits/pkcs11/libcknfast.so
storageSlot	1	The HSM slot where the 3DES keys used for encrypting the data are present.
accelSlot	0	The slot for internal use by Arcot.
sessionCount	200	The maximum number of sessions that can be established with the HSM device.
Watchdog Configurations in [arcot/watchdog] Section		
ServerStartsTimeout	25	The time period from the Server startup. If watchdog brings up the Server for 5 times within the specified duration of ServerStartsTimeout (25 minutes), then the Server is not restarted again. The time is in minutes.
ServerStartsCount	5	The maximum count for restarting the Server. After this, the Server is not be restarted again.
RestartSleepTime	5000	The sleep time after which watchdog restarts the Server. The sleep time is in milliseconds.

Changing Server Startup Logging Parameters

If you want to change the logging parameters that you see when AuthMinder Server starts up, then:

1. Navigate to the conf directory in ARCOT_HOME.
2. Open arcotcommon.ini in a text editor of your choice.
3. Add the following section at the end of the file:

```
[arcot/WebFort/startup]
LogFile=
LogFileSize=10485760
BackupLogFileDir=
LogLevel=
LogTimeGMT=0
```

The following table explains these parameters:

Parameter	Default	Description
LogFile		The file path to the default directory and the file name of the log file. Note: This path is relative to ARCOT_HOME (<install_location>/arcot/).
LogFileSize	10485760	The maximum number of bytes the log file can contain. When a log file reaches this size, a new file is started and the old file is moved to the location specified for BackupLogFileDir.
BackupLogFileDir		The location of the directory where backup log files are maintained, after the current file exceeds LogFileSize bytes. Note: This path is relative to ARCOT_HOME (<install_location>/arcot/).
LogLevel		The default logging level for the server, unless an override is specified. The possible values are: <ul style="list-style-type: none">■ 0 FATAL■ 1 WARNING■ 2 INFO■ 3 DETAIL

Parameter	Default	Description
LogTimeGMT	0	The parameter which indicates the time zone of the time stamp in the log files. The possible values are: <ul style="list-style-type: none"> ■ 0 Local Time ■ 1 GMT

1. Set the required values for the parameters that you want to change.
2. Save and close the file.
3. Restart AuthMinder Server.

Parameters Used by Administration Console and User Data Service

The following table describes the database setting parameters in the arcotcommon.ini file:

Parameter	Default	Description
Common Database Parameters in[arcot/db/dbconfig] Section		
DbType	No default	The type of database applicable to all database connections. The supported values are: <ul style="list-style-type: none"> ■ oracle ■ mssqlserver ■ db2 ■ mysql
Driver	No default	The fully-qualified name of the JDBC driver class that is supplied by the JDBC driver vendor. Consult your JDBC vendor documentation for the right driver name. <ul style="list-style-type: none"> ■ For Oracle - oracle.jdbc.driver.OracleDriver ■ For SQLServer - com.microsoft.sqlserver.jdbc.SQLServerDriver ■ For DB2 - jcom.ibm.db2.jcc.DB2Driver ■ For MySQL - com.mysql.jdbc.Driver
MinConnections	4	The minimum number of connections to initially create between AuthMinder components and the database.

Parameter	Default	Description
MaxConnections	32	The maximum number of connections that can be created between AuthMinder components and the database.
IncConnections	2	The number of connections that are created when a new connection is needed between AuthMinder components and the database.
MaxIdleConnections	4	The maximum number of idle database connections that server can maintain.
MaxWaitTimeForConnection	30000	Maximum amount of time (in milliseconds) the Server must wait (when there are no available connections) for a connection to become available, before timing out.
AutoRevert	1	Specifies whether or not the system attempts to connect to the primary database after a failover occurs. Set AutoRevert=1 if you have a backup database configured and if you want the server to try to connect to the primary database after a failover occurs.
MaxTries	3	The number of times the server will attempt to connect to the database before aborting the connection.
ConnRetrySleepTime	100	The number of milliseconds to delay between attempts to connect to the database.
MonitorSleepTime	50	The amount of time in seconds the Monitoring Thread sleeps between heartbeat checks on all databases.
Profiling	0	Specifies whether to log the database messages. Set the value to 1 if you want to enable logging of database messages.
EnableBrandLicensing	0	Specifies whether a branded ODBC driver is in use.
BrandLicenseFile	IVWF.LIC	The license file name when you use a branded ODBC driver. This parameter is required if the value of EnableBrandLicensing is 1. Otherwise it is ignored. If present, this value must <i>not</i> be edited.

Parameter	Default	Description
MaxTransactionRetries	3	The maximum number of times the transaction is retried with a database instance for pre-defined error conditions.
TransactionRetrySleepTime	10	The interval in milliseconds between two consecutive transaction retries.
Primary and Backup Database Connection Parameters in [arcot/db/primarydb] and [arcot/db/backupdb] Sections		
Datasource.N		The name of the ODBC System Data Source Name (DSN) pointing to the primary database hosting the server data.
AppServerConnectionPoolName.N	No default	<p>The JNDI name used to look up the connection pool object, if the database connection pooling feature of the application server is being used.</p> <p>A pool by this JNDI name should be created in the containing application server, and sufficient access right must be given to AuthMinder Web applications for it to use the connection pool:</p> <ul style="list-style-type: none"> ■ If the JNDI name is configured in Apache Tomcat, then use a fully qualified JNDI name. For example: AppServerConnectionPoolName.1=java:comp/env/SampleDS ■ For other application servers, specify only the JNDI name. For example: AppServerConnectionPoolName.1=SampleDS <p>See appendix, "Configuring Application Server" for more information.</p> <p>If the application server connection pool is <i>not</i> required, then leave this configuration empty.</p>

Parameter	Default	Description
URL. <i>N</i>	No default	<p>The name of the JDBC data source.</p> <ul style="list-style-type: none"> ■ For Oracle - jdbc:oracle:thin:@<server>:<oracle_port>:<sid> ■ For SQLServer - jdbc:sqlserver://<server>:<sql_port>;databaseName=<databaseName>;selectMethod=cursor ■ For DB2 - jdbc:db2://<server>:<db2_port>/<database> ■ For MySQL - jdbc:mysql://<server>:<mysql_port>/<database>
Username. <i>N</i>	No default	The User Name used by the server to access the database.
TrustStorePath. <i>N</i>	No default	<p>The SSL certificate truststore path corresponding to Datasource.<i>N</i>.</p> <p>The path (including the filename) refers to the certificate truststore file, which contains the list of certificates that the client trusts.</p> <p>Note: The password corresponding to TrustStore Path.<i>N</i> must to be securely stored in securestore.enc with value of TrustStorePath.<i>N</i> as the key by using DBUtil tool. Refer to <i>CA AuthMinder Administration Guide</i> for more information on this tool.</p>
KeyStorePath. <i>N</i>	No default	<p>Note: This attribute is used only for MySQL.</p> <p>If you want to configure one-way SSL between RiskMinder and a MySQL Database, this is one of the parameters for which you must specify a value. This parameter holds the SSL Certificate Keystore Path corresponding to Datasource.<i>N</i>. The path (including the filename) refers to the certificate keystore file. The password corresponding to KeyStorePath.<i>N</i> must be securely stored in securestore.enc with the value of KeyStorePath.<i>N</i> as the key.</p>

Parameter	Default	Description
HostNameInCertificate.N	No default	The value of Common Name (CN) in the subject Distinguished Name (DN) of Datasource.N SSL certificate in truststore. Note: This parameter is required <i>only</i> if you are using Microsoft SQL Server.
Encryption Mode Setting Parameter in [arcot/crypto/device] Section		
HSMDevice	s/w	The mode that sets whether the information must be encrypted with a key stored in the database or with one in a Hardware Security Module (HSM). Supported values are: <ul style="list-style-type: none"> ■ s/w: Indicates that the data will be encrypted with the key label stored in database. ■ chrysalis: Indicates the Chrysalis (Luna) HSM will be used to encrypt the data. ■ nfast: Indicates nFast (nCIPHER netHSM) will be used to encrypt the data.
Chrysalis (Luna) HSM Configuration Parameters in [crypto/pkcs11modules/chrysalis] Section		
sharedLibrary	No Default	The absolute path to the PKCS#11 shared library corresponding to the HSM. The default value for Chrysalis (Luna) is: /usr/lunasa/lib/libCryptoki2.so
storageSlot	0	The HSM slot where the 3DES keys used for encrypting the data are present.
accelSlot	0	The slot for internal use by Arcot.
sessionCount	20	The maximum number of sessions that can be established with the HSM device.
nFast (nCIPHER netHSM) HSM Configuration Parameters in [crypto/pkcs11modules/nfast] Section		
sharedLibrary	No Default	The absolute path to the PKCS#11 shared library corresponding to the HSM. The default value for nFast (nCIPHER netHSM) is: /opt/nfast/toolkits/pkcs11/libcknfast.so

Parameter	Default	Description
storageSlot	1	The HSM slot where the 3DES keys used for encrypting the data are present.
accelSlot	0	The slot for internal use by Arcot.
sessionCount	200	The maximum number of sessions that can be established with the HSM device.
Instance ID Configuration Parameter in [arcot/system] Section		
InstanceID	1	A parameter that can be used to identify Administration Console or User Data Service instance. It is mandatory that you provide unique values for every instance of the server. The instance ID is also displayed in the transaction reports. You must specify an integer value for this parameter.

adminsserver.ini

The adminsserver.ini file contains the parameters to set the Administration Console log information. Table B-4 lists the log file information of Administration Console.

Table B-4. Log Parameters

Parameter	Default Value	Description
Log Configuration Parameters in [arcot/admin/logging] Section		
log4j.rootCategory	ERROR, roothandle Important! roothandle is the name of the Administration Console log handle and <i>must</i> be specified.	The root logger that resides at the top of the logger hierarchy. All children loggers inherit this value, if no value is specified.

Parameter	Default Value	Description
<ul style="list-style-type: none"> ■ log4j.logger.com.arcot.euds ■ log4j.logger.com.arcot.admin ■ log4j.logger.com.arcot.admin.framework ■ log4j.logger.com.arcot.adminconsole ■ log4j.logger.com.arcot.common.cache ■ log4j.logger.com.arcot.common.crypto ■ log4j.logger.com.arcot.crypto.impl.SecurityStoreUtil ■ log4j.logger.com.arcot.common.database ■ log4j.logger.com.arcot.common.ldap 	INFO	<p>Specify the log level that must be used to write Administration Console logs. The supported log levels are:</p> <ul style="list-style-type: none"> ■ FATAL ■ WARNING ■ INFO ■ DEBUG <p>Note: Refer to <i>CA AuthMinder Administration Guide</i> for more information on the log levels.</p>
log4j.appender.roothandle.Encoding	UTF-8	The encoding to use when writing the entries in the log file.
log4j.appender.roothandle.File	\${arcot.home}/logs/arcotadmin.log	<p>The log file name and the location where the Administration Console logs will be created. By default, the Administration Console log file name is arcotadmin.log and is created in the following location:</p> <p><i><install_location>/arcot/logs/</i></p>
log4j.appender.roothandle.MaxFileSize	10 MB	The maximum allowed file size of the log file.
log4j.appender.roothandle.MaxBackupIndex	100	<p>The maximum number of backup files that can be created.</p> <p>When the number of backup files reaches this number, then the application starts to overwrite from the first log file.</p>

Parameter	Default Value	Description
log4j.appender. rootHandler.layout	org.apache. log4j. PatternLayout	The output format, as specified by ConversionPattern.
log4j.appender. rootHandler.layout. ConversionPattern	%d{yyyy-MM-dd hh:mm:ss,SSS z} : [%t] : %-5p : %-5c{3} : %m%n	<p>The format in which the Administration Console log file entries are written:</p> <ul style="list-style-type: none"> ■ Time Stamp (%d{yyyy-MM-dd hh:mm:ss,SSS z} :) ■ Thread ID ([%t] :) ■ Log Level (or Severity) (%-5p :) ■ Logger Class (%-5c{3} :) ■ Message (%m%n) <p>This pattern is similar to the C language printf function.</p>

udserver.ini

The udserver.ini file contains the parameters to set the User Data Service (UDS) log information. The following table lists the log file information of UDS:

Parameter	Default Value	Description
Log Configuration Parameters in [arcot/uds/logger] Section		
log4j.rootCategory	ERROR, debuglog	The root logger that resides at the top of the logger hierarchy. All children loggers inherit this value, if no value is specified.

