CA RiskMinder™

Installation and Deployment Guide for Microsoft Windows

r3.1.01

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.

Contents

Chapter 1: Understanding the Basics

CA RiskMinder (referred to as RiskMinder later in the guide) is an adaptive authentication solution that evaluates each online transaction (shopping, banking, or corporate access) in real time by examining a wide range of collected data against the out-of-the-box rules. RiskMinder then assigns each transaction a risk score and advice. The higher the risk score, the greater the possibility of a fraud. Based on your business policies, your application can then use this risk score and advice to approve or decline the transaction, ask for extra authentication, or alert a customer service representative.

RiskMinder is highly configurable, and offers you the flexibility to modify the configuration parameters of any of the risk evaluation rules in keeping with your policies and risk-mitigation requirements. It also gives you the flexibility to modify the default risk score, scoring configuration, and scoring priorities of individual rules and selectively enable or disable the execution of one or more rules.

Besides preconfigured out-of-the-box rules, the Rule Builder capability in RiskMinder enables you to create your rules on the fly.

This guide provides information for planning the deployment of CA RiskMinder 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.

This section introduces you to the basic concepts of RiskMinder, explains its architecture, and then walks you through the features and enhancements that have been introduced in this release:

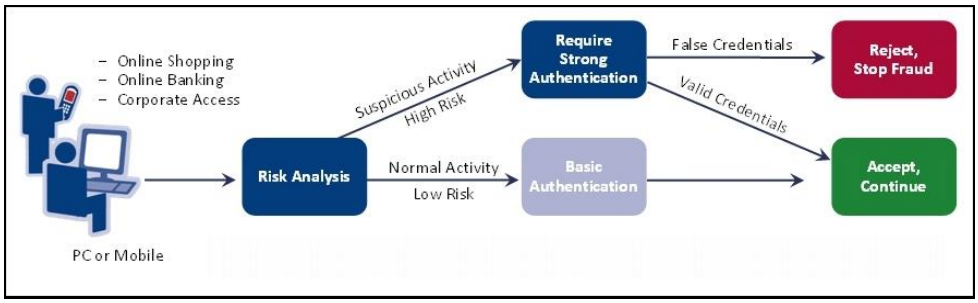
Note: CA RiskMinder still contains the terms Arcot and RiskFort in some of its code objects and other artifacts. Therefore, you will find occurrences of Arcot and RiskFort in all CA RiskMinder documentation. In addition, some of the topics in this guide do not follow the standard formatting guidelines.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

Introduction to RiskMinder

RiskMinder collects a wide range of data for risk evaluation (as discussed in "Data Used for Risk Evaluations" (see page 15).) This data is then evaluated with the help of configured rules (see "Rules and Risk Processing" (see page 19)). The result of each rule is then evaluated in the order of priority that is set by a RiskMinder administrator and a score and advice is generated corresponding to the first rule that matched ("Risk Score and Advice" (see page 23)). RiskMinder then creates a user-device association in the RiskMinder database ("User-Device Associations" (see page 25)).

How RiskMinder Works

The following figure illustrates how RiskMinder broadly assesses risk and detects fraud for each transaction.



You can implement the risk analysis capability either *before* the user logs in to your online application or *after* they have successfully logged in, as discussed in the following subsections.

Pre-Login Risk Assessment and Fraud Detection

When a user accesses your online application, you can assess them for a potential risk even before they log in.

If you call the risk analysis capability even before a user logs in to your online application, then the risk evaluation workflow is as follows:

1. User accesses your online application.
2. Your application invokes RiskMinder to analyze the risk that is associated with the transaction.
3. RiskMinder evaluates the risk by using the incoming IP address of the user and the configured rules. It uses the data that is discussed in the section, "Location Information" (see page 17) for the purpose.
4. Based on the result of the rules that were executed and whether the assessed information matched, RiskMinder generates a Risk Score and Advice (see page 23).
5. Your application validates the user, as follows:
   - If the risk is low, the user is allowed to access your online application.
   - If the risk is high, the user is denied access to your online application.
   - If the transaction is tagged as suspicious, then your application challenges the user for secondary authentication to prove their identity.

Post-Login Risk Assessment and Fraud Detection

When a user accesses your online application, you can first log them in and then comprehensively assess them for potential risks when they try to perform predefined actions (such as wire transactions).

If you call the risk analysis capability *after* you authenticate a user into your online application, then the risk evaluation workflow is as follows:

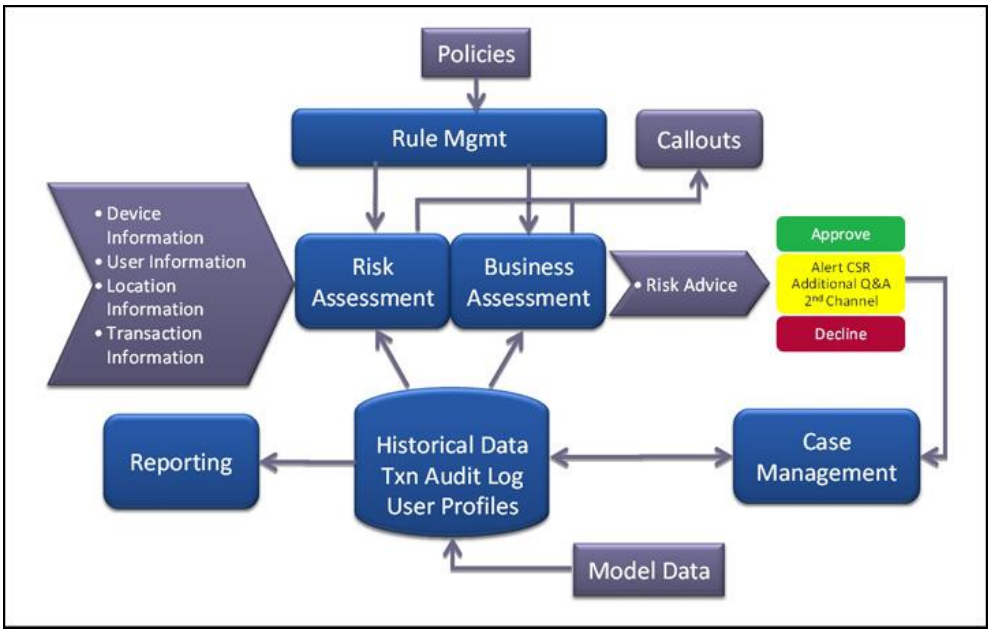
1. User logs in to your online application.
2. User tries to perform certain actions that were identified by you.
3. Your application invokes RiskMinder to analyze the risk that is associated with the transaction.
4. RiskMinder evaluates the risk by using the incoming inputs and the configured rules. It uses the data categories that are discussed in the section, "Data Used for Risk Evaluations" (see page 15) for the purpose.
5. Based on the result of rules that were executed and whether the assessed information matched, RiskMinder generates a Risk Score and Advice (see page 23).
6. Your application allows the user to continue with the transaction, as follows:
   - If the risk is low, the user is allowed to continue.
   - If the risk is high, the user is denied the transaction.
   - If the transaction is tagged as suspicious, then your application challenges the user for secondary authentication to prove their identity.

Data Used for Risk Evaluations

RiskMinder bases the result of a risk analysis by comparing the following incoming information, if available, with the historical data for the user:

- End-User Device Identification Information (see page 16)
- Location Information (see page 17)
- User and Transaction Information (see page 17)
- Case Management (see page 18)
- Fraud Model (see page 18)

The following figure illustrates how RiskMinder uses this data. The following subsections provide a quick overview of each of the data categories.



End-User Device Identification Information

The following subsections briefly walk you through the device identification and analytics technique that is used by RiskMinder:

- Device ID
- Machine FingerPrint (MFP)
- DeviceDNA

Device ID

The *Device ID* is a device identifier string that RiskMinder generates on the end-user device to identify and track the device that the end user uses for logging in to your online application and performing transactions. The Device ID information is in encrypted format.

Machine FingerPrint (MFP)

Machine FingerPrint (also referred to as Device fingerprinting or PC fingerprinting in industry terms) represents the browser information and device identification attributes (such as operating system, installed software applications, screen display settings, multimedia components, and other attributes) that are gathered from the end-user system and are analyzed to generate a risk profile of a device in real time. Some of the attributes that are collected from the end-user device include:

- Browser information (such as the name, UserAgent, major version, minor version, JavaScript version, HTTP headers)
- The operating system name and version.
- Screen settings, such as height, width, color depth.
- System information such as the time zone, language, system locale.

For every transaction performed by the end user, RiskMinder matches the corresponding MFP stored in its database with the incoming information. If this match percentage (%) is equal to or more than the value specified for the Device-MFP Match rule in Administration Console, then it is considered "safe".

DeviceDNA

DeviceDNA is a device identification and analytics technique that uses both Machine FingerPrint (MFP) and Device ID for more accurate information analyses. For accuracy, more information is collected than in case of MFP. For example:

- Extra system information (such as platform, CPU, MEP, system fonts, camera, and speaker information)
- Extra browser information (such as vendor, VendorSubID, BuildID)
- Extra screen settings (such as buffer depth, pixel depth, DeviceXDPI, DeviceYDPI)

- Plug-in information (such as QuickTime, Flash, Microsoft Windows Media Player, ShockWave, Internet Explorer plug-ins)
- Network information (such as the connection type)

Location Information

Derived from the end-user system IP address, this information includes geo-location information such as the locale, ISP, time zone, and related geographical information. To obtain this information, RiskMinder is integrated with Quova®, which specializes in providing detailed geographic information for each IP address by mapping it to a region.

To know more about Quova and their services, go to:
http://www.quova.com

For every transaction performed by the end user, RiskMinder matches the incoming IP address and the information that is derived from this IP address with the related information stored in the RiskMinder database. This information is then used as an input for Negative IP Address List, Negative Country List, and Zone Hopping rules.

User and Transaction Information

Typically, the login ID for a user identifies a user uniquely in the system. RiskMinder uses this information as one of the attributes to identify a user uniquely.

In addition, if configured, RiskMinder can also accept contextual or transaction information (such as the transaction amount, transaction type, and date) for analyzing the risk that is associated with a transaction. However, you can use your own custom rules to enable RiskMinder to evaluate this contextual information.

Book: See the *CA RiskMinder Administration Guide* for more information about adding new rules.

Case Management

The Case Management feature of RiskMinder provides administrators and fraud analysts a single unified view of the data that is related to cases. This view helps them analyze the data more efficiently and take faster, better-informed decisions toward resolving the cases. In addition, analysts can also constantly track the status and progress of their cases and maintain complete case histories with instant access to all related information. As a result, Case Management can be used to analyze data and identify new patterns of fraud from historical data. These patterns, in turn, can be used to configure new rules to reduce fraud.

Case Management enables you to:

- Efficiently manage customer service and support
- Manage large numbers of cases and investigations
- Create actions and tasks with due dates
- Assign actions with due dates
- Record investigation notes and the resolution that is provided to the user
- Handle cases and tasks more efficiently
- Keep clear audit trail or history of actions on a case
- Analyze trends
- Generate fraud-related reports

Fraud Model

RiskMinder offers an advanced fraud modeling capability. Based on the historical data, this modeling capability can be built and created in RiskMinder. By using the available transaction data and system data, the model generates a score that describes the extent to which the model suspects the genuineness of a transaction. This score typically ranges from 0 through 100, where the higher the number, the greater the possibility of fraud. RiskMinder can be configured to send different responses to your calling application based on this model score.

The model score is available as a part of the system parameters (as ModelScore) while configuring rules on the Rules and Scoring Management page in Administration Console. This score can be used with other data elements to arrive at a risk advice.

Rules and Risk Processing

After the required data is collected, it is forwarded to *Rules Engine* (a module of RiskMinder Server). The Rules Engine is a set of configured rules that evaluate this information, which is based on incoming information and historical data, if available.

A *rule*, in turn, is a condition or a set of conditions that must be true for a rule to be invoked. By default, each rule is assigned a priority and is evaluated in the specific order of its priority level. However based on your business requirements, you can change this priority of rule scoring.

The out-of-the-box rules that are provided by RiskMinder are explained in the following table.

Rule Name	**Description**
Exception User Check	An organization may choose to exclude a user from risk evaluation during a certain time interval. For example if a user travels to a country that is configured as negative in RiskMinder, then for the specified interval their status can be changed to an *exception user*. RiskMinder returns a low risk score for transactions originating from exception users and the advice is typically Allow.
Untrusted IP Check	This list constitutes the IP addresses that originate from anonymizer proxies or have been the origin of known fraudulent or malicious transactions in the past. Transactions originating from configured negative IP addresses receive a high score and the advice is Deny.
Negative Country Check	This list comprises the countries that have been known to be origins of significant number of frauds in the past. RiskMinder derives the country information based on the input IP address, and then uses this data to return a high risk score for online transactions originating from these "negative" countries. Transactions originating from configured negative countries receive a high score and the advice is Deny.

Rule Name	Description
Trusted IP/Aggregator Check	Transactions originating from IP addresses "trusted" to the organization receive a low score, by default, and the advice is Allow. Many enterprises use the services of account and data aggregation service providers to expand their online reach. The originating IP addresses when users log in from a protected portal versus when they come in through such aggregators are different. Transactions originating from aggregators "trusted" to the organization receive a low score, by default, and the advice is Allow.
Unknown User	An *unknown user* is not registered in the RiskMinder database. If the user is unknown to RiskMinder, then by default an Alert is returned. A Customer Support Representative (CSR) can then choose to further authenticate the user based on the advice.
Unknown DeviceID	The Device ID is a device identifier string that RiskMinder generates and stores on the end user's device to identify and track the device that the end user uses for logging in to your online application to perform transactions. RiskMinder returns a low risk score for transactions originating from known devices and the advice is typically Allow.
User Not Associated with DeviceID	■ Transactions originating from a known device that is associated with a user, and whose DeviceDNA matches, receive a low score, and the advice is Allow. ■ Transactions originating from a known device that is not associated with a known user receive a medium score, and the advice is IncreaseAuth. **Note:** See the sections "User-Device Associations" (see page 25), "Machine FingerPrint (MFP)", and "DeviceDNA" for more information about these topics.

Rule Name	Description
Device MFP Not Match	■ Transactions originating from a known device whose DeviceDNA does not match receive a medium score, and the advice is IncreaseAuth. ■ Transactions originating from an unknown device that is not associated with a known user receive a high score, and the advice is Deny. **Note:** See the sections "User-Device Associations" (see page 25), "Machine FingerPrint (MFP)", and "DeviceDNA" for more information about these topics.
User Velocity Check	Frequent use of the same user ID could be an indication of risky behavior. For example, a fraudster might use the same user ID and password from different devices to watch a specific activity in a targeted account. Too many transactions originating from the same user within a short (configurable) interval receive a high score and the advice is Deny.
Device Velocity Check	Frequent use of the same device could also be an indication of risky behavior. For example, a fraudster might use the same device to test multiple combinations of user IDs and passwords. Administrators can now configure RiskMinder to track this behavior, as well. Too many transactions originating from the same user device within a short (configurable) interval receive a high score and the advice is Deny.
Zone Hopping Check	If a user logs in from two long-distance locations within a short time span by using the same user ID, this might be a strong indication of fraudulent activity. In addition, a User ID can also be shared, in which case, RiskMinder understands that the two people sharing the same User ID can be in geographically different locations and responds with an appropriate response. Transactions originating from the same user from locations that are far apart from each other within a short (configurable) interval receive a high score and the advice is Deny.

The Rules Engine executes these rules in the order of their precedence. The evaluation result is then forwarded to another module of the RiskMinder Server, which is known as the **Scoring Engine**. Between Rules Engine and Scoring Engine, the rules are run in the following two phases:

- Execution Phase

  RiskMinder Server does a first parse of all the rules in the active ruleset. In this phase, the Server:

  a. Executes all the rules in the list in the order of execution priority.

     This execution priority is internal, and is defined by the Server.

  b. Generates an individual risk score and advice for each rule it executes.

- Scoring Phase

  RiskMinder Server now does the second parse of the rules. In this phase, the Server:

  a. Uses the result for each rule in the first parse, and parses the rules in the ruleset based on the scoring priority.

     The Global Administrator (GA) configures the scoring priority by using the Administration Console.

  b. Stops the scoring at first matched rule.

  c. Returns the score and advice of the rule that matched as final.

Note: Depending on when the first rule matched, the second parse may not be run completely.

Risk Score and Advice

Based on the result of the execution of each rule that Rules Engine provides, the *Scoring Engine* evaluates the score of each rule in the order of priority set (by the administrator) and returns the score corresponding to the first rule that matched.

For example, consider that you have configured these rules in the following order:

1. Negative IP (say, with a score of 85)
2. User Velocity (say, with a score of 70)
3. High Amount Check (say with a score of 80)
4. Device Velocity (say, with a score of 65)

Note: High scores are typically assigned to rules that are more critical.

If RiskMinder determines that a transaction is coming from a negative IP address, then it returns a score of 85 (Deny), based on the first configured rule that matched. If another transaction exceeds the configured Device Velocity, then RiskMinder returns a score of 65.

The *risk score* that is generated by the Scoring Engine is an integer from 1 through 100. RiskMinder then uses this risk score to generate the corresponding *advice* and returns this advice to your application.

The following table shows the default out-of-the-box risk score and corresponding advice matrix. You can configure these ranges according to your organization policies and requirements.

Score Value (From)	Score Value (To)	Advice	Default Recommended Action
0	--	--	The rule is executed but is not used for scoring.
1	30	ALLOW	Allow the transaction to proceed.
31	50	ALERT	Take an appropriate action. For example, if the user name is currently unknown, then on getting an alert you can either redirect it to a Customer Support Representative (CSR) or you can create a user in RiskMinder.

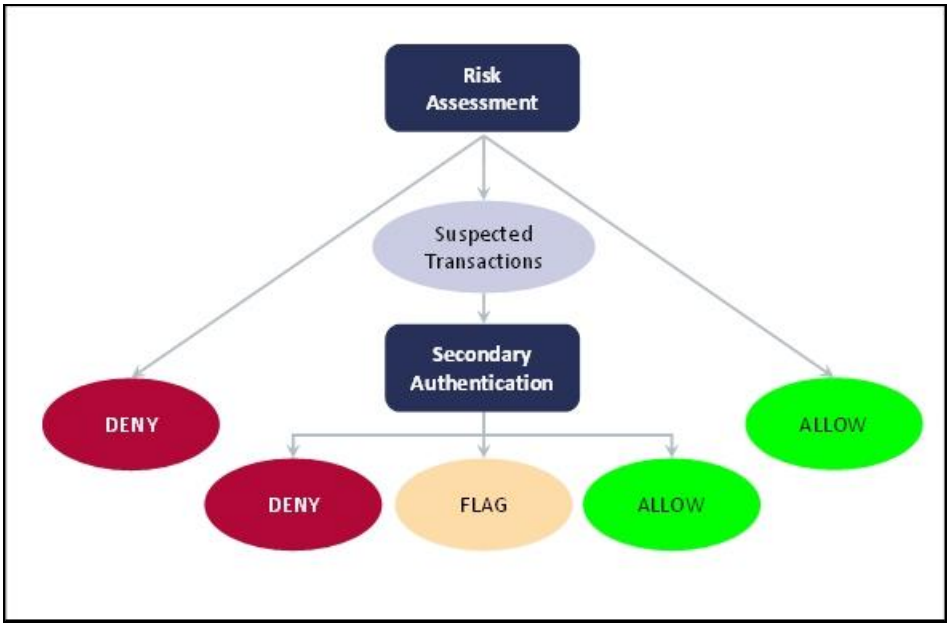
Score Value (From)	Score Value (To)	Advice	Default Recommended Action
51	70	INCREASEAUTH	Perform additional authentication before proceeding any further.
71	100	DENY	Deny the transaction.

Based on the data that is received, RiskMinder generates any of the following advice:

- **ALLOW:** RiskMinder returns ALLOW, if the risk score associated with the transaction is low.
- **ALERT:** If a user who is not registered with RiskMinder tries to log in, then ALERT is returned.
- **INCREASE AUTHENTICATION:** When RiskMinder detects a suspicious transaction, it flags the transaction with INCREASE AUTHENTICATION and it advises the application to force the user for secondary authentication.

  For example, when a user registered with RiskMinder attempts a transaction from a device that is not yet recognized by RiskMinder, then the user must undergo secondary authentication (such as OTP or QnA) with your application.
- **DENY:** RiskMinder returns the DENY advice when a high risk score is associated with the transaction.

The following figure illustrates the advice that RiskMinder returns.



User-Device Associations

For subsequent evaluations, RiskMinder uniquely identifies a user as a valid user by automatically associating (or binding) a user to the device that they use to access your application. This is referred to as an *association* (or device binding) in RiskMinder terminology. Users who are not bound are more likely to receive the Increase Authentication advice to be authenticated.

RiskMinder also allows users to be bound to more than one device. For example, a user can use a work and a home computer to access your application. Similarly, you can bind a single device to more than one user. For example, members of a family can use one computer to access your application.

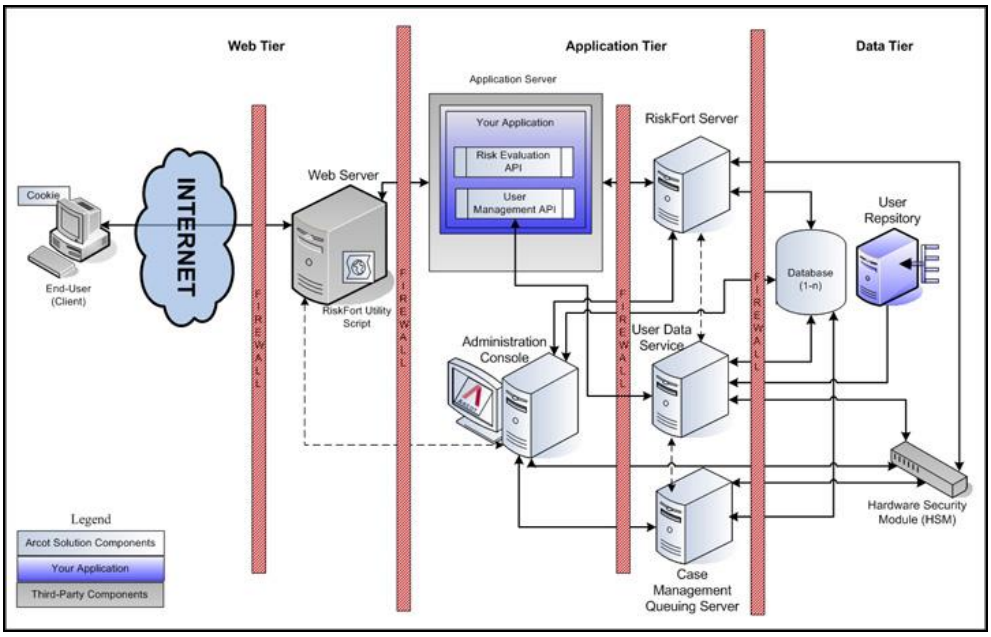
RiskMinder Architecture and Component Communication

This section covers:

RiskMinder Architecture

You can install RiskMinder on a single system or you can distribute its components across multiple systems, as discussed in later chapters in the guide. However to ensure maximum security and integrity of data and transactions, the three-tier architecture in the following figure is recommended:

- Web Tier (see page 26)
- Application Tier (see page 27)
- Data Tier (see page 28)



Web Tier

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

The *RiskMinder Utility Script* (ArcotDeviceDNA.js), which is a client-side JavaScript that must be included in your application, is served to the end-user browser through the web servers that reside in this layer. This script enables you to:

- Set the Device ID on the end-user device.
- Collect the Machine FingerPrint (MFP), DeviceDNA, and Device ID information.

Book: See "Collecting Device ID and DeviceDNA" in the *CA RiskMinder Java Developer's Guide* for detailed information about DeviceDNA, Device ID, and using the utility script.

Application Tier

This layer constitutes all application server components in the system. These include RiskMinder Server, UDS, Administration Console, and the RiskMinder SDKs:

Note: All components in this layer can be installed on one system or can be distributed across multiple systems, as discussed in the following chapters.

- RiskMinder Server

  A server component that processes risk evaluation requests from your application through RiskMinder SDKs.

- Case Management Queuing Server

  A server component that schedules and dispatches cases to Customer Support Representatives (CSRs) and then manages the lifecycle of these cases.

- Administration Console

  A web-based console for configuring server instances, communication mode between RiskMinder components, business rules and the corresponding data, 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 databases (RDBMSs) and directory servers (LDAPs).

- Risk Evaluation SDK

  APIs and web services that your application can invoke to forward risk-analysis requests to RiskMinder Server.

- Risk Evaluation Web Service

  The web-based interface that enables interaction over a network between RiskMinder Server and your application. It consists of the web services that can be invoked by your web application to perform risk evaluation.

- User Management Web Service

  Web services that can be invoked by your application to forward requests to User Data Service for enrolling users and for managing user details in RiskMinder.

- Sample Application

  Sample Application demonstrates the usage of RiskMinder Java APIs and how your application can be integrated with RiskMinder. Sample Application can also be used to verify if RiskMinder was installed successfully, and if it is able to perform risk-evaluation operations.

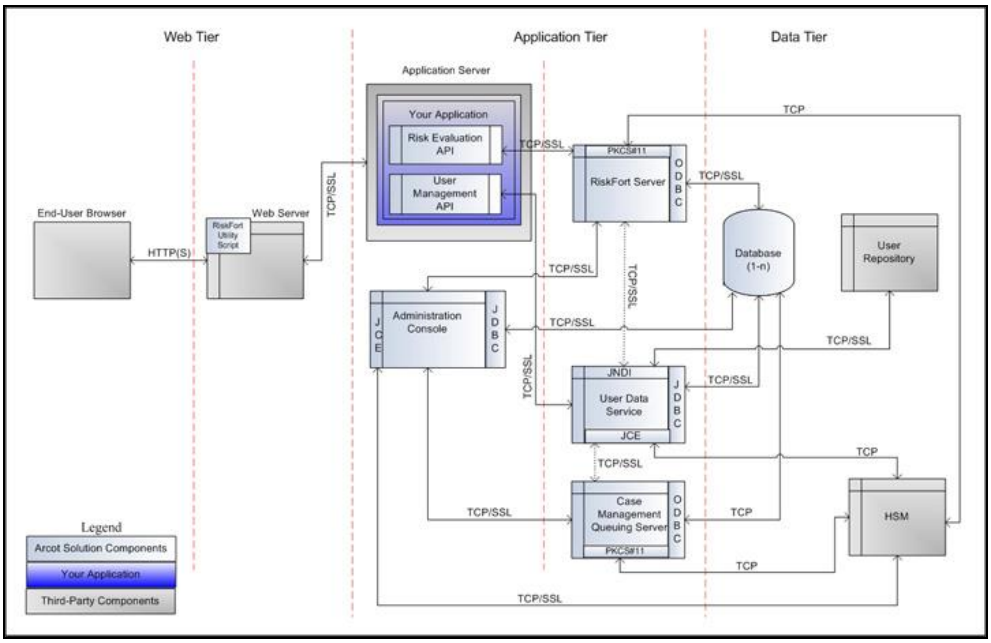
Data Tier

This layer comprises the instances of relational databases that store the configuration, user, and historical data that is used by RiskMinder to analyze each transaction. In addition, this layer also constitutes any directory servers (LDAPs) that you have configured for storing user details.

If you are planning to use any Hardware Security Modules (HSMs) for encrypting sensitive user data, then the HSM is also a part of this layer.

Communication Between RiskMinder Components

The following figure illustrates the possible communication modes that are supported between RiskMinder and its components.



As shown in the figure, the default mode of communication between components is TCP, RiskMinder Server supports SSL communication (two-way and one-way communication) with the following components to ensure integrity and confidentiality of the data being exchanged during a transaction:

- Case Management Queuing Server
- RiskMinder Database
- User Data Service
- RiskMinder SDK (Risk Evaluation)
- Sample Application
- Evaluation Callout
- Scoring Callout

Note: RiskMinder enables you to write your own custom Evaluation rule based on your business requirements. This custom rule is known as **Evaluation Callout**. Similarly, RiskMinder also enables you to write your own custom Scoring logic, which is known as **Scoring Callout**. See *CA RiskMinder Administration Guide* for more information about these callouts.

What's New in this Release

See the release notes for information about the key features and enhancements that have been introduced in release 3.1.01.

Chapter 2: Planning the Deployment

This section helps you decide on a deployment model, and determine which RiskMinder components and prerequisite software to install on each system. The 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 software that is run on the system.

The section covers the following topics:

Deployment Overview

You can either perform a fresh installation of RiskMinder or upgrade from a previous release. The following sections describe the deployment procedures for both the cases:

If You are Performing a Fresh Installation

This section provides a quick overview of steps for deploying a fresh instance of RiskMinder and provides pointers for choosing a deployment model based on your requirements:

1. Choose a deployment model. RiskMinder can be installed on a single system or across multiple systems.

   See "Choosing a Deployment Model" (see page 34) for more information.

2. Ensure that the system where you plan to install RiskMinder and its components meets all hardware requirements.

   See "Hardware Requirements" (see page 45) for more information.

3. Install the prerequisite software products.

   See "Software Requirements" for more information.

4. Create a database user in the SQL database.

   See "Configuring Database Server" (see page 51) for more information.

5. Install RiskMinder:

   - See "Performing Complete Installation" (see page 66) for more information about installing in a single-system environment.
   - See "Installing on the First System" (see page 102) for more information about installing in a distributed environment.

6. To create the RiskMinder schema and set initial configuration preferences, run SQL scripts in the database.

   - See "Running Database Scripts" (see page 74) for more information about running SQL scripts for single-system deployments.
   - See "Running Database Scripts" (see page 113) for more information about running SQL scripts for distributed deployments.

7. Copy the required files and JARs on your application server. Administration Console and User Data Service (UDS) use the JARs for proper functioning:

   - See "Preparing Your Application Server" (see page 75) for more information about deploying and starting UDS and Administration Console for single-system deployments.
   - See "Preparing Your Application Server" (see page 114) for more information about deploying and starting UDS and Administration Console for distributed deployments.

8. Deploy Administration Console:

   - See "Logging In to Administration Console" (see page 84) on deploying Administration Console for single-system deployments.

- See "Logging In to Administration Console" (see page 123) on deploying Administration Console in a distributed environment.

9. Log in to Administration Console as a Master Administrator to initialize it:
   - See "Bootstrapping the System" (see page 85) for more information about initializing Administration Console for single-system deployments.
   - See "Bootstrapping the System" (see page 124) for more information about initializing Administration Console in a distributed environment.
10. Start RiskMinder Server and Case Management Queuing Server, and verify that the services are coming up correctly:
    - See "Starting RiskMinder Server" (see page 87), "Starting the Case Management Queuing Server" (see page 88), and "Verifying the Installation" (see page 91) for more information about initializing Administration Console for single-system deployments.
    - See "Starting RiskMinder Server" (see page 126), "Starting the Case Management Queuing Server" (see page 127), and "Verifying the Installation" (see page 127) for more information about initializing Administration Console in a distributed environment.
11. **(Optional)** Deploy User Data Service (UDS). Perform this step *only if* you want to use your LDAP as the user repository:
    - See "Deploying User Data Service (UDS)" (see page 89) for more information about deploying and starting UDS for single-system deployments.
    - See "Deploying User Data Service (UDS)" (see page 128) for more information about deploying and starting UDS for single-system deployments.
12. (**For Distributed Installation Only**) Install RiskMinder on the subsequent systems.

    See "Installing on the Second System" (see page 130) for more information.
13. To test RiskMinder installation, deploy and run Sample Application:
    - See Deploying Sample Application (see page 91) and Using Sample Application (see page 92) for more information about performing this task in a single-system environment.
    - See "Deploying Sample Application" (see page 131), "Configuring Sample Application for Communication with RiskMinder Server" (see page 132), and "Using Sample Application" (see page 133) for more information about performing this task in a distributed environment.
14. **(Optional)** Change the HSM settings that you specified during installation:

    See "Changing Hardware Security Module Information After the Installation" (see page 217) for more information.

If You are Upgrading from a Previous Release

This section provides a quick overview of steps for upgrading your RiskMinder instance and provides pointers for choosing a deployment model based on your requirements:

1. Ensure that the system where you plan to install RiskMinder and its components meets all hardware requirements.

   See "Hardware Requirements" (see page 45) for more information.

2. Install the prerequisite software products.

   See "Software Requirements" for more information.

3. Upgrade RiskMinder.

   See "Upgrading RiskMinder" (see page 147) for more information.

Choosing a Deployment Model

As a part of the RiskMinder deployment, RiskMinder Server is the primary component that you must install. The server provides the risk evaluation service, which includes transaction risk evaluation. Your applications that must use RiskMinder Server can integrate with it by using Java SDKs or web services shipped with it.

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

Typically, all RiskMinder components are installed on a single system for development and simple testing. However, in production deployments and staging environments, RiskMinder Server must 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.

RiskMinder is also shipped with a Sample Application, which can be used to verify if RiskMinder was installed properly and to perform risk evaluation. Sample Application also serves as a code sample for integrating RiskMinder with your existing applications.

The high-level deployment types that are supported by RiskMinder are:

- **Single-System Deployment** - For development or testing
- **Distributed-System Deployment** - For production or staging environments
- **High-Availability Deployment** - For high availability and scalability, production, or staging environments

Deploying on a Single System

In a single-system deployment, all components of RiskMinder and the application, which users log in to, are installed on a single system. The database may be on the same system where RiskMinder 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. The prerequisite software for these components is identical.

The simplest way to perform a single-system deployment is to select the **Complete Installation** option (see "Performing Complete Installation" (see page 66) for more information) while running the RiskMinder installer.

Component Diagrams

The diagrams in this section depict possible deployment options for prerequisite software and RiskMinder components. Note, that if you perform a **Complete Installation**, then both Java SDKs and Web services are present on the system. You can select one or both integration methods, in this case.

- Deploying Java SDKs
- Deploying Web Services

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

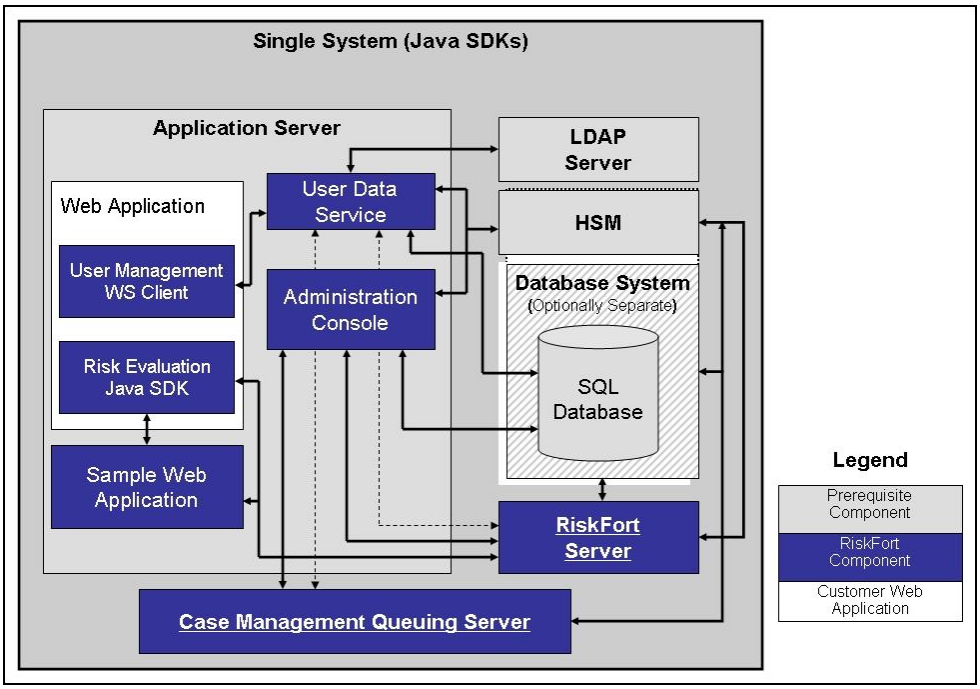
- Install a database server on the system which has RiskMinder Server, or use an existing database on a separate system?
- Use Sample Application or write my own Web application?

  Important! Sample Application must *not* be used in production deployments. it is 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 my own Web application.

The following sections help you achieve your deployment decision.

Deploying Java SDKs

The following figure illustrates RiskMinder Server and Java SDKs deployed on a single system.

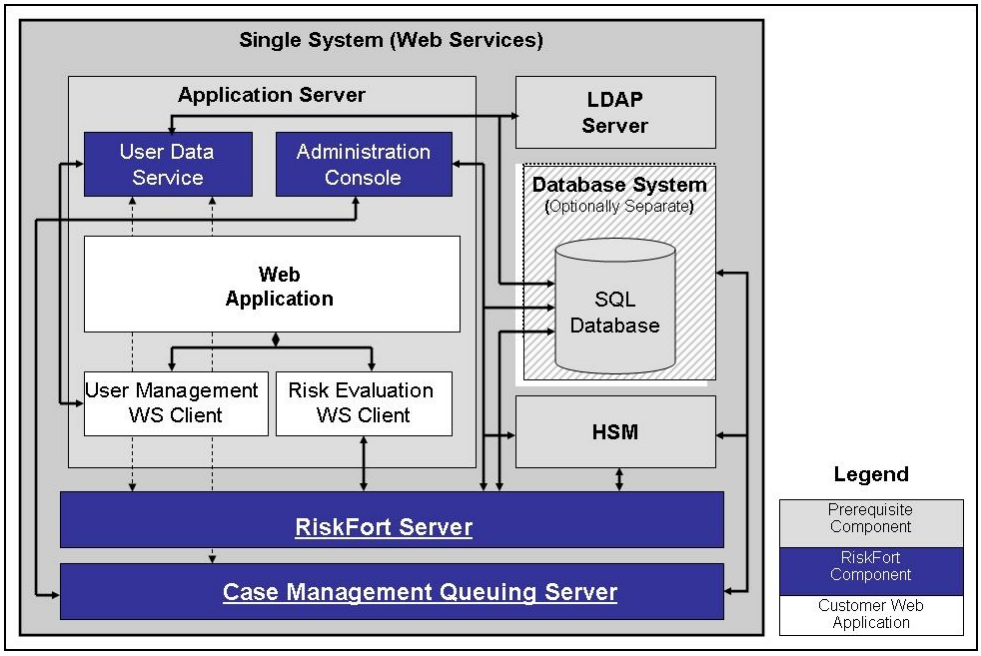

Single System (Java SDKs)
Application Server
Web Application
User Management WS Client
Risk Evaluation Java SDK
Sample Web Application
User Data Service
Administration Console
LDAP Server
HSM
Database System
(Optionally Separate)
SQL Database
RiskFort Server
Case Management Queuing Server
Legend
Prerequisite Component
RiskFort Component
Customer Web Application

Note: The use of a web server to deliver HTML pages for the application server is optional and is transparent to RiskMinder. In production deployments, this approach is typically used to improve the application server performance and security. See the documentation of your application server for detailed information.

Deploying Web Services

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

Note: Because all Web services are now built into the RiskMinder Server module itself, you simply install RiskMinder Server on the target system and generate the requisite client stubs. No further configuration is required.



Deploying on Distributed Systems

The distributed model is typical of web-based applications whose components are distributed across the web tier, application tier, and data tier, and require a secure zone between its web servers and application servers. The other reasons for deploying RiskMinder in a distributed model are:

- High availability (failover and load balancing)
- High performance
- Increase in throughput

In a distributed-system deployment, RiskMinder components are installed on different servers. This is done for security, performance, and to enable multiple applications to use the risk-evaluation functionality.

This deployment model is typically used for production deployments or for staging environments.

For example, the most common deployment is to install RiskMinder Server on one system and one or more web applications on additional systems. Because the deployment covers more than one system, an architecture diagram is included that indicates which systems must be able to communicate with each other.

To perform a distributed-system deployment, you must select the **Custom** installation option (See "Installing on the First System" (see page 102) for more information) in the RiskMinder installer.

Component Diagrams

The diagrams in this section depict several possible options, where prerequisites and RiskMinder 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

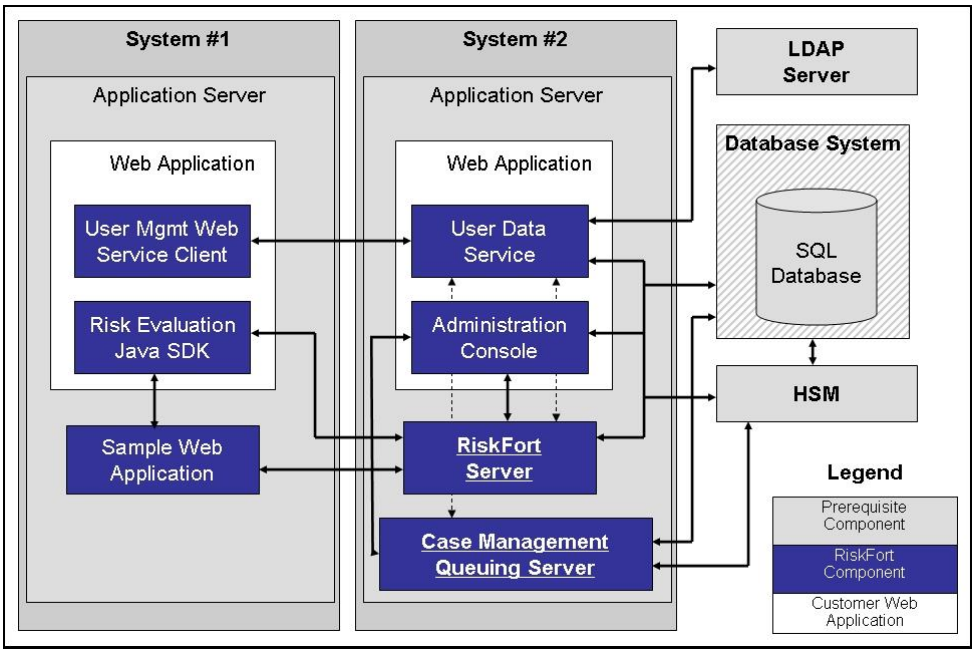
Decision Point

- Which RiskMinder components must be installed on each system?

The following sections help you achieve your deployment decision.

Deploying Single Application with Java SDKs

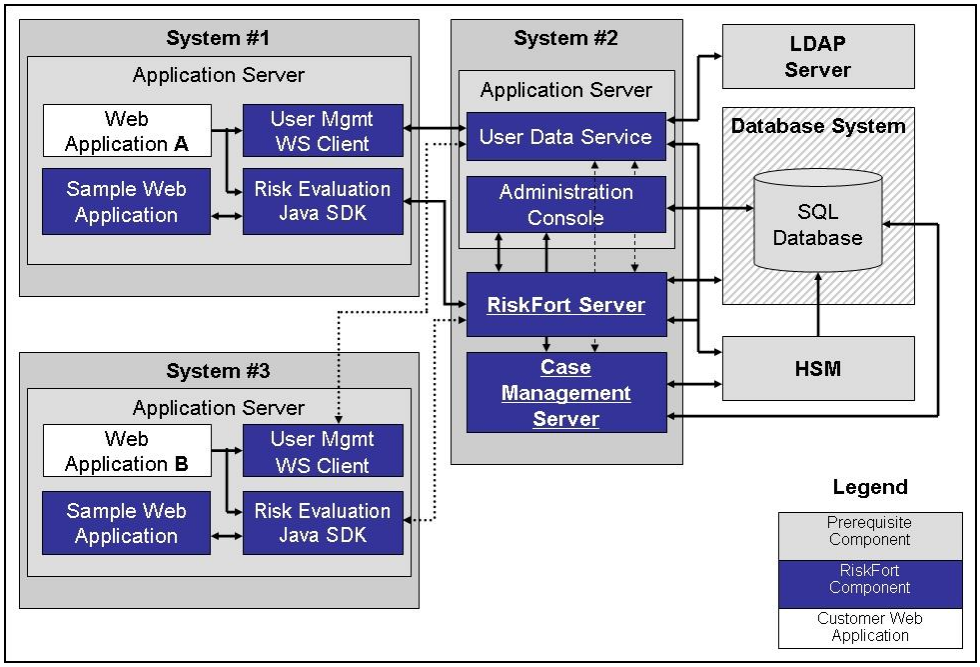
The following figure illustrates RiskMinder using Java SDKs with a single application.



Note: Administration Console can be installed on any individual system, every system, or on a system that is not listed in the diagrams.

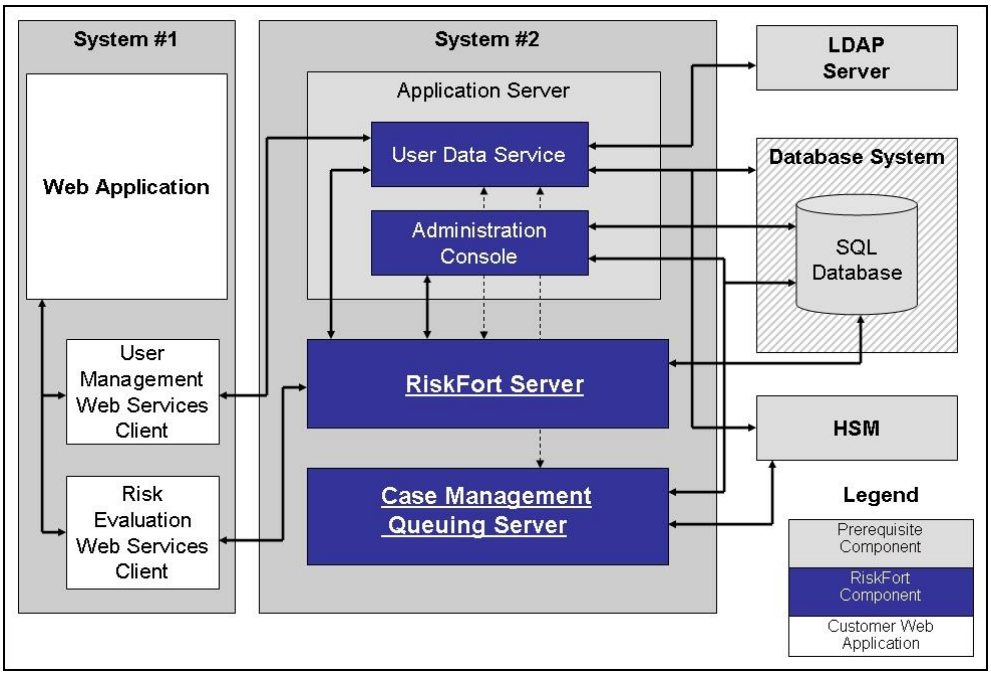
Deploying Multiple Applications with Java SDKs

The following figure illustrates RiskMinder deployment using Java SDK with multiple applications.



Deploying Single Application with Web Services

The following figure illustrates RiskMinder deployment using Web services on a single application.



Deploying in a High-Availability Environment

In a high-availability deployment, RiskMinder components are installed on more than one server to provide high availability and scalability. This section discusses the component diagrams for deploying in a high-availability environment.

Component Diagrams

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

Decision Points

- When do I add a server instance?

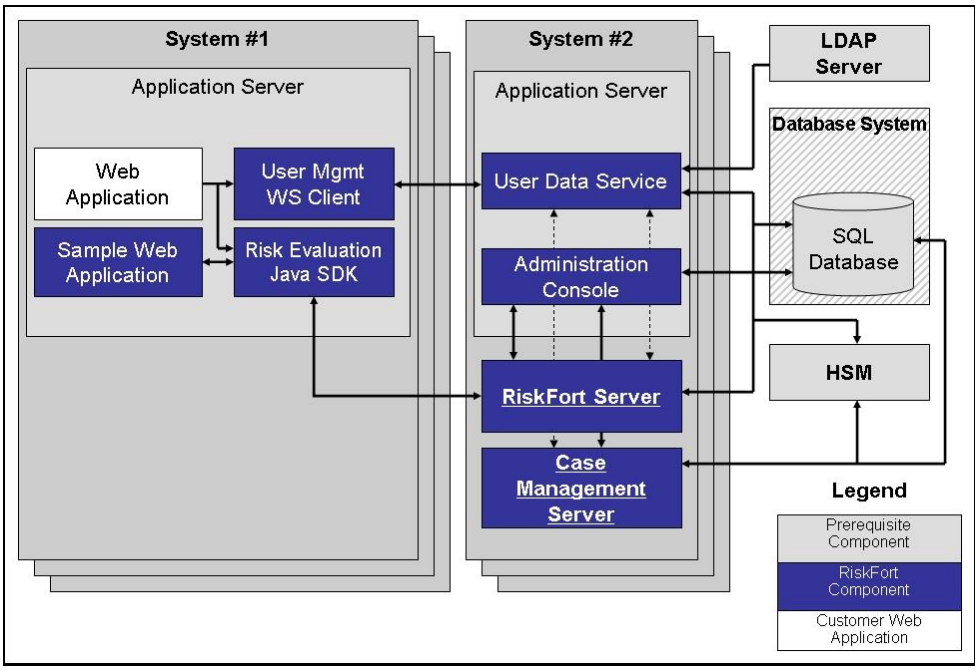
  Typically, when your transaction rate exceeds the permissible threshold (as decided by your organizational policies), then you must add a server instance.

- How many RiskMinder Server, Case Management Server, UDS, and SDK instances can I have?
  - **RiskMinder Servers:** Multiple instances are supported. The number depends on the transaction rate you want to achieve.
  - **Case Management Queuing 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 simultaneously.
  - **UDS Servers:** Currently, only one is supported.
  - **SDKs:** Multiple instances are supported. This number depends on the number of your application instances that you plan to support.

The following sections help you achieve your deployment decision.

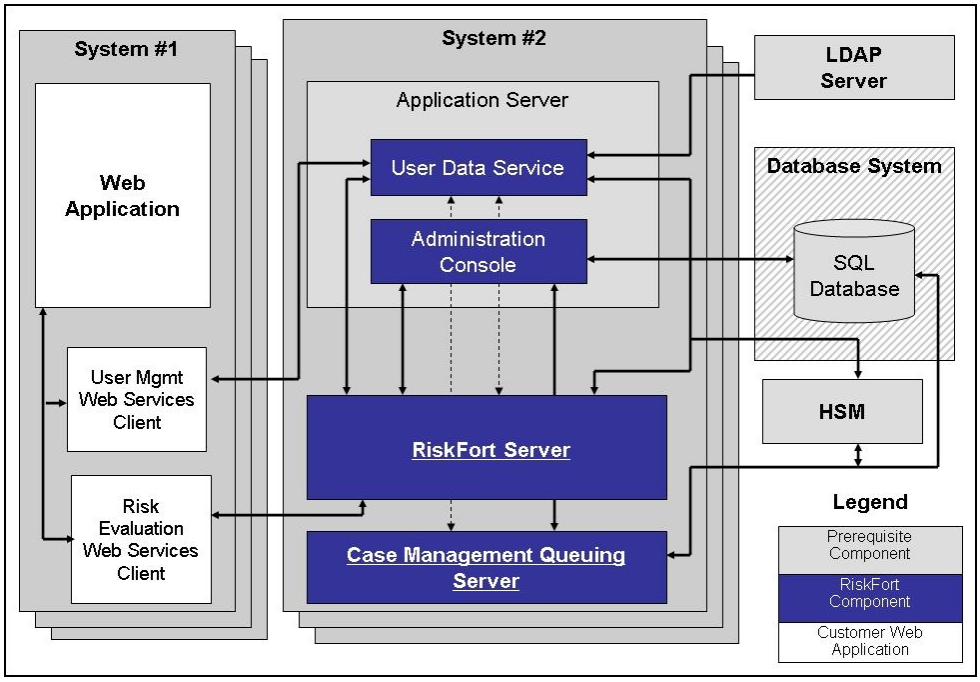
High-Availability Deployment Using Java SDK

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



High-Availability Deployment Using Web Services

The following figure illustrates multiple-instance deployment of RiskMinder using Web services.



Chapter 3: Preparing for Installation

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

This section contains the following sections:

Hardware Requirements

The *minimum* hardware requirements for installing RiskMinder include:

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

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 that is required for the installation. While you perform load-testing, you must remember 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. See your platform vendor documentation for information about configuring 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 RiskMinder 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 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 necessary, must be made while you are still in the planning and preparation stages. Otherwise, you must initialize the database again later, because all your current encryption would use keys in software.

Software Requirements

This section describes the following requirements:

- Minimum Software Requirements (see page 47)
- RiskMinder Component-Specific Prerequisites (see page 50)

Minimum Software Requirements

The following table lists the minimum software that is required to install RiskMinder.

Note: For all the 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	■ Microsoft Windows Server 2003 Enterprise Edition (32-bit) ■ Microsoft Windows Server 2008 Standard Edition (32-bit) ■ Microsoft Windows Server 2008 Enterprise Edition (32-bit) ■ Microsoft Windows Server 2008 Standard Edition (64-bit) ■ Microsoft Windows Server 2008 Enterprise Edition (64-bit) ■ Microsoft Windows Server 2008 Release 2 (64-bit)
Service Pack	SP2 or higher
Database Server	■ Microsoft SQL Server 2005, Standard Edition (SP2) or higher ■ Microsoft SQL Server 2008 Enterprise Edition
	■ Oracle 10*g* ■ Oracle 11*g* Release 2
	■ MySQL Enterprise Edition 5.1
JDBC Drivers (JARs)	The JDBC driver best compatible with your database. **Important!** It is strongly recommended that the JDBC JAR version is same as or higher than your database server version.

Software Type	Version
Directory Server	The following Directory Servers are supported: ■ Microsoft Windows Active Directory Server 2003 ■ Microsoft Windows Active Directory Server 2008 ■ SunOne Directory Server 5.2 ■ SunOne Directory Server 6.3 ■ Oracle Directory Server 11*g* ■ CA Directory Server r12.0 Service Pack 10
Application Server	The following Application Servers are supported: ■ Apache Tomcat 5.5.*x* (Here, *x* is 31 or higher) ■ Apache Tomcat 6.*x* (32-bit and 64-bit) ■ Apache Tomcat 7.*x* (32-bit and 64-bit) ■ IBM WebSphere 6.1.*x* **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**. ■ IBM WebSphere 7.0 (32-bit and 64-bit) ■ JBoss Application Server 5.1.*x* ■ Oracle WebLogic 10.1.*x* (32-bit and 64-bit) ■ Oracle WebLogic 11*g* (WebLogic Server 10.3) (32-bit and 64-bit) The **JVM Memory** Settings (Heap Size) for the application server must be a minimum of 512 MB. However, if you plan to use an LDAP repository with large user base (for example, 100,000 users), then it is strongly recommended that you increase the Heap Size to 1 GB or higher. To set the Heap Size to 512 MB, use the -Xmx512M JVM memory setting.Similarly, to set the Heap Size to 1 GB, use the -Xmx1024M setting. **Note:** Do not set the -Xms parameter.

Software Type	Version
JDK **Note:** If you perform a fresh installation of JDK, 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 Administration Console and other JDK-dependent components may fail to start.	The JDK version that is best compatible with the Application Server that you are using: ■ 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 **Important!** If you are using JRockit, ensure that *JROCKIT_HOME*/jre/bin/ is included in the PATH environment variable. In addition, this change in the PATH variable *must* be effective before you start the WebLogic application server.
Web Service Clients	The following clients are supported: ■ Axis2 1.5 ■ .NET Framework 4
Browsers	The following Web browsers are supported: ■ Internet Explorer 7.0 ■ Internet Explorer 8.0 ■ Internet Explorer 9.0 ■ Mozilla Firefox 18 or higher ■ Apple Safari 5.0 or higher ■ Google Chrome 20 or higher

RiskMinder Component-Specific Prerequisites

The RiskMinder components to be installed on a system determines the prerequisite software. Refer to "Planning the Deployment" (see page 31) to determine which RiskMinder components to install for each deployment type.

The following table lists the prerequisite software that is required by each RiskMinder component:

Component	Prerequisite		
	Database Server	**JDK**	**Application Server**
RiskMinder Server	✓		
Case Management Queuing Server	✓		
Administration Console	✓	✓*	✓
User Data Service	✓	✓*	✓
Risk Evaluation Java SDK		✓*	✓
User Management Web Service		✓*	✓
Administration Web Service		✓*	✓
Transaction Web Service		✓*	✓
Sample Application		✓*	✓

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

Configuring Database Server

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

RiskMinder supports a primary database and 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 RiskMinder installation, when the installer edits the arcotcommon.ini (see page 198) file with the database information you supply.
- Later, manually by:
  a. Editing the arcotcommon.ini (see page 198) file.
  b. Editing the DSN, as required for the server component.
  c. Updating securestore.enc by using the DBUtil.exe tool.

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

Important! To protect the database, it is recommended that the database server be protected with a firewall or any other access control mechanism and is set to the same time-zone as all dependent products.

- Configuring Microsoft SQL Server (see page 51)
- Configuring Oracle Database (see page 53)
- Configuring MySQL (see page 54)

Configuring Microsoft SQL Server

This section provides the following configuration information for SQL Server:

Note: See the SQL Server documentation for detailed information about performing the tasks listed in this section.

1. Verifying Authentication Mode (see page 52)
2. Creating a Database (see page 52)
3. Creating a Database User (see page 52)

Verifying Authentication Mode

Verify that SQL Server is configured to use the **"SQL Server and Windows Authentication mode"** for Server authentication. RiskMinder cannot connect to the database if SQL Server is configured to "**Windows Authentication Mode"**. You can verify this by right-clicking the server in the Object Explorer window and selecting the Security page.