Parameter	Default Value	Description
<ul style="list-style-type: none"> ■ log4j.logger.com.arcot.euds ■ log4j.logger.com.arcot.crypto.impl.SecureStoreUtil ■ log4j.logger.com.arcot.common.database ■ log4j.logger.com.arcot.common.cache 	INFO	<p>Specify the log level that must be used to write UDS logs. The supported log levels are:</p> <ul style="list-style-type: none"> ■ FATAL ■ WARNING ■ INFO ■ DEBUG <p>Note: Refer to <i>CA AuthMinder Administration Guide</i> for more information on the log levels.</p>
log4j.appender.debuglog.File	\${arcot.home}/logs/arcotuds.log	<p>Specify the log file name and the location where the UDS logs must be written to.</p> <p>By default, the UDS log file name is <code>arcotuds.log</code> and is created in the logs folder present in <code><install_location>/arcot/</code>.</p>
log4j.appender.debuglog.MaxFileSize	10MB	Specify the size of the log file. By default, it is 2 MB.
log4j.appender.debuglog.MaxBackupIndex	100	Specify the number of backup files that can be created. When the number of backup files is equivalent to this number, the application starts to overwrite from the first log file.
log4j.appender.debuglog.layout	org.apache.log4j.PatternLayout	The output format, as specified by <code>ConversionPattern</code> .
log4j.appender.debuglog.Encoding	UTF-8	The encoding to use when writing the entries in the log file.

Parameter	Default Value	Description
log4j.appender. debuglog.layout. ConversionPattern	%d{yyyy-MM-dd hh:mm:ss,SS z} : [%t] : %-5p : %-5c{3} : %m%n	The format in which the UDS log file entries are written: <ul style="list-style-type: none"> ■ Time Stamp (%d{yyyy-MM-dd hh:mm:ss,SSS z} :) ■ Thread ID ([%t] :) ■ Log Level (or Severity) (%-5p :) ■ Logger Class (%-5c{3} :) ■ Message (%m%n) This pattern is similar to the C language printf function.

Properties Files

webfort.authentication.properties

The webfort.authentication.properties file provides the parameters for the Authentication Java SDK to read AuthMinder Server information. The following table lists the configuration parameters.

By default, the configuration parameters are appended with **.1**, which indicate that these configurations are for the primary AuthMinder Server. If you have multiple instances of AuthMinder Servers and want to enable failover, then you must duplicate the sections based on the number of servers you plan to support and configure the parameters accordingly.

Parameter	Default	Description
pool.maxActive	64	Maximum number of connections allowed in the pool from the SDK to the AuthMinder Server.
pool.maxIdle	16	The maximum number of idle connections allowed in the pool from the SDK to the AuthMinder Server.
pool.maxWaitTimeMillis	-1	The maximum amount of time (in milliseconds) that a request will wait for the connection. Default -1 indicates that the thread will wait for infinite time.

Parameter	Default	Description
pool.minEvictableIdleTimeMillis	-1	The minimum amount of time a connection might be idle in the pool before it is evicted by the idle connection evictor (if any).
pool.timeBetweenEvictionRunsMillis	-1	The amount of time in milliseconds to wait before checking the pool to evict the idle connections.
authentication.host. <i>n</i>	localhost	Host name or the IP address of AuthMinder Server.
authentication.port. <i>n</i>	9742	Port number configured for the Authentication Native protocol.
authentication.transport. <i>n</i>	tcp	To enable the SSL communication between AuthMinder Authentication SDK and AuthMinder Server set this parameter to 1SSL or 2SSL. Note: If you change the transport mode to SSL, then you must restart AuthMinder Server.
authentication.connectionTimeout. <i>n</i>	10000	Maximum time in milliseconds before the AuthMinder Server is considered unreachable.
authentication.readTimeout. <i>n</i>	30000	The maximum time in milliseconds allowed for a response from AuthMinder Server.
authentication.serverCACertPEMPath. <i>n</i>	No Default	Provide the path for the CA certificate file of the server. The file <i>must</i> be in PEM format. Provide the complete path for the file. For example: server.CACertPEMPath=<%SystemDrive%>/certs/webfort_ca.pem
authentication.clientCertKeyP12Path. <i>n</i>	No Default	Provide the path for the client certificate, which is in the p12 format.
authentication.clientCertKeyPassword. <i>n</i>	No Default	Enter the client key pair password to open the p12 file.

webfort.issuance.properties

The webfort.issuance.properties file provides the parameters for the Issuance Java SDK to read AuthMinder Server information. The following table lists the configuration parameters.

By default, the configuration parameters are appended with **.1**, which indicate that the configurations are for the primary AuthMinder Server. If you have multiple instances of AuthMinder Servers and want to enable failover, then you must duplicate the sections based on the number of servers you plan to support and configure the parameters accordingly.

Parameter	Default	Description
pool.maxActive	64	Maximum number of connections allowed in the pool from the SDK to the AuthMinder Server.
pool.maxIdle	16	The maximum number of idle connections allowed in the pool from the SDK to the AuthMinder Server.
pool.maxWaitTimeMillis	-1	The maximum amount of time (in milliseconds) that a request will wait for the connection. Default -1 indicates that the thread will wait for infinite time.
pool.minEvictableIdleTimeMillis	-1	The minimum amount of time a connection might be idle in the pool before it is evicted by the idle connection evictor (if any).
pool.timeBetweenEvictionRunsMillis	-1	The amount of time in milliseconds to wait before checking the pool to evict the idle connections.
issuance.host.n	localhost	Host name or the IP address of AuthMinder Server.
issuance.port.n	9744	Port number configured for the Transaction Web services protocol.
issuance.transport.n	tcp	To enable the SSL communication between AuthMinder Issuance SDK and AuthMinder Server set this parameter to 1SSL or 2SSL. Note: If you change the transport mode to SSL, then you must restart AuthMinder Server.
issuance.connectionTimeout.n	10000	Maximum time in milliseconds before the AuthMinder Server is considered unreachable.

Parameter	Default	Description
issuance.readTimeout. <i>n</i>	30000	The maximum time in milliseconds allowed for a response from AuthMinder Server.
issuance.serverCACertPEMPath. <i>n</i>	No Default	Provide the path for the CA certificate file of the server. The file <i>must</i> be in PEM format. Provide the complete path for the file. For example: server.CACertPEMPath=<%SystemDrive%>/certs/webfort_ca.pem
issuance.clientCertKeyP12Path. <i>n</i>	No Default	Provide the path for the client certificate, which is in p12 format.
issuance.clientCertKeyPassword. <i>n</i>	No Default	Enter the client key pair password to open the p12 file.

Appendix C: Changing HSM Configurations

This appendix lists the steps that you must perform, if you want to change the Hardware Security Module (HSM) configurations that you have specified during installation.

Note: Before proceeding with the configurations explained in this section, ensure that you have set up the HSM server and client, and generated the 3DES key in the HSM. Refer to "[\(Optional, Only If You are Using HSMs\) Requirements for HSM](#)" (see page 54) for more information.

As mentioned in "[Hardware Security Module \(HSM\) Requirements](#)" (see page 35), AuthMinder now supports Hardware Security Module (HSM) to secure your data. If you choose to encrypt the data using HSM, then the data stored in the database is encrypted with the key that resides in the HSM.

AuthMinder supports Luna and nCipher netHSM for data encryption using hardware. The configurations for the HSMs are available in the `arcotcommon.ini` file. This file provides separate sections for configuring the required HSM, which in the current release are:

- Luna HSM (`[crypto/pkcs11modules/chrysalis]`)
- nCipher netHSM (`[crypto/pkcs11modules/nfast]`)

Based on the HSM you are configuring, you must specify the `sharedLibrary` parameter in the corresponding section. After specifying the HSM information, you must then re-create the `securestore.enc` file with the HSM key label, initialize the HSM, and then initialize AuthMinder to use the HSM key.

To change the HSM information that AuthMinder needs:

1. Navigate to the following location:
`<install_location>/arcot/conf`
2. Take a backup of `securestore.enc`.
3. Delete the existing `securestore.enc` file from `<install_location>/arcot/conf`.

4. To change the HSM information that AuthMinder needs:
 - a. Navigate to the following location:
`<install_location>/arcot/conf`
 - b. Open `arcotcommon.ini` in a text editor.
 - c. Ensure the `HSMDevice` parameter in the `[arcot/crypto/device]` section is set to the HSM that you plan to use:
 - `chrysalis` for Luna HSM.or
 - `nfast` for nCipher netHSM.
 - d. Depending on the HSM that you are configuring, set the `sharedLibrary` parameter to the location where the HSM library file is located.

For Luna (`libCryptoki2.so`) and for nCipher netHSM (`libcknfast.so`), enter the absolute path and full name of the file.

Note: Refer to "[arcotcommon.ini](#)" (see page 181) for more information on the other HSM configuration parameters available in this section.
 - e. Save and close the `arcotcommon.ini` file.
5. Navigate to the following location, where the DBUtil tool is available:
`<install_location>/arcot/tools/<platform_name>`
6. Run the DBUtil tool with the following commands:
 - a. `dbutil -init <HSM_key_label>`

Note: The `<HSM_key_label>` corresponds to the 3DES key that resides in the HSM.

The preceding command creates a new `securestore.enc` file with the specified key label. The generated file is stored in the `<install_location>/arcot/conf` location.
 - b. `dbutil -i <HSM_module_name> <HSM_password>`

Note: The `<HSM_module_name>` is `chrysalis` for Luna HSM, and `nfast` for nCipher netHSM.

The preceding command initializes the HSM.
 - c. `dbutil -pi <DSN_Name> <Database_password> -h <HSM_password> -d <HSM_module_name>`

Note: `<DSN_NAME>` refers to the ODBC DSN that AuthMinder Server uses to connect to the AuthMinder database. `<Database_password>` refers to password used to connect to the database.

The preceding command initializes the AuthMinder Server data to be encrypted using HSM.

- d. `dbutil -pi <Database_Username> <Database_password> -h <HSM_password> -d <HSM_module_name>`

Note: *<Database_Username>* refers to the user name used to connect to the AuthMinder database. The database user name is case-sensitive, therefore ensure that you provide the correct value. *<Database_password>* refers to password used to connect to the database.

The preceding command initializes the Administration Console and the User Data Service data to be encrypted by using HSM.

Appendix D: Preparing for Localization

AuthMinder now supports *localization*, which is the process of adapting internationalized software for a region or language of your choice, by adding locale-specific components and translating the text. In this release, the following aspects can be localized:

- Date-Time formatting
- Number formatting
- Display formatting in the user interfaces

This appendix provides the steps that you must perform to localize your AuthMinder setup and the list of locales that AuthMinder supports. It covers the following topics:

- [Adding New Locales](#) (see page 204)
- [Configuring Supported Locales](#) (see page 205)
- [Setting Your Default Locale](#) (see page 206)
- [List of Available Locales](#) (see page 206)

Adding New Locales

AuthMinder supports English (US) out-of-the-box. To add a custom language other than English, you must perform the following tasks:

1. Navigate to the appropriate location in your `<APP_SERVER_HOME>` where the `arcotadmin/WEB-INF/classes/` directory is available, after you deployed the `arcotadmin.war`.