Creating a Database

To create a database, use the following criteria:

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

Creating a Database User

To create a database user, use the following steps:

Note: SQL Server refers to a 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.
   c. Ensure that you specify other password settings on this page according to the password policies in your organization.
   d. Default the database to the database (arcotdb) you created.
   e. User Mapping for the login (in the Users mapped to this login section).
   f. 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 RiskMinder Server.

Note: See the Oracle database documentation for details on performing the tasks that are listed in the following sections.

Required Tablespaces

Running RiskMinder 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 RiskMinder database.

  See "Creating a New Database" **(see page 53)** for creating a database.

- The second tablespace is used to run reports. For high performance, it is recommended that you use a separate tablespace.

RiskMinder Database Configuration Script

The RiskMinder 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 must manually create this tablespace and delete the section in this script, which creates the reports tablespace.

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

To set up the Oracle database, perform the following steps:

1. Creating a New Database (see page 53)
2. Creating a Database User (see page 54)

Creating a New Database

Create a database that stores information in the UTF-8 character set. This allows RiskMinder to use international characters including double-byte languages. To enable UTF-8 support for your Oracle database:

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

   Note: If the deployment is for the production environment, staging, or other intensive testing, see "Database Reference" (see page 221) to determine the quota that is required for a user.
3. Grant the DBA role to the user.

Configuring MySQL

This section provides the following configuration information for MySQL:

Note: See the MySQL documentation for detailed information about performing the tasks that are listed in this section.

Enabling Support for the InnoDB Transaction Engine

RiskMinder uses the InnoDB storage engine of MySQL. To check whether your MySQL installation supports this storage engine, 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 the MySQL documentation.

Setting the lower_case_table_names Variable

If you are running MySQL on any non-Windows platform, set the lower_case_table_names variable to 1.

Note: For detailed information about this variable, see the MySQL documentation.

Creating a Database

To create a database:

1. Open a MySQL command window.
2. To create the database schema, run the following command:
   `CREATE SCHEMA '<schema-name>' DEFAULT CHARACTER SET utf8;`
3. To create the database user, run the following command:
   `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 following privileges to the user:
   - 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 the RiskMinder installation, set up the RiskMinder data store, the Database Client, and gather the required database information for use during the installation later. Also ensure that the prerequisite JDK version and application server that are required by RiskMinder components are installed.

This section discusses the following topics:

Configure UTF-8 Support on Client Systems

Enable the UTF-8 support on the systems (for example, RiskMinder Server, Administration Console, and User Data Service) where you plan to install RiskMinder components that communicate with the database server. This section walk you through the steps to do so.

To enable UTF-8 support, perform the following steps:

1. Install the required language package. See the vendor documentation for more information about how to do this.
2. Navigate to the following location:

   Start -> **Settings** -> **Control Panel** -> **Regional and Language Options**

   The Regional and Language Options dialog appears.
3. Activate the **Languages** tab.
4. Select the following options:
   - Install files for complex script and right-to-left languages (including Thai)
   - Install files for East Asian Language
5. Click **Apply** to save the changes.
6. Click **OK** to close the dialog.

Database Information that You Need for Installing RiskMinder

Perform the tasks that are described in this section on the system where you plan to install RiskMinder or the system that uses RiskMinder components.

SQL Database

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

- Server
- Database
- User Name
- Password
- Port Number

Oracle Database

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

- Service ID (Instance identifier of the Oracle database)
- Host Name
- Port Number
- User Name
- Password

MySQL

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

- Server
- Database
- User Name
- Password
- Port Number

Requirements for Java-Dependent Components

Install the following components required by Administration Console, RiskMinder 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**\. If you fail to do so, then Administration Console and other JDK-dependent components might fail to start.

- Application Server

(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 components before you proceed:

1. HSM Server
2. HSM Client
3. At least one 3DES key created in HSM (You will need this 3DES key for encrypting information in the database).

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

See 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 the following checklist before you proceed with the installation and setup of RiskMinder.

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

Your Information	Example Entry	Your Entry
HARDWARE		
Processor	Intel Xeon X5450 3 GHz	
RAM	2 GB	
Disk Space	20 GB	

Your Information	Example Entry	Your Entry
SOFTWARE		
Operating System	Microsoft Windows Server 2003	
Distribution	Enterprise Edition	
Service Pack (or Patch)	SP3	
DATABASE		
Type	Oracle	
Database Name *(MS SQL Only)*	arcotdb	
DSN Name (If created)	arcotdsn	
Host Name (or Server IP Address)	51.100.25.24	
Port	1521	
Service ID (*Oracle Databases Only*)	oradb1	
User Name	rfdbadmin	
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		
ALTER EXTENT PARAMETERS		
CREATE TABLESPACE (*For Reports*)		

Your Information	Example Entry	Your Entry
UNLIMITED TABLESPACE (*For Reports,* Optional)		
DROP TABLESPACE		
MS SQL Server **Note:** The user performing these actions must belong to the ddladmin role.		
CREATE TABLE		
CREATE INDEX		
CREATE PROCEDURE		
REFERENCES		
DML PRIVILEGES		
CONNECT PRIVILEGES		
ALTER		
MySQL		
SELECT		
INSERT		
UPDATE		
DELETE		
EXECUTE		
CREATE		
ALTER		
CREATE ROUTINE		
ALTER ROUTINE		
DROP		
GRANT OPTION		
APPLICATION SERVER		
Type	Apache Tomcat 5.5.31	
Host Name	localhost	

Your Information	Example Entry	Your Entry
Port	8080	
JDK	1.5.0_10	
DIRECTORY SERVICE		
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=dscc	
Password	mypassword1234!	
WEB SERVER (OPTIONAL)		
Type	IIS 6	
Host Name	mywebserver.com	
Port	443	

Chapter 4: Deploying RiskMinder On a Single System

Use the **Arcot RiskFort 3.1.01 InstallAnywhere Wizard** to install RiskMinder components. This Wizard supports *Complete* and *Custom* installation types. To install and configure RiskMinder on a single computer, use the **Complete** option when you run the installer.

The following steps provide a quick overview of the process:

1. To install RiskMinder components and configure them to access your SQL database, run the RiskMinder installer.

   See "Performing Complete Installation" (see page 66) for install instructions.

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

   See "Running Database Scripts" (see page 74) and "Verifying the Database Setup" (see page 74) for more information.

3. Copy to your application server the files that are required by UDS and Administration Console to function correctly.

   See "Preparing Your Application Server" (see page 75) for more information.

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

   See "Deploying Administration Console" (see page 82) for more information.

5. Log in to Administration Console with the Master Administrator credentials to initialize RiskMinder.

   See "Logging In to Administration Console" (see page 84) and "Bootstrapping the System" (see page 85) for more information.

6. Start RiskMinder Server and Case Management Queuing Server and verify if the services start successfully.

   See "Starting RiskMinder Server" (see page 87), "Starting the Case Management Queuing Server" (see page 88), and "Verifying the Installation" (see page 91) for more information.

7. Deploy User Data Service (UDS) on the application server and verify the deployment.

   See "Deploying User Data Service (UDS)" (see page 89) for more information.

8. To test the RiskMinder configuration, deploy and use Sample Application.

   Note: Sample Application is automatically installed as a part of Complete installation.

   See "Deploying Sample Application" (see page 91) and "Using Sample Application" (see page 92) for more information.

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

   Book: See *CA RiskMinder Administration Guide* for more information.

10. Complete the installation checklist.

    See "Post-Installation Checklist" (see page 96) for more information.

11. (Optional) Change the HSM settings that you specified during the installation process.

    See "Changing Hardware Security Module Information After the Installation" (see page 217) for more information.

Important Notes Related to the Installation

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

- Ensure that the *<install_location> does not contain* any special characters (such as ~ ! @ # $ % ^ & * ( ) _ + = { }[ ] ’ ").
- The MySQL database name should not contain dot(.) characters.
- Currently, you cannot modify or repair RiskMinder 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 the installation (*especially during the last stages*), you click the **Cancel** button to abort the installation, then the installer may not remove *all* the directories that it has created so far. You must manually clean up the installation directory, <*install_location*>\Arcot Systems\, and its subdirectories.
- If you run the installer on a system that already contains an instance of an existing ARCOT_HOME, then:
  - You are not prompted for an installation directory.
  - You re not prompted for the database setup. The installer uses the existing database.
  - You are not prompted to set up encryption.
  - If you have already installed CA AuthMinder release 7.1.01, you will be shown the screens for performing a custom RiskMinder installation.

    You can install and use CA AuthMinder along with CA RiskMinder. Both products use certain common components, which are copied during the installation of either product. If you have already installed AuthMinder 7.1.01 and you are now starting the RiskMinder installation procedure, the RiskMinder installer can detect the presence of the common components that were copied during the AuthMinder installation. The RiskMinder installer then displays the screens for performing a custom installation.

Performing Complete Installation

To install (and later configure) RiskMinder on Microsoft Windows successfully, the user account that you plan to use for installation *must* belong to the Administrators group. Else, some critical steps in the installation, such as DSN creation and configuration, and RiskMinder service creation, do not complete successfully, though the installation may complete without any errors.

Complete installation allows you to install all components of the RiskMinder package. These components include RiskMinder Server and the scripts that are required for setting up the database that you intend to use for RiskMinder.

Note: Before you proceed with the installation, ensure that all prerequisite software components are installed and the database is set up, as described in "Preparing for Installation" (see page 45).

To install the RiskMinder components, perform the following tasks:

1. To run the installation wizard, navigate to the directory where the Arcot-RiskFort-3.1.01-Windows-Installer.exe file is located and double-click the file.

   The Welcome screen appears.

2. Click Next to continue.

   The License Agreement screen appears.

3. Carefully read the license agreement, select the I accept the terms of the License Agreement option, and click Next.

   The installer now checks if any other CA product is installed on the computer.

   If it does not find an existing CA product installation, then you are prompted for an installation directory. In this case, the Installation Location screen appears.

   If the installer detects an existing CA product installation (an existing ARCOT_HOME), then:

   - You are not prompted for an installation directory.
   - In addition, you are not prompted for the database and encryption setup. The installer uses the existing database and encryption settings. As a result, you see the screen in Step 6, though the configuration is disabled, and the screen corresponding to Step 10 is not displayed.

4. Click Next to install in the specified directory.

   The Installation Type screen appears.

5. Select Complete to install all components in one ARCOT_HOME and then click Next to continue.

   The Database Type screen appears.

6. Depending on the type of database you have, you can select Microsoft SQL Server, Oracle Database, or MySQL. Click Next to proceed.

If you selected Microsoft SQL Server, then the SQL Server Database Details screen appears.

Note: If you are using a SQL database, ensure that the ODBC Driver version you are using is the same as the one mentioned in "Preparing for Installation" (see page 45).

If you selected Oracle on the Database Type screen, then the Oracle Database Details screen appears.

Note: CA RiskMinder release 3.1.01 is now certified to work with Oracle Real Application Clusters (Oracle RAC). To use Oracle RAC with your RiskMinder Installation, select Oracle Database in this step, perform the next step (Step 7), and then perform the steps in Configuring CA RiskMinder for Oracle RAC (W) (see page 247).

If you selected MySQL on the Database Type screen, then the MySQL Database Details screen appears.

7. Based on your database choice in the preceding screen:

   - If you selected Microsoft SQL Server, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.
Server	The host name or IP address of the RiskMinder datastore. Default Instance ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase Named Instance ■ **Syntax**: *<server_name>**<instance_name>* ■ **Example**: demodatabase\instance1

Parameter	Description
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.
Password	The password associated with the User Name you specified in the previous field and which is used by RiskMinder to access the database. This password is specified by the database administrator.
Database	The name of the MS SQL database instance.
Port Number	The port at which the database listens to the incoming requests. The default port at which an MS SQL database listens is 1433. However, if you would like to specify another port, enter the port value in this field.

- If you selected Oracle Database, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.

Parameter	Description
Password	The password associated with the **User Name** you specified in the previous field and which is used by RiskMinder 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. The default port at which an Oracle database listens is 1521. However, if you would like to specify another port, enter the port value in this field.
Host Name	The host name or IP address of the RiskMinder datastore. ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase

- If you selected MySQL, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.
Server	The host name or IP address of the RiskMinder datastore. Default Instance ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase Named Instance ■ **Syntax**: *<server_name>**<instance_name>* ■ **Example**: demodatabase\instance1

Parameter	Description
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.
Password	The password associated with the User Name you specified in the previous field and which is used by RiskMinder to access the database. This password is specified by the database administrator.
Database	The name of the MySQL database instance.
Port Number	The port at which the database listens to the incoming requests. The default port at which an MySQL database listens is 3306. However, if you would like to specify another port, enter the port value in this field.

1. After you specify the database details, test if you can successfully connect to the database by clicking the **Test Data Source** button and verify the result of the same in the field below the button.

   Note: If the connection was not successful, ensure that you have specified the correct database details and click **Test Data Source** again. Proceed only if the database connectivity is successful.

2. Click **Next** to continue.

   The Encryption Configuration screen appears. Use this screen to select the encryption mode and configure the information that is used for encryption.

3. Specify the following information:

Field Name	Description
Master Key	Specify the password for the Master Key, which is stored at *<install_location>*\Arcot Systems\conf\securestore.enc and will be used to encrypt the data stored in the database. By default, this value is set to MasterKey. **Note:** If you want to change the value of Master Key *after* the installation, then regenerate securestore.enc with a new Master Key value. See "Changing Hardware Security Module Information After the Installation" (see page 217) for more information.

Field Name	Description
Configure HSM	Select this option only if you will use a Hardware Security Module (HSM) to encrypt the sensitive data. If you do not select this option, then, by default, the data is encrypted by using the Software Mode.
PIN	Enter the password to connect to the HSM.
Choose Hardware Module	Choose one of the following HSMs that you plan to use: ■ Luna HSM ■ nCipher netHSM
HSM Parameters **Tip:** The HSM parameter values are recorded in arcotcommon.ini, which is available in <install_location>\Arcot Systems**conf**\. To change these values *after* installation, edit this file, as discussed in Configuration Files and Options" (see page 195).	Set the following HSM information: ■ **Shared Library:** The absolute path to the PKCS#11 shared library corresponding to the HSM. For Luna (cryptoki.dll) and for nCipher netHSM (cknfast.dll), specify the absolute path and name of the file. ■ **Storage Slot Number:** The HSM slot where the 3DES keys used for encrypting the data are available. – For **Luna**, the default value is 0. – For **nCipher netHSM**, the default value is 1.

The Pre-Installation Summary screen appears.

1. Review the information about this screen, and if you must change a previous selection, then click **Previous** to do so. After you change the required selection, click **Next** to go to the next screen.
2. Click **Install** to begin the installation process.

   The Microsoft Visual C++ 2010 x86 Redistributable Setup screen appears. *This screen appears only if the current system where you are installing RiskMinder does not have Microsoft Visual C++ 2010 x86.*
3. On the Microsoft Visual C++ 2010 x86 Redistributable Setup screen:
   a. Select the **I have read and accept the license terms** option, and click **Install**.

      The Installation Progress screen appears. This may take a few seconds. After some time the Installation Is Complete screen appears.

   b. Click **Finish** to close the Microsoft Visual C++ 2010 x86 Redistributable Setup dialog box and continue with the RiskMinder installation.

   The Installing Arcot RiskFort screen appears. This may take several minutes.

   After some time the Installation Complete screen appears.
4. Click **Done** to complete the RiskMinder installation.

Note: After the installation is completed, perform the post-installation tasks that are discussed in "Performing Post-Installation Tasks" (see page 73).

Installation Logs

After installation, you can access the installation log file (Arcot_RiskFort_Install_<*timestamp*>.log) in the <*install_location*> directory. For example, if you had specified the C:\Program Files directory as the installation directory, then the installation log file is created in the C:\Program Files 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 RiskMinder. These steps are required for configuring RiskMinder correctly and must be *performed in the following order*:

1. Running Database Scripts (see page 74)
2. Verifying the Database Setup (see page 74)
3. Preparing Your Application Server (see page 75)
4. Deploying Administration Console (see page 82)
5. Logging In to Administration Console (see page 84)
6. Bootstrapping the System (see page 85)
7. Starting RiskMinder Server (see page 87)
8. Starting the Case Management Queuing Server (see page 88)
9. Deploying User Data Service (UDS) (see page 89)
10. Deploying Sample Application (see page 91)
11. Verifying the Installation (see page 91)
12. Using Sample Application (see page 92)

Note: After you complete these post-installation tasks, perform the SDK and Web services configuration tasks that are discussed in "Configuring RiskMinder SDKs and Web Services" (see page 139).

Running Database Scripts

Important! Before you run the scripts that are discussed in this section, ensure that you are logged in as the same database user that you created in the section, "Configuring Database Server" (see page 51).

RiskMinder is shipped with scripts that are required to create necessary tables in the RiskMinder database. To run the required database scripts:

1. Navigate to the following directory:
   `<install_location>\Arcot Systems\dbscripts\`
2. Based on the database that you are using, navigate to one of the following subdirectories:
   - For Oracle: oracle\
   - For Microsoft SQL: mssql\
   - For MySQL: mysql\
3. Irrespective of whether you have a single database for reports *and* transactions, or a separate database for reports, run the scripts *in the following order*:
   a. arcot-db-config-for-common-2.0.sql

      Important! If you have installed CA AuthMinder 7.1.01, do not run arcot-db-config-for-common-2.0.sql because you have already run it while installing CA AuthMinder 7.1.01.

   b. arcot-db-config-for-riskfort-3.1.01.sql

   c. (**Optional**, only if you must create the 3D Secure Channel) arcot-db-config-for-3dsecure-3.1.01.sql

Verifying the Database Setup

After you run the required database scripts, verify that the RiskMinder schemas were seeded correctly. To do so:

1. Log in to the RiskMinder database as the user who installed the database.

   Note: If you are following the upgrade path, then log in to the database as the user who upgraded the database.

2. Run the following query:
   ```
   SELECT SERVERNAME, VERSION FROM ARRFSERVERS;
   ```
   You must see the following output as a result of the preceding query:
   ```
   SERVERNAME                VERSION
   ------------------------  ----------------
   RiskFort                  3.1.01
   RiskFortCaseManagement    3.1.01
   ```
3. Log out of the database console.

Preparing Your Application Server

Two components of RiskMinder, 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 Application Server

Before you deploy the WAR files for these web applications on the application server of your choice, copy the files that are 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 and to deploy the WAR files of these web applications:

- Step 1: Setting Java Home (see page 75)
- Step 2: Copying Database Access Files to Your Application Server (see page 76)
- Step 3: Copying JDBC JAR Files to Your Application Server (see page 79)
- Step 4: (Mandatory for Oracle WebLogic 10.1) Creating Enterprise Archive Files (see page 81)

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. This JAVA_HOME must be your application server JAVA_HOME.

In addition, %JAVA_HOME%**bin**\ must be added to the PATH variable. If you fail to do so, then Administration Console, UDS, 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 RiskMinder database securely:

- arcot-crypto-util.jar available at:
  `<install_location>\Arcot Systems\java\lib\`
- ArcotAccessKeyProvider.dll available at:
  `<install_location>\Arcot Systems\native\win\<32bit-or-64bit>\`

As a result, these files must be copied to the appropriate location on the application server where you have deployed these RiskMinder 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 files that are required for database access:

1. Copy arcot-crypto-util.jar *to* *<Tomcat_JAVA_HOME>*\jre\lib**ext**\.

   Here, *<Tomcat_JAVA_HOME>* represents the JAVA_HOME used by your Apache Tomcat instance.

2. Copy ArcotAccessKeyProvider.dll *to* *<Tomcat_JAVA_HOME>*\jre**bin**\.
3. Restart the application server.

IBM WebSphere

To copy the files that are required for database access:

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 or 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, C:\Program Files\Arcot Systems\java\lib**arcot-crypto-util.jar**.

   e. Enter the JNI Library path.

This path must point to the location where the ArcotAccessKeyProvider.dll file is present.

3. Click **Apply** to save the changes.
4. Configure the 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 must be performed.
   c. Click **Java and Process Management** and then click **Class Loader**.
   d. Click **New**.
   e. Select default **Classes loaded with parent class loader first** and click **OK.**
   f. Click the auto-generated **Class Loader ID**.
   g. On the class loader **Configuration** page, click **Shared Library References**.
   h. Click **Add**, select **ArcotJNI,** and then click **Apply**.
   i. Save the changes.
5. Copy ArcotAccessKeyProvider.dll to <*WebSphere_JAVA_HOME*>\jre**bin**\.

   Here, <*WebSphere_JAVA_HOME*> represents the JAVA_HOME used by your IBM WebSphere instance.
6. Restart the application server.

Oracle WebLogic

To copy the files that are required for database access:

1. Copy ArcotAccessKeyProvider.dll to <*WebLogic_JAVA_HOME*>\jre**bin**\.

   Here, <*Weblogic_JAVA_HOME*> represents the JAVA_HOME used by your Oracle WebLogic instance.
2. Copy arcot-crypto-util.jar to <*WebLogic_JAVA_HOME*>\jre\lib**ext**\.

   Note: Ensure that you use the appropriate <*JAVA_HOME*> used by WebLogic.
3. Log in to WebLogic Administration Console.
4. Navigate to **Deployments**.
5. Enable the **Lock and Edit** option.
6. Click **Install** and navigate to the directory that contains the arcot-crypto-util.jar file.
7. Click **Next** to open the Application Installation Assistant.
8. Click **Next** to display the Summary page.
9. Click **Finish**.
10. Activate the changes.

11. Restart the application server.

JBoss Application Server

To copy the files that are required for database access:

1. Copy ArcotAccessKeyProvider.dll *to* <*JBoss_JAVA_HOME*>\jre**bin**\.

   Here, <*JBoss_JAVA_HOME*> represents the JAVA_HOME used by your JBoss Application Server instance.

2. Copy arcot-crypto-util.jar *to* <*JBoss_JAVA_HOME*>\jre\lib**ext**\.

3. Restart the application server.

Step 3: Copying JDBC JAR Files to Your Application Server

RiskMinder requires the following JDBC JAR files for the supported databases:

- **Oracle 10g:** Oracle JDBC Driver (10.2.0.1.0)
- **Oracle 11g:** Oracle JDBC Driver (11.2.0.2.0)
- **Microsoft SQL Server:** MSSQL JDBC Driver (1.2.2828)
- **MySQL:** MySQL JDBC Driver (5.1.22)

The following subsections 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 required JDBC JAR file:

1. Navigate to the location where you have downloaded the *<Database_JAR>* file.
2. Copy the *<Database_JAR>* file to the following directory:
   - **On Apache Tomcat 5.5.x:** *<TOMCAT_HOME>*\common\lib\
   - **On Apache Tomcat 6.x and 7.x:** *<TOMCAT_HOME>*\lib\
3. Restart the server.

IBM WebSphere

To copy the required JDBC JAR file:

1. Log in to the WebSphere Administration Console.
2. Click Environment, and then click Shared Libraries.
   a. From the Scope list, select a valid visibility scope. The scope must include the target server or node on which the application is deployed.
   b. Click New.
   c. Enter the Name, say, JDBCJAR.
   d. Specify the Classpath.

      Important! This path *must* point to the location where the *<Database_JAR>* file is present and *must* include the file name.

   e. Click Apply to save the changes that were made.
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 is performed.

c. Click **Java and Process Management**, and then click **Class Loader**.

d. Click **New**.

e. Select default **Classes loaded with parent class loader first** and click **OK.**

f. Click the auto-generated **Class Loader ID**.

g. In the class loader **Configuration** page, click **Shared Library References**.

h. Click **Add**, select **JDBCJAR**, and then click **Apply**.

i. Save the changes that were made.

4. Restart the application server.

Oracle WebLogic

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

To copy the required JDBC JAR file in the case of Microsoft SQL Server:

1. Copy the *<Database_JAR>* file to *<Weblogic_JAVA_HOME>*\lib**ext**\.

   Here, *<WebLogic_JAVA_HOME>* represents the JAVA_HOME used by your Oracle WebLogic instance.

2. Log in to the 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 required *<Database_JAR>* file.
6. Click **Next** to display the Application Installation Assistant page.
7. Click **Next** to display the Summary page.
8. Click **Finish**.
9. Activate the changes.
10. Restart the application server.

JBoss Application Server

To copy the required JDBC JAR file:

1. Copy the JDBC JAR file to the following location on the JBOSS installation directory: *<JBOSS_HOME>*`\server\default\lib\`
2. Restart the application server.

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

Most enterprise Application Servers (such as WebSphere and Weblogic) enable you to bundle the related Java ARchive (JAR) or Web ARchive (WAR) files from one vendor (say, CA) to 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, WAR files are provided to deploy UDS and Administration Console. However if necessary, you can also change the format of these files to Enterprise ARchive (EAR) and then deploy the EAR files.

As discussed in the following subsections, you can either generate separate EAR files for both UDS and Administration Console, or you can generate a single EAR file that contains both web archives.

Generating Separate EAR Files

To create a separate EAR file each for UDS and Administration Console:

1. Open the Command Prompt window.
2. Navigate to the <*install_location*>\Arcot Systems\tools\common**bundlemanager**\ directory.
3. To create the EAR file, run the following command:
   `java -jar bundle-manager.jar -ear <filename.ear> -warList <filename.war>`

   The preceding command generates individual EAR files that are available at:
   `<install_location>\Arcot Systems\java\webapps\`

Generating a Single EAR File

To create a single EAR file that contains UDS and Administration Console Web archives:

1. Open the Command Prompt window.
2. Navigate to the <*install_location*>\Arcot Systems\tools\common**bundlemanager**\ directory.
3. To create the EAR file, run the following command:
   `java -jar bundle-manager.jar -ear <filename.ear> -warList arcotadmin.war arcotuds.war`

   The preceding command generates a single EAR file that is available at:
   `<install_location>\Arcot Systems\java\webapps\`

Deploying Administration Console

Note: If you are deploying the Administration Console on IBM WebSphere 7.0, then see the instructions in appendix, "Deploying Administration Console on IBM WebSphere 7.0" (see page 273) instead of the following instructions.

Administration Console is a browser-based interface to RiskMinder that enables you to customize the server configurations and manage the deployed system.

You need the **arcotadmin.war** file to deploy the RiskMinder Administration Console. All Administration Console information is logged in the arcotadmin.log file. After you deploy arcotadmin.war, you can verify if it was correctly deployed by using this log file (arcotadmin.log), which is available in the $ARCOT_HOME/arcot/logs directory.

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

To deploy the Administration Console WAR file on your application server and verify if it was successfully deployed, follow these steps:

1. Deploy arcotadmin.war in the appropriate directory on the application server.

   Note: The deployment procedure depends on the application server that you are using. See 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**\.

2. **(For 32-bit WebSphere Only)** Configure reload of the Admin class when the application files are updated.

   a. Navigate to Application, Enterprise Applications, and then access the Admin 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.

   e. Restart the Admin application.