For example, on Apache Tomcat this location is `<TOMCAT_HOME>/webapps/arcotadmin/WEB-INF/classes/`.

2. Append the locale of the language that you want to localize to as the suffix to the following files:

- `adminconsole-admin-error.properties`
- `adminconsole-admin-message.properties`
- `adminconsole-admin-report.properties`
- `framework-admin-error.properties`
- `framework-admin-message.properties`
- `framework-common-message.properties`
- `framework-report-messages.properties`
- `framework-useraccount-status.properties`
- `message.properties`
- `webfort-admin-error.properties`
- `webfort-admin-message.properties`
- `webfort-admin-report.properties`

For example, if you are localizing to Japanese, then you must add the suffix `_ja_JP` to all the preceding properties files. In this case, the `adminconsole-admin-error.properties` file name will change to `adminconsole-admin-error_ja_JP.properties`.

See [List of Available Locales](#) (see page 206) to identify the locale string for a specific language and country.

3. Convert and save *all* the `*<locale>.properties` files to UTF-8 by using the `native2ascii` tool, which is available in `<JAVA_HOME used by Application Server>/bin`.
4. Navigate to `<install_location>/arcot/resourcepacks/i18n/`, and:
5. Create a subdirectory called `webfort`.
6. Copy the converted and saved properties files to the `<install_location>/arcot/resourcepacks/i18n/webfort` directory.
7. Open the Command Prompt window.
8. Navigate to the `<install_location>/arcot/tools/common/bundlemanager/` directory.

9. Run the following command to integrate the newly-ported resource bundles with the adminframework.war file and re-create arcotadmin.war under `<install_location>/arcot/java/webapps/`:

```
java -jar bundle-manager.jar
```
10. Re-deploy the arcotadmin.war. See "[Deploying Administration Console](#)" (see page 79) and "[Verifying the Administration Console](#)" (see page 81) for more information.

Configuring Supported Locales

To configure custom locales and set the default locale and date time format using Administration Console:

1. Access the Administration Console in a Web browser.
2. Ensure that you are logged in as the MA.
3. Activate the **Services and Server Configurations** tab.
4. Click the **Administration Console** option on the submenu of the tab. Under the **System Configuration** section on the side-bar menu, click the **Localization Configuration** link to display the page.
5. In the **Configure Supported Locales** section, select the locales that you want to add from the **Available** list to the **Selected** list.

Click the > button to move selected locales to the desired list. You can also click the >> button to move all locales to the desired lists.

Note: Hold the **Ctrl** key to select more than one locale at a time.

6. In the **Configure Default Locale** section, select the **Default Locale** from the drop-down list.
7. In the **Configure Default Date Time Format** section, specify the **Date Time Format** you want to use.

Note: The Administrator can change the Locale and Date Time Format on the My Profile page.

8. Click **Save** to save your changes.

The locale that you added will appear in the **Available** list on the Localization Configuration page of the Administration Console.

9. Refresh the AuthMinder Server instance:
 - a. Activate the **WebFort** tab.
 - b. Click the Instance Management link under the **Instance Configurations** section.
 - c. Select the same AuthMinder Server instance.
 - d. Click the **Refresh** button.

Setting Your Default Locale

As an administrator, you need to change your current locale to the new configured locale. Perform the following steps to do so:

1. Access the Administration Console in a Web browser.
2. Log in to the Administration Console.
3. Click the **<username>** link in the header of the Administration Console.
The My Profile page appears.
4. In the **Administrator Preferences** section, specify the preferred **Locale** for your instance of Administration Console.
5. Click **Save**.

List of Available Locales

The following table lists the identifier string for each supported locale:

Language	Language	Language Code	Country Code	Locale
Arabic		ara		ar
	United Arab Emirates	ara	ARE	ar_AE
	Bahrain	ara	BHR	ar_BH
	Algeria	ara	DZA	ar_DZ
	Egypt	ara	EGY	ar_EG
	Iraq	ara	IRQ	ar_IQ
	Jordan	ara	JOR	ar_JO
	Kuwait	ara	KWT	ar_KW
	Lebanon	ara	LBN	ar_LB
	Libya	ara	LBY	ar_LY
	Morocco	ara	MAR	ar_MA
	Oman	ara	OMN	ar_OM
	Qatar	ara	QAT	ar_QA
	Saudi Arabia	ara	SAU	ar_SA
	Sudan	ara	SDN	ar_SD
Syria	ara	SYR	ar_SY	
Tunisia	ara	TUN	ar_TN	

Language	Language	Language Code	Country Code	Locale
	Yemen	ara	YEM	ar_YE
Hindi	India	hin	IND	hi_IN
Hebrew		heb		iw
	Israel	heb	ISR	iw_IL
Japanese		jpn		ja
	Japan	jpn	JPN	ja_JP
Korean		kor		ko
	South Korea	kor	KOR	ko_KR
Thai		tha		th
	Thailand	tha	THA	th_TH
	Thailand,TH	tha	THA (Variant: TH)	th_TH_TH
Vietnamese		vie		vi
	Vietnam	vie	VNM	vi_VN
Chinese		zho		zh
	China	zho	CHN	zh_CN
	Hong Kong	zho	HKG	zh_HK
	Taiwan	zho	TWN	zh_TW
Belarusian		bel		be
	Belarus	bel	BLR	be_BY
Bulgarian		bul		bg
	Bulgaria	bul	BGR	bg_BG
Catalan		cat		ca
	Spain	cat	ESP	ca_ES
Czech		ces		cs
	Czech Republic	ces	CZE	cs_CZ
Danish		dan		da
	Denmark	dan	DNK	da_DK
German		deu		de
	Austria	deu	AUT	de_AT

Language	Language	Language Code	Country Code	Locale
	Switzerland	deu	CHE	de_CH
	Germany	deu	DEU	de_DE
	Luxembourg	deu	LUX	de_LU
Greek		ell		el
	Greece	ell	GRC	el_GR
English	Australia	eng	AUS	en_AU
	Canada	eng	CAN	en_CA
	United Kingdom	eng	GBR	en_GB
	Ireland	eng	IRL	en_IE
	India	eng	IND	en_IN
	New Zealand	eng	NZL	en_NZ
	South Africa	eng	ZAF	en_ZA
Spanish		spa		es
	Argentina	spa	ARG	es_AR
	Bolivia	spa	BOL	es_BO
	Chile	spa	CHL	es_CL
	Colombia	spa	COL	es_CO
	Costa Rica	spa	CRI	es_CR
	Dominican Republic	spa	DOM	es_DO
	Ecuador	spa	ECU	es_EC
	Spain	spa	ESP	es_ES
	Guatemala	spa	GTM	es_GT
	Honduras	spa	HND	es_HN
	Mexico	spa	MEX	es_MX
	Nicaragua	spa	NIC	es_NI
	Panama	spa	PAN	es_PA
	Peru	spa	PER	es_PE
	Puerto Rico	spa	PRI	es_PR
Paraguay	spa	PRY	es_PY	
El Salvador	spa	SLV	es_SV	

Language	Language	Language Code	Country Code	Locale
	Uruguay	spa	URY	es_UY
	Venezuela	spa	VEN	es_VE
Estonian		est		et
	Estonia	est	EST	et_EE
Finnish		fin		fi
	Finland	fin	FIN	fi_FI
French		fra		fr
	Belgium	fra	BEL	fr_BE
	Canada	fra	CAN	fr_CA
	Switzerland	fra	CHE	fr_CH
	France	fra	FRA	fr_FR
	Luxembourg	fra	LUX	fr_LU
Croatian		hrv		hr
	Croatia	hrv	HRV	hr_HR
Hungarian		hun		hu
	Hungary	hun	HUN	hu_HU
Icelandic		isl		is
	Iceland	isl	ISL	is_IS
Italian		ita		it
	Switzerland	ita	CHE	it_CH
	Italy	ita	ITA	it_IT
Lithuanian		lit		lt
	Lithuania	lit	LTU	lt_LT
Latvian		lav		lv
	Latvia	lav	LVA	lv_LV
Macedonian		mkd		mk
	Macedonia	mkd	MKD	mk_MK
Dutch		nld		nl
	Belgium	nld	BEL	nl_BE
	Netherlands	nld	NLD	nl_NL

Language	Language	Language Code	Country Code	Locale
Norwegian		nor		no
	Norway	nor	NOR	no_NO
	Norway,Nynorsk	nor	NOR (Variant: NY)	no_NO_NY
Polish		pol		pl
	Poland	pol	POL	pl_PL
Portuguese		por		pt
	Brazil	por	BRA	pt_BR
	Portugal	por	PRT	pt_PT
Romanian		ron		ro
	Romania	ron	ROU	ro_RO
Russian		rus		ru
	Russia	rus	RUS	ru_RU
Slovak		slk		sk
	Slovakia	slk	SVK	sk_SK
Slovenian		slv		sl
	Slovenia	slv	SVN	sl_SI
Albanian		sqi		sq
	Albania	sqi	ALB	sq_AL
Swedish		swe		sv
	Sweden	swe	SWE	sv_SE
Turkish		tur		tr
	Turkey	tur	TUR	tr_TR
Ukrainian		ukr		uk
	Ukraine	ukr	UKR	uk_UA
English	United States	eng	USA	en_US
		eng		en

Appendix E: Database Reference

AuthMinder database contains a number of tables, some of which grow with increased usage. Some tables grow in direct relation to the number of users, while others grow in direct relation to the usage of the product. Also, a user accessing the system multiple times will cause the tables to grow. Because of restricted disk space, as a database administrator managing AuthMinder deployments, you might not want these tables to grow indefinitely. In this case, you can use the information in this appendix to trim some tables to manage your disk space and improve the database performance.

You must only trim the tables that capture transaction details, such as audit log information. You *must not* trim tables that capture user information, which is necessary to authenticate users.

Note: It is recommended that you make appropriate adjustments to the SQL databases based on the configuration and the need for reporting data. For example, deleting a large volume of data will adversely impact performance during the delete process. Depending on the size of the rollback segments, this may even cause the system to fail. It is also highly recommended that you archive older records and not delete them completely.

This appendix discusses the recommendations on database table replication, how to calculate the database size while you are planning to set up the database for AuthMinder, and lists all tables used by AuthMinder along with some trimming recommendations:

- [AuthMinder Database Tables](#) (see page 211)
- [Database Sizing Calculations](#) (see page 220)
- [Database Tables Replication Advice](#) (see page 221)
- [Database Tuning Parameters](#) (see page 226)

AuthMinder Database Tables

This section briefly explains all the database tables:

- [Used by AuthMinder](#) (see page 212)
- [Used by Administration Console](#) (see page 214)
- [Used by User Data Service](#) (see page 217)

Used by AuthMinder

The following table lists the database tables used by the AuthMinder Server:

Table Name	Description
ARWFADMINAUDITLOG	Contains the audit log information for the AuthMinder administration activities.
ARWFARCOTEMV	Contains the ArcotID OTP-EMV credentials for the users. It contains an individual entry for each user.
ARWFARCOTEMVHISTORY	Contains all the ArcotID OTP-EMV credentials that are in the <i>reissue</i> state.
ARWFARCOTID	Contains the ArcotID PKI credentials for the users. It contains an individual entry for each user.
ARWFARCOTIDHISTORY	Contains all the ArcotID PKIs that are in the <i>reissue</i> state.
ARWFARCOTOTP	Contains the ArcotID OTP-OATH credentials for the users. It contains an individual entry for each user.
ARWFARCOTOTPHISTORY	Contains all the ArcotID OTP-OATH credentials that are in the <i>reissue</i> state.
ARWFAUTHAUDITLOG	Contains the audit log information for the authentication activities.
ARWFAUTHTOKENS	Contains the authentication tokens that are generated after a successful authentication. One entry is made in this table for every successful authentication irrespective of the type of token requested.
ARWFCONFIG	Contains the AuthMinder configuration information. The information in this table contains version information and therefore has multiple entries per configuration.
ARWFDATABASEERRORCODES	Contains the database error codes that indicate the communication failure.
ARWFDATABASEQUERIES	Contains the list of database queries used by the AuthMinder Server.
ARWFDISPLAYNAMES	Contains names and values of different keys used in AuthMinder.
ARWFGENERICCRED	Contains the information about the miscellaneous credentials of the user. For example, the credentials supported by custom APIs.