3. **(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 Systems\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>
```

Add the following category for cryptographic operations:

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

f. Save and close the file.

g. Take a backup of the existing JBoss logging libraries. These library files are available at:
<*JBOSS_HOME*>\lib\

h. 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/*

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

4. Restart the application server.
5. Verify that the console was successfully deployed:
   a. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`
   b. Open the arcotadmin.log file in any editor and locate the following lines:
      - 2.0.3
      - Arcot Administration Console Configured Successfully.

      These lines indicate that your Administration Console was deployed successfully.
   c. Also ensure that the log files *do not* contain any FATAL *and* WARNING messages.
   d. Close the file.

Logging In to Administration Console

When you log in to Administration Console for the first time, use the Master Administrator (MA) credentials that are configured automatically in the database during the deployment.

To log in to Administration Console as MA:

1. Launch the Administration Console in a Web browser window. The default URL for Administration Console is:
   `http://<host>:<appserver_port>/arcotadmin/masteradminlogin.htm`

   Note: The *host* and *port* information that you specify in the preceding URL must be of the application server where you deployed Administration Console.
   For example, in case of Apache Tomcat, the default *host* is localhost and *port* is 8080.
2. Log in by using the default Master Administrator account credentials. The credentials are:
   - **User Name: masteradmin**
   - **Password: master1234!**

Bootstrapping the System

Before you start using Administration Console to manage RiskMinder, perform the following mandatory steps to initialize the system:

- Change the default Master Administrator password
- Configure the Global Key label
- Specify the configuration settings for the out-of-the-box organization

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

Before you proceed with "Performing the Bootstrapping Tasks" (see page 86), understand the related concept of "Default Organization".

Default Organization

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

Performing Bootstrapping Tasks

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

To bootstrap the system by using the wizard:

1. Click Begin to start the process.

   The Change Password screen appears.

2. Specify the Current Password, New Password, Confirm Password, and click Next.

   The Configure Global Key Label screen appears.

3. On the Configure a Global Key Label page:

   - Specify the Global Key Label, and click Next.

     RiskMinder enables you to use hardware- or software-based encryption of your sensitive data. (You can enable hardware-based encryption by using arcotcommon.ini (see page 198) file, while software-based encryption is enabled by default.) Irrespective of hardware or software encryption, *Global Key Label* is used 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 *must* match the HSM key label. However in case of software-based encryption, this label acts as the key.

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

   - Specify the Storage Type to indicate whether the encryption key is stored in the database (Software) or the HSM (Hardware).

4. Click Next to continue.

   The Configure Default Organization screen appears.

5. 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 who belong to the Default Organization. Administration Console supports three types of authentication methods for the administrators to log in:

     - LDAP User Password

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

Note: If this mechanism is used for authenticating administrators, then deploy UDS, as discussed in the section, "Deploying User Data Service (UDS)" (see page 89).

- Basic

  If you select this option, then the built-in authentication method that is provided by Administration Console is used for authenticating the administrators.

- WebFort Password

  If you select the WebFort Password option here, then the credentials are issued and authenticated by the AuthMinder Server. For this, the CA AuthMinder Server must be installed.

Note: For information about installing and configuring AuthMinder, see the *CA AuthMinder Installation and Deployment Guide*.

6. Under the Key Label Configuration section of the Configure Default Organization screen, specify the following values:
   - **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 label for encryption.
   - **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).
7. Click Finish to complete the bootstrapping process.

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

Starting RiskMinder Server

To start RiskMinder Server:

1. Click the **Start** button on your desktop window.
2. Navigate to **Settings, Control Panel, Administrative Tools,** and then **Services.**
3. Locate and double-click **Arcot RiskFort Service**.
4. Click **Start** in the service window.

Note: If you want to stop RiskMinder Server, then follow the Steps 1 through 3, and click the **Stop** button in the service window.

Starting the Case Management Queuing Server

To start Case Management Queuing Server:

1. Click the **Start** button on your desktop window.
2. Navigate to **Settings, Control Panel, Administrative Tools,** and then **Services.**
3. Locate and double-click the **Arcot RiskFort Case Management Queuing Service**.
4. Click **Start** in the service window.

Note: If you want to stop the Case Management Queuing Server, then follow the Steps 1 through 3, and click the **Stop** button in the service window.

Deploying User Data Service (UDS)

RiskMinder can access user data either from a relational database (RDBMS) or directly from an LDAP server by using UDS, which is an abstraction layer that provides RiskMinder seamless access to the third-party data repositories deployed by your organization.

You need the arcotuds.war file to deploy UDS, as follows:

1. Deploy arcotuds.war on the application server. This file is available at: <*install_location*>\Arcot Systems\java\webapps\

   For example, in the case of Apache Tomcat, deploy the WAR file at <*APP_SERVER*_HOME>\webapps\.

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

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 to save the changes.

3. **(For JBoss Only)** Perform the following steps, if you have deployed UDS on a JBoss application server:

   a. Copy the Bouncy Castle JAR file (bcprov-jdk15-146.jar) from <*install_location*>\Arcot Systems\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 while Deploying Administration Console (see page 325):

```
<appender-ref ref="arcotudslog"></appender-ref>
```

f. Add the following line in the cryptographic category that you created while Deploying Administration Console (see page 325):

```
<appender-ref ref="arcotudslog"></appender-ref>
```

g. Save and close the file.

4. Restart the application server.
5. Verify if UDS was deployed successfully:

   Note: The arcotuds.log file is used for logging UDS-related information.

   a. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`

   b. Open the arcotuds.log file in any editor and locate the following line:

   - User Data Service (Version: 2.0.3) initialized successfully.

   This line indicates that UDS was deployed successfully.

   c. Also ensure that the log files *do not* contain any FATAL *and* WARNING messages.

   d. Close the file.

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 RiskMinder was installed and configured properly. In addition, it demonstrates:

- The typical RiskMinder workflows
- The basic operations (invocation and post-processing) of RiskMinder APIs
- Integration of your application with RiskMinder

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

1. Deploy the riskfort-3.1.01-sample-application.war file from the following location: `<install_location>\Arcot Systems\samples\java\`
2. If necessary, restart the application server.
3. Access Sample Application in a Web browser window. The default URL for Sample Application is: *http://<host>:<appserver_port>/riskfort-3.1.01-sample-application/index.jsp*

Verifying the Installation

After you have seeded the database schema, deployed UDS and Administration Console, and bootstrapped the system, and started the Server, ensure that all these components have started correctly. The log files that you must verify for this purpose is arcotriskfort.log.

To verify if the server started correctly:

1. Navigate to the following location: `<install_location>\Arcot Systems\logs\`
2. Open the arcotriskfortstartup.log file in any editor and locate the following lines:
   - STARTING Arcot RiskFort 3.1.01_w
   - Arcot RiskFort Service READY
3. Open the arcotriskfortcasemgmtserverstartup.log file in any editor and locate the following lines:
   - STARTING Arcot RiskFort Case Management 3.1.01_w
   - Arcot RiskFort Case Management Service READY

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

Using Sample Application

This subsection describes the risk-evaluation operations that can be performed by using Sample Application. Each operation in Sample Application is designed to run without error when RiskMinder is installed and functional.

Sample Application demonstrates the following operations that RiskMinder Server can perform:

- Performing Risk Evaluation and Post Evaluation for a First-Time User (see page 93)
- Creating Users (see page 94)
- Performing Risk Evaluation and Post Evaluation for a Known User (see page 95)
- Editing the Default Profile and Performing Risk Evaluation (see page 96)

Performing Risk Evaluation and Post Evaluation for a First-Time User

To perform risk evaluation on the default profile of a user:

1. Ensure that Sample Application is open (in a Web browser window.) The default URL for Sample Application is:
   http://<host>:<appserver_port>/riskfort-3.1.01-sample-application/index.jsp
2. Click Evaluate Risk to open the Risk Evaluation page.
3. On the page, specify the name of the user (who you want to evaluate) in the User Name field.
4. If necessary, specify the name of the organization to which the user belongs in the User Organization field.
5. If necessary, specify the Channel from which the transaction originated.
6. Click Evaluate Risk to open the Risk Evaluation Results page.

   This page displays the Risk Score, the associated Risk Advice, and lists the rules that are configured for the specified organization. For a first-time user, the result is ALERT.
7. Click Next Step to open the Post Evaluation page and perform post-evaluation on the specified user profile.

   By using Post evaluation, your application provides feedback to RiskMinder Server about the current user and the device they are using. RiskMinder updates user and device attributes and the user-device association based on this feedback, and accordingly assesses the risk that is associated with the transactions for the user in future.
8. Select the result of secondary authentication by selecting the appropriate option from the Result of Secondary Authentication list.
9. Specify the name for the user name-device association in the Association Name field.
10. Click Post Evaluate to complete the post evaluation process and to display the result of the same in the Post Evaluation Results section.

Creating Users

To create a user:

1. Create a GA account:
    a. Log in to Administration Console as the MA.
    b. Ensure that the **Users and Administrators** tab is active.
    c. In the menu on the left side, click the **Create Administrator** link to display the Create Administrator page.
    d. Specify the details on the page and click **Next**.
    e. On the Create Administrator page, select **Global Administrator** from the **Role** list.
    f. Specify the **Password** and **Confirm Password**.
    g. Select the **All Organizations** option in the **Manages** section.
    h. Click **Create**.
    i. Click **Logout** in the top right-hand corner of the page to log out as the MA.
2. Log in to Administration Console as a Global Administrator (GA) or an Organization Administrator (OA). The URL for the purpose is:
http://<host>:<appserver_port>/arcotadmin/adminlogin.htm
3. Follow the instructions that are displayed to change your password.
4. If already not activated, activate the **Manage Users and Administrators** subtab under the **Users and Administrators** tab.
5. In the pane on the left side, under **Manage Users and Administrators**, click **Create User** to open the Create User page.
6. 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 necessary, 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.
7. Return to the RiskMinder Sample Application page.

Performing Risk Evaluation and Post Evaluation for a Known User

1. On the Main Page of Sample Application, click **Evaluate Risk** to open the Risk Evaluation page.
2. In the **User Name** field, specify the name of the user that you created in the section, "Creating Users" (see page 94).
3. Specify the user's organization in the **User Organization** field.
4. If necessary, specify the **Channel** from which the transaction originated.
5. Click **Evaluate Risk** to open the Risk Evaluation Results page.

   The Risk Advice typically is **INCREASEAUTH**.
6. Click **Store DeviceID** to store the specified type of Device ID information on the end user's device.
7. Click **Next Step** to perform Post Evaluation:
   - Select the **Result of Secondary Authentication** from the list.
   - Edit the **Association Name**, if necessary.
8. Click **Post Evaluate** to display the final advice.

   If you repeat Step 1 through Step 5, the **Risk Advice** changes to **ALLOW** on the Risk Evaluation Results page.

Editing the Default Profile and Performing Risk Evaluation

Using Sample Application, you can change the DeviceDNA, IP address, and the Device ID of the computer that you are using to simulate various scenarios. To edit the default profile of a user:

1. On the Main Page of Sample Application, click Evaluate Risk to open the Risk Evaluation page.
2. Specify the user name whose profile you want to edit in the User Name field.
3. Specify the user's organization in the User Organization field.
4. Click Edit Inputs to open the Edit Risk-Evaluation Inputs page.
5. On the page, all fields are prepopulated. Change the values for one or more of the required fields:
   - My User Name
   - My Org
   - My Channel
   - Machine Finger Print of My Device
   - Short Form of Machine Finger Print of My Device
   - IP Address of My Machine
   - Device ID of My Machine
6. Click Evaluate Risk to open the Risk Evaluation Results page.
7. Click Next Step to open the Post Evaluation page and perform postevaluation on the specified user profile.
8. Select the result of secondary authentication by selecting the appropriate option from the Result of Secondary Authentication list.
9. Click Post Evaluate to complete postevaluation and display the result of the same.

Note: To ensure secure communication between the RiskMinder components, you can configure them to support SSL (Secure Socket Layer) transport mode. For more information, see "Configuring SSL" in the *CA RiskMinder Administration Guide*.

Applying the Post-Installation Checklist

It is recommended that you fill the following checklist with the installation and setup information for RiskMinder. This information is useful when you perform various administrative tasks.

Your Information	Example Entry	Your Entry
ARCOT_HOME	C:\Program Files\Arcot Systems	

Your Information	Example Entry	Your Entry
SYSTEM INFORMATION		
Host Name	my-bank	
User Name	administrator	
Password	password1234!	
Configured Components	RiskFort 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 RiskMinder on a Distributed System

To install the RiskMinder components, use the Arcot RiskFort 3.1.01 InstallAnywhere Wizard. This Wizard supports *Complete* and *Custom* installation types. To install and configure RiskMinder in a distributed environment, use the **Custom** option when you run the installer.

The following steps provide a quick overview of the process:

1. To install RiskMinder Server and Administration Console and to configure them to access your SQL database, run the RiskMinder installer. You can also install the web services on the same system.

   See Installing on the First System (see page 102) for installation instructions.

2. To create RiskMinder schema and database tables, execute the database scripts. Also ensure that the database setup was successful.

   See Running Database Scripts (see page 113) and Verifying the Database Setup (see page 113) for more information.

3. Copy to your application server the files that UDS and Administration Console require to function correctly.

   See Preparing Your Application Server (see page 114) for more information.

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

   See Deploying Administration Console (see page 121) for more information.

5. Log in to Administration Console with the Master Administrator credentials to initialize RiskMinder.

   See Logging In to Administration Console (see page 123) and Bootstrapping the System (see page 124) for more information.

6. Start RiskMinder Server and the Case Management Queuing Server, and verify if they start successfully.

See Starting RiskMinder Server (see page 126), Starting the Case Management Queuing Server (see page 127), and Verifying the Installation (see page 127) for more information.

1. Deploy User Data Service (UDS) on the application server and verify the deployment.

   See Deploying User Data Service (UDS) (see page 128) for more information.

2. Install the Java SDKs and web services on one or more systems.

   See Installing on the Second System (see page 130) for more information.

3. To test the RiskMinder configuration, deploy, configure, and use Sample Application.

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

   See Deploying Sample Application (see page 131), Configuring Sample Application for Communication with RiskMinder Server (see page 132), and Using Sample Application (see page 133) for more information.

4. (Optional) To ensure secure communication between the RiskMinder components, you can configure them to support SSL (Secure Sockets Layer) transport mode.

   Book: See *CA RiskMinder Administration Guide* for more information.

5. Complete the installation checklist.

   See Post-Installation Checklist (see page 138) for more information.

6. (Optional) Change the HSM settings that you specified during the installation.

   See Changing Hardware Security Module Information After the Installation (see page 217) for more information.

Important Notes Related to Installation

Keep the following points in mind while installing RiskMinder either on a single system or in a distributed environment:

- Ensure that the *<install_location> does not contain* any special characters, such as ~ ! @ # $ % ^ & * ( ) _ + = { }[ ] ’ ".
- The MySQL database name should not contain dot(.) characters.
- Currently, you cannot modify or repair the RiskMinder 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 the installation (*especially during the last stages*), you click the Cancel button to abort the installation, then the installer may not remove *all* the directories that it has created so far. You must manually clean up the installation directory, <*install_location*>\Arcot Systems\, and its subdirectories.
- If you run the installer on a system that already contains an instance of an existing ARCOT_HOME, then:
  - You are not prompted for an installation directory.
  - You are not prompted for the database setup. The installer uses the existing database.
  - You are not prompted to set up encryption.

Installing on the First System

To install (and later configure) RiskMinder on Microsoft Windows successfully, the user account that you plan to use for the installation *must* belong to the Administrators group. Else, some critical steps in the installation, such as DSN creation and configuration, and RiskMinder service creation, are not completed successfully, though the installation may complete without any errors.

In a distributed scenario, irrespective of how many systems you are distributing RiskMinder, Administration Console, Java SDKs, and web services across, you typically install RiskMinder 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 you proceed with the installation, ensure that all prerequisite software components are installed and the database is set up, as described in "Preparing for Installation" (see page 45).

To install the RiskMinder components, perform the following tasks:

1. Navigate to the directory where the Arcot-RiskFort-3.1.01-Windows-Installer.exe file is located and double-click the file to run the installation wizard.

   The Welcome screen appears.

2. Click Next to continue.

   The License Agreement screen appears.

3. Carefully read the license agreement, select the I accept the terms of the License Agreement option, and click Next.

   The installer now checks if any other CA product is installed on the computer.

   If it does not find an existing CA product installation, then you are prompted for an installation directory. In this case, the Installation Location screen appears.

   If the installer detects an existing CA product installation (an existing ARCOT_HOME), then:

   - You are not prompted for an installation directory.
   - In addition, you are not prompted for the database and encryption setup. The installer uses the existing database and encryption settings. As a result, you will see the screen in Step 8, though the configuration is disabled, and the screen corresponding to Step 12 is not displayed at all.

4. Click Next to install in the specified directory.

   The Installation Type screen appears.

5. Select Custom to install the selected components in one ARCOT_HOME.

   The Component Selection screen appears.

6. Deselect the components that are not required. By default, all components are selected for the installation.

   For example, to install RiskMinder Server, Case Management Queuing Server, and Administration Console (*without* the SDKs and Sample Application) on the current system, you must select *only* the following options:

   a. Arcot Risk Evaluation Server

   b. Arcot Case Management Queuing Server

   c. Arcot Administration Console

   d. Arcot User Data Service

   Note: To install Sample Application *only*, select the Arcot RiskFort SDKs and Sample Application option and then proceed with the installation.

   The following table describes all components that are installed by the RiskMinder installer.

Component	**Description**
Arcot Risk Evaluation Server	This option installs the core Processing engine (RiskMinder Server) that serves the following requests from Administration Console: ■ Risk Evaluation ■ Configuration In addition, this component also installs the following Web services that have been built into the server: ■ **Risk Evaluation Web Service:** Provides the web-based programming interface for risk evaluation with RiskMinder Server. ■ **User Management Web Service**: Provides the web-based programming interface for the creation and management of users. ■ **Administration Web Service**: Provides the web-based programming interface that is used by RiskMinder Administration Console.
Arcot Case Management Queuing Server	This option installs the core Queuing engine (Case Management Queuing Server) that allocates cases to the Customer Support Representatives (CSRs) who work on these cases. **Note:** At any given point in time, *all* instances of Administration Console can only connect to this single instance of Case Management Queuing Server.

Component	Description
Arcot RiskFort SDKs and Sample Application	This option provides programming interfaces (in form of APIs and Web services) that can be invoked by your application to forward risk evaluation requests to RiskMinder Server. This package comprises the following sub-components: ■ **Risk Evaluation SDK**: Provides the Java programming interface for risk evaluation with RiskMinder Server. ■ **Sample Application**: Demonstrates the usage of RiskMinder Java APIs. In addition, it can also be used to verify if RiskMinder was installed successfully, and if it is able to perform risk evaluation requests. Refer to "Configuring RiskMinder SDKs and Web Services" (see page 139) for more information on configuring these components.
Arcot Administration Console	This option provides the Web-based interface for managing RiskMinder Server and risk evaluation-related configurations.
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.)

Note: If you did not select the Arcot Risk Evaluation Server option on this screen, then screens in Step 7 through Step 9 does not appear.

1. Select Next to continue.

   The Database Type screen appears.

2. Depending on the type of database you have, you can select Microsoft SQL Server, Oracle Database, or MySQL. Click Next to proceed.

   If you selected Microsoft SQL Server on the Database Type screen, then the Microsoft SQL Server Database Details screen appears.

   Note: If you are using Microsoft SQL Server database, then ensure that the ODBC Driver version you are using is the same as the one mentioned in the Configuring Database Server (see page 51).

   If you selected Oracle Database on the Database Type screen, then the Oracle Database Details screen appears.

   Note: CA RiskMinder release 3.1.01 is now certified to work with Oracle Real Application Clusters (Oracle RAC). To use Oracle RAC with your RiskMinder Installation, select Oracle Database in this step, perform the next step (Step 9), and then perform the steps in Configuring CA RiskMinder for Oracle RAC (W) (see page 247).

   If you selected MySQL on the Database Type screen, then the MySQL Database Details screen appears.

3. Based on your database choice in the preceding screen:

   - If you selected Microsoft SQL Server, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.

Parameter	Description
Server	The host name or IP address of the RiskMinder datastore. Default Instance ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase Named Instance ■ **Syntax**: *<server_name>**<instance_name>* ■ **Example**: demodatabase\instance1
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.
Password	The password associated with the User Name you specified in the previous field and which is used by RiskMinder to access the database. This password is specified by the database administrator.
Database	The name of the MS SQL database instance.
Port Number	The port at which the database listens to the incoming requests. The default port at which an MS SQL database listens is 1433. However, if you would like to specify another port, enter the port value in this field.

■ If you selected Oracle Database, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.

Parameter	Description
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.
Password	The password associated with the User Name you specified in the previous field and which is used by RiskMinder 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. The default port at which an Oracle database listens is 1521. However, if you would like to specify another port, enter the port value in this field.
Host Name	The host name or IP address of the RiskMinder datastore. ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase

- If you selected MySQL, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.

Parameter	Description
Server	The host name or IP address of the RiskMinder datastore. Default Instance ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase Named Instance ■ **Syntax**: *<server_name>**<instance_name>* ■ **Example**: demodatabase\instance1
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.
Password	The password associated with the User Name you specified in the previous field and which is used by RiskMinder to access the database. This password is specified by the database administrator.
Database	The name of the MySQL database instance.
Port Number	The port at which the database listens to the incoming requests. The default port at which an MySQL database listens is 3306. However, if you would like to specify another port, enter the port value in this field.

1. After you specify the database details, test if you can successfully connect to the database by clicking the Test Data Source button and then verify the result of the same in the field below the button.

   Note: If the connection was not successful, ensure that you have specified the correct database details and click Test Data Source again. Proceed only if the database connectivity is successful.

2. Click Next to continue.

   The Encryption Setup screen appears. Use this screen to select the encryption mode and configure the information that is used for encryption.

3. Specify the following information:

Field Name	Description
Master Key	Specify the password for the Master Key, which is stored at *<install_location>*\Arcot Systems\conf\securestore.enc and will be used to encrypt the data stored in the database. By default, this value is set to MasterKey. **Note:** If you want to change the value of Master Key *after* the installation, then you must regenerate securestore.enc with a new Master Key value. See Changing Hardware Security Module Information After the Installation (see page 217) for more information.
Configure HSM	Select this option only if you will use a Hardware Security Module (HSM) to encrypt the sensitive data. If you do not select this option, then, by default, the data is encrypted by using the Software Mode.
PIN	Enter the password to connect to the HSM.
Choose Hardware Module	Choose one of the following HSMs that you plan to use: ■ Luna HSM ■ nCipher netHSM

Field Name	Description
HSM Parameters **Note:** The HSM parameter values are recorded in arcotcommon.ini, that is available at <install_location>\Arcot Systems\conf\. To change these values *after* installation, edit this file, as discussed in appendix, "Configuration Files and Options" (see page 195).	Set the following HSM information: ■ **Shared Library:** The absolute path to the PKCS#11 shared library corresponding to the HSM. For Luna (cryptoki.dll) and for nCipher netHSM (cknfast.dll), specify the absolute path and name of the file. ■ **Storage Slot Number:** The HSM slot where the 3DES keys used for encrypting the data are available. For Luna, the default value is 0. For nCipher netHSM, the default value is 1.

The Pre-Installation Summary screen appears.

1. Review the information on this screen, and to change a previous selection, click Previous. After you change the required selection, click Next to go to the next screen.
2. Click Install to begin the installation process.

   The Microsoft Visual C++ 2010 x86 Redistributable Setup screen appears. This screen appears only if the current system where you are installing RiskMinder does not have Microsoft Visual C++ 2010 x86.
3. On the Microsoft Visual C++ 2010 x86 Redistributable Setup screen:
   a. Select the I have read and accept the license terms option, and click Install.

      The Installation Progress screen appears. This may take a few seconds. After some time the Installation Is Complete screen appears.
   b. Click Finish to close the Microsoft Visual C++ 2010 x86 Redistributable Setup dialog and continue with the RiskMinder installation.

   The Installing Arcot RiskFort screen appears. This may take several minutes. After some time the Install Complete screen appears.
4. Click Done to complete the installation.

Note: After the installation is completed, perform the post-installation tasks that are discussed in the following sections.

Installation Logs

After you complete the installation, you can access the installation log file (Arcot_RiskFort_Install_*<timestamp>*.log) in the *<install_location>* directory. For example, if you had specified the C:\Program Files directory as the installation directory, then the installation log file is created in the C:\Program Files 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

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

1. Running Database Scripts (see page 113)
2. Verifying the Database Setup (see page 113)
3. Preparing Your Application Server (see page 114)
4. Deploying Administration Console (see page 121)
5. Logging In to Administration Console (see page 123)
6. Bootstrapping the System (see page 124)
7. Starting RiskMinder Server (see page 126)
8. Starting the Case Management Queuing Server (see page 127)
9. Verifying the Installation (see page 127)
10. Deploying User Data Service (UDS) (see page 128)

Note: After you complete these post-installation tasks, perform the SDK and web services configuration tasks that are discussed in Configuring RiskMinder SDKs and Web Services (see page 139).

Running Database Scripts

Important! Before you run the scripts that are discussed in this section, ensure that you are logged in as the same database user that you created in the section, Configuring Database Server. (see page 51)

RiskMinder is shipped with scripts that are required to create necessary tables in the RiskMinder database. To run the required database scripts:

1. Navigate to the following directory:
   `<install_location>\Arcot Systems\dbscripts\`
2. Based on the database that you are using, navigate to one of the following subdirectories:
   - For Oracle: oracle\
   - For Microsoft SQL Server: mssql\
   - For MySQL: mysql\
3. Run the scripts *in the following order*:
   a. arcot-db-config-for common-2.0.sql

      Important! If you have installed CA AuthMinder 7.1.01, you need not run arcot-db-config-for-common-2.0.sql because you have already run it while installing CA AuthMinder 7.1.01.

   b. arcot-db-config-for-riskfort-3.1.01.sql

   c. (**Optional**, only if you want to create the 3D Secure Channel) arcot-db-config-for-3dsecure-3.1.01.sql.

Verifying the Database Setup

After you run the required database scripts, verify that the RiskMinder schemas were seeded correctly. To do so:

1. Log in to the RiskMinder database as the user who installed the database.

   Note: If you are following the upgrade path, then log in to the database as the user who upgraded the database.

2. Run the following query:
   ```
   SELECT SERVERNAME, VERSION FROM ARRFSERVERS;
   ```
   You must see the following output as a result of the preceding query:
   ```
   SERVERNAME                    VERSION
   ------------------------  ----------------
   RiskFort                       3.1.01
   RiskFortCaseManagement         3.1.01
   ```
3. Log out of the database console.

Preparing Your Application Server

Two components of RiskMinder, 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 Application Server

Before you deploy the WAR files for these web applications on the application server of your choice, copy the files that UDS and Administration Console require 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 and to deploy the WAR files of these web applications:

- Step 1: Setting Java Home (see page 114)
- Step 2: Copying Database Access Files to Your Application Server (see page 115)
- Step 3: Copying JDBC JAR Files to Your Application Server (see page 118)
- Step 4: (Mandatory for Oracle WebLogic 10.1) Creating Enterprise Archive Files (see page 120)

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. This JAVA_HOME must be your application server JAVA_HOME.

In addition, %JAVA_HOME%**bin**\ must be added to the PATH variable. If you fail to do so, then Administration Console, UDS, 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 RiskMinder database securely:

- arcot-crypto-util.jar available at:
  `<install_location>\Arcot Systems\java\lib\`
- ArcotAccessKeyProvider.dll available at:
  `<install_location>\Arcot Systems\native\win\<32bit-or-64bit>\`

As a result, these files must be copied to the appropriate location on the application server where you have deployed these RiskMinder 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 files:

1. Copy arcot-crypto-util.jar *to* *<Tomcat_JAVA_HOME>*\jre\lib**ext**\.

   Here, *<Tomcat_JAVA_HOME>* represents the JAVA_HOME used by your Apache Tomcat instance.
2. Copy ArcotAccessKeyProvider.dll *to* *<Tomcat_JAVA_HOME>*\jre**bin**\.
3. Restart the application server.

IBM WebSphere

To copy the files:

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 or 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, C:\Program Files\Arcot Systems\java\lib**arcot-crypto-util.jar**.
   e. Enter the JNI Library path.

This path must point to the location where the ArcotAccessKeyProvider.dll file is present.

3. Click **Apply** to save the changes.
4. Configure the 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 must be performed.
   c. Click **Java and Process Management** and then click **Class Loader**.
   d. Click **New**.
   e. Select default **Classes loaded with parent class loader first** and click **OK.**
   f. Click the auto-generated **Class Loader ID**.
   g. On the class loader **Configuration** page, click **Shared Library References**.
   h. Click **Add**, select **ArcotJNI,** and then click **Apply**.
   i. Save the changes.
5. Copy ArcotAccessKeyProvider.dll to <*WebSphere_JAVA_HOME*>\jre**bin**\.

   Here, <*WebSphere_JAVA_HOME*> represents the JAVA_HOME used by your IBM WebSphere instance.
6. Restart WebSphere.

Oracle WebLogic

To copy the files:

1. Copy ArcotAccessKeyProvider.dll to <*WebLogic_JAVA_HOME*>\jre**bin**\.

   Here, <*WebLogic_JAVA_HOME*> represents the JAVA_HOME used by your Oracle WebLogic instance.
2. Copy arcot-crypto-util.jar to <*WebLogic_JAVA_HOME*>\jre\lib**ext**\.

   Note: Ensure that you use the appropriate <*JAVA_HOME*> used by WebLogic.
3. Log in to WebLogic Administration Console.
4. Navigate to **Deployments**.
5. Enable the **Lock and Edit** option.
6. Click **Install** and navigate to the directory that contains the arcot-crypto-util.jar file.
7. Click **Next** to open the Application Installation Assistant.
8. Click **Next** to display the Summary page.
9. Click **Finish**.
10. Activate the changes.

11. Restart the server.

JBoss Application Server

To copy the files:

1. Copy ArcotAccessKeyProvider.dll *to* <*JBoss_JAVA_HOME*>\jre**bin**\.

   Here, <*JBoss_JAVA_HOME*> represents the JAVA_HOME used by your JBoss Application Server instance.

2. Copy arcot-crypto-util.jar *to* <*JBoss_JAVA_HOME*>\jre\lib**ext**\.

3. Restart the application server.

Step 3: Copying JDBC JAR Files to Your Application Server

RiskMinder requires the following JDBC JAR files for the supported databases:

- **Oracle 10g:** Oracle JDBC Driver (10.2.0.1.0)
- **Oracle 11g:** Oracle JDBC Driver (11.2.0.2.0)
- **Microsoft SQL Server:** MSSQL JDBC Driver (1.2.2828)
- **MySQL:** MySQL JDBC Driver (5.1.22)

The following subsections 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 required JDBC JAR file:

1. Navigate to the location where you have downloaded the *<Database_JAR>* file.
2. Copy the *<Database_JAR>* file to the following directory:
   - **On Apache Tomcat 5.5.x:** *<TOMCAT_HOME>*\common\lib\
   - **On Apache Tomcat 6.x and 7.x:** *<TOMCAT_HOME>*\lib\
3. Restart the application server.

IBM WebSphere

To copy the required JDBC JAR file:

1. Log in to the WebSphere Administration Console.
2. Click Environment, and then click Shared Libraries.
   a. From the Scope list, select a valid visibility scope. The scope must include the target server or node on which the application is deployed.
   b. Click New.
   c. Enter the Name, say, JDBCJAR.
   d. Specify the Classpath.

      Important! This path must point to the location where the <Database_JAR> file is present and must include the file name.

   e. Click Apply to save the changes that were made.
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 is performed.

c. Click Java and Process Management, and then click Class Loader.

d. Click New.

e. Select default Classes loaded with parent class loader first and click OK.

f. Click the auto-generated Class Loader ID.

g. In the class loader Configuration page, click Shared Library References.

h. Click Add, select JDBCJAR, and then click Apply.

i. Save the changes that were made.

4. Restart the application server.

Oracle WebLogic

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

To copy the required JDBC JAR file in case of Microsoft SQL Server:

1. Copy the <*Database_JAR*> file to <*Weblogic_JAVA_HO*ME>\lib\ext\.

   Here, <WebLogic_JAVA_HOME> represents the JAVA_HOME used by your Oracle WebLogic instance.

2. Log in to the 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 required <Database_JAR> file.
6. Click Next to display the Application Installation Assistant page.
7. Click Next to display the Summary page.
8. Click Finish.
9. Activate the changes.
10. Restart the application server.

JBoss Application Server

To copy the required JDBC JAR file:

1. Copy the JDBC JAR file to the following location on the JBOSS installation directory: <*JBOSS_HOME*>\server\default\lib\
2. Restart the application server.

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

Most enterprise Application Servers (such as WebSphere and WebLogic) enable you to bundle the related Java ARchive (JAR) or Web ARchive (WAR) files from one vendor (say, CA) to 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, WAR files are provided to deploy UDS and Administration Console. However if necessary, you can also change the format of these files to Enterprise ARchive (EAR) and then deploy the EAR files.

As discussed in the following subsections, you can either generate separate EAR files for both UDS and Administration Console, or you can generate a single EAR file that contains both Web archives.

Generating Separate EAR Files

To create a separate EAR file each for UDS and Administration Console, follow these steps:

1. Open the Command Prompt window.
2. Navigate to the <*install_location*>\Arcot Systems\tools\common**bundlemanager**\ directory.
3. To create the EAR file, run the following command:
   `java -jar bundle-manager.jar -ear <filename.ear> -warList <filename.war>`

   The preceding command generates individual EAR files that are available at:
   `<install_location>\Arcot Systems\java\webapps\`

Generating a Single EAR File

To create a single EAR file that contains UDS and Administration Console Web archives:

1. Open the Command Prompt window.
2. Navigate to the <*install_location*>\Arcot Systems\tools\common**bundlemanager**\ directory.
3. To create the EAR file, run the following command:
   `java -jar bundle-manager.jar -ear <filename.ear> -warList arcotadmin.war arcotuds.war`

   The preceding command generates a single EAR file that is available at:
   `<install_location>\Arcot Systems\java\webapps\`

Deploying Administration Console

Note: If you are deploying the Administration Console on IBM WebSphere 7.0, see the instructions in Deploying Administration Console on IBM WebSphere 7.0 (see page 273) instead of the following instructions.

Administration Console is a browser-based interface to RiskMinder that enables you to customize the server configurations and manage the deployed system.

You need the **arcotadmin.war** file to deploy the RiskMinder Administration Console. All Administration Console information is logged in the arcotadmin.log file. After you deploy arcotadmin.war, you can verify if it was correctly deployed by using this log file (arcotadmin.log). This log file is in the %ARCOT_HOME%\Arcot Systems\logs directory.

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

To deploy the Administration Console WAR file on your application server, and to verify if it was successfully deployed, follow these steps:

1. Deploy arcotadmin.war in the appropriate directory on the application server.

   Note: The deployment procedure depends on the application server that you are using. See 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**\.

2. **(For 32-bit WebSphere Only)** Configure reload of the Admin class when the application files are updated.

   a. Navigate to Application, Enterprise Applications, and then access the Admin 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.

   e. Restart the Admin application.

3. **(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 Systems\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>
Add the following category for cryptographic operations:
<category name="com.arcot.crypto.impl.NCipherCrypter">
<priority value="FATAL" />
<appender-ref ref="arcotadminlog"></appender-ref>
</category>
```

f. Save and close the file.

g. Take a backup of the existing JBoss logging libraries. These library files are available at:
<*JBOSS_HOME*>\lib\

h. 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/*

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

4. Restart the application server.
5. Verify that the console was successfully deployed:
   a. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`
   b. Open the arcotadmin.log file in any editor and locate the following lines:
      - 2.0.3
      - Arcot Administration Console Configured Successfully.

      These lines indicate that your Administration Console was deployed successfully.
   c. Also ensure that the log files *do not* contain any FATAL *and* WARNING messages.
   d. Close the file.

Logging In to Administration Console

When you log in to Administration Console for the first time, use the Master Administrator (MA) credentials that are configured automatically in the database during the deployment.

To log in to Administration Console as MA:

1. Launch the Administration Console in a Web browser window. The default URL for Administration Console is:
   `http://<host>:<appserver_port>/arcotadmin/masteradminlogin.htm`

   Note: The *host* and *port* information that you specify in the preceding URL must be of the application server where you deployed Administration Console.
   For example, in case of Apache Tomcat, the default *host* is localhost and *port* is 8080.
2. Log in by using the default Master Administrator account credentials. The credentials are:
   - **User Name: masteradmin**
   - **Password: master1234!**

Bootstrapping the System

Before you start using Administration Console to manage RiskMinder, perform the following mandatory steps to initialize the system:

- Change the default Master Administrator password
- Configure the Global Key label
- Specify the configuration settings for the out-of-the-box organization

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

Before you proceed with Performing Bootstrapping Tasks (see page 125), you must understand the related concept of Default Organization.

Default Organization

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

Performing Bootstrapping Tasks

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

To bootstrap the system by using the wizard:

1. Click Begin to start the process.

   The Change Password screen appears.

2. Specify the Current Password, New Password, Confirm Password, and click Next.

   The Configure Global Key Label screen appears.

3. On the Configure a Global Key Label page:

   - Specify the Global Key Label.

     RiskMinder enables you to use hardware- or software-based encryption of your sensitive data. (You can enable hardware-based encryption by using the arcotcommon.ini (see page 198) file, while software-based encryption is enabled by default.) Irrespective of hardware or software encryption, *Global Key Label* is used 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 *must* match the HSM key label. However in case of software-based encryption, this label acts as the key.

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

   - Specify the Storage Type.

     Use this option to indicate whether the encryption key is stored in the database (Software) or the HSM (Hardware).

4. Click Next to continue.

   The Configure Default Organization screen appears.

5. 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 who belong to the Default Organization. Administration Console supports three types of authentication methods for the administrators to log in:
     - LDAP User Password

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

Note: If this mechanism is used for authenticating administrators, then deploy UDS as discussed in Deploying User Data Service (UDS) (see page 128).

- Basic

  If you select this option, then the built-in authentication method that is provided by Administration Console is used for authenticating the administrators.

- WebFort Password

  If you select the WebFort Password option here, then the credentials are issued and authenticated by the AuthMinder Server. To use this option, CA AuthMinder must be installed.

Note: For information about installing and configuring AuthMinder, see the *CA AuthMinder Installation and Deployment Guide*.

6. Under the Key Label Configuration section of the Configure Default Organization screen, specify the following options:
   - **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 then specify a new label for encryption.
   - **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).
7. Click Finish to complete the bootstrapping process.

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

Starting RiskMinder Server

To start RiskMinder Server:

1. Click the **Start** button on your desktop window.
2. Navigate to **Settings, Control Panel, Administrative Tools,** and then **Services.**
3. Locate and double-click **Arcot RiskFort Service**.
4. Click **Start** in the service window.

Note: If you want to stop RiskMinder Server, then follow the Steps 1 through 3, and click the **Stop** button in the service window.

Starting the Case Management Queuing Server

To start Case Management Queuing Server:

1. Click the **Start** button on your desktop window.
2. Navigate to **Settings, Control Panel, Administrative Tools,** and then **Services.**
3. Locate and double-click the **Arcot RiskFort Case Management Queuing Service**.
4. Click **Start** in the service window.

Note: If you want to stop the Case Management Queuing Server, then follow the Steps 1 through 3, and click the **Stop** button in the service window.

Verifying the Installation

To verify if the server started correctly:

1. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`
2. Open the arcotriskfortstartup.log file in any editor and locate the following lines:
   - STARTING Arcot RiskFort 3.1.01_w
   - Arcot RiskFort Service READY
3. Open the arcotriskfortcasemgmtserverstartup.log file in any editor and locate the following lines:
   - STARTING Arcot RiskFort Case Management 3.1.01_w
   - Arcot RiskFort Case Management Service READY

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

Deploying User Data Service (UDS)

RiskMinder can access user data either from a relational database (RDBMS) or directly from an LDAP server by using UDS, which is an abstraction layer that provides RiskMinder seamless access to the third-party data repositories deployed by your organization.

You need the arcotuds.war file to deploy UDS, as follows:

1. Deploy arcotuds.war on the application server. This file is available at:

   <*install_location*>\Arcot Systems\java\webapps\

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

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

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 to save the changes.

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 Systems\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 Step of "Deploying Administration Console" (see page 121) section:
`<appender-ref ref="arcotudslog"></appender-ref>`

f. Add the following line in the cryptographic category that you created in Step of "Deploying Administration Console" (see page 121) section:
`<appender-ref ref="arcotudslog"></appender-ref>`

g. Save and close the file.

4. Restart the application server.

5. Verify if UDS was deployed successfully:

   Note: The arcotuds.log file is used for logging UDS-related information.

   a. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`

   b. Open the arcotuds.log file in any editor and locate the following line:

      - User Data Service (Version: 2.0.3) initialized successfully.

      This line indicates that UDS was deployed successfully.

   c. Also ensure that the log files *do not* contain any FATAL *and* WARNING messages.

   d. Close the file.

Installing on the Second System

After installing RiskMinder Server and Administration Console, you must now install the other remaining components on the second system in this distributed environment. The specific components to install must have been determined when you performed your planning in "Planning the Deployment" (see page 31).

Note: Before proceeding with the installation, ensure that all prerequisite software components are installed on this system as described in "Preparing for Installation" (see page 45).

To install RiskMinder components on the subsequent system:

1. Copy the installer file Arcot-RiskFort-3.1.01-Windows-Installer.exe on the target (second) system.
2. Double-click the installer to run it.
3. Follow the installer instructions from Step 2 in Installing on the First System (see page 102) until you reach the Choose Install Set screen.
4. Select the components you wish to install.

   Typically, you will be installing the Java SDKs for Risk Evaluation and Sample Application.
5. After you have selected all the components, follow the steps from Step 7 through Step 16 in Installing on the First System (see page 102) 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 and web services:

1. Deploying Sample Application (see page 131)
2. Configuring Sample Application for Communication with RiskMinder Server (see page 132)
3. Using Sample Application (see page 133)

Note: After you complete these configurations, configure RiskMinder SDKs (and web services) as discussed in Configuring RiskMinder SDKs and Web Services (see page 139).

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 RiskMinder was installed and configured properly. In addition, it demonstrates:

- The typical RiskMinder workflows
- The basic operations (invocation and post-processing) of RiskMinder APIs
- Integration of your application with RiskMinder

Note: If you did not install Sample Application during the installation, then you can install *only* Sample Application by running the installer again and by selecting the SDKs and Sample Application options and proceed with the installation.

To deploy Sample Application on your application server:

1. Deploy the riskfort-3.1.01-sample-application.war file from the following location: `<install_location>\Arcot Systems\samples\java\`
2. If necessary, restart the application server.
3. Access Sample Application in a Web browser window. The default URL for Sample Application is: *http://<host>:<appserver_port>/riskfort-3.1.01-sample-application/index.jsp*

Configuring Sample Application for Communication with RiskFort Server

The riskfort.risk-evaluation.properties file provides the parameters for the Java SDK and Sample Application to read RiskMinder Server information. Therefore, after deploying Sample Application, you must now configure it to communicate with RiskMinder Server. This file is only available *after* you deploy the RiskFort Sample Application WAR file, riskfort-3.1.01-sample-application.war.

To configure the riskfort.risk-evaluation.properties file:

1. Navigate to the riskfort.risk-evaluation.properties file on your application server.

   In Apache Tomcat, this file is available at:
   `<App_Home\riskfort-3.1.01-sample-application>\WEB-INF\classes\properties\`

   Here, <App_Home\riskfort-3.1.01-sample-application\> represents the directory path where RiskMinder application WAR files are deployed.

2. Open the riskfort.risk-evaluation.properties file in an editor window and set the value for the following parameters:

   - HOST.1
   - PORT.1

   A default value is specified for the remaining parameters in the file. You can change these values, if necessary. See "riskfort.risk-evaluation.properties" (see page 212) for more information about configuration parameters.

3. **Optional:** Perform this step only if you configured SSL-based communication in "Configuring SSL".

   Set the following parameters:

   - TRANSPORT_TYPE=SSL (By default, this parameter is set to TCP.)
   - CA_CERT_FILE=<*absolute_path_of_Root_Certificate_in_PEM_FORMAT*>

     For example, you can specify one of the following values:

     - CA_CERT_FILE=<*install_location*>/certs/<ca_cert>.pem
     - CA_CERT_FILE=<install_location>\\certs\\<ca_cert>.pem

   Important! In the absolute path that you specify, ensure that you use \\ or / instead of \. This is because the change may not work if you use the conventional \ that is used in Microsoft Windows for specifying paths.

4. Save the changes and close the file.

5. To ensure that these changes are reflected, restart the application server.

Using Sample Application

This subsection describes the risk-evaluation operations that can be performed by using Sample Application. Each operation in Sample Application is designed to run without error when RiskMinder is installed and functional.

Sample Application demonstrates the following operations that RiskMinder Server can perform:

- Performing Risk Evaluation and Post Evaluation for a First-Time User (see page 134)
- Creating Users (see page 135)
- Performing Risk Evaluation and Post Evaluation for a Known User (see page 136)
- Editing the Default Profile and Performing Risk Evaluation (see page 137)

Performing Risk Evaluation and Post Evaluation for a First-Time User

To perform risk evaluation on the default profile of a user:

1. Ensure that Sample Application is open (in a Web browser window.) The default URL for Sample Application is: *http://<host>:<appserver_port>/riskfort-3.1.01-sample-application/index.jsp*
2. Click Evaluate Risk to open the Risk Evaluation page.
3. On the page, specify the name of the user (who you want to evaluate) in the User Name field.
4. Specify the name of the organization to which the user belongs in the User Organization field.
5. If necessary, specify the Channel from which the transaction originated.
6. Click Evaluate Risk to open the Risk Evaluation Results page.

   This page displays the Risk Score, the associated Risk Advice, and lists the rules that are configured for the specified organization. For a first-time user, the result is ALERT.
7. Click Next Step to open the Post Evaluation page and to perform postevaluation on the specified user profile.

   By using postevaluation, your application provides feedback to RiskMinder Server about the current user and the device they are using. RiskMinder updates the user and device attributes and the user-device association based on this feedback, and assesses the risk associated with the transactions for the user in future.
8. Select the result of secondary authentication by selecting the appropriate option from the Result of Secondary Authentication list.
9. Specify the name for the user name-device association in the Association Name field.
10. Click Post Evaluate to complete the post evaluation process and to display the result of the same in the Post Evaluation Results section.

Creating Users

To create a user:

1. Create a GA account:
    a. Log in to Administration Console as the MA.
    b. Ensure that the **Users and Administrators** tab is active.
    c. On the menu on the left side, click the **Create Administrator** link to display the Create Administrator page.
    d. Specify the details on the page and click **Next**.
    e. On the Create Administrator page, select **Global Administrator** from the **Role** list.
    f. Specify the **Password** and **Confirm Password**.
    g. Select the **All Organizations** option in the **Manages** section.
    h. Click **Create**.
    i. Click **Logout** in the top right-hand corner of the page to log out as the MA.
2. Log in to Administration Console as a Global Administrator (GA) or an Organization Administrator (OA). The URL for the purpose is:
http://<host>:<appserver_port>/arcotadmin/adminlogin.htm
3. Follow the instructions that are displayed to change your password.
4. If already not activated, activate the **Manage Users and Administrators** subtab under the **Users and Administrators** tab.
5. In the left pane, under **Manage Users and Administrators**, click **Create User** to open the Create User page.
6. 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 necessary, 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.
7. Return to the RiskFort Sample Application page.

Performing Risk Evaluation and Post Evaluation for a Known User

1. On the Main Page of Sample Application, click **Evaluate Risk** to open the Risk Evaluation page.
2. In the **User Name** field, specify the name of the user that you created in the section Creating Users (see page 135).
3. In the **User Organization** field, specify the organization to which the user belongs.
4. If necessary, specify the **Channel** from which the transaction originated.
5. Click **Evaluate Risk** to open the Risk Evaluation Results page.

   The Risk Advice typically is **INCREASEAUTH**.
6. Click **Store DeviceID** to store the specified type of Device ID information on the end user's device.
7. Click **Next Step** to perform Post Evaluation:
   - Select the **Result of Secondary Authentication** from the list.
   - Edit the **Association Name**, if necessary.
8. Click **Post Evaluate** to display the final advice.

   If you repeat Step 1 through Step 5, the **Risk Advice** changes to **ALLOW** on the Risk Evaluation Results page.

Editing the Default Profile and Performing Risk Evaluation

Using Sample Application, you can change the DeviceDNA, IP address, and the Device ID of the computer that you are using to simulate various scenarios. To edit the default profile of a user:

1. On the Main Page of Sample Application, click Evaluate Risk to open the Risk Evaluation page.
2. Specify the user name whose profile you want to edit in the User Name field.
3. In the User Organization field, specify the name of the organization to which the user belongs.
4. If necessary, specify the Channel from which the transaction originated.
5. Click Edit Inputs to open the Edit Risk-Evaluation Inputs page.
6. On the page, all fields are prepopulated. Change the values for one or more of the required fields:
   - My User Name
   - My Org
   - My Channel
   - Machine Finger Print of My Device
   - Short Form of Machine Finger Print of My Device
   - IP Address of My Machine
   - Device ID of My Machine
7. Click Evaluate Risk to open the Risk Evaluation Results page.
8. Click Next Step to open the Post Evaluation page and perform postevaluation on the specified user profile.
9. Select the result of secondary authentication by selecting the appropriate option from the Result of Secondary Authentication list.
10. Click Post Evaluate to complete postevaluation and display the result of the same.

Note: To ensure secure communication between the RiskMinder components, you can configure them to support SSL (Secure Sockets Layer) transport mode. For more information, see "Configuring SSL" in the *CA RiskMinder Administration Guide*.

Applying the Post-Installation Checklist

It is recommended that you fill the following checklist with the installation and setup information for RiskMinder. This information is useful when you perform various administrative tasks.

Your Information	Example Entry	Your Entry
ARCOT_HOME	C:\Program Files\Arcot Systems	
SYSTEM INFORMATION		
Host Name	my-bank	
User Name	administrator	
Password	password1234!	
Configured Components	RiskFort 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 RiskMinder SDKs and Web Services

This section describes the steps to configure the Application Programming Interfaces (APIs) and web services that are provided by RiskMinder.

The section covers the following topics:

RiskMinder APIs

RiskMinder is shipped with a set of Java APIs that are available at <*install_location*>\Arcot Systems\sdk\java\lib\arcot\. In this location, the core JAR that you must implement is the Risk Evaluation SDK, **arcot-riskfort-evaluaterisk.jar**. In addition, you also find the following JARs that this core JAR is dependent on:

- arcot_core.jar
- arcot-pool.jar
- arcot-riskfort-mfp.jar

Important! At this location, you also see the JAR for Issuance SDK, arcot-riskfort-issuance.jar. However, this API has been deprecated in this release and only has been included for backward compatibility.
Instead of this API, you must use the **User Management Web Service**. See the *CA RiskMinder Web Services Developer's Guide* for detailed information.

The arcot-riskfort-evaluaterisk.jar file comprises the com.arcot.riskfortAPI package, which provides the logic for risk assessment. Operations that this package enables include:

- Evaluate and assess risk
- Generate advice
- List user-device associations
- Delete associations

Configuring Java APIs

This section provides the procedure to configure the Java APIs so that they can be used with your application.

Important! Before proceeding with the configuration steps in this section, ensure that the JARs required for implementing the Java APIs are installed at <*install_location*>\Arcot Systems\sdk\java\lib\.

To configure RiskMinder Risk Evaluation APIs for using with a J2EE application:

Note: The following instructions are based on Apache Tomcat Server. The configuration process may vary depending on the application server you are using. See the application server documentation for detailed information about these instructions.

1. Copy the listed JAR files *from* the following location:
   `<install_location>\Arcot Systems\`

   Paste them in the appropriate location in your <*APP_SERVER_HOME*> directory. For example, on Apache Tomcat this location is <*Application_Home*>\WEB-INF\lib\.

   - /sdk/java/lib/arcot/arcot_core.jar
   - /sdk/java/lib/arcot/arcot-pool.jar
   - /sdk/java/lib/arcot/arcot-riskfort-evaluaterisk.jar
   - /sdk/java/lib/arcot/arcot-riskfort-mfp.jar
   - /sdk/java/lib/external/bcprov-jdk15-146.jar
   - /sdk/java/lib/external/commons-lang-2.0.jar
   - /sdk/java/lib/external/commons-pool-1.5.5.jar

   For example, on Apache Tomcat 5.5.*x*, you must copy these files to C:\Program Files\Apache Software Foundation\Tomcat 5.5.31\webapps\<*Your_Application*>\WEB-INF\lib\.

2. Configure the log4j.properties.risk-evaluation and riskfort.risk-evaluation.properties files as follows:

   - If the application *already has* a configured log4j.properties.risk-evaluation file, then merge it with the following log configuration files:
     `<install_location>\Arcot Systems\sdk\java\properties\log4j.properties.risk-evaluation`

     and
     `<install_location>\Arcot Systems\sdk\java\properties\riskfort.risk-evaluation.properties`
   - If the application *does not have* the log4j.properties file configured, then:

     a. Rename log4j.properties.risk-evaluation to log4j.properties.

b. Merge riskfort.risk-evaluation.properties with log4j.properties.

c. Copy the log4j.properties file to:
<*Application_Home*>\WEB-INF\classes\properties\

For example, on Apache Tomcat 5.5.*x*, you must copy log4j.properties to C:\Program Files\Apache Software Foundation\Tomcat 5.5.31\webapps\<Your_Application>\WEB-INF\classes\.

Note: To know more about APIs and their initialization, refer to the RiskFort Javadocs at <*install_location*>\Arcot Systems\docs\riskfort\Arcot-RiskFort-3.1.01-risk-evaluation-sdk-javadocs.zip.

Working with RiskMinder Web Services

Important! To use the RiskMinder web services, deploy the **arcotuds.war** file. See Deploying User Data Service (UDS) (see page 89) for more information.

RiskMinder provides web services for managing users, organizations, administration, and for performing risk assessments. The WSDLs for these web services are available at: <*install_location*>\Arcot Systems\wsdls\

Generating Client Code Using the WSDLs

Important! Before you proceed with the client code generation, ensure that the RiskMinder package was installed successfully and that the Server is up and running.

After the installation, generate the client stub in the language you want to code in by using the WSDL files that are shipped with RiskMinder. These WSDLs enable the web services client to communicate with RiskMinder Server.

To generate the client code:

1. Stop the application server.
2. Navigate to the following location:
<*install_location*>\Arcot Systems\wsdls\<*required_folder*>
3. Use the required WSDL file (listed in the following table) to generate the client code.

WSDL File	Description
admin/**ArcotRiskFortAdminWebService.wsdl**	Used for creating and managing rule configurations that are typically done by using Administration Console.
riskfort/**ArcotRiskFortEvaluateRiskService.wsdl**	Used for performing risk evaluation.

WSDL File	Description
uds/**ArcotUserRegistryMgmtSvc.wsdl**	Used for creating and managing organizations in your setup.
uds/**ArcotConfigRegistrySvc.wsdl**	Used for creating and managing user account types.
uds/**ArcotUserRegistrySvc.wsdl**	Used for creating and managing users and user accounts.

4. Restart the application server.
5. In a browser window, access the end-point URLs (listed in the following table) to verify if the client can access the Web Service.

Web Service	URL
ArcotRiskFortAdminWebService	*http://<rf_hostname>:<rf_port>/services/ArcotRiskFortAdminSvc* The default *port* here is **7777**.
ArcotRiskFortEvaluateRiskService	*http://<rf_hostname>:<rf_port>/services/RiskFortEvaluateRiskSvc* The default *port* here is **7778**.
ArcotUserRegistryMgmtSvc	*http://<app_server_hostname>:<appserver_port>/arcotuds/services/ArcotUserRegistrySvc*
ArcotConfigRegistrySvc	*http://<app_server_hostname>:<appserver_port>/arcotuds/services/ArcotConfigRegistrySvc*
ArcotUserRegistrySvc	*http://<app_server_hostname>:<appserver_port>/arcotuds/services/ArcotUserRegistryMgmtSvc*

Book: See the *CA RiskMinder Web Services Developer's Guide* for more information about generating the Java client.

Configuring Device ID and DeviceDNA

RiskMinder uses Device ID and DeviceDNA to register and identify the device that is used by a user during transactions. The Device ID is stored on the end user's device. The Device ID information is in encrypted format.

The following options are available for storing the Device ID on the end user's device. The plugin store is the most persistent storage option.

- Plugin store: The plugin store is a permanent store on the end user's device. A Device ID that is placed in the plugin store cannot be deleted by common end-user actions such as clearing browser cache and deleting browser cookies. The plugin store is supported from CA RiskMinder Client release 2.1 onward.
- Local storage that is provided in HTML5
- UserData store: This store is available only in Microsoft Internet Explorer
- Cookie store: Typically, on Microsoft Windows, the Device ID is stored in one of the following folders:
  - **Internet Explorer on Microsoft Windows 7 or 2008:**
    `C:\Documents and Settings\<user_profile>\Application Data\Microsoft\Windows\Cookies\`
  - **Internet Explorer on Microsoft Windows 2003 or XP:**
    `C:\Documents and Settings\<user_profile>\Cookies\`
  - **Mozilla Firefox:**
    `C:\Documents and Settings\<user_profile>\Application Data\Mozilla\Firefox\Profiles\<random_dirname>\cookies.sqlite`
  - **Safari:**
    `C:\Documents and Settings\<user_name>\Application Data\Apple Computer\Safari\cookies.plist`

Important! From CA RiskMinder Client version 2.0 onward, the Device ID is not stored as a Flash cookie. If you have existing Flash cookies from an earlier release, then these cookies are automatically migrated to one of the stores that is listed earlier in this section.

File You Will Need for Device ID and DeviceDNA Collection

When you perform a complete installation (see Performing Complete Installation (see page 66) for more information) or select to install RiskFort Evaluation SDK or Web Service in the Choose Install Set screen, the following file is automatically installed:
<install_location>\Arcot Systems\sdk**devicedna\riskminder-client.js**

This file provides the functions to get and set the Device ID and DeviceDNA.

- Enabling Device ID and DeviceDNA Collection (see page 144)
- Migrating Flash Cookies from Preceding Releases (see page 144)

Enabling Device ID and DeviceDNA Collection

To configure for a cookie to be set on the end-user computer, include riskminder-client.js in your application pages that get or set the cookies. To do so:

1. Copy the entire devicedna directory from *<install_location>*\Arcot Systems\sdk\ to the appropriate web application directory. Typically, the web application folder is at the following location:
   <APP_SERVER_HOME>\<Your_Application_Home>
2. Include the riskminder-client.js file in the required application pages. We assume that these files are located in a folder that is relative to the folder containing index.jsp.
   ```
   <script type="text/javascript"
   src="devicedna/riskminder-client.js"></script>
   ```

Migrating Flash Cookies from Preceding Releases

As mentioned earlier, Flash cookies are not supported any more for storing the Device ID. However, if you have existing Flash cookies from an earlier release, then these cookies are automatically migrated to one of the supported stores on the end-user device when you complete the tasks described in "Collecting Device ID and DeviceDNA" in one of the following guides:

- CA RiskMinder Java Developer's Guide
- CA RiskMinder Web Services Developer's Guide

Enabling SSL Communication

RiskMinder supports Secure Sockets Layer (SSL) communication between RiskMinder Server and its Java SDKs. For information about setting SSL as the transport mode between RiskMinder Server and its clients, see "Configuring SSL" in the *CA RiskMinder Administration Guide*.