Table Name	Description
ARWFGENERICCREDHISTORY	Contains all the miscellaneous (for example, custom API supported) credentials that are in <i>re-issue</i> state.
ARWFINSTANCES	Contains information about all the instances of AuthMinder Servers that communicate with a specific database.
ARWFISSUANCEAUDITLOG	Contains the audit log information for the credential issuance activities.
ARWFMESSAGES	Contains the messages that are posted by the AuthMinder Server.
ARWFMODULEREGISTRY	Contains information about the internal modules of the AuthMinder Server and about the plug-ins.
ARWFOATH	Contains the OATH One-Time Password (OTP) credentials of the users. It contains an individual entry for each user.
ARWFOATHHISTORY	Contains all the OATH OTP credentials that are in <i>re-issue</i> state.
ARWFOATHTOKENREGISTRY	Contains the OATH token details such as, seed value, token ID, and token type.
ARWFORGACTIVECONFIG	Contains configuration mapping of the currently active organization. The information in this table contains version information and therefore has multiple entries per configuration.
ARWFORGCONFIG	Contains configuration mapping per organization. The information in this table contains version information and therefore has multiple entries per configuration.
ARWFOTP	Contains the One-Time Password (OTP) credentials for the users. It contains an individual entry for each user.
ARWFPASSWORD	Contains the user name-password credentials for the users. It contains an individual entry for each user.
ARWFPASSWORDHISTORY	Contains all the user-name password credentials that are in <i>re-issue</i> state.
ARWFPROTOCOLCONFIGURATION	Contains configuration of each listener port of the AuthMinder Server.
ARWFQNA	Contains the Question and Answer (QnA) credentials for the users. It contains an individual entry for each user.

Table Name	Description
ARWFQNAHISTORY	Contains all the question and answer credentials that are in <i>re-issue</i> state.
ARWFSEQUENCE	Contains information about sequences used for version configurations.
ARWFSSLTRUSTSTORE	Contains the SSL root certificates that are trusted by the AuthMinder Server.
ARWFSVRMGMTAUDITLOG	Contains the audit log information for the server management activities.
ARWFUNIQUEFIELDS	Enforces uniqueness of certain fields in the configuration. For example, a RADIUS client IP address cannot be used by two organizations.
ARWFVERIFIEDCHALLENGES	Contains information about the challenges for which the ArcotID PKI signature is successfully verified. An entry is made for successful ArcotID PKI authentication, provided No Replay for challenge is turned on. It is turned off by default.

Used by Administration Console

The following table lists all database tables used by the Administration Console:

Table Name	Description
ARADMINAUDITTRAIL	Stores administrator activity audit.
ARADMINAUTHOKEN	Stores the tokens that Administration Console uses for pluggable authentication. Every time you log in to the Console by using password, a token is internally generated after password match and stored in this table.
ARADMINBASICAUTHPWDHISTORY	Stores the last <i>n</i> occurrences of password of all administrators in all organizations that use Basic Authentication (for administrators) to log in to the Administration Console. This information is stored to prevent password reuse.
ARADMINBASICAUTHUSER	Stores the basic authentication credentials of all administrators in all organizations that use Basic Authentication (for administrators) to log in to the Administration Console.
ARADMINCONFIG	Stores the Administration Console configurations.

Table Name	Description
ARADMINCUSTOMROLE	Stores the configurations for all custom-defined roles.
ARADMINMANAGEROLE	Stores the list of roles that a specified role can manage.
ARADMINMAP	Stores the information of the AuthMinder Server instance, which is entered as a key-value pair.
ARADMINPAFCONFIG	Stores the authentication configurations for all administrators in all organizations in the system.
ARADMINPREDEFINEDROLE	Stores the role information for all supported administrators.
ARADMINPWDPOLICY	Stores the details of password policies for all administrators in all organizations.
ARADMINROLEPRIVILEGE	Stores the mapping of all administrative actions (or tasks) supported by the Administration Console, the scope of each task, and which role can perform the task.
ARADMINSCOPE	Stores the list of organizations over which each administrator has control (scope).
ARADMINSCOPEALL	Stores the list of all administrators who have control (scope) over <i>all</i> the existing organizations in the system.
ARADMINSUPPORTEDAUTHMECHANISM	Stores the information about all supported authentication mechanisms to log in to the Administration Console.
ARADMINSUPPORTEDTIMEZONE	Stores the list of all available time zones. Note: This is an internal table.
ARADMINTURNEDOFFPRIVILEGE	Stores the list of the privileges that are not available for the given custom role.
ARADMINTXID	Stores the information required to generate a unique ID for each transaction.
ARADMINUITAB	Stores the information about the tabs that are available and the order in which they are available in the Administration Console.
ARADMINUITASK	Stores the information about all the tasks that are available and the order in which they are available through the Administration Console.

Table Name	Description
ARADMINUITASKATTRIBUTES	Stores the details of the tasks that are displayed, when the first-level and the second-level tabs in the Administration Console are clicked. These tasks are referred to as landing pages.
ARADMINUITASKCONTAINER	Stores the information related to available <i>task containers</i> . A task container can either be a second-level tab ID or the task group in the Administration Console.
ARADMINUSER	Stores the detailed information (such as organization to which they belong, current status, time zone, locale, last login time) of all existing administrators.
ARADMINUSER_ARCHIVE	Stores the information related to all deleted users.
ARADMINWIZARDTASK	Stores the information about all the tasks that can be performed by using the Bootstrap Wizard.
ARCMNBULKOPERATION	Stores information related to all supported bulk operations that include uploading users and uploading user accounts.
ARCMNBULKOPERATIONATTRIBUTE	Stores attributes for all bulk operations in the ARCMNBULKOPERATION table.
ARCMNBULKREQUEST	Stores details (such as organization name, Request ID, status of the request, data uploaded, and operation) for each bulk-upload request.
ARCMNBULKTASKPARAM	Stores the name and the value of each attribute for each task supported in the system.
ARCMNBULKUPLOADTASK	Stores the status of each task for every bulk-upload request.
ARCMNCACHEREFRESH	Stores cache-related housekeeping information that indicates whether the Administration Console needs to be refreshed or not.
ARCMNCONFIG	Stores common Administration Console configuration information. Some of these include whether Bootstrap is complete, whether the cache refresh is automatic or manual, whether attribute encryption is enabled, and whether the bulk upload feature is enabled or not.

Table Name	Description
ARCMNDBERRORCODES	Stores vendor-specific database error codes and SQL state values that signify whether the database is down or non-responsive. This information is used by the system to decide if database should be failed over, in case a backup database is configured.
ARCMNMAPDATATYPE	Stores CA product-specific information that the Administration Console uses for rendering the Console pages.
ARCMNKEY	Stores the key configurations used while creating or updating organizations.
ARPCFMNCACHEREFRESHEVENT	Stores details of all cache refresh events for all instances in the system.
ARPCFMNCACHEREFRESHSCOPE	Stores information about all organizations that will be affected if a server cache refresh event occurs.
ARPCFMNCACHEREFRESHSTATUS	Stores the status of each cache refresh event for every instance for which it was triggered.
ARPCFMNINSTANCE	Stores detailed information for all AuthMinder Server instances configured in the system. This also includes the last time the instance was refreshed.
ARPCFMNORGCONFIGDATA	Stores configuration details for each organization. This includes global configurations that can be, typically, overridden at the organization-level.
ARPCFMNORGCONFIGSTATE	Stores the status of each assigned configuration from the ARPCFMNORGCONFIGDATA table.
ARPCFMNPRIVILEGEMAPPING	Stores the details of each privilege available through the Administration Console.
ARREPORTTABLES	Stores the information about the reports that are generated using Administration Console.
ARSEQUENCETABLE	Simulates sequences using stored procedures. Note: This table is used <i>only</i> by MS SQL Server.

Used by User Data Service

The following table explains the database tables used by UDS:

Table Name	Description
ARUDSACCOUNTTYPE	Stores the details of all account types that are configured in the system.

Table Name	Description
ARUDSATTRMAP	Stores the configuration details that describe the field names of custom attributes for accounts, specific to each organization.
ARUDSAUTHSESSION	Stores authentication session details for currently active sessions. If this table is not replicated, then active authentication sessions can be lost.
ARUDSCALLOUT	Stores user-specific Callout configurations. These Callouts are called, if configured, for specific events, such as user creation and update.
ARUDSCALLOUTINTERNAL	Stores the configuration information (SDK method to be invoked), for Callouts when a delete event with cascade effect is triggered or enabled.
ARUDSCALLOUTINTERNALPARAMS	Stores the details, such as parameters and types specific to internal Callouts.
ARUDSCALLOUTPARAM	Stores the details, such as parameters and types specific to external Callouts.
ARUDSCONFIG	Stores the UDS configuration parameters and their values.
ARUDSCONFIGAUDITLOG	Stores the audit log information for the User Data Source (UDS) operations and their return status.
ARUDSCONCONTACTTYPE	Stores additional contact information (such as secondary email and telephone number) that can be configured at the organization or global level.
ARUDSCUSTOMATTREXT	Stores the additional user account custom attributes. By default, up to 10 user account custom attributes are stored in the ARUDSUSERACCOUNT table. Any additional attributes (after the first 10) are stored in this table.
ARUDSCUSTOMATTREXT_ARCHIVE	Stores the archived information related to user account custom attributes when a user account is deleted.
ARUDSLDAPREPOSITORYCONFIG	Stores the LDAP Repository configurations, such as LDAP host and port details.
ARUDSORGANIZATION	Stores organization definitions, their attributes and repository connectivity details.
ARUDSORGANIZATIONAUDITLOG	Stores detailed organization-specific UDS audit logging information.

Table Name	Description
ARUDSORGREPOATTRIBUTES	Stores the organization-specific repository mapping information. For example, if you are using LDAP as the user repository, then a CA attribute (say FNAME) might be mapped to a corresponding LDAP attribute (say GIVENNAME).
ARUDSORGSECUREATTRIBUTES	Stores organization-specific attributes that need to be encrypted, such as PII fields. Note: These attributes are configured using the Administration Console.
ARUDSREPOCLONESTATUS	Stores the status of temporary cloning of user information from an external repository (such as LDAP) to the ARUDSREPOSITORYUSER table.
ARUDSREPOSITORYTYPES	Stores the definitions of all repositories supported by UDS.
ARUDSREPOSITORYUSER	Temporarily stores user information from an external repository (such as LDAP) to increase performance. This is typically done when user data for a large number of users must be retrieved from the external repository.
ARUDSRESOURCESCOPE	Stores the resource-to-organization mapping. In other words, this table specifies which resource is applicable for which organizations. For example, specific account types might be applicable only for specific organizations.
ARUDSRESOURCESCOPEALL	Stores the resource-to-organization mapping. However, it is different from the ARUDSRESOURCESCOPE table, because it specifies which resource is applicable for <i>all</i> organizations.
ARUDSSECUREATTRIBUTES	Stores information related to attributes that need to be encrypted, such as PII fields. Note: These attributes are configured using the Administration Console.
ARUDSUSER	Stores user details and attributes of all users who belong to the organization.
ARUDSUSER_ARCHIVE	Stores user details for all user accounts that have been deleted from the system.
ARUDSUSERACCOUNT	Stores user account information for specific users.