Chapter 7: Upgrading RiskMinder

This section walks you through the steps for upgrading your previous versions of RiskMinder to version 3.1.01. It includes the following topics:

Upgrade Overview

You can upgrade to RiskMinder 3.1.01 from any of the following versions:

- 1.5.1, 1.5.1.*x*, 1.6, 1.6.0.*x*, 1.7, 1.7.0.*x* (collectively referred to as 1.*x* in this chapter)
- 2.0, 2.2, 2.2.5.3 through 2.2.5.11, 2.2.6, 2.2.7, and 2.2.8 (collectively referred to as 2.*x* in this chapter)
- 3.0, 3.0.1, or 3.1 (collectively referred to as 3.*x* in this chapter)

Important! If you have a version of RiskMinder that is not listed here, then you must apply the required patches to upgrade to one of these versions, and then proceed with the upgrade. See the corresponding *Release Notes* for the patch upgrade instructions.

If you are upgrading from version 1.*x*, then you must first upgrade to version 2.2.7, and then upgrade to version 3.1.01. However, if you are upgrading from version 2.*x* or 3.x, then you can directly upgrade to version 3.1.01.

Database Privileges Required for Upgrade

The following table lists the database privileges that you must have for performing the database procedures that are related to upgrading to RiskMinder 3.1.01:

Database Type	Upgrade Privileges	Run time Privileges
Oracle	CREATE TABLE	CREATE TABLE
	CREATE ANY INDEX	DML Privileges
	CREATE ANY SEQUENCE	
	CREATE TABLESPACE (*for Reports*)	
	ALTER TABLESPACE	

Database Type	Upgrade Privileges	Run time Privileges
	CREATE PROCEDURE	
	UNLIMITED TABLESPACE (*for Reports, optional*)	
	DROP TABLESPACE	
	ALTER ANY TABLE	
	DML Privileges (including CREATE SESSION privilege)	
MS SQL Server **Note:** UserID must also have the database role of ddladmin. If the database user is dbowner, then the database user already has ddladmin privileges.	CREATE TABLE	CREATE TABLE
	CREATE INDEX	DML Privileges
	CREATE PROCEDURE	
	EXECUTE PROCEDURE	
	REFERENCES	
	ALTER TABLE	
	DML Privileges	

Upgrading to Release 3.1.01

Upgrading from 1.x to 3.1.01 is a two-stage procedure. You first upgrade from 1.x to 2.2.7 and then upgrade from 2.2.7 to 3.1.01. In contrast, if you are upgrading from 2.x or 3.x, you directly upgrade to 3.1.01.

Perform the following steps to upgrade to RiskMinder 3.1.01:

1. Performing Pre-Upgrade Tasks (see page 150)
2. If you are upgrading from 1.x, then perform the steps that are described in Migrating the Database to Release 2.2.7 for Arcot Common Components (see page 155). Do not perform this step if you are upgrading from 2.x or 3.x.
3. If you are upgrading from 1.x, then perform the steps that are described in Migrating the Database to Release 2.2.7 for RiskMinder Components (see page 156). Do not perform this step if you are upgrading from 2.x or 3.x.
4. Preparing for the Upgrade to Release 3.1.01 (see page 157)
5. Migrating the Database to Release 3.1.01 for Arcot Common Components (see page 158)
6. Migrating the Database to Release 3.1.01 for RiskMinder Components (see page 161)
7. Uninstalling the Existing Release of RiskMinder (see page 162)
8. Reinstalling RiskMinder (see page 163)
9. If you encounter any warnings during the Server startup and if your transactions fail, then perform the steps described in (In Error Scenario Only) Reverting to Your Initial Setup (see page 165).
10. Performing Post-Upgrade Tasks (see page 166)
11. Replacing Deprecated Rules with New Rules (see page 166)
12. Reviewing Configuration Changes After Upgrade (see page 169)

Performing Pre-Upgrade Tasks

Important! Perform the upgrade procedure on the system where the Administration Console is installed.

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

- If you have earlier installed both CA AuthMinder and CA RiskMinder and you plan to upgrade both products, then ensure that you follow the guidelines that are given at various places in this document.
- Ensure that the account that you plan to use for the upgrade operation belongs to the Administrators group.
- If you are upgrading from RiskMinder 1.*x* to 3.1.01, migrate all your proposed configuration data to production. Only active data is migrated and available after upgrade.

  Note: If you are upgrading from RiskMinder 2.*x* or 3.x to 3.1.01, both proposed and active configuration data get migrated.
- From release 3.1 onward, a rule with a score of 0 no longer carries the ALLOW advice. Instead, a score of 0 implies SILENT, which means that the rule is executed but is not used for scoring. In addition, if the default score was 0 before the upgrade, then the default rule score is changed to 1 during the upgrade.

  Note: For information about changing the score of a rule, see the administration guide for your RiskMinder release.
- Custom add-on rule types that you created in release 2.x or earlier releases are not migrated during the upgrade. The feature to create a custom add-on rule type by importing an XML file has been deprecated. If there are custom add-on rule types in your RiskMinder deployment, then delete them before the upgrade.
- If the mnemonic of an existing rule is the same as the mnemonic of a rule that is newly introduced or modified by the upgrade, then the upgrade will fail. The same issue is encountered if the name of an existing rule is the same as the name of a new rule. To avoid this issue:

  1. Use the administration console to compare the mnemonics of your existing rules with the mnemonics of the rules that are newly introduced or modified by the upgrade.

     The following tables shows the rules that are newly introduced or modified by the upgrade:

Rule Name	Rule Mnemonic
Unknown DeviceID	UNKNOWNDEVICEID
Device MFP Not Match	MFPMISMATCH
User Not Associated with DeviceID	USERDEVICENOTASSOCIATED
Unknown User	UNKNOWNUSER

2. If the mnemonic of an existing rule matches the mnemonic of a new rule, delete the existing rule and then re-create it. While re-creating the rule, give it a different mnemonic. The system allows the rule name to be the same for two different rules, but it is recommended that you change the name of the existing rule to avoid confusion.

   Note: For information about deleting and creating rules, see the administration guide for your current RiskMinder release.

- In release 2.x, you can have a rule, ruleset, or miscellaneous rule configuration refer to another rule, ruleset, or miscellaneous rule configuration. This feature is not available in release 3.1. Perform the following steps for each rule, ruleset, or miscellaneous rule configuration that refers to another rule, ruleset, or miscellaneous rule configuration:

  a. Log in to the Administration Console as a GA or OA.

  b. If you have logged in as the GA and you want to perform this procedure for a system ruleset, click the Services and Server Configurations tab.

  c. If you have logged in as the GA or OA to perform this procedure for a single organization:

     Activate the Organizations tab.

     Click the Search Organization link under Manage Organizations.

     Click the Search button on the Search Organization page to display the list of organizations.

     Click the name of the organization.

     Click the RiskFort Configuration tab.

  d. Under the Rules Management section on the side-bar menu, click the link for the rule, ruleset, or miscellaneous rule configuration that refers to another rule, ruleset, or miscellaneous rule configuration.

  e. Select Use Own.

  f. Select Copy from an Existing Ruleset.

  g. From the Ruleset Name list, select the ruleset to which this rule, ruleset or miscellaneous rule configuration was referring.

  h. Click Save.

  i. Migrate the changes to production.

     Note: For detailed information about migrating the changes to production, see the *CA RiskMinder Administration Guide*.

- The upgrade process is supported only in the offline mode. Shut down the following gracefully:
  - RiskFort Server
  - Case Management Queuing Server
  - Any application servers where Administration Console and User Data Service are deployed

- If Administration Console is open, close it.
- 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 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.
- 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 plan to upgrade is available throughout the upgrade process.
- Ensure that the database on which you plan to upgrade is disabled for replication.
- Back up the database containing the RiskMinder schema.
- If you require multi-byte character or internationalization support in RiskMinder 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 (see page 51).
- Consider requirements such as rollback segment size, based on data volume, before running the upgrade tool.
- Ensure that you have the database privileges required to upgrade RiskMinder. For the complete list of privileges, see Database Privileges Required for Upgrade (see page 147).
- 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 RiskMinder is installed.
- 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 that is created by the existing RiskMinder installation. Typically, ARCOT_HOME is <install_location>\Arcot Systems\.

ARCOT_HOME_BACKUP refers to the backup directory into which you copy the contents of the existing the ARCOT_HOME directory. If you encounter any errors during upgrade, use the ARCOT_HOME_BACKUP directory to revert to the initial setup.

Migrating the Database to Release 2.2.7 for Arcot Common Components

Note: Perform the tasks in this section only if you are upgrading from release 1.x. If you are upgrading from release 2.x or 3.x, you can ignore this topic.

Important! If you installed CA AuthMinder with CA RiskMinder and you have completed the upgrade to CA AuthMinder release 7.1.01, then do not migrate the database for Arcot common components. This step has already been performed during the AuthMinder upgrade process.

Migrate the database to the release 2.2.7 state for Arcot common components.

Follow these steps:

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

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

   - arcot-common-upgrade-0.*x*-1.0.zip
   - arcot-riskfort-upgrade-1.*x*-2.2.7.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>`

   Here, *db_type* can be mssql or oracle.

5. Run the arcot-db-config-for-common-1.0.sql script.

   Note: In the case of 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.

6. Navigate to the following directory:
   `%ARCOT_HOME%\dbscripts\<db_type>\upgrade-scripts\`

   Here, *db_type* can be mssql or oracle.

7. Run the arcot-upgrade-for-common-1.0.sql script.

   Note: In the case of 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 your database to the following directory:
   `%ARCOT_HOME%\java\lib`

9. Back up the existing ArcotAccessKeyProvider.dll file if it is in <*JAVA_HOME used by APP_SERVER*>\jre\bin. Then, copy the %ARCOT_HOME%\native\win\<*32bit-or-64bit*>\ArcotAccessKeyProvider.dll file to <*JAVA_HOME used by APP_SERVER*>\jre**bin**.
10. Set the PATH variable to include the directory where ArcotAccessKeyProvider.dll is copied.
11. Copy the file %ARCOT_HOME%\java\lib\arcot-crypto-util.jar to <*JAVA_HOME used by APP_SERVER*>\jre\lib**ext**\.
12. Navigate to the %ARCOT_HOME%\tools\upgrade directory.
13. Run the upgrade-common.bat tool.
14. To ensure that the common database upgrade operation was run successfully, see the %ARCOT_HOME%\logs\upgrade-common.log file.

Migrating the Database to Release 2.2.7 for RiskMinder Components

Important! Perform the steps in this section only if you are upgrading from release 1.x. If you are upgrading from release 2.x or 3.x, you can ignore this procedure.

After you migrate the database for Arcot common components, migrate the database to the release 2.2.7 state for RiskMinder components.

Follow these steps:

1. Copy the arcot-riskfort-upgrade-1.*x*-2.2.7.zip file to the ARCOT_HOME directory.
2. Extract the contents of the arcot-riskfort-upgrade-1.*x*-2.2.7.zip file in this directory.
3. Navigate to the following directory:
   `%ARCOT_HOME%\dbscripts\<db_type>\upgrade-scripts`

   Here, *db_type* can be mssql or oracle.
4. Run the SQL script corresponding to your current release of RiskMinder, as listed in the following table.

Current RiskMinder Release	**SQL Script to Run**
1.5.1 or 1.5.1.*x*	arcot-riskfort-upgrade-1.5.1.8-2.2.7.sql
1.6 or 1.6.0.*x*	arcot-riskfort-upgrade-1.6.0.3-2.2.7.sql
1.7 or 1.7.0.*x*	arcot-riskfort-upgrade-1.7.0.3-2.2.7.sql

5. Navigate to the following directory:
   `%ARCOT_HOME%\dbscripts\<db_type>\upgrade-scripts\`

   Here, *db_type* can be mssql or oracle.

6. Run the arcot-post-upgrade-for-common-1.0.sql script.

   This script ensures the following configurations:

   - The user ID for Master Administrator is changed from MASTER_ADMIN to MASTERADMIN.
   - The password for the MASTERADMIN account is **master1234!**
   - The organization that MASTERADMIN belongs to is MASTERADMIN. This feature 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.

Preparing for the Upgrade to Release 3.1.01

This section describes the steps that you must perform to prepare your setup for upgrading to 3.1.01.

Follow these steps:

1. If application server connection pooling was being used in your existing RiskMinder deployment, navigate to the %ARCOT_HOME%\bin directory, and update the securestore.enc file by running the following command for the primary database:
   `DBUtil -pi <DB_username> <DB_password>`

   Note: To determine whether database connection pooling is being used, open the %ARCOT_HOME%\conf\arcotcommon.ini file. Check the value of the AppServerConnectionPoolName parameter.

2. If SSL has been configured for the connection with the database, navigate to the %ARCOT_HOME%\bin directory and set the TrustStore password using DBUtil, as follows:
   `DBUtil -pi TrustStorePath.1 <truststore-password>`

   Note: To determine whether SSL has been configured, check the value of the TrustStorePath parameter in the arcotcommon.ini file.

Migrating the Database to Release 3.1.01 for Arcot Common Components

Migrate the database to the release 3.1.01 state for Arcot common components.

Follow these steps:

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

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

   - arcot-common-upgrade-1.0.*x*-2.0.zip
   - arcot-riskfort-upgrade-2.x-3.x-3.1.01.zip

2. Copy the arcot-common-upgrade-1.0.*x*-2.0.zip file to the ARCOT_HOME directory.
3. Extract the contents of the arcot-common-upgrade-1.0.*x*-2.0.zip file in this directory.

   Note: Click **Yes** if you are prompted to overwrite any existing files.

4. Navigate to the following directory:
   `%ARCOT_HOME%\tools\common\upgrade\`
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 **exact** name, as follows:

   - ORACLE: ojdbc.jar
   - SQL Server: sqljdbc.jar

7. Locate the JAVA_HOME used by the existing installation and ensure that you use the same JAVA_HOME to run the upgrade tool.
8. Set the PATH variable to include the directory where ArcotAccessKeyProvider.dll is copied.

   Important! If you are upgrading from release 3.x to 3.1.01, do not perform the remaining steps of this procedure. Instead, directly proceed to the next section.

9. At the command prompt, change your working directory to:
   `%ARCOT_HOME%\tools\common\upgrade\`
10. 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	The following JVM options are required only if LDAP organizations are configured: ■ -Xmx<*heap_memory_size_in_MB*>M: Sets the maximum heap size to 1GB. If there are more than 1,00,000 users in the configured LDAP, then it is strongly recommended that you increase the heap size to 2048M (2GB). ■ -Dcom.arcot.ldap.migration.timeout=<*duration*>: The migration of an LDAP organization involves fetching all the users from the LDAP server and migrating the users to the RiskMinder database. This parameter sets the maximum time (in minutes) taken to fetch all users from the LDAP server, beyond 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, the timeout would depend on the type of hardware configuration being used. The default value of this parameter is 240 minutes. **Note:** Ensure that the java command executable belongs to JAVA_HOME identified in Step 7. If JAVA_HOME is not set, modify the PATH environment variable to include %*JAVA_HOME*%\bin.
log-file	Specifies the path to the log file: ■ 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%\logs\ 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.

Option	Description
product-name	Specifies the name of the product for which the upgrade is run. If you do not specify the product name, the product name is assumed to be common. Possible values are: ■ common: Indicates the Arcot common components. ■ riskfort: Indicates RiskMinder. **Note:** Upgrade the Arcot common components before you upgrade RiskMinder.
prompt	Prompts whether to proceed further after each phase of the upgrade process is completed successfully. The upgrade process happens in the following phases: ■ Pre-upgrade: Involves performing various DDL and DML operations to migrate the database schema. ■ Upgrade: Involves migrating the data to the new schema. ■ Post-upgrade: Involves cleanup or follow-up actions that are required to be performed after the upgrade. ■ Verification: Involves the verification of whether the upgrade is successful. This option You can choose to run the upgrade 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. If you are upgrading from release 1.0.x, then check for the following line in the %ARCOT_HOME%\logs\arcot_common_upgrade.log file:

   Upgrade for common from version 1.0.x to version 2.0 run successfully.

   The presence of this line in the log confirms that the database was upgraded successfully.

Migrating the Database to Release 3.1.01 for RiskMinder Components

After you migrate the database for Arcot common components, migrate the database to the release 3.1.01 state for RiskMinder components.

Follow these steps:

1. Extract the contents of the arcot-riskfort-upgrade-2.x-3.x-3.1.01.zip file in the ARCOT_HOME directory.
2. Navigate to the following directory:
   `%ARCOT_HOME%\tools\common\upgrade\`
3. Run the following command:
   `java -jar arcot-common-upgrade-framework.jar --product-name riskfort`

   See the table in Migrating the Database to Release 3.1.01 for Arcot Common Components (see page 158) for a description of the command options.
4. Depending on the release that you are upgrading from, locate one of the following lines in the arcot_common_upgrade.log file in the %ARCOT_HOME%\logs directory:
   `Upgrade for riskfort from version <your-RiskMinder-release> to version 3.1.01 run successfully.`

   For example, if you upgraded from release 3.0, then locate the following line:
   `Upgrade for riskfort from version 3.0 to version 3.1.01 run successfully.`

   The presence of this line in the log confirms that the database was upgraded successfully.

Uninstalling the Existing Release of RiskMinder

Uninstall the existing release of RiskMinder. Also uninstall the RiskMinder components 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 RiskMinder installation, follow the uninstallation instructions that are given in the installation guide for your existing release of RiskMinder.

Follow these steps:

1. Uninstall the existing release of RiskMinder as follows:
   a. Ensure that the following components have been shut down gracefully:
      - RiskFort Server
      - Case Management Queuing Server
      - Any application servers where other RiskFort components are deployed.
   b. Ensure that the Administration Console is not open.
   c. Ensure that all INI and other files that are related to the RiskMinder configuration are closed.
   d. On the desktop, click Start, Settings, Control Panel, Add/Remove Programs to open the Add or Remove Programs window.
   e. From the Currently installed programs list, select Arcot RiskFort, and click Change/Remove.

      The Uninstall Arcot RiskFort window appears.

      Note: You can also uninstall RiskMinder by running Uninstall Arcot RiskFort.exe available in the <*install_location*>\Arcot Systems\Uninstall Arcot RiskFort\ directory.
   f. Select Complete Uninstall.

      Note: You may have to wait for a few minutes for the uninstallation process to complete.

      After the software is uninstalled successfully, the Uninstallation Complete screen appears with a success message.
   g. Click Done to exit the wizard and complete the uninstallation process.
   h. If you have installed CA AuthMinder and CA RiskMinder and you are deleting both products, delete any files that are left over in the ARCOT_HOME directory.
2. Undeploy the Administration Console, User Data Service, and Sample Application Web applications from the application server. For detailed information, see the application server documentation.

Reinstalling RiskMinder

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

- Reinstalling RiskMinder on a Single System (scenario) (see page 310)
- Reinstalling RiskMinder on a Distributed System (scenario) (see page 333)

Reinstalling RiskMinder on a Single System

Perform the tasks that are described in the following sections to reinstall RiskMinder on a single system:

1. Install RiskMinder.

   Note: When you install release 3.1.01, ensure that you specify the same primary and backup database details from arcotcommon.ini in the %ARCOT_HOME%\conf\ directory.

2. Verify the database setup.
3. Prepare the application server.
4. Deploy Administration Console.
5. Log in to Administration Console.

   Important! Ensure that you use the current MA password and *not* the default password, because the MA password has been reset during the bootstrap process that you performed during 2.x installation.

6. Start the RiskMinder Server.
7. Start the Case Management Queuing Server.
8. Verify the installation.

   Note: If there are any warnings during the Server startup and if your transactions fail, then the upgrade has not been performed successfully. You can revert to your initial setup by following the steps that are listed in (In Error Scenario Only) Reverting to Your Initial Setup.

9. Deploy User Data Service.
10. Deploy Sample Application.
11. Use the Sample Application to test the migration by verifying whether the user accounts and the related data from the earlier setup have been successfully migrated to the new database.
12. Apply the post-installation checklist.

Reinstalling RiskMinder on a Distributed System

Perform the tasks that are described in the following sections to reinstall RiskMinder on a distributed system:

Important! The information in these sections applies to both a fresh installation of RiskMinder and an upgrade of an existing RiskMinder installation. Some of the steps that are mentioned in these sections do not apply in an upgrade scenario. For example, MySQL-related steps apply only for an upgrade from release 3.1 because MySQL is supported only from release 3.1 onward.

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

1. Installing on the First System

   Note: When you install release 3.1.01, ensure that you specify the same primary and backup database details from arcotcommon.ini in the %ARCOT_HOME%\conf\ directory.

2. Verifying the Database Setup
3. Preparing Your Application Server
4. Deploying Administration Console
5. Logging In to Administration Console

   Important! Ensure that you use the current MA password and *not* the default password, because the MA password has been reset during the bootstrap process that you performed during 2.x installation.

6. Starting RiskMinder Server
7. Starting the Case Management Queuing Server
8. Verifying the Installation

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

9. Deploying User Data Service
10. Installing on the Second System
11. Deploying Sample Application
12. Configuring Sample Application for Communication with RiskMinder Server (scenario).
13. Using Sample Application (scenario) (distributed) to test the upgrade by verifying whether the user accounts and the related data from the earlier setup have been successfully migrated to the new database.

14. Post-Installation Checklist.

(In Error Scenario Only) Reverting to Your Initial Setup

During upgrade, if there are any warnings during the RiskMinder Server startup and if your transactions fail, then you may want to revert to your initial setup.

Follow these steps:

1. Uninstall RiskMinder 3.1.01.

   Refer to "Uninstalling RiskMinder" (see page 179) for more information.

2. Install the RiskMinder version that you want to revert to. For example, 1.*x* or 2.*x*.

   Note: For installation instructions, see the *CA RiskMinder Installation and Deployment Guide* that is shipped with the corresponding release.

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. Replace the ArcotAccessKeyProvider.dll file in *<JAVA_HOME used by Application Server>*\jre\bin with the backup that you created while performing the procedure described in Migrating the Database to Release 2.2.7 for Arcot Common Components (see page 155).
6. Deploy the web components, such as the Administration Console and UDS.
7. Restore the database from the backup that you had taken before you began the upgrade procedure.
8. Start RiskMinder Server and Case Management Queuing Server.
9. Test the installation.

Performing Post-Upgrade Tasks

This section describes the tasks that you must perform after upgrading to release 3.1.01.

Follow these steps:

1. If you disabled database replication before upgrade, then after you upgrade to RiskMinder 3.1.01 enable replication for the backup database.
2. If you configured SSL for the following ports in RiskMinder 2.2.7, then reconfigure SSL.
   - Port 7980 for Server Management protocol of the RiskMinder Server instance
   - Port 7780 for Case Management Queuing Administration protocol of the Case Management Queuing Server instance

   Reconfigure SSL as follows:
   - Between Administration Console and RiskMinder Server: Port 7980
   - Between Administration Console and Case Management Queuing Server: Port 7780

   This configuration is required because most administrative tasks, such as instance management and protocol configuration, are done using these ports in Administration Console in release 3.1.01.

   Note: For instructions on setting up SSL between Administration Console and RiskMinder Server or Case Management Queuing Server, see "Configuring SSL" in the *CA RiskMinder Administration Guide*.
3. Set the Base Currency Code for your organization from the Miscellaneous Configurations screen.

   Note: For more information about setting the organization-specific base currency code, see Managing Global Configurations" in the *CA RiskMinder Administration Guide*.
4. If there are any rules with a score of 0 and you want to use these rules for scoring, then change the score to a nonzero value, like 1 or 2.

Replacing Deprecated Rules with New Rules

Four of the predefined rules have been deprecated in release 3.1. Alternative rules have been introduced for these deprecated rules. The following table lists the deprecated and new rules and rule mnemonics:

Deprecated Rule Name and Rule Mnemonic	**New Rule Name and Rule Mnemonic**
DeviceID Known (DEVICEIDCHECK)	Unknown DeviceID (UNKNOWNDEVICEID)

Deprecated Rule Name and Rule Mnemonic	New Rule Name and Rule Mnemonic
Device MFP Match (SIGMATCH)	Device MFP Not Match (MFPMISMATCH)
User Associated with DeviceID (USERDEVICEASSOCIATED)	User Not Associated with DeviceID (USERDEVICENOTASSOCIATED)
User Known (USERKNOWN)	Unknown User (UNKNOWNUSER)

Important! Although these rules have been deprecated, they are still available and can be used after the upgrade. However, it is recommended that you replace each deprecated rule with the corresponding new rule by making the required changes in the rule expression.

For any of the four deprecated rules, if the rule evaluates to No, then the rule is considered to have matched and it is used for scoring. In contrast, each of the other predefined rules is considered to have matched when they evaluate to Yes.

In each of the four new rules that is introduced in release 3.1, if the rule evaluates to Yes, then the rule is considered to have matched. In this way, the four new rules are consistent with the other predefined rules.

The following table lists examples that highlight the difference between the deprecated rules and new rules:

Sample Use Case	Deprecated Rule	Deprecated Rule Result	New Rule	New Rule Result
User does not exist in the RiskMinder database.	USERKNOWN	No	UNKNOWNUSER	Yes
DeviceID does not exist in the RiskMinder database.	DEVICEIDCHECK	No	UNKNOWNDEVICEID	Yes
MFP does not exist in the RiskMinder database.	SIGMATCH	No	MFPMISMATCH	Yes
User is not associated with the DeviceID.	USERDEVICEASSOCIATED	No	USERDEVICENOTASSOCIATED	Yes

Follow these steps:

1. Log in to the administration console.
2. In the Rule Configurations Report for all organizations and rulesets, verify whether any of the mnemonics that are listed in the Rule expression column of the report belong to the list of deprecated mnemonics.
3. If a rule uses a deprecated mnemonic and if you do not want to use the deprecated mnemonic, use the corresponding new mnemonic.

   To modify a rule expression:

   a. Log in to the administration console as the GA or OA.

   b. If you have logged in as the GA and you want to perform this procedure for a system ruleset, click the Services and Server Configurations tab.

   c. If you have logged in as the GA or OA to perform this procedure for a single organization:

      Activate the Organizations tab.

      Click the Search Organization link under Manage Organizations.

      Click the Search button on the Search Organization page to display the list of organizations.

      Click the name of the organization.

      Click the RiskFort Configuration tab.

   d. Under the Rules Management section on the side-bar menu, click the Rules and Scoring Management link.

      The Rules and Scoring Management page appears.

   e. From the Select a Ruleset list, select the ruleset for which this configuration is applicable.

      The configuration information for the specified ruleset appears.

   f. Click the rule that you want to modify.

      The Rule Builder page opens.

   g. Make the required changes in the Rule being developed text field.

   h. Save the changes and close the Rule Builder page.

4. Migrate the modified rule to the production environment, and then refresh the cache.

   Note: For detailed information about migrating a rule to the production environment and refreshing the cache, see the *CA RiskMinder Administration Guide*.

Reviewing Configuration Changes After Upgrade

This section lists the changes that you can expect to see after you upgrade to release 3.1.01.

Silent Execution of Rules

A rule whose score is 0 is considered to be a silent rule. Such a rule is not used for scoring. This feature allows you to observe how a rule would execute during transactions, without the risk of unintended effects on end users. In earlier releases, a rule whose score is 0 always generates the ALLOW advice.

Deleted Rules Listed in the Active Set

When you delete a rule, it continues to be displayed in the active set. In addition, a message stating the rule is deleted is displayed in the proposed set.

Deleting Rulesets

You can delete rulesets that are not currently assigned to any organization.

Custom Actions

You can add custom actions and then use these actions to build rules. For more information, see Adding Custom Actions (see page 274).

Instance and Protocol Configuration

Logging parameters, such as log directory, log file size, log backup directory, log level, and log timestamps, which existed in the riskfortserver.ini file can now be edited from the Instance Management page in Administration Console.

Server parameters, such as maximum threads and minimum threads, which existed in the riskfortserver.ini file can now be edited from the Protocol Configuration page in Administration Console.

Model Configuration

Model Configuration is performed at the global level and is no longer specific to rulesets. Only the Master Administrator can edit the model configuration parameters. The Global Administrator can enable or disable the model at the global level and at the organization level.

User Creation Mode

The User Creation Mode configuration that was available at the ruleset level in previous releases is now available as User Enrollment Mode at the organization level.

Machine FingerPrint (MFP) Threshold

MFP Threshold configuration parameter in previous releases is now part of the Device MFP Not Match rule.

Reverse Lookup Configuration

After upgrade, reverse lookup configuration for Device MFP and IP address are available at the channel level. You can configure these parameters, **Enable Reverse Lookup for Device Identification** and **Use IP Address for Reverse Lookup,** on the Miscellaneous Configurations page.