Table Name	Description
ARUDSUSERACCOUNT_ARCHIVE	Stores user account information for all user accounts that have been deleted from the system.
ARUDSUSERATTRIBUTE	Stores all user attribute definitions. This table is expected to change rarely, only when new user attributes are added by individual products.
ARUDSUSERAUDITLOG	Stores user operation-specific detailed audit logging information.
ARUDSUSERCONTACT	Stores the secondary contact information (such as email or telephone numbers) for users.
ARUDSUSERCONTACT_ARCHIVE	Stores the secondary contact information (such as email or telephone numbers) for the user accounts that have been deleted from the system.

Database Sizing Calculations

This section helps the database administrator to calculate the approximate size of the database that has to be set up for AuthMinder.

Denotations Used in Sample Calculations

The following denotations are used in the calculations:

- Number of users = N
- Average number of transactions per day = T
- Computation timeframe (in days) = D

Value Assumptions

The following assumptions have been made for calculation purposes:

- Number of users (**N**) = 1,000,000 (one million)
- Average number of transactions per day (**T**) = 24,000
- Computation timeframe (**D**) = 90 days

Sample Calculation Based on Assumptions

Considering the figures assumed in section, "[Value Assumptions](#)" (see page 220) the final requirement should be:

- Based on **total number of users**, the database size = $(21 * N)$ KB
- Based on **daily activity**, the database size = $(T * D * 5)$ KB

Database Tables Replication Advice

This section provides information on how frequently the tables must be replicated between the primary and the backup databases. It covers the following topics:

- [Tables That Need Real-Time Synchronization](#) (see page 221)
- [Tables That Need Periodic Synchronization](#) (see page 222)
- [Tables That Do Not Need Synchronization](#) (see page 225)

Tables That Need Real-Time Synchronization

The following table lists the database tables that need real-time synchronization between the primary and the backup databases. This category mainly includes the tables that contain user-related information and this data is required for authentication, therefore you must perform real-time synchronization of these tables.

AuthMinder Component	Table
Administration Console	ARADMINAUDITTRAIL
	ARADMINBASICAUTHUSER
	ARADMINSCOPE
	ARADMINSCOPEALL
	ARADMINUSER
	ARADMINTXID
	ARPCFMNINSTANCE
	ARSEQUENCETABLE
User Data Service	ARUDSORGANIZATION
	ARUDSORGREPOATTRIBUTES
	ARUDSORGSECUREATTRIBUTES
	ARUDSLDAPREPOSITORYCONFIG

AuthMinder Component	Table
	ARUDSACCOUNTTYPE
	ARUDSRESOURCESCOPE
	ARUDSRESOURCESCOPEALL
	ARUDSATTRMAP
	ARUDSCONTACTTYPE
	ARUDSUSER
	ARUDSUSERACCOUNT
	ARUDSCUSTOMATTREXT
	ARUDSAUTHSESSION
	ARUDSUSERCONTACT
	ARUDSREPOSITORYUSER
WebFort Server	ARWFARCOTEMV
	ARWFARCOTID
	ARWFARCOTOTP
	ARWFAUTHTOKENS
	ARWFINSTANCES
	ARWFGENERICCRED
	ARWFOATH
	ARWFOTP
	ARWFQNA
	ARWFPASSWORD
	ARWFVERIFIEDCHALLENGES

Tables That Need Periodic Synchronization

The following table lists the database tables that need periodic synchronization between the primary and the backup databases. These database tables are synchronized when there is any change in the configurations.

Component	Table
Administration Console	ARADMINCONFIG

Component	Table
	ARADMINCUSTOMROLE
	ARADMINMAP
	ARADMINPAFCONFIG
	ARADMINPWDPOLICY
	ARADMINBASICAUTHPWDHISTORY
	ARADMINTURNEDOFFPRIVILEGE
	ARADMINCACHEREFRESH
	ARADMINAUDITTRAIL
	ARADMINUSER_ARCHIVE
	ARADMINMANAGEROLE
	ARADMINROLEPRIVILEGE
	ARPCMNORGCONFIGDATA
	ARPCMNORGCONFIGSTATE
	ARPCMNCACHEREFRESHSTATUS
	ARPCMNCACHEREFRESHEVENT
	ARPCMNCACHEREFRESHSCOPE
	ARCMNBULKTASKPARAM
	ARCMNBULKUPLOADTASK
	ARCMNBULKREQUEST
	ARCMNBULKOPERATIONATTRIBUTE
	ARCMNBULKOPERATION
UDS	ARUDSUSERAUDITLOG
	ARUDSORGANIZATIONAUDITLOG
	ARUDSCONFIGAUDITLOG
	ARUDSCONFIG
	ARUDSREPOSITORYTYPES
	ARUDSUSERATTRIBUTE
	ARUDSUSERACCOUNT_ARCHIVE
	ARUDSCUSTOMATTREXT_ARCHIVE

Component	Table
	ARUDSUSER_ARCHIVE
	ARUDSUSERCONTACT_ARCHIVE
	ARCMNCONFIG
	ARUDSREPOCLONESTATUS
	ARUDSCALLOUTINTERNAL
	ARUDSCALLOUTINTERNALPARAMS
	ARUDSCALLOUT
	ARUDSCALLOUTPARAM
WebFort	ARWFADMINAUDITLOG
	ARWFARCOTEMVHISTORY
	ARWFARCOTIDHISTORY
	ARWFARCOTOTPHISTORY
	ARWFAUTHAUDITLOG
	ARWFCONFIG
	ARWFGENERICCREDHISTORY
	ARWFISSUANCEAUDITLOG
	ARWFMODULEREGISTRY
	ARWFOATHHISTORY
	ARWFOATHTOKENREGISTRY
	ARWFORGACTIVECONFIG
	ARWFORGCONFIG
	ARWFPASSWORDHISTORY
	ARWFPROTOCOLCONFIGURATION
	ARWFQNAHISTORY
	ARWFSEQUENCE
	ARWFSSSLTRUSTSTORE
	ARWFSVRMGMTAUDITLOG
	ARWFUNIQUEFIELDS

Tables That Do Not Need Synchronization

The following table lists the database tables that do not need any synchronization between the primary and the backup databases:

Component	Table
Administration Console	ARCMNDBERRORCODES
	ARADMINAUTHTOKEN
	ARADMINMANAGEROLE
	ARADMINPREDEFINEDROLE
	ARADMINSUPPORTEDAUTHMECH
	ARADMINSUPPORTEDTIMEZONE
	ARADMINUITAB
	ARADMINUITASK
	ARADMINUITASKATTRIBUTES
	ARADMINUITASKCONTAINER
	ARADMINWIZARDTASK
	ARCMNMAPDATATYPE
	ARCMNCACHEREFRESH
	ARPCMNPRIVILEGEMAPPING
ARREPORTTABLES	
User Data Service	ARUDSSECUREATTRIBUTES
WebFort	ARWFDBERRORCODES
	ARWFDBQUERIES
	ARWFDISPLAYNAMES
	ARWFMESSAGES

Database Tuning Parameters

The following table lists the common parameters that you can use to tune the connection between AuthMinder Server and the database. These configurations are made using the Instance Management screen of the Administration Console.

Field	Description
Minimum Connections	Enter the minimum number of connections that will be created between AuthMinder Server and the database when the server starts up.
Maximum Connections	Enter the maximum number of connections that can be created between the AuthMinder Server and the database. Note: You must set this value depending on the maximum connections that the database supports, because this overrides the MaxConnections parameter. See your database vendor documentation for more information.
Increment Connections By	Enter the number of connections that will be added to the existing connections at a time, when the need arises. Important! The total number of connections <i>cannot</i> exceed the maximum number of connections.
Monitor Thread Sleep Time (in Seconds)	Enter the time for which the monitoring thread will sleep between successive heartbeat checks for all databases.
Monitor Thread Sleep Time in Fault Condition (in Seconds)	Enter the interval at which the database monitor thread will check the health of the connection pool in case of faulty database connections.
Log Query Details	Enable this option if you want to log the details for all database queries.
Monitor Database Connectivity	Enable checking of the pools proactively in the database monitor thread.
Auto-Revert to Primary	Enable this option if you want server to switch from the backup database to the primary database when the primary database becomes available again after a failover condition.

Appendix F: Configuring Alternate Schema for IBM DB2 Universal Database

IBM DB2 Universal Database (UDB) enables you to use a different schema other than the default schema. This schema is referred to as alternate schema. This appendix explains the steps that you must perform to set up and use the alternate schema. It covers the following topics:

- [Creating Schema](#) (see page 228)
- [Editing Configuration File](#) (see page 229)
- [Configuring ODBC DSN](#) (see page 229)

Creating Schema

Perform the following steps to set up the alternate schema:

1. Log in to IBM DB2 UDB database.

If the login name is arcotuser, then the default schema for this user will be arcotuser.

2. Create an alternate schema.

For example, the alternate schema can be arcotalternateuser.

Note: Refer to IBM DB2 UDB database documentation for more information how to create the alternate schema.

3. Run the following queries to set up the alternate schema (arcotalternateuser):

```
set current schema ARCOTALTERNATEUSER  
set current path ARCOTALTERNATEUSER
```

Note: The alternate schema *must* be specified in upper case.

4. Navigate to the following location:

```
<install_location>/arcot/dbscripts/db2
```

5. Run the *scripts in the following order*:

- a. arcot-db-config-for-common-2.0.sql

- b. arcot-db-config-for-webfort-7.1.sql

Note: If you encounter any error while executing the scripts, then check with your database administrator whether you have the required privileges.

6. Ensure that the database user name (arcotuser) and the corresponding password are set in the securestore.enc file. If not, you need to use the DBUtil tool to insert the database user name and password.

Refer to chapter, "Tools for System Administrators" in *CA AuthMinder Installation and Deployment Guide* for more information.

Editing Configuration File

If you plan to use alternate schema for IBM DB2 UDB, then you must edit the arcotcommon.ini file to configure these settings. Perform the following steps to do so:

1. Navigate to the following locations:
`<install_location>/arcot/conf`
2. Open the arcotcommon.ini file in a text editor.
3. In the [arcot/db/primarydb] and [arcot/db/backupdb] sections, set the following parameters:
 - **URL.1:** Set this parameter to the JDBC data source, as follows:
`jdbc:db2://<server>:<database_port>/<database>:currentSchema=<alternateID>;currentFunctionPath="SYSIBM", "SYSFUN", "SYSPROC", "SYSIBMADM", "<alternateID>"`

 For example
`jdbc:db2://db2server:50000/arcotdb:currentSchema=ARCOTALTERNATEUSER;currentFunctionPath="SYSIBM", "SYSFUN", "SYSPROC", "SYSIBMADM", "ARCOTALTERNATEUSER"`
 - **Username.1:** Set this parameter to the user name that is used to access the database. For example, arcotuser.
4. Save and close the arcotcommon.ini file.

Configuring ODBC DSN

AuthMinder Server uses the Open Database Connectivity (ODBC) to connect to the database. You must edit the ODBC settings for AuthMinder Server to connect to the database using the alternate schema.

Perform the following steps to edit the ODBC settings:

1. Navigate to the following location:
`<install_location>/arcot/odbc32v70wf`
2. Open the odbc.ini file in a file editor.
3. In the [*<Database_name>*] section that corresponds to the database you are using, you must edit the parameters as listed in the following table:

Parameter	Description
AlternateID	The alternate schema created in Step 2.
CurrentFuncPath	Set this field to SYSIBM, SYSFUN, SYSPROC, SYSIBMADM, <i><Alternate_ID></i> . For example, SYSIBM, SYSFUN, SYSPROC, SYSIBMADM, ARCOTALTERNATEUSER.

4. Save and close the odbc.ini file.

Appendix G: Default Port Numbers and URLs

This appendix lists the default port numbers that AuthMinder uses. It contains the following sections:

- [Default Port Numbers](#) (see page 231)
- [URLs for AuthMinder Components](#) (see page 232)

Default Port Numbers

AuthMinder supports four protocols that are configured at different ports. The following table lists the default port numbers used by AuthMinder:

Protocol	Default Port Number	Default Status	Description
Server Management Web Services	9743	Enabled	This protocol is used for managing AuthMinder server. Administration Console and arwfutil clients communicate using this port for server management activities.
Transaction Web Services	9744	Enabled	This protocol is used by the Authentication and the Issuance Web services clients to connect to AuthMinder Server.
Transaction Native	9742	Enabled	This is a proprietary, binary protocol used by AuthMinder for the purpose of authentication and credential issuance. This port is used by Authentication and Credential Management SDKs.
Administration Web Services	9745	Enabled	This protocol is used to create and manage configurations, such as profiles, policies, SAML, and ASSP configurations.
RADIUS	1812	Disabled	This is used to support the Remote Authentication Dial In User Service (RADIUS) protocol. When configured to support RADIUS protocol, AuthMinder server acts as a RADIUS server.

Protocol	Default Port Number	Default Status	Description
ASSP	9741	Disabled	This protocol is used with Adobe® Reader and Adobe® Acrobat® to authenticate user for server-side digital signing of the PDF documents.
Transaction HTTP	9746	Disabled	This protocol is used to transfer the HTTP request packets from the HTTP Client to the AuthMinder Server.

If some other service is already running on the default port, then you must set a new port for the protocols, as follows:

- **To set a new port number for Server Management Web Services protocol**, use webfortserver tool.

Note: Refer to the topic titled "Tools for System Administrators" in *CA AuthMinder Administration Guide* for more information on the tool.
- **To set a new port number for other protocols**, use Protocol Configuration screen in the Administration Console.

Note: Refer to the topic titled "Managing WebFort Server Instances" in *CA AuthMinder Administration Guide* for more information on changing the port numbers.

URLs for AuthMinder Components

Use the URLs listed in the following table to access AuthMinder components after installation:

Component or Service	URL
Administration Console URL for Master Administrator	<p><code>http://<Apphost>:<app_server_port>/arcotadmin/masteradminlogin.htm</code></p> <p>Note: Apphost refers to the system where Administration Console is deployed. App_Server_Port refers to the port number of the application server on which the Administration Console is deployed.</p>
Administration Console URL for other administrators	<p><code>http://<Apphost>:<app_server_port>/arcotadmin/adminlogin.htm</code></p> <p>Note: Apphost refers to the system where Administration Console is deployed. App_Server_Port refers to the port number of the application server on which the Administration Console is deployed.</p>

Component or Service	URL
Organization Management Web Service	<p>http://<Apphost>:<app_server_port>/arcotuds/services/ArcotUserRegistryMgmtSvc</p> <p>Note: Apphost refers to the system where User Data Service (UDS) Console is deployed. App_Server_Port refers to the port number of the application server on which UDS is deployed.</p>
User Management Web Services	<p>http://<Apphost>:<app_server_port>/arcotuds/services/ArcotUserRegistrySvc</p> <p>Note: Apphost refers to the system where User Data Service (UDS) Console is deployed. App_Server_Port refers to the port number of the application server on which UDS is deployed.</p>
Configuration Registry Web Service	<p>http://<Apphost>:<app_server_port>/arcotuds/services/ArcotConfigRegistrySvc</p> <p>Note: Apphost refers to the system where User Data Service (UDS) Console is deployed. App_Server_Port refers to the port number of the application server on which UDS is deployed.</p>
Credential Management Web Services	<p>http://<Apphost>:<Protocol_port>/WebFortIssuanceSvc</p> <p>Note: Apphost refers to the system where AuthMinder Server is installed. Protocol Port refers to the port number of the Transaction Web Services protocol. By default, this value is 9744.</p>
Administration Web Service	<p>http://<Apphost>:<Protocol_port>/ArcotWebFortAdminSvc</p> <p>Note: Apphost refers to the system where AuthMinder Server is installed. Protocol Port refers to the port number of the Administration Web Services protocol. By default, this value is 9745.</p>
Authentication Web Services	<p>http://<Apphost>:<Protocol_port>/WebFortAuthSvc</p> <p>Note: Apphost refers to the system where AuthMinder Server is installed. Protocol Port refers to the port number of the Transaction Web Services protocol. By default, this value is 9744.</p>

Component or Service	URL
Bulk Operation Web Service	<p><i>http://<Apphost>:<Protocol_port>/WebFortBulkOperationsSvc</i></p> <p>Note: Apphost refers to the system where AuthMinder Server is installed.</p> <p>Protocol Port refers to the port number of the Administration Web Services protocol. By default, this value is 9745.</p>
Sample Application	<p><i>http://<Apphost>:<app_server_port>/webfort-7.1-sample-application/</i></p> <p>Note: App_Server_Port refers to the port number of the application server on which the Sample Application is deployed.</p>

Appendix H: Configuring Application Server

This appendix outlines the steps that you must perform to set up the database connection pooling on the application server, where the AuthMinder components will be deployed, lists the steps for setting up connection pool used by LDAP, enable Apache Tomcat security manager. It covers the following topics:

- [Enabling Database Connection Pooling](#) (see page 235)
- [Enabling LDAP Connection Pooling](#) (see page 242)
- [Enabling Apache Tomcat Security Manager](#) (see page 246)

Enabling Database Connection Pooling

Typically, accessing the database might not be a bottleneck, but setting up a new connection for each request can be an overhead and can bring down the performance of the system. By creating a database connection pool, you can avoid potential bottlenecks. This is because connection pooling helps you avoid the overhead of making a new database connection every time a AuthMinder component that is deployed on application server requires access to the database.

This section covers the configuration steps for the following application servers:

- [Apache Tomcat](#) (see page 236)
- [IBM WebSphere](#) (see page 238)
- [Oracle WebLogic](#) (see page 240)
- [JBoss](#) (see page 241)

Apache Tomcat

This section provides the steps to enable Apache Tomcat for JNDI-based database operations.

Perform the following steps to create a JNDI connection in Apache Tomcat:

1. Install the Apache Tomcat application server and test the installation using the following URL:

http://localhost:8080/

The preceding URL must open the Apache Tomcat home page.

2. Open the `server.xml` file present in the `<TOMCAT-HOME>/conf` directory.
3. Collect the following information required to define a data source:

- JNDI Name

The JNDI name used by the Arcot components. This name must match with the `AppServerConnectionPoolName.N` in `arcotcommon.ini` (without the `java:comp/env/` prefix).

- User ID

The database user ID.

- Password

The database password.

- JDBC Driver Class

The JDBC driver class name. For example, `oracle.jdbc.driver.OracleDriver`.

- JDBC URL

The JDBC URL for the database server. For example, if you are using Oracle driver, then URL would be: `jdbc:oracle:thin:@<server>:<database_port>:<sid>`.

4. Add the following entry to define the `datasource` within the `<GlobalNamingResources>` tag:

```
<Resource name="SampleDS"
auth="Container"
type="javax.sql.DataSource"
factory="org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory"
username="<userid>"
password="<password>"
driverClassName="<JDBC driver class>"
url="<jdbc-url>"
maxWait="30000"
maxActive="32"
maxIdle="8"
initialSize="4"
timeBetweenEvictionRunsMillis="300000"
minEvictableIdleTimeMillis="30000"/>
```

5. Open the context.xml file present in the `<TOMCAT-HOME>/conf` directory.
6. Add the following entry to define the datasource within the `<Context>` tag:
`<ResourceLink global="SampleDS" name="SampleDS"
type="javax.sql.DataSource"/>`
7. Copy the following database connection pooling (DBCP) dependencies to `<TOMCAT-HOME>/common/lib` directory.
 - commons-dbc1.2.2.jar
 - ojdbc14-10.2.0.1.0.jar (for Oracle database)
 - sqljdbc.jar (Microsoft JDBC driver for MS SQL Server 2005 - version 1.2.2828)
 - db2jcc-9.5.jar (for DB2 database)
 - mysql-connector-java-5.1.22-bin.jar (for MySQL)

IBM WebSphere

This section provides the steps to enable IBM WebSphere for JNDI-based database operations:

Perform the following steps to configure IBM WebSphere for deploying Java-dependant components of AuthMinder

1. Log in to WebSphere Administration Console.
2. Click **Resources** and expand the **JDBC** node.
3. Click **JDBC Providers**.

The JDBC Providers page appears.

4. In the **Preferences** section, click **New**.

The Create a new JDBC Provider page appears.

5. Perform the following steps to create a JDBC provider:

Note: Refer to

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.base.iseries.doc/info/iseres/ae/tdat_crtprov.html for more information on JDBC providers.

- a. Specify the **Database Type** and **Provider Type**.
 - b. Select **Connection pool data source** from the **Implementation Type** drop-down list.
 - c. Enter a **Name** for the JDBC provider. You can also enter a **Description** for the JDBC provider.
 - d. Click **Next**.
The Enter database class path information screen appears.
 - e. Enter the absolute path for the JAR file.
 - f. Click **Next**.
The Summary screen appears.
 - g. After reviewing the summary of the information that you have entered, click **Finish**.
6. Set the CLASSPATH for the JDBC provider that you created in Step 5.
 - a. Click **Resources** and expand the **JDBC** node.
 - b. Click **JDBC Providers**.
The JDBC Providers page appears.
 - c. Click the JDBC provider that you created in Step 5.
 - d. Set the **Class Path** for the JDBC JAR.
 - e. Click **Apply** to save the changes.

7. Create a Data Source, as follows:
 - a. Go to **Resources**, and then click **JDBC**.
 - b. Under **JDBC**, open **Data Sources** and click **New**. Perform the following steps to create a data source:
 - c. Specify the **Data source name**.
 - d. Specify the **JNDI name**. This name must match with the `AppServerConnectionPoolName.N` in `arcotcommon.ini`.
 - e. Click **Next**.
 - f. **Select an existing JDBC provider** created in Step 3.
 - g. Click **Next**.

The Enter database specific properties for the data source screen appears.
 - h. Depending on the database, enter the following information:
 - **For Oracle:**

Specify the **Value** for JDBC URL. This URL would be of the following type:
`jdbc:oracle:thin:@<server>:<oracle_port>:<sid>`

Select the **Data store helper class name**.
 - **For MS SQL Server:**
`jdbc:sqlserver://<server>:<sql_port>;databaseName=<database name>;selectMethod=cursor`
 - **For IBM DB2:**
`jdbc:db2://<server>:<db2_port>/<database>`
 - **For MySQL:**
`jdbc:mysql://<server>:<mysql_port>/<database>`
 - i. Click **Next**.

The Setup Security aliases screen appears.
 - j. Click **Next** to view the Summary screen, and then click **Finish**.
8. Click the data source created in Step 7.
9. In the **Related Items** section, click **JAAS - J2C authentication data**.
10. Click **New** to create a new credential.
11. Enter login credentials that are used to connect to the database and save the credential.
12. Click **Apply**, and then click **OK** to save the changes made.
13. Click **Data Sources** and select the data source that you created in Step 7.
14. Under **Security Settings** -> **Component-managed authentication alias**, select the JAAS credential that you created in Step 11 and click **Apply**, and then **OK**.

15. Click **Data Sources** and select the check box for the data source you created in Step 7.
16. Click **Test connection** to verify that you have specified the connection correctly.