Annotation in Risk Evaluation API Response

The Risk Evaluation API response contains all rule results in a field called *annotation*. In RiskMinder 1.7.0.3, the annotation field contained the rule results, USERDIDMATCH=Y or USERDIDMATCH=N, though USERDIDMATCH was not a rule available on Administration Console. This issue was resolved in RiskMinder 3.0 and now the annotation field contains only the results of rules that are configured through Administration Console. If your calling application used this annotation in RiskMinder 1.7.0.3 and you require this feature after upgrade, you can use the USERDEVICEIDASSOCIATED rule, which is equivalent to the USERDIDMATCH rule.

Rules Based on User Device Association and DeviceID-MFP Match

RiskMinder 1.*x* had Base Combination rules. From RiskMinder 1.7, you could configure these rules in Administration Console. These rules were based on User-DeviceID match and Machine FingerPrint match rules and were separate from Standalone rules in Administration Console. After upgrading to RiskMinder 3.1, you can use the **User Associated with DeviceID** (USERDEVICEASSOCIATED) and **Device MFP Match** (SIGMATCH) rules to re-create these combination rules with appropriate rule mnemonics.

The default score for the combination rules in RiskMinder versions 1.5.1.6 and earlier was as follows:

- USERDEVICEASSOCIATED AND SIGMATCH: 10
- NOT (USERDEVICEASSOCIATED) AND SIGMATCH: 65
- USERDEVICEASSOCIATED AND NOT (SIGMATCH): 65
- NOT (USERDEVICEASSOCIATED) AND NOT (SIGMATCH): 85

The default score for the combination rules in RiskMinder versions 1.5.1.7 and later until 2.0 was as follows:

- USERDEVICEASSOCIATED AND SIGMATCH: 10
- NOT (USERDEVICEASSOCIATED) AND SIGMATCH: 65
- USERDEVICEASSOCIATED AND NOT (SIGMATCH): 65
- NOT (USERDEVICEASSOCIATED) AND NOT (SIGMATCH): 65

Ruleset Configuration

The following ruleset configurations are not available after the upgrade:

- Creating a ruleset by referring to another ruleset
- Editing rule configurations to refer to another ruleset
- Edit a rule to copy from another ruleset

Amount Check Rule

If you had an Amount Check rule for an organization associated with a channel that does not have the AMOUNT element, then manually re-create this rule after the upgrade. If your rule needs to set different thresholds for different currencies, then you must add AMOUNT as a channel element. If you expect transactions that are based on only a single currency, then you can create a simple numeric comparison rule using the CUSTOM element in Rule Builder.

Note: The DEFAULT channel does not have the AMOUNT element. Note the configuration of the Amount Check rule before upgrade and re-create after upgrade, if required.

New and Deprecated Rules

Four of the predefined rules have been deprecated in this release. Alternative rules have been introduced for these deprecated rules. The following table lists the deprecated and new rules and rule mnemonics:

Deprecated Rule Name and Rule Mnemonic	**New Rule Name and Rule Mnemonic**
DeviceID Known (DEVICEIDCHECK)	Unknown DeviceID (UNKNOWNDEVICEID)
Device MFP Match (SIGMATCH)	Device MFP Not Match (MFPMISMATCH)
User Associated with DeviceID (USERDEVICEASSOCIATED)	User Not Associated with DeviceID (USERDEVICENOTASSOCIATED)
User Known (USERKNOWN)	Unknown User (UNKNOWNUSER)

Important! Although these rules have been deprecated, they are still available and can be used after you upgrade to release 3.1. However, it is recommended that you replace each deprecated rule with the corresponding new rule by making the required changes in the rule expression.

For any of the four deprecated rules, if the rule evaluates to No, then the rule is considered to have matched. It is then used for scoring. In contrast, each of the other predefined rules are considered to have matched when they evaluate to Yes.

In each of the four new rules introduced in release 3.1, if the rule evaluates to Yes, then the rule is considered to have matched. In this way, the four new rules are consistent with the other predefined rules.

Rule Migration

All rules are migrated to the DEFAULT channel for all Actions that are supported by default in the system.

Rule Execution Priority

After you upgrade from RiskMinder 2.*x*, you do not have to enable or disable the rules for execution. The execution priority is automatically determined by the system.

Secondary Authentication Result

Transactions in RiskMinder 2.2.5.11 that had the **Secondary Authentication Result** status Not Available now appear with the status Abandoned after the upgrade.

Untrusted IP Type Configuration After Upgrade

RiskMinder versions prior to 1.7 allowed you to configure Negative IP Types as Active, Suspect, or Private from the Administration Console. From RiskMinder 1.7, the Active, Suspect, Private, Inactive, and Unknown negative type categories are derived from the data that is provided by our Intelligence Partner. So, if you had configured any Negative IP Type as Active, Suspect, or Private in your RiskMinder 1.6.0.*x* or earlier deployments, then after upgrading to 3.1, these IP types are migrated to the "Negative" category of the Untrusted IP Type.

Cache Refresh

After you upgrade from RiskMinder version 2.*x* or later, you can refresh the cache of all server instances from Administration Console. If you choose to use the command-line option, refresh the server instances using the arrfclient tool instead of the arrfadmin tool.

Case Assignment After Upgrade

After you upgrade from RiskMinder version 2.*x* or later, all cases that were generated in the previous version continue to be assigned to the Default Queue for the organization.

Note: In RiskMinder 2.0, cases were assigned to each Customer Support Representative (CSR), but from RiskMinder 2.2 onwards, cases are not assigned to individual CSRs. For more information, see the *CA RiskMinder Administration Guide*.

All new cases that are generated after you upgrade to RiskMinder 3.1 are assigned to the relevant Queue according to the Queue criteria defined by the Queue Manager in RiskMinder 3.1.

Calling Application Code Changes

The following list describes the changes to the calling application code after upgrade:

- The older Java SDK client will continue to work with the new installation of RiskMinder Server. The client code will not require any modification if you continue to use the old SDK. However, the new SDK provides additional functionality and it is recommended that you integrate the calling application code with the latest version of SDK.
- Applications integrated using old Risk Evaluation WSDL will continue to work without modification in the code.

  Note: For RiskMinder 1.x versions, Web services were built as a WAR implementation. The client must continue to point to the old Web service even after upgrading to RiskMinder 3.1.
  In RiskMinder versions 2.0 and later including RiskMinder 3.1, Web services are implemented as part of RiskMinder Service and not as a WAR. It is recommended that you integrate your applications using the new WSDL and configure your application to RiskMinder Service according to the new architecture.
- In previous releases, the Issuance Java API provided a programmable interface, which could be used by Java clients (such as Java Servlets and JSP pages) to send Issuance-related requests to RiskMinder Server. In RiskMinder 3.0, the Issuance API (Issuance) has been deprecated. Now, you must use the User Management Web service (ArcotUserRegistrySvc) for this purpose.
- If you were using the Exception User Web service that was shipped earlier, use the new RiskMinder Administration Web service WSDL implemented in RiskMinder Service that provides the Exception User API.
- It is recommended that you use the enhanced RiskMinder 3.1 risk evaluation APIs that now return response codes and reason codes.

Role Privileges

After you upgrade from RiskMinder 2.*x*, review the privileges that are associated with the various roles. The following table lists the privileges that have been deleted after upgrade for the Master Administrator, Global Administrator, and Organization Administrator roles.

Role	Scope	Privileges Deleted

Role	Scope	Privileges Deleted
Master Administrator (MA)	Global	■ Update RiskMinder Protocols ■ Add Add-On Rule Type
Global Administrator (GA)	Global	■ Manage Negative Countries ■ Manage Negative IP Addresses ■ Manage User Velocity Configuration ■ Manage Device Velocity Configuration ■ Manage Trusted IPs/Aggregators ■ Manage IP Velocity Configuration ■ Manage Scoring Configuration ■ Manage Miscellaneous Rule Configurations ■ Manage Negative IP Types ■ Configure Add-on Rules ■ Show GDP URL ■ View Trusted IP Addressed/Aggregators Report ■ View Negative IP Address Report ■ View Negative Country Report ■ Manage Category Based Rule Data ■ View Mapping Data Report

Role	Scope	Privileges Deleted
	Organization	■ Manage Negative Countries ■ Manage Negative IP Addresses ■ Manage User Velocity Configuration ■ Manage Device Velocity Configuration ■ Manage Trusted IPs/Aggregators ■ Manage IP Velocity Configuration ■ Manage Scoring Configuration ■ Manage Miscellaneous Rule Configurations ■ Manage Negative IP Types ■ Configure Add-on Rules ■ Manage Category Based Rule Data
Organization Administrator (OA)	Global	■ Manage Queues ■ View Trusted IP Addressed/Aggregators Report ■ View Negative IP Address Report ■ View Negative Country Report ■ View Mapping Data Report
	Organization	■ Manage Negative Countries ■ Manage Negative IP Addresses ■ Manage User Velocity Configuration ■ Manage Device Velocity Configuration ■ Manage Trusted IPs/Aggregators ■ Manage IP Velocity Configuration ■ Manage Scoring Configuration ■ Manage Miscellaneous Rule Configurations ■ Manage Negative IP Types ■ Configure Add-on Rules ■ Manage Category Based Rule Data

The following table lists the privileges that have been added after upgrade for the Master Administrator, Global Administrator, Organization Administrator, and User Administrator roles.

Role	Target	Privileges Added
Master Administrator (MA)	Global	■ Instance Management ■ View Instance Management Report ■ Model Configuration
	API	■ Get User Profile (web service) ■ Add User to Exception List (web service) ■ Delete User from Exception List (web service) ■ Get Location and Connection Info (web service)
Global Administrator (GA)	Global	■ Manage Miscellaneous Configurations ■ Reports Summary ■ Rules and Scoring Management ■ Model Configuration ■ Rebuild Queues
	Organization	■ Manage Miscellaneous Configurations ■ Assign Channel and Configure Default Account ■ Rules and Scoring Management ■ Model Configuration
	API	■ Get User Profile (web service) ■ Add User to Exception List (web service) ■ Delete User from Exception List (web service) ■ Get Location and Connection Info (web service)
Organization Administrator (OA)	Global	■ Reports Summary ■ Manage Queues ■ Rebuild Queues

Role	Target	Privileges Added
	Organization	■ Manage Miscellaneous Configurations ■ Rules and Scoring Management
	API	■ Get User Profile (web service) ■ Add User to Exception List (web service) ■ Delete User from Exception List (web service) ■ Get Location and Connection Info (web service)
User Administrator (UA)	Global	■ Reports Summary
	API	■ Get User Profile (web service) ■ Add User to Exception List (web service) ■ Delete User from Exception List (web service) ■ Get Location and Connection Info (web service)

Chapter 8: Uninstalling RiskMinder

Before you uninstall RiskMinder, first remove its schema and then proceed with the uninstallation process. You can either use the **Add/Remove Programs** utility in the Microsoft Windows Control Panel to uninstall RiskMinder or run the uninstaller file (Uninstall Arcot RiskFort.exe) to remove RiskMinder from your system. After you complete the uninstallation process, perform the post-uninstallation tasks to clean up the residual WAR files and entries.

If you installed both CA AuthMinder with CA RiskMinder and you are uninstalling only CA RiskMinder, then pay special attention to the guidelines given in the following sections. These guidelines are aimed at ensuring that you do not remove or modify the common components that are also being used by CA AuthMinder.

This section guides you through the steps for uninstalling RiskMinder and its components. The section covers the following sections:

Dropping RiskMinder Schema

Note: For some reason, if you must retain the RiskMinder database schema, then *do not* perform the procedure described in this section.

Important! You may be using both CA RiskMinder and CA AuthMinder. If you plan to uninstall only RiskMinder, then:

- If you had first installed RiskMinder and then installed AuthMinder, first drop the AuthMinder schema and then drop the RiskMinder schema.
- If you had first installed AuthMinder and then installed RiskMinder, drop only the RiskMinder schema. You need not drop the AuthMinder schema.

Note that if you plan to uninstall both products, then you can drop the schemas in any order.

Refer to the section, "Uninstalling RiskMinder Server" (see page 181) to proceed with the uninstallation process.

To uninstall the RiskMinder database:

1. Navigate to the following directory:
   `<install_location>\Arcot Systems\dbscripts\`
2. Based on the database that you are using, navigate to one of the following subdirectories:
   - For Oracle:
     `<install_location>\Arcot Systems\dbscripts\oracle\`
   - For Microsoft SQL Server:
     `<install_location>\Arcot Systems\dbscripts\mssql\`
   - For MySQL:
     `<install_location>\Arcot Systems\dbscripts\mysql\`
3. Run the scripts in the *following* order to drop all database tables of RiskMinder and related components:
   a. Run drop-riskfort-3.1.01.sql.
   b. If applicable, run drop-riskfort-3dsecure-3.1.01.sql.
   c. If you have not installed or upgraded to CA AuthMinder release 7.1.01, run drop-arcot-common-2.0.sql.

      Note: The drop-arcot-common-2.0.sql script is used to remove the schema for Arcot common components. This schema is used by both AuthMinder and RiskMinder. If you have already (successfully) installed AuthMinder, you must not drop this schema because AuthMinder can continue to use it.
4. If you have no further use for the database user account that you had created for the RiskMinder schema, delete that user account.

Uninstalling RiskMinder Server

To uninstall RiskMinder Server, you need to remove the files shipped with RiskMinder. Uninstallation also deletes the scripts required to uninstall the database. If you need to remove the RiskMinder database, then see Dropping RiskMinder Schema **(see page 180)** before proceeding.

Important! If you had first installed RiskMinder and then installed AuthMinder, first uninstall AuthMinder Server and then uninstall RiskMinder Server. In other words, uninstall these products in the reverse of the order in which you installed them.

To uninstall RiskMinder Server:

1. Shut down the following gracefully:
   a. RiskMinder Server
   b. Case Management Queuing Server
   c. Any application servers where other RiskMinder components are deployed.
2. Close Administration Console, if open.
3. Ensure that all INI and other RiskMinder configuration files are closed.
4. On the desktop, click **Start, Settings**, **Control Panel, Add/Remove Programs** to open the Add or Remove Programs window.
5. From the **Currently installed programs** list, select **Arcot RiskFort**, and click **Change/Remove.**

   The Uninstall Arcot RiskFort window appears.

   Note: You can also uninstall RiskMinder by running Uninstall Arcot RiskFort.exe available in the <*install_location*>\Arcot Systems**Uninstall Arcot RiskFort**\ directory.
6. In the wizard window, select one of the following options:
   - To uninstall *all* components of RiskMinder, select **Complete Uninstall**, and go to Step 8.

     Note: You may have to wait for a few minutes for the uninstallation process to complete.
   - To uninstall the selected components, select **Uninstall Specific Features**, and click **Next** to display the Choose Product Features screen.

     This screen displays the RiskMinder components that are installed on the system. Go to Step 7.
7. Deselect the components that you want to uninstall and click **Uninstall** to display the Uninstall Arcot RiskFort window.

Important! To **Uninstall Specific Features**, follow the reverse sequence in which you installed the components. For example, if you installed Arcot RiskFort Server and then Arcot Administration Console, then first uninstall Arcot Administration Console and only then the RiskFort Server.

You may have to wait for a few minutes for the uninstallation process to complete.

After the software is uninstalled successfully, the *Uninstallation Complete* screen appears with a success message.

8. Click **Done** to exit the wizard and complete the uninstallation process.

Performing Post-Uninstallation Tasks

The post-uninstallation steps that you must perform to ensure that all RiskMinder components are removed are:

1. Delete the <*install_location*>**\Arcot Systems** directory, if not required after the uninstallation process.

   Note: If multiple CA products are installed on this system, then delete this directory *only if* RiskMinder is the last product to be uninstalled.

2. Stop the application server.
3. Uninstall the following WAR files from the appropriate subdirectory in <*APP-SERVER-HOME*>.

   Note: Here, *APP-SERVER-HOM*E represents the directory path where the application server (for example, Apache Tomcat) is installed.

   See the application server vendor documentation for detailed information about uninstalling the WAR files.

   - arcotadmin.war: Administration Console
   - arcotuds.war: User Data Service, if deployed
   - riskfort-3.1.01-sample-application.war: Sample Application
   - riskfort-3.1.01-sample-callouts.war: Sample Callout

   Note: If you have a distributed-system deployment, then locate these files on the system where you have deployed the particular application.

4. If you used Oracle Database for the database setup, then delete the tabspace_arreports_<*time_database_was_created*>.dat file from the system running the RiskMinder database.
5. Delete the DSN entry that was created during the RiskMinder installation.

   To delete this entry, open the Control Panel, navigate to **Administrative Tools, Data Sources (ODBC), System DSN**, select the required DSN, and click **Remove.**

Appendix A: RiskMinder Directory Structure

This appendix provides the information about the location of all files that are installed by the RiskMinder installer. It covers:

RiskMinder Risk Evaluation Java SDK Files

The following table lists the main directories, files, and JARs that are created by the RiskMinder installer. The table also describes specific subdirectories and files that have been referred to in this guide.

In addition to the files and directories that are discussed in the table, you also see a blank file named arcotkey in the installation directory. This file is used by the installer to detect previously installed CA products. If you delete this file, then the installer cannot detect previously installed CA products, and allows new installations to be performed in any location. As a result, the installer cannot ensure the same destination directory for multiple CA products and components, in which case, the products (or components) may not work, as expected. This file has no impact on patches and upgrade.

Directory	Used By	File Names and Description

Directory	Used By	File Names and Description
<*install_location*>\Arcot Systems**bin**\ **Book:** See *CA RiskMinder Administration Guide* for more	■ RiskMinder Server ■ Case Management Queuing Server	Contains the following executables used by RiskMinder Server: ■ arrfcasemgmtserver.exe (Tool for refreshing and gracefully shutting down the Case Management Queuing Server.) ■ arrfclient.exe (Tool for refreshing and gracefully shutting down RiskMinder Server.) ■ arrfserver.exe (Tool for setting the server management port and other server-related operations.) ■ arrfupload.exe (Tool for uploading Quova data to the RiskMinder database.) ■ arrfversion.exe (Tool for determining the version of the library files that are provided by CA.) Also contains the following library files used by RiskMinder Server: ■ aradminprotocol.dll ■ aradminwsprotocol.dll ■ arrfuds.dll ■ arrfudswrapper.dll ■ arRiskEngine.dll ■ NameValueXref.dll ■ srvmgrwsprotocol.dll ■ transwsprotocol.dll

Directory	Used By	File Names and Description
<*install_location*>\Arcot Systems**conf**\ **Note:** See Configuration Files and Options (see page 195) for more details on the configuration files that you see in this directory.	■ Administration Console	Contains the following configuration files used by Administration Console: ■ adminserver.ini (Used for reading Administration Console logging configurations.) ■ arcotcommon.ini (Used for connecting to RiskMinder database, RiskMinder instances, and Hardware Security Module (HSM), if configured.)
	■ RiskMinder Server	Contains the following configuration files for use by RiskMinder Server and other RiskMinder components: ■ arcotcommon.ini(Used for connecting to RiskMinder database, RiskMinder instances, and Hardware Security Module (HSM), if configured.) ■ riskfortdataupload.ini (Used for uploading Quova data to the RiskMinder database.) ■ securestore.enc(Used for storing the encrypted information that is required to connect to the RiskMinder database.)
	■ UDS	Contains the udsserver.ini file for use by UDS for reading UDS logging configurations.
	■ UDS ■ Administration Console	The resourcebundles directory contains the properties files for common errors thrown by Administration Console and UDS.
<*install_location*>\Arcot Systems**dbscripts**\	■ Administration Console ■ RiskMinder Server ■ UDS	Contains the database scripts to create and drop RiskMinder schemas for the Database Type that you specified during installation.

Directory	Used By	File Names and Description
<*install_location*>\Arcot Systems\docs**riskfort**\	■ Administration Console ■ RiskMinder Server	Contains the following zipped WSDLdoc: ■ Arcot-RiskFort-3.1.01-AdminWeb Service-wsdldocs.zip (The WSDLDocs for the Admin Web Service.)
	■ RiskMinder Server	Contains the following zip and XSD files for writing Callouts, and the Javadocs and WSDLdocs for Risk Evaluation SDKs: ■ Arcot-RiskFort-3.1.01-CallOutInterface-xsds.zip (The Evaluation and Scoring Request and Response files that are required for writing a Callout.) ■ Arcot-RiskFort-3.1.01-risk-evaluation-sdk-javadocs.zip ■ Arcot-RiskFort-3.1.01-risk-evaluation-wsdldocs.zip
<*install_location*>\Arcot Systems\docs**uds**\	■ UDS	Contains the following zipped WSDLdoc: ■ arcot-uds-2_0-wsdl-docs.zip (The WSDLDocs for UDS Web Services.)
<*install_location*>\Arcot Systems**java****lib**\	■ Administration Console	Contains an empty directory called **sdk** and the following WAR and JAR files required by the Administration Console Framework and UDS: ■ adminframework.jar ■ adminframework.war ■ arcot-common.jar ■ arcot-crypto-util.jar ■ arcot-euds.jar ■ bcprov-jdk15-146.jar ■ udsframework.war

Directory	Used By	File Names and Description
<*install_location*>\Arcot Systems**java****webapps**\	■ Administration Console	Contains the following WAR file required by Administration Console: ■ arcotadmin.war (The WAR file that is required to deploy Administration Console.)
	■ UDS	Contains the arcotuds.war file required to deploy UDS for: ■ LDAP connectivity ■ Access to UDS web services ■ Authentication and Authorization for web services
<*install_location*>\Arcot Systems**logs**\ **Book:** See "RiskMinder Logging" in the *CA RiskMinder Administration Guide* for detailed information about these log files.		Contains the log files used by Administration Console, Case Management, RiskMinder, and UDS. You can use the **backup** subdirectory to store the older logs, if available.
	■ Administration Console	■ arcotadmin.log
	■ RiskMinder Server	■ arcotriskfort.log ■ arcotriskfortstartup.log
	■ Case Management Queuing Server	■ arcotriskfortcasemgmtserver.log ■ arcotriskfortcasemgmtstartup.log
	■ UDS	■ arcotuds.log **Note:** This log appears only if you deployed the UDS WAR file (arcotuds.war) for LDAP connectivity.
<*install_location*>\Arcot Systems**native**\	■ Administration Console ■ UDS	Contains ArcotAccessKeyProvider.dll (in appropriate subdirectories) used for reading the contents of securestore.enc for your 32-bit or 64-bit OS platform (RHEL, Solaris SPARC, or Microsoft Windows).

Directory	Used By	File Names and Description
<*install_location*>\Arcot Systems**odbc32v70wf**\	■ RiskMinder Server	Contains the CA Arcot-branded DataDirect ODBC libraries for all the databases supported by RiskMinder.
<*install_location*>\Arcot Systems**plugins****rules**\	■ RiskMinder Server	Contains DLL (library binary) files to support all out-of-box RiskMinder rules, and Scoring.
<*install_location*>\Arcot Systems**resourcepacks**\	■ Administration Console ■ UDS	Contains the required Administration Console and product pack bundles: ■ bundle_adminconsole.zip ■ bundle_riskfort.zip Also contains the **i18n** subdirectory, which is where you will store the required files for internationalization. **Note:** See Preparing for Localization for more information about how to localize RiskMinder.
<*install_location*>\Arcot Systems**samples****java**\	■ RiskMinder Server ■ RiskMinder Risk Evaluation SDK	The **java** subdirectory contains the sample WAR files for: ■ riskfort-3.1.01-sample-application.war to deploy the RiskFort Sample Application. ■ riskfort-3.1.01-sample-callouts.war to deploy the RiskFort Sample Callout.
<*install_location*>\Arcot Systems**sdk**\	■ RiskMinder Risk Evaluation SDK	Contains SDKs and dependent files supported by RiskMinder in the **c**, **devicedna**, and the **java** flavors. The **devicedna** subdirectory contains the accompanying JavaScripts and Flash files that are used by these SDKs and the MFP and DeviceDNA modules. See RiskMinder Risk Evaluation Java SDK Files (see page 183) for detailed explanation of the contents of this directory.

Directory	Used By	File Names and Description
<*install_location*>\Arcot Systems**tools**\	■ Administration Console	The **common** subdirectory contains the following subdirectories: ■ The **arreporttool** subdirectory contains the report command-line utility that enables you to export (or download) reports. ■ The **bundlemanager** subdirectory contains the files that are required by Administration Console Resourcepack. ■ The **uds-monitor** subdirectory contains the script for checking the health of UDS.
<*install_location*>\Arcot Systems**tools\<platform>**\ The <*platform*> can be: linux, solsparc, and win.	■ Administration Console ■ User Data Service (UDS)	Contains the DBUtil.exe tool for your OS platform (RHEL, Solaris SPARC, or Microsoft Windows). This tool is required for editing securestore.enc, which stores the encrypted information needed by RiskMinder Server to connect to the RiskMinder database.
<*install_location*>\Arcot Systems\ **Uninstall_Arcot RiskFort**\	■ RiskMinder Server	Contains the files required to uninstall RiskMinder. In addition, ■ The **jre** subdirectory contains all files that are required for Java Runtime Environment (JRE) support: – Java Virtual Machine – Runtime Class Libraries – Java Application Launcher ■ The **resource** directory contains all files that are required by the installer for the uninstallation of RiskMinder.

Directory	Used By	File Names and Description
<*install_location*>\Arcot Systems **wsdls**\	■ RiskMinder Server	Contains the WSDL files required by Administration Console (the **admin** subdirectory), RiskMinder (the **riskfort** subdirectory), and UDS (the **uds** subdirectory). See section, "RiskMinder Risk Evaluation Java SDK Files" (see page 183) later in this appendix for detailed explanation of the contents of this directory.

The following table lists the directory location of the files that are used by Risk Evaluation Java SDK.

Directory	File Description
<*install_location*>\Arcot Systems**docs****riskfort**\	Contains the Arcot-RiskFort-3.1.01-risk-evaluation-sdk-javadocs.zip file, which contains the Javadocs for Risk Evaluation SDK.
<*install_location*>\Arcot Systems**samples****java**\	Contains the following: ■ riskfort-3.1.01-sample-application.war (For deploying Sample Application.) ■ riskfort-3.1.01-sample-callouts.war (For deploying the Sample Callout Server shipped with the product.) **Book:** See *CA RiskMinder Administration Guide* for more information about deploying and use this Sample Callout.
<*install_location*>\Arcot Systems**sdk**\	Contains SDKs and dependent files supported by RiskMinder.
<*install_location*>\Arcot Systems**sdk****c**\	Contains the library and included files required for C SDK.

Directory	File Description
<*install_location*>\Arcot Systems**sdk****devicedna**\	The directory contains: ■ riskminder-client.js, which is required for collecting DeviceDNA information at the client-end. ■ riskminder-client.swf, which is required for migrating Flash-based cookies from the preceding releases to the Browser (HTTP) cookie store that this release supports.
<*install_location*>\Arcot Systems**sdk****flash**\	The directory contains: ■ arcot-devicedna.swf, which manages the Device ID Flash object. ■ crossdomain.txt, which specifies the list of domains that can access the Flash object.
<*install_location*>\Arcot Systems**sdk****java**\	■ The **lib** subdirectory contains the CA-supplied and third-party JAR files that are used by the product. **Book:** See the Third-Party Software Licenses document in the package for the licensing information of these third-party JARs. ■ The **properties** directory contains the property files that are required for configuration of RiskMinder.
<*install_location*>\Arcot Systems**sdk****java****lib****arcot**\	Contains the following JAR files used by Risk Evaluation Java SDK. ■ arcot_core.jar ■ arcot-pool.jar ■ arcot-riskfort-evaluaterisk.jar ■ arcot-riskfort-issuance.jar ■ arcot-riskfort-mfp.jar **Note:** The Issuance API has been deprecated in this release. However, arcot-riskfort-issuance.jar ensures backward compatibility with the preceding releases.
<*install_location*>\Arcot Systems**sdk****java****lib****external**\	Contains the third-party JAR files required by the Risk Evaluation Java SDK. ■ bcprov-jdk15-146.jar ■ commons-lang-2.0.jar ■ commons-pool-1.5.5.jar

Directory	File Description
<*install_location*>\Arcot Systems**sdk\java\properties**\	Contains the following files: ■ log4j.properties.risk-evaluation ■ riskfort.risk-evaluation.properties

RiskMinder WSDL Files

The following table lists the directory location of the files that are used by Risk EvaluationWSDLs.

Directory	File Description
<*install_location*>\Arcot Systems**docs\riskfort**\	Contains the zipped WSDLdocs for RiskMinder Risk Evaluation and Administration Console: Arcot-RiskFort-3.1.01-AdminWebService-wsdldocs.zip Arcot-RiskFort-3.1.01-risk-evaluation-wsdldocs.zip
<*install_location*>\Arcot Systems**docs\uds**\	Contains the arcot-uds-2_0-wsdl-docs.zip file required by UDS. This WSDL describes the UDS Web services and how to access them.
<*install_location*>\Arcot Systems**wsdls\admin**\	Contains the ArcotRiskFortAdminWebService.wsdl file required by Administration Console. This WSDL describes the RiskMinder Administration Web services and how to access them. In addition, it can be used to add Exception Users.
<*install_location*>\Arcot Systems**wsdls\riskfort**\	Contains the following file required by RiskMinder: ■ ArcotRiskFortEvaluateRiskService.wsdl The WSDLdoc describes the Risk Evaluation Web Service and how to access it.

Directory	File Description
<*install_location*>\Arcot Systems**wsdls****uds**\	Contains the WSDLs and XML Schema files required by UDS. These WSDLs describe the UDS Web services and how to access them: ■ ArcotConfigManagementSvc.wsdl (WSDL for creating and managing user account types.) ■ ArcotOrganizationManagementSvc.wsdl (WSDL for creating and managing organizations.) ■ ArcotUserManagementSvc.wsdl (WSDL for creating and managing users and user accounts.) ■ ArcotUserSchema.xsd (XML Schema Definition that serves as the reference library that can be uses by your code for working with the UDS web services.)

Appendix B: Configuration Files and Options

This appendix discusses the configuration files that RiskMinder uses and the parameters that you must configure in these files. It also includes samples of these default configuration files.

The configuration files important for RiskMinder can be categorized as:

- INI Files (see page 195)
- Properties Files (see page 212)

INI Files

The plain-text INI files that are used for configuring RiskMinder include:

- adminserver.ini (see page 196)
- arcotcommon.ini (see page 198)
- riskfortdataupload.ini (see page 208)
- udsserver.ini (see page 210)

All RiskMinder configuration files are available at the following default location:
`<install_location>\Arcot Systems\conf\`

adminserver.ini

The adminserver.ini file contains the parameters to set the Administration Console log information.

Logging Configurations

The following table lists the log file information that is used by Administration Console. The common log-level values that can be set in this file are:

- FATAL
- WARNING
- INFO
- DEBUG

Book: See *CA RiskMinder Administration Guide* for more information about the log levels.

Parameter	Default Value	Description
log4j.rootCategory	ERROR, roothandle **Important!** roothandle is the name of the Administration Console log handle and *must* 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.
log4j.logger.com.arcot.euds	INFO	The log level for writing the User Data Service (UDS) information.
log4j.logger.com.arcot.admin	INFO	The log level that must be used to write the Administration Console logs.
log4j.logger.com.arcot.admin.framework	INFO	The log level that must be used to write the Administration Console Framework logs.
log4j.logger.com.arcot.adminconsole	INFO	The log level that must be used to write the Administration Console logs.
log4j.logger.com.arcot.common.cache	INFO	The log level for writing the cache-related information.
log4j.logger.com.arcot.common.crypto	INFO	The log level for writing the information related to HSM.

Parameter	Default Value	Description
log4j.logger.com.arcot.crypto.impl.SecureStoreUtil	INFO	The log level that must be used to write the logs, if you are using a hardware-based or software-based HSM.
log4j.logger.com.arcot.common.database	INFO	The log level that must be used to write the database information.
log4j.logger.com.arcot.common.ldap	INFO	The log level that must be used to write the LDAP information.
log4j.appender.roothandle	org.apache.log4j.RollingFileAppender	The root logger that resides at the top of the logger hierarchy. All children loggers inherit this value, if no value is specified.
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**	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: <*install_location*>\Arcot Systems**logs**\
log4j.appender.roothandle.MaxFileSize	10 MB	The maximum allowed file size of the log file.
log4j.appender.roothandle.MaxBackupIndex	100	The maximum number of backup files that can be created. When the number of backup files reaches this number, then the application starts to overwrite from the first log file.
log4j.appender.roothandle.layout	org.apache.log4j.PatternLayout	The output format, as specified by ConversionPattern.

Parameter	Default Value	Description
log4j.appender.roothandle.layout.ConversionPattern	%d{yyyy-MM-dd HH:mm:ss,SSS z} : [%t] : %-5p : %-5c{3} : %m%n	The format in which the Administration Console log file entries are written: ■ 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) **Note:** This pattern is similar to the C language printf function.

arcotcommon.ini

The arcotcommon.ini file contains the parameters for database and instance settings for RiskMinder Sever and other components (Administration Console and User Data Service) of RiskMinder. Typically, you must edit the following sections in this file:

- Database Settings (see page 199)
- HSM Encryption Settings (see page 205)
- Instance Settings (see page 206)

You can also change the default startup logging settings for RiskMinder Server and Case Management Queuing Server by using arcotcommon.ini. See Changing Server Startup Logging Parameters (see page 207) for more information.

Database Settings

The database settings in arcotcommon.ini allow you to identify the database to which the server connects and the backup database to use for failover. These settings also enable you to configure database communications resources available between the server and the database.

Note: For notes and recommendations for database settings, refer to the "Configuring Database Server" (see page 51) section in "Preparing for Installation" (see page 45).

You must edit the following sections, which are related to database settings in the arcotcommon.ini file:

- [arcot/db/dbconfig]
- [arcot/db/primarydb]
- [arcot/db/backupdb]

[arcot/db/dbconfig]

This section enables you to specify the type of database and generic information about this database type. The following table lists the database setting parameters in the [arcot/db/dbconfig] section.

Parameter	Default	Description
DbType	--	The type of database applicable to all database connections. The supported values are: ■ oracle ■ mssqlserver ■ mysql
Driver	--	The fully-qualified name of the database driver class that is supplied by the JDBC driver vendor. **Note:** Consult your JDBC vendor documentation for the right driver name. For example: – **Oracle:** oracle.jdbc.driver.OracleDriver – **Microsoft SQL Server:** com.microsoft.sqlserver.jdbc.SQLServerDriver – **MySQL**: com.mysql.jdbc.Driver
MinConnections	4	The minimum number of connections to initially create between the server and the database.

Parameter	Default	Description
MaxConnections	64	The maximum number of connections that will be created between the server and the database. **Note:** There is a limit to how many connections a database allows and that limit may limit the server from creating the MaxConnections number of connections. See your database driver documentation for more information about the limit on the number of inbound connections.
IncConnections	2	The number of connections that will be created when a new connection is needed between the RiskMinder components and the database.
MaxIdleConnections	64	The maximum number of idle database connections that the server can maintain.
MaxWaitTimeFor Connection	30000	The maximum time (in **milliseconds**) the server must wait for a connection to become available (when there are no available connections) before timing out.
AutoRevert	1	Whether or not the system will attempt 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 back 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.
ConnRetrySleep Time	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 heartbeats checks on all databases.

Parameter	Default	Description
Profiling	0	Whether the database messages are being logged. Set the value to 1 if you want to enable logging of database messages.
EnableBrandLicensing	1	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. **Important!** If present, this value must *not* be edited.
MaxTransactionRetries	3	The maximum number of times the transaction is retried with a database instance for pre-defined error conditions.
TransactionRetrySleep Time	10	The interval in milliseconds between two consecutive transaction retries.

[arcot/db/primarydb]

This section enables you to specify the primary database to which RiskMinder Server is connected. You can configure more than one primary database by specifying the required number, *N* in the following parameters:

- Datasource.*N*
- AppServerConnectionPoolName.*N*
- URL.*N*
- Username.*N*
- TrustStorePath.*N*
- KeyStorePath.*N*
- HostNameInCertificate.*N*

The following table lists the database setting parameters in the [arcot/db/primarydb] section.

Parameter	Default	Description
Datasource.*N*	No default	The name of the ODBC System Data Source Name (DSN) pointing to the primary database hosting the server data.

Parameter	Default	Description
AppServerConnection PoolName.*N*	No default	The JNDI name used to look up the connection pool object, if the database connection pooling feature of the application server is being used. A pool by this JNDI name should be created in the containing application server, and sufficient access right must be given to Web applications for it to use the connection pool. 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 See appendix, "Configuring Application Server for Database Connection Pooling" (see page 257) for more information. If the application server connection pool is *not* required, then leave this configuration empty.
URL.*N*	No default	The name of the JDBC data source. For ■ **Oracle** -> jdbc:oracle:thin:<*server*>:<*database_port*>:<*sid*> ■ **Microsoft SQLServer** -> jdbc:sqlserver://<*server*>:<*database_port*>;databaseName=<*databasename*>;selectMethod=cursor ■ **MySQLServer** -> jdbc:mysql://<*server*>:<*database_port*>/<*database*>
Username.*N*	No default	The user ID used by the server to access the database.

Parameter	Default	Description
TrustStorePath.*N* **Note:** To be used only if you have SSL configured between RiskMinder and the database.	No default	The SSL Certificate Truststore Path corresponding to Datasource.*N*. The path (including the filename) refers to the certificate Truststore file, which contains the list of certificates that the client trusts. **Important!** The password corresponding to TrustStorePath.*N* must be securely stored in securestore.enc, with the value of TrustStorePath.*N* as the key. The DBUtil tool is used to achieve this. **Note:** See the *CA RiskMinder Administration Guide* for more information about DBUtil.
KeyStorePath.*N*		**Note:** This attribute is used only for MySQL. 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.*N*. The path (including the filename) refers to the certificate keystore file. The password corresponding to KeyStorePath.*N* must be securely stored in securestore.enc with the value of KeyStorePath.*N* as the key.

Parameter	Default	Description
HostNameInCertificate.*N* **Note:** To be used only if you have SSL configured between RiskMinder and the database.	No default	The value of Common Name (CN) in the subject Distinguished Name (DN) of Datasource.*N* SSL Certificate in Truststore.

[arcot/db/backupdb]

This section [arcot/db/backupdb] enables you to specify the backup database to use for failover. You can configure more than one failover database by specifying the required number, *N* in the following parameters:

- Datasource.*N*
- AppServerConnectionPoolName.*N*
- URL.*N*
- Username.*N*
- TrustStorePath.*N*
- KeyStorePath.*N*
- HostNameInCertificate.*N*

This section uses the same parameters as the [arcot/db/primarydb] section. Refer to the table in the previous section for the list of database setting parameters in this section.

HSM Encryption Settings

The arcotcommon.ini file enables you to specify the configurations for your Hardware Security Module (HSM). As a result, you can store the Private Keys that are used for RiskMinder in an encrypted format. The following HSMs are supported:

- Chrysalis-ITS Luna SA
- Thales nFast (nCipher netHSM)

The following table lists the common configurations for secure storage, as specified in the [arcot/crypto/device] section.

Parameter	Default	Description
HSMDevice	S/W	The mode that sets whether the RiskMinder information must be encrypted with a key stored in database or with the one in stored the HSM. Supported values are: ■ S/W: Indicates that the data is encrypted with the key label that is stored in database. ■ chrysalis: Indicates that the Chrysalis (Luna) HSM is used to encrypt the data. ■ nfast: Indicates nFast (nCipher netHSM) is used to encrypt the data.

The following table lists the configuration parameters for Chrysalis-ITS Luna SA, as specified in the [crypto/pkcs11modules/chrysalis] section.

Parameter	Default	Description
sharedLibrary	<*location/to/cryptoki*.dll>	The absolute path to the PKCS#11 shared library corresponding to the HSM. The default value for Chrysalis (Luna) is: C:\Program Files\LunaSA\cryptoki.dll
storageSlot	0	The HSM slot where the encryption keys (symmetric as well as asymmetric) are present.
accelSlot	0	The slot for internal use by CA.
sessionCount	20	The maximum number of sessions that can be established with the HSM device.

The following table lists the configuration parameters for nCipher netHSM, as specified in the [crypto/pkcs11modules/nfast] section.

Parameter	Default	Description
sharedLibrary	<*location/to/cknfast*.dll>	The absolute path to the PKCS#11 shared library corresponding to the HSM. The default value for nFast (nCipher netHSM) is: C:\nfast\bin\cknfast.dll
storageSlot	1	The HSM slot where the encryption keys (symmetric as well as asymmetric) are present.
accelSlot	0	The slot for internal use by CA.
sessionCount	200	The maximum number of sessions that can be established with the HSM device.

Instance Settings

In a farm of servers, it is recommended that every instance of the server has its own unique identification. RiskMinder supports a parameter to set and identify every instance of the servers. This section enables you to configure these system-wide settings for unique instances. The following table lists the instance setting parameters in the [arcot/system] section.

Parameter	Default	Description
InstanceId	1	The parameter that can be used to identify any server instance. **Important!** It is mandatory that you provide unique values for every instance of the server. The server instance is also displayed in the transaction reports, making it easier to trace the server instance to the transaction.

Changing Server Startup Logging Parameters

If you want to change the logging parameters that you see when RiskMinder Server or Case Management Queuing 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. (**For RiskMinder Server**) Add the following section at the end of the file:

```
[arcot/riskfort/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 Systems\).
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 Systems\).
LogLevel		The default logging level for the server, unless an override is specified. The possible values are: ■ 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: ■ 0: Local Time ■ 1: GMT

1. (**For Case Management Queuing Server**) Add the following section at the end of the file:

```
[arcot/riskfortcasemgmtserver/startup]
LogFile=
LogFileSize=10485760
BackupLogFileDir=
LogLevel=
LogTimeGMT=0
```

   The table in the previous step explains these parameters.

2. Set the required values for the parameters that you want to change.
3. Save and close the file.
4. Restart RiskMinder Server.

riskfortdataupload.ini

RiskMinder uses Quova data to identify the geolocation of a user by using the IP address of the system from which the transaction originated. It then uses this data to evaluate Negative Country, Negative IP, and Zone Hopping rules.

RiskMinder is shipped with the *Arcot RiskFort Data Upload Tool* (arrfupload) to enable you to upload the geolocation data from Quova files to the RiskMinder database. The riskfortdataupload.ini file controls the behavior of the Arcot RiskFort Data Upload tool and is available at the following location:
`<install_location>\Arcot Systems\conf\`

The following table lists the configuration parameters in this file.

Parameter	Default	Description
Tables	Do Not Load	The tables that the user can work with. Possible values are: ■ GeoPoint ■ Anonymizer

Parameter	Default	Description
Load	0	The indicator whether to upload the data to the table or not. Possible values are: ■ 0: Do not load) ■ 1: (Load)
Swap	0	The indicator whether to swap the tables or not. Possible values are: ■ 0: (Do not swap) ■ 1: (Swap)
Filename	--	The name of the file from which the Quova data has to be loaded. **Important!** Specify the absolute path to the file, with the file name.

Note: If both, Load and Swap are set to 1, then first the table is loaded and then swapped.

udsserver.ini

The udsserver.ini file contains the parameters to set the User Data Service (UDS) log information. The following table provides information about parameters that you must configure for RiskMinder.

The common log-level values that can be set in this file are:

- FATAL
- WARNING
- INFO
- DEBUG

Book: See *CA RiskMinder Administration Guide* for more information about the log levels.

Parameter	Default Value	Description
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.
log4j.logger.com.arcot.euds	INFO	The log level that must be used to write the UDS information.
log4j.logger.com.arcot.crypto.impl.SecureStoreUtil	INFO	The log level that must be used to write the logs, if you are using a hardware-based or software-based HSM.
log4j.logger.com.arcot.common.database	INFO	The log level that must be used to write the database information.
log4j.logger.com.arcot.common.cache	INFO	The log level that must be used to write the UDS cache information.
log4j.appender.debuglog	org.apache.log4j.RollingFileAppender	The name of the UDS log handle that specifies the mode in which the log file is opened and the offset pointer where the next operation will begin.

Parameter	Default Value	Description
log4j.appender.debuglog.File	${arcot.home}/logs/arcotuds.log	The log file name and the location where the UDS logs will be created. By default, the UDS log file name is arcotuds.log and is created in the following location: <*install_location*>\Arcot Systems\logs\
log4j.appender.debuglog.MaxFileSize	10MB	The maximum allowed file size of the log file.
log4j.appender.debuglog.MaxBackupIndex	100	The maximum number of backup files that can be created. When the number of backup files reaches this number, then the application starts to overwrite from the first log file.
log4j.appender.debuglog.layout	org.apache.log4j.PatternLayout	The output format, as specified by the ConversionPattern parameter.
log4j.appender.debuglog.Encoding	UTF-8	The encoding to use when writing the entries in the log file.
log4j.appender.debuglog.layout.ConversionPattern	%d{yyyy-MM-dd HH:mm:ss,SSS z} : [%t] : %-5p : %-5c{3} : %m%n	The format in which the UDS log file entries are written: ■ 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) **Note:** This pattern is similar to the C language printf function.

Properties Files

RiskMinder primarily uses the properties files that are discussed in the following subsections:

- riskfort.risk-evaluation.properties (see page 212)
- log4j.properties.risk-evaluation (see page 215)

These files are available at:
`<install_location>\Arcot Systems\sdk\java\properties\`

riskfort.risk-evaluation.properties

The riskfort.risk-evaluation.properties file provides the parameters for the RiskMinder Risk Evaluation Java SDK and Sample Application to read RiskMinder Server information. The following table lists the configuration parameters that are used in this file.

Parameter	Default	Description
HOST.1	localhost	IP address of RiskMinder Server.
PORT.1	7680	Port number where RiskMinder Server is listening to incoming requests.
CONNECTION_TIMEOUT	10000	Time in **milliseconds** before RiskMinder Server is considered unreachable.
CONNECTION_RETRIES	3	Maximum number of retries allowed with RiskMinder Server.
READ_TIMEOUT	30000	Maximum time in **milliseconds** allowed for a response from RiskMinder Server.
USE_CONNECTION_POOLING	1	Parameter for enabling or disabling connection pooling to RiskMinder Server: ■ 0: Disabled ■ 1: Enabled
MAX_ACTIVE	128	Maximum number of active connections (from the pool) allowed with RiskMinder Server. It controls the maximum number of connections that can be borrowed from the pool at one time. When negative, there is no limit on the number of objects that might be active at a time.

Parameter	Default	Description
TIME_BETWEEN_CONNECTION_EVICTION	900000 (15 minutes)	Time in **milliseconds** between consecutive runs of the Idle Connection Evictor thread. **Note:** If this parameter is set to -1, then connections are not evicted. **Important!** Ensure that TIME_BETWEEN_CONNECTION_EVICTION + IDLE_TIME_OF_CONNECTION is less than the connection timeout of your firewall (between SDK and RiskMinder Server.) This ensures that no connection is abruptly dropped by the firewall because of idle time, which ensures smooth functioning of the system.
IDLE_TIME_OF_CONNECTION	1800000 (30 minutes)	Idle time (in **milliseconds**) after which a connection will be closed. **Note:** If this parameter is set to -1, then connections are not evicted.
WHEN_EXHAUSTED_ACTION	BLOCK	The SDK behavior when all connections are exhausted: ■ BLOCK: The SDK waits for a connection to be free. This behavior is the default behavior. ■ FAIL: The transaction is considered as failed. ■ GROW: The SDK can increase the pool.
TRANSPORT_TYPE	TCP	Default value for RiskMinder Server to start up is TCP. Set this parameter to SSL, if RiskMinder Native protocol is set to SSL. In other words, set this parameter to SSL, if you want to enable SSL-based secure communication between Administration Console and RiskMinder Server. **Note:** Restart RiskMinder Server, if you change the value to SSL.

Parameter	Default	Description
CA_CERT_FILE		Path for the CA certificate file of the server. The file *must* be in .PEM format. Provide the complete path for the file. For example: <*install_location*>/certs/ca.pem or <*install_location*>\\certs\\ca.pem **Note:** – Use CLIENT_P12_FILE for the client PKCS#12 file (which contains the Client key and the Certificate pair.) – Use CLIENT_P12_PASSWORD for the password of the specified PKCS#12 file.
LIFO	false	Indication whether or not the connection pool returns idle objects in Last-In-First-Out order. Set it to false to ensure that each connection is used in a round-robin manner and is not idle. For high-load deployments, the recommended value is false.
NUM_PRE_CREATE	32	Number of connections that must be created during the initialization of the pool.
NUM_CONNECT_FAILURES_TO_TRIGGER_FAILOVER	2	Number of consecutive connection failures that will trigger the failover to another pool.
MAX_IDLE	-1	The maximum number of idle connections from the SDK to a given server instance allowed in the pool.
MAX_WAIT_TIME_MILLIS	3000	The maximum time (**in milliseconds**), a connection request will wait for a connection from the pool. **Note:** If this parameter is set to -1, the request waits indefinitely.

log4j.properties.risk-evaluation

The log4j.properties.risk-evaluation file specifies the logging behavior of RiskMinder and its Risk Evaluation components. The following table provides information about the parameters that you may need to change for Risk Evaluation.

Parameter	Default Value	Description
log4j.rootLogger	INFO, debuglog	Specify the log level that must be used to write the logs. The supported log levels are: ■ FATAL ■ WARNING ■ INFO ■ DEBUG **Book:** See *CA RiskMinder Administration Guide* for more information about the log levels.
log4j.logger.com.arcot	INFO	
log4j.logger.com.arcot.riskfortAPI	DEBUG	
log4j.appender.debuglog.File	arcot-riskfort-evaluaterisk.log	The name of the log file. Possible values for this parameter are: ■ riskfortsdk.log (for RiskMinder Java SDK) ■ arriskfortws.log (for RiskMinder Web Service)
log4j.appender.debuglog.MaxFileSize	1MB	The maximum allowed file size of the log file.
log4j.appender.debuglog.MaxBackupIndex	3	The maximum number of backup files that can be created. When the number of backup files reaches this number, then the application starts to overwrite from the first log file.

Appendix C: Changing Hardware Security Module Information After the Installation

Note: Before you proceed 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 58) for more information.

As mentioned in "Hardware Security Module (HSM) Requirements" (see page 46), RiskMinder now supports Hardware Security Module (HSM) to secure your data. If you choose to encrypt the data by using HSM, then the data that is stored in the database is encrypted with the key that resides in the HSM.

By default, RiskMinder uses the software (**S/W**) mode to encrypt data. Therefore, you must change the mode to hardware (**chrysalis** or **nfast**). You do so by using the [arcot/crypto/device] section in arcotcommon.ini. This file also 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, specify the sharedLibrary parameter in the corresponding section. After you specify the HSM information, re-create the securestore.enc file with the HSM key label, initialize the HSM, and then initialize RiskMinder to use the HSM key.

Changing HSM Configuration Post-Installation

During the installation process, the RiskMinder installer prompts you to specify this HSM-related information. However, if you want to change the HSM configurations later, such as changing the data encryption mode and configuring other HSM information that is needed by RiskMinder, then perform the following steps:

1. Navigate to the following location:
   `<install_location>\Arcot System\conf\`
2. Take a backup of securestore.enc.
3. Delete the existing securestore.enc file from *<install_location>*\Arcot System\conf\.
4. To change the data encryption mode from software (S/W) to hardware (chrysalis or nfast), and configure the HSM information that RiskMinder needs:
   a. Navigate to the following location:
      `<install_location>\Arcot System\conf\`
   b. Open arcotcommon.ini in a text editor.
   c. In the [arcot/crypto/device] section:
      - Set the HSMDevice parameter to chrysalis for Luna HSM.

      or

      - Set the HSMDevice parameter to 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:
      - The default location of the Luna HSM library is <*SYSTEM_DRIVE*>:\Program Files\LunaSA\cryptoki.dll.

      or

      - The default location of the nCipher netHSM is <*SYSTEM_DRIVE*>:\nfast\bin\cknfast.dll.

      Note: See arcotcommon.ini (see page 198) for more information about 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 System\tools\platform\`
6. Run the DBUtil tool with the following commands:

   Note: The database user (<*Database_Username*>) that you specify in the following commands is case-sensitive.

   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 securestore.enc file with the specified key label. The generated file in stored in the *<install_location>*\Arcot System\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 RiskMinder Server uses to connect to the RiskMinder database. <*Database_Password*> refers to the password used to connect to the database.

The preceding command initializes the RiskMinder Server data to be encrypted by 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 RiskMinder database. <*Database_Password*> refers to password used to connect to the database.

The preceding command initializes Administration Console and the User Data Service data to be encrypted by using HSM.

Appendix D: Database Reference

The RiskMinder 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 causes the tables to grow. Because of restricted disk space, as a database administrator managing the RiskMinder deployments, you may 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.

Trim only the tables that capture transaction details, such as audit log information. *Do not* trim tables that capture user information, which is necessary to assess the risk evaluation.

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 adversely impacts performance during the delete process. Depending on the size of the rollback segments, this deletion may even cause the system to fail. It is also strongly recommended that you archive older records and not delete them completely.

This section discusses the recommendations on database table replication, how to calculate the database size while you are planning to set up the database for RiskMinder, and lists all tables used by RiskMinder, with some trimming recommendations:

RiskMinder Database Tables

This section briefly explains all the database tables:

Used by RiskMinder

The following table lists all RiskMinder database tables and their description.

Table Name	Description
ARQGEOANONYMIZER1	Stores the known IP addresses of the anonymizers that do not propagate the end-user IP addresses. This is the primary table. **Note:** While reloading data to this table, RiskMinder Server refers to ARQGeoAnonymizer2.
ARQGEOANONYMIZER2	Stores the known IP addresses of the anonymizers that do not propagate the end-user IP addresses. This is the secondary table. **Note:** While reloading data to this table, RiskMinder Server refers to ARQGeoAnonymizer1.
ARQGEOPOINT1	Stores the geo-location information for different ranges of IP addresses. This information is obtained from Quova. **Note:** While reloading data to this table, RiskMinder Server refers to ARQGEOPOINT2.
ARQGEOPOINT2	Stores the geo-location information for different ranges of IP addresses. This information is obtained from Quova. **Note:** While reloading data to this table, RiskMinder Server refers to ARQGEOPOINT1.
ARQUOVAVERSION	Tracks the files from Quova that were uploaded to ARQ* tables.
ARRF_CASE_TXN	Contains the Case-to-Transaction mapping and related details of the default Channel. If you define a specific Channel for your deployment, then another database table is created with the Channel name appended to the default table, for example, ARRF_CASE_TXN_<*channel_name*>.
ARRF_CMA	Contains the repeated transactions of the same combination of Cardholder-Merchant-Amount (CMA). **Note:** If the rule is not used, then the table is empty.
ARRF_IMA	Contains the repeated transactions of the same combination of IP-Merchant-Amount. **Note:** If the rule is not used, the table is empty.

Table Name	Description
ARRFADDONEXPOSEDPARAMS	Stores the parameter details used by the custom rules you deploy. This table also stores the information whether specific parameters can be modified by a custom rule during processing. **Note:** It is recommended that you consult CA Support before modifying any parameters.
ARRFADDONRULELISTDATA	Contains list data and corresponding dataset version. This is used by rules or rule fragments that use IN_LIST and IN_CATEGORY operators.
ARRFADDONRULEMAPPINGDATA	Contains mapping of the elements and the category to which it belongs to. This data is used by rules that use the IN_CATEGORY operator to store the DATA-to-Category mapping. For example, Merchant rules in a 3-D Secure deployment.
ARRFADDONRULETYPE	Stores the detailed configuration information for custom rules implemented for each organization in the system.
ARRFADVICECODE	Stores the list of available risk advices.
ARRFADVICECONFIG	Stores mapping of risk score ranges and corresponding advice. **Note:** Currently, this mapping is same for all the organizations.
ARRFBASECHANNELELEMENTS	Stores the mapping of all common elements and their configurations across channels.
ARRFBUCKETCONFIG	Stores the details of all categories used for signature matching by MFP and DeviceDNA. In other words, this table contains the master list of all classifications, and their details, such as attributes and relative weight that is used in the DeviceDNA algorithm.
ARRFBUCKETELEMENTCONFIG	Stores the configuration details for all elements in all categories used for signature matching by DeviceDNA. In addition, this table contains the classification for these elements.
ARRFCASEAUDITLOG	Stores the case details and other case-related activities that are logged.
ARRFCASEQUEUES	Stores the definitions of each case queue.
ARRFCASES	Stores the details of all the open cases in the system, irrespective of the queue they belong to.

Table Name	Description
ARRFCHANNEL	Stores the basic definition (such as, case transaction table name and audit log table name) of all Channels that exist in the system.
ARRFCHANNELDETAILCATEGORY	Stores the details on various categories that GUI display elements belong to, for each channel.
ARRFCHANNELELEMENTS	Stores the details of all Channel elements.
ARRFCHANNELMSGPROPERTIES	Stores channel-specific localization information, such as channel display names and keys. **Note:** Localization is not supported in the current release.
ARRFCHANNELTXNTYPE	Stores the mapping details of all transactions supported for each channel in the system.
ARRFCHANNELTXNTYPEELEMENTS	Stores the channel-wise details of all possible element types that can come as a part of the request.In other words, this table stores the mapping of channel elements to action.
ARRFCLIENTCERTSANDKEYS	Stores SSL keys and certificates required for communicating with a de-tokenization service. **Note:** Currently, this table is applicable only for TransFort-RiskMinder integration deployments.
ARRFCLIENTSSLROOTCAS	Stores the client trust stores and the corresponding root CA certificates for two-way SSL authentication.
ARRFCONFIGURATION	Stores global- and organization-level miscellaneous RiskMinder configurations