Note: This test only checks the connection to the database server, not necessarily the correct definition of the data source.

Oracle WebLogic

This section provides the steps to enable Oracle WebLogic for JNDI-based database operations.

Perform the following steps to create a data source in Oracle WebLogic:

1. Log in to WebLogic Administration Console.
2. Click the **Lock & Edit** button, if it is not done.
3. Go to **Resources**, and then click **JDBC**.
4. Under **JDBC**, open **Data Sources** and click **New** to create a new data source. Perform the following steps to create a data source.
5. Set the following JNDI and the database information:
 - a. Set **Name** = ArcotDB.
 - b. Set **JNDI Name** = ArcotDB.
 - c. Choose a suitable **Database Type**, for example Oracle.
 - d. Select a suitable **Database Driver**, for example Oracle Thin Driver.
6. Click **Next**, retain the default values and click **Next** again.
7. In the Connection Properties page, set the database details. The values mentioned here are for Oracle database:
 - **Database:** SID or service name of the DB server
 - **Hostname:** The DB server's IP address or host name
 - **Port:** 1521 or any other port the DB server is running
 - **Database User Name**
 - **Database Password / Confirm Password**
8. Click **Test Configuration** to verify that you have specified the correct database parameters.
9. Click **Next** and set the data source target to the preferred WebLogic server instance.
10. Click **Finish** to return to the data source list page.
11. Click **Activate** to enable the data source settings.

JBoss

This section provides the steps to enable JBoss for JNDI-based database operations.

1. Navigate to the location where you have deployed the WAR files. For example:


```
<JBOSS_HOME>/server/default/deploy
```
2. Create a data source descriptor file called `arcotdatabase-ds.xml`.
3. Collect the following information required to define a data source in the `arcotdatabase-ds.xml` file:
 - **JNDI Name**
The JNDI name used by the Arcot components. This name must match with the `AppServerConnectionPoolName.N` in `arcotcommon.ini` (without the `java:comp/env/` prefix).
 - **User ID**
The database user ID.
 - **Password**
The database password.
 - **JDBC Driver Class**
The JDBC driver class name. For example, `oracle.jdbc.driver.OracleDriver`.
 - **JDBC URL**
The JDBC URL for the database server. For example, if you are using Oracle driver, then URL would be: `jdbc:oracle:thin:<server>:<database_port>:<sid>`.
 - **Exception Sorter Class**
Class that implements the `org.jboss.resource.adapter.jdbc.ExceptionSorter` interface, which determines whether the exception indicates a connection error.

Use this parameter for Oracle database *only*. Set it to `org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter`.

4. Open the `arcotdatabase-ds.xml` in a text editor.

5. Add the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
<local-tx-datasource>
<jndi-name>SampleDS</jndi-name>
<connection-url><jdbcurl></connection-url>
<driver-class><JDBC Driver class></driver-class>
<user-name><database_userid></user-name>
<password><database_password></password>
<exception-sorter-class-name><Exception Sorter
Class></exception-sorter-class-name>
</local-tx-datasource>
```

```
</datasources>
```

6. Save and close the file.

Enabling LDAP Connection Pooling

This section covers the configuration steps for enabling LDAP connection pool for the following application servers:

- [Apache Tomcat](#) (see page 242)
- [IBM WebSphere](#) (see page 243)
- [Oracle WebLogic](#) (see page 243)
- [JBoss](#) (see page 245)

Apache Tomcat

Perform the following steps to create a LDAP connection pool:

1. Install the Apache Tomcat application server and test the installation using the following URL:

```
http://localhost:8080/
```

The preceding URL must open the Apache Tomcat home page.

2. Navigate to the following location:
`<TOMCAT-HOME>/conf`
3. Open the `catalina.properties` file in a text editor.
4. Add the following entries to the file:
 - `com.sun.jndi.ldap.connect.pool.protocol=plain ssl`
 - `com.sun.jndi.ldap.connect.pool.authentication=simple`
 - `com.sun.jndi.ldap.connect.pool.maxsize=64`
 - `com.sun.jndi.ldap.connect.pool.prefsiz=32`
 - `com.sun.jndi.ldap.connect.pool.timeout=240000`
 - `com.sun.jndi.ldap.connect.pool.initsize=8`
5. Save and close the file.
6. Restart the application server.

IBM WebSphere

Perform the following steps to create a LDAP connection pool:

1. Log in to WebSphere Administration Console.
2. Navigate to **Servers -> Server Types -> WebSphere application servers**.
3. The Application servers page appears.
4. Click the Server that you want to configure.
5. In the **Server Infrastructure** section, click **Java and Process Management**.
6. Click the **Process Definition** link.
7. In the **Additional Properties** section, click **Java Virtual Machine**.
8. In the **Additional Properties** section, click **Custom Properties**.
9. Click **New** to add custom properties.

The **General Properties** section appears.

10. Add the configurations listed in the following table as name-value pairs in the **General Properties** section. You have to repeat the process for every name-value pair:

Name	Value
com.sun.jndi.ldap.connect.pool.maxsize	64
com.sun.jndi.ldap.connect.pool.prefsize	32
com.sun.jndi.ldap.connect.pool.initsize	8
com.sun.jndi.ldap.connect.pool.timeout	240000
com.sun.jndi.ldap.connect.pool.protocol	plain ssl
com.sun.jndi.ldap.connect.pool.authentication	simple

11. Click **Apply**.
12. Restart WebSphere.

Oracle WebLogic

This section covers the following topics for enabling LDAP connection pool for Oracle WebLogic:

- [Including LDAP Options in Startup Script](#) (see page 244)
- [Specifying LDAP Pool Options Using Managed Server](#) (see page 245)

Including LDAP Options in Startup Script

This section provides the steps to include the LDAP connection pool parameters in WebLogic server startup script:

1. Log in to the system
2. Create a backup copy of the WebLogic Server startup script. This script is available at the following location:
domain-name/bin/startWebLogic.sh
3. Open the script in a text editor.
4. Add the following entries in the section that is used to start the WebLogic server.
 - -Dcom.sun.jndi.ldap.connect.pool.maxsize=64
 - -Dcom.sun.jndi.ldap.connect.pool.prefsiz=32
 - -Dcom.sun.jndi.ldap.connect.pool.initsize=8
 - -Dcom.sun.jndi.ldap.connect.pool.timeout=240000
 - -Dcom.sun.jndi.ldap.connect.pool.protocol="plain ssl"
 - -Dcom.sun.jndi.ldap.connect.pool.authentication=simple

The following code snippet shows a sample script with LDAP connection pool parameters:

```
# START WEBLOGIC
echo "starting weblogic with Java version:"
${JAVA_HOME}/bin/java ${JAVA_VM} -version
if [ "${WLS_REDIRECT_LOG}" = "" ] ; then
echo "Starting WLS with line:"
echo "${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
-Dweblogic.Name=${SERVER_NAME} -Djava.security.poli
cy=${WL_HOME}/server/lib/weblogic.policy ${PROXY_SETTINGS} ${SERVER_CLASS}"
${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
-Dcom.sun.jndi.ldap.connect.pool.maxsize=64
-Dcom.sun.jndi.ldap.connect.pool.prefsiz=32
-Dcom.sun.jndi.ldap.connect.pool.initsize=8
-Dcom.sun.jndi.ldap.connect.pool.timeout=240000
-Dcom.sun.jndi.ldap.connect.pool.authentication=simple
-Dcom.sun.jndi.ldap.connect.pool.protocol="plain ssl"
-Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy
${PROXY_SETTINGS} ${SERVER_CLASS}
else
echo "Redirecting output from WLS window to ${WLS_REDIRECT_LOG}"
${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
-Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy
${PROXY_SETTINGS} ${SERVER_CLASS} >"${WLS_REDIRECT_LOG}" 2>&1
fi
```

5. Save the close the file.
6. Restart WebLogic server.

Specifying LDAP Pool Options Using Managed Server

1. Log in to WebLogic Administration Console.
2. Click the **Lock & Edit** button, if it is not done.
3. In the **Domain Structure** pane, Navigate to **Environment > Servers**.
4. Click the server you want to configure.
5. In the right pane, click **Server Start**.
6. In the **Arguments** field, include the following JVM options:
 - -Dcom.sun.jndi.ldap.connect.pool.maxsize=64
 - -Dcom.sun.jndi.ldap.connect.pool.prefsiz=32
 - -Dcom.sun.jndi.ldap.connect.pool.initsize=8
 - -Dcom.sun.jndi.ldap.connect.pool.timeout=240000
 - -Dcom.sun.jndi.ldap.connect.pool.protocol="plain ssl"
 - -Dcom.sun.jndi.ldap.connect.pool.authentication=simple
7. Click **Save** and then **Activate Changes**.
8. Restart WebLogic server.

JBoss

Perform the following steps to create a LDAP connection pool:

1. Navigate to the following location:
`<JBoss_HOME>/server/<Profile>/deploy/`
2. Open properties-service.xml file in a text editor.
3. Add the following properties to the <attribute name="Properties"> section:
 - com.sun.jndi.ldap.connect.pool.protocol=plain ssl
 - com.sun.jndi.ldap.connect.pool.authentication=simple
 - com.sun.jndi.ldap.connect.pool.maxsize=64
 - com.sun.jndi.ldap.connect.pool.prefsiz=32
 - com.sun.jndi.ldap.connect.pool.timeout=240000
 - com.sun.jndi.ldap.connect.pool.initsize=8
4. Save and close the file.
5. Restart JBoss.

Enabling Apache Tomcat Security Manager

Perform the following steps to enable Tomcat Security Manager:

1. Add the security manager entries to the **JAVA_OPTS** environment variable, as follows:

```
export CATALINA_OPTS="-Djava.security.manager
-Djava.security.policy=<Tomcat_Home>/conf/catalina.policy"
```
2. Navigate to the following Apache Tomcat installation location:
`<Tomcat_Home>/conf/`
3. Open `catalina.policy` file in a text editor.
4. Add the following code in the **WEB APPLICATION PERMISSIONS** section.

```
grant {
permission java.io.FilePermission
"${catalina.base}${file.separator}webapps${file.separator}arcotuds${file.sepa
rator}-", "read";
permission java.util.PropertyPermission "adb.converterutil", "read";
permission java.lang.RuntimePermission "accessDeclaredMembers";
permission java.security.SecurityPermission "putProviderProperty.BC";
permission java.security.SecurityPermission "insertProvider.BC";
permission java.security.SecurityPermission "putProviderProperty.SHAProvider";
permission java.io.FilePermission "${arcot.home}${file.separator}-",
"read,write";
permission java.net.SocketPermission "*:1024-65535", "connect,accept,resolve";
permission java.net.SocketPermission "*:1-1023", "connect,resolve";
};
```
5. Add the following section to grant permission for Administration Console (arcotadmin) and User Data Service (arcotuds).

```
grant codeBase "file:${catalina.home}/webapps/arcotuds/-" {
permission java.lang.RuntimePermission "getenv.ARCOT_HOME", "";
permission java.lang.RuntimePermission
"accessClassInPackage.org.bouncycastle.asn1.*";
permission java.security.AllPermission;
};
grant codeBase "file:${catalina.home}/webapps/arcotadmin/-" {
permission java.lang.RuntimePermission "getenv.ARCOT_HOME", "";
permission java.security.AllPermission;
};
```
6. Save and close the file.
7. Restart Apache Tomcat.

Appendix I: Deploying Administration Console on IBM WebSphere 7.0

If you plan to deploy Administration Console on IBM WebSphere 7.0, then perform the following steps:

1. Change the working directory to:
`<install_location>/arcot/sbin`
2. Type source arwfenv and press **Enter** to set the \$ARCOT_HOME environment variable.
3. Restart the application server for the changes to take effect.
4. Navigate to the following location where the Administration Console WAR file is located:
`<install_location>/arcot/java/webapps`
5. Copy the arcotadmin.war file to a temporary directory. For example, `opt/arcot_temp`.
6. Extract the arcotadmin.war file contents.

Of the JARs that are extracted to the `/opt/arcot_temp/WEB_INF/lib` directory, the following JARs are used to create the shared library in IBM WebSphere:

- axiom-api-1.2.10.jar
 - axiom-impl-1.2.10.jar
 - axis2-java2wsdl-1.5.2.jar
 - backport-util-concurrent-3.1.jar
 - commons-httpclient-3.1.jar
 - commons-pool-1.5.5.jar
 - axiom-dom-1.2.10.jar
 - axis2-adb-1.5.2.jar
 - axis2-kernel-1.5.2.jar
 - commons-codec-1.3.jar
 - commons-logging-1.1.1.jar
 - log4j-1.2.16.jar
 - axis2-transport-http-1.5.2.jar
 - axis2-transport-local-1.5.2.jar
7. Log in to IBM WebSphere Administration Console.

8. Click Environment, and then click Shared Libraries.
 - a. From the Scope drop-down, select a valid visibility scope. The scope must include the target server/node on which the application is deployed.
 - b. Click New.
 - c. Enter the Name, for example, ArcotAdminSharedLibrary.
 - d. Specify the Classpath. Enter the path and file name for all the JAR files extracted in Step 3.

For example, /opt/arcot_temp/WEB_INF/lib/axiom-api-1.2.10.jar
 - e. Click Apply to save the changes made.
9. Navigate to the following location where the Administration Console WAR file is located:
<install_location>/arcot/java/webapps
10. Deploy arcotadmin.war in the application server.
11. Configure shared library, as follows:
 - a. Click Applications, and then click WebSphere enterprise applications.
 - b. Click arcotadmin_war.
 - c. In the References section, click Shared library references.
 - d. Select arcotadmin_war and click Reference shared libraries.
 - e. Select the ArcotAdminSharedLibrary from the Available list and move it to the Selected list.
 - f. Click OK to save the configurations.
12. Configure the class loader order and policy as follows:
 - a. Click Applications, Application Types, and then click WebSphere enterprise applications.
 - b. Click arcotadmin_war.
 - c. Click Class loading and update detection link.
 - d. In the Class loader order section, select the Classes loaded with local class loader first (parent last) option.
 - e. In the WAR class loader policy section, select the Single class loader for application option.
 - f. Click OK to save the configurations.
13. Ensure that the application is restarted.

Appendix J: Troubleshooting AuthMinder Errors

This appendix describes the troubleshooting steps, which will help you resolve the errors that you might face while using AuthMinder. The troubleshooting topics are classified based on different AuthMinder components, as follows:

- [Installation Errors](#) (see page 250)
- [Database-Related Errors](#) (see page 255)
- [AuthMinder Server Errors](#) (see page 257)

Before you perform any troubleshooting tasks, check the AuthMinder log files to see if there were any errors. By default, all the log files are saved in the `<install_location>/arcot/logs/` directory. The following table lists the default log file names of the AuthMinder components.

AuthMinder Component	File Name	Description
AuthMinder Server	arcotwebfortstartu p.log	This file records all the start-up (or boot) actions. The information in this file is very useful in identifying the source of the problems if the AuthMinder service does not start up.
	arcotwebfort.log	This file records all requests processed by the server.
Administration Console	arcotadmin.log	This file records the Administration Console operations.
User Data Service	arcotuds.log	This file records the User Data Service operations.

Note: Refer to the topic titled "AuthMinder Logging" in the *CA AuthMinder Installation and Deployment Guide* for detailed information on the AuthMinder log files.

Installation Errors

Problem:

I do not find arcotadmin.war in *<install_location>/arcot/java/webapps* directory.

Cause:

The file might not have been created during installation.

Solution:

Perform the following steps to create the arcotadmin.war file:

1. Open command window.
2. Ensure ARCOT_HOME environment variable is set.
3. Navigate to *<install_location>/arcot/tools/common/bundlemanager* directory.
4. Run bundlemanager with the following command:
`java -jar bundle-manager.jar`

The preceding command generates the arcotadmin.war file in *<install_location>/arcot/java/webapps* directory.

Problem:

I cannot start the AuthMinder Server (Arcot WebFort Authentication Service). I see the following error in arcotwebfortstartup.log:

Failed to initialize DB Pool Manager

or

Data source name not found and no default driver specified

Cause:

The possible causes for this issue might be:

- The DSN for your database was not created as System DSN.
- You are using a 64-bit platform. As a result, the DSN was created by using the 64-bit ODBC Manager.

Solution:

You can verify the DSN-related issues in the `arcotcommon.ini` file. If the problem is DSN-related, then:

1. To resolve the first cause, you must ensure that the DSN is a System DSN. To do so:
 - a. Open the Control Panel, navigate to Administrative Tools, and Data Sources (ODBC).
 - b. Activate the System DSN tab, and verify that your DSN exists here. If not, then you must re-create the DSN with the same name as earlier.
 - c. Restart the service.
2. To resolve the second issue (if you are using a 64-bit platform), you must use the 32-bit version of the ODBC Manager.

Note: For detailed information on `arcotcommon.ini` and other configuration files, see the topic titled "Configuration Files and Options" in *CA AuthMinder Installation and Configuration Guide*.

Problem:

I cannot start the AuthMinder Server (Arcot WebFort Authentication Service). The error message indicates that the service starts and stops automatically.

Cause:

A possible cause for this issue might be that you specified details for a Database during installation, but the data source was not created successfully.

Solution:

To resolve this issue:

1. Verify if there is a corresponding entry for the DSN in `arcotcommon.ini`.

- If entry not found, manually create the DSN.
2. If you found the entry, then clean up the database (see ["Uninstalling AuthMinder Schema"](#) (see page 167)) and reseed the database, as described in section, ["Running Database Scripts"](#) (see page 70).
 3. Restart the AuthMinder Server.

Problem

When I launch the Administration Console for the first time (section, "Performing the Bootstrapping Tasks") as the Master Administrator, I see the following message: The server encountered an internal error that prevented it from fulfilling this request."

I see the following error in the arcotadmin.log file:
adminLog: java.lang.UnsatisfiedLinkError: no ArcotAccessKeyProvider in java.library.path

Cause:

The JAVA library does not include the path to one of the following files:

- libArcotAccessKeyProvider.so
- arcot-crypto-util.jar

Solution:

Do the following:

1. Ensure that the LD_LIBRARY_PATH variable includes the absolute path to the following files:
 - libArcotAccessKeyProvider.so
 - arcot-crypto-util.jar
2. Restart the application server.

Problem:

I do not see the log file (arcotadmin.log, arcotuds.log, or webfortserver.log) in the logs directory in ARCOT_HOME.

Cause:

Some of the probable causes for this issue might be:

- ARCOT_HOME might not be set correctly during installation.
- The application server JAVA HOME might be pointing to JRE instead of the JDK HOME.

Solution:

To resolve these issues, you must:

- Ensure that you reset the ARCOT_HOME to point to the correct location. Typically, this is `<installation_location>/arcot/`.
As a result of this, when you use the `cd $ARCOT_HOME` command in the command prompt window, your current directory must change to `<installation_location>/arcot/`.
- Ensure that you copy the `libArcotAccessKeyProvider.so` and `arcot-crypto-util.jar` files in the application server JAVA HOME location.

Problem:

I deployed the UDS, but it is not coming up.

Cause:

One of the possible causes might be that the application server JAVA HOME might be pointing to JRE instead of the JDK HOME.

Solution:

Ensure that you have copied the `libArcotAccessKeyProvider.so` and `arcot-crypto-util.jar` files in the application server JAVA HOME location.

Problem:

Connection to the UDS fails. I see the following error message:
Unable to contact User Data Service

Cause:

The possible causes might be:

- You might not have specified the UDS **Host**, **Port**, and **Application Context** information for UDS correctly.
- The UDS service might not be initialized.

Solution:

To resolve this issue:

1. Verify whether the UDS information that you have provided in the "User Data Service Configuration" page of the Administration Console is correct. The details in **Host, Port, and Application Context** fields must be correct.
2. Check the UDS log file to ensure that the service was initialized successfully.

Database-Related Errors

Problem:

Connection to the database fails with the following entry in the AuthMinder Server log file:
ReportError: SQL Error State:08001, Native Error Code: 30FD, ODBC Error: [DataDirect][ODBC Oracle driver][Oracle]ORA-12541: TNS:no listener

Solution:

Check for the following:

- Listener service on your database server.
- The TNSnames.ora file settings on the system where AuthMinder Server is installed.

Problem:

Connection to the database fails with the following entry in the AuthMinder Server log file:
TNS:listener could not resolve SERVICE_NAME given in connect descriptor

Solution:

Check for the following:

- Database is started. If it is not, you will get the above message.
- If the database is running, probably the database has not registered yet with the listener. This occurs when the database or listener just starts up. Normally this problem should be solved by waiting a minute or so.
- If you are using static registration, make sure the SERVICE_NAME entry used in the connection string (TNSNAMES.ORA, NAMES, OID, ...) matches a valid service known by the listener.
- You can use C:>tnsping SERVICE_NAME - to check the status or C:>lsnrctl services - to verify all the services known to listener.

Problem:

Connection to the database fails with the following entry in the AuthMinder Server log file:
Database password could not be obtained from securestore.enc

Cause:

The database details might not be available in securestore.enc file.

Solution:

Use the DBUtil tool to update the securestore.enc file with the database details. Refer to *CA AuthMinder Administration Guide* for more information on how to use DBUtil.

Problem:

Connection to the database fails with the following entry in the AuthMinder Server log file:

```
ORA-03113: end-of-file on communication channel
```

Cause:

This is a generic error that indicates that the connection has been lost. This can be caused by many reasons such as:

- Network issues or problems
- Forceful disconnection of a Server session
- Oracle Database crash
- Database Server crash
- Oracle internal errors, such as ORA-00600 or ORA-07445, causing aborts
- Oracle Client or TNS layer inability to handle the connections

Solution:

Check for the possible causes mentioned in the preceding list.

AuthMinder Server Errors

Problem:

I am trying to start AuthMinder Server, but it is not coming up. The last lines in `arcotwebfortstartup.log` shows the following error:

```
WARN  STARTUP      -161388848 00WFMAIN - [11]: Protocol module  
[SVRMGMT_WS] received portType error [bind: Address already in use]
```

Cause:

The possible cause for this issue is that the Server Management Port (Default Port Number: 9743) is already open on the host by some other process. While, AuthMinder Server requires a minimum of Server Management port to start up.

Solution:

Do the following:

1. Open the command prompt window.
2. Navigate to `$ARCOT_HOME/bin`.
3. Run the following command:
`arwfserver -i`
4. Enter `setsvrnmgmtport <new_port_number>`

After the Server Management port is set, the Master Administrator can log in to the Administration Console and configure the other ports.

Problem:

RADIUS requests fail with the "Access-Reject" message.

Cause:

Check for the following:

- Shared secret is configured correctly.
- AuthMinder Server logs for any errors.

Solution:

Log in to the Administration Console as Global Administrator or Organization and set the shared secret using the RADIUS Configuration page.

Problem:

ArcotID PKI authentication fails with the following entry in the AuthMinder Server log file:
[Arcot Exception, No valid issuer certificate is available for this certificate: unknown or invalid certificate issuer in ArcotVerifier], Challenge verification failed -

Cause:

The Domain Key might have expired.

Solution:

Log in to the Administration Console as Global Administrator or Organization and configure the Domain Key using the Credential Key Management Configuration page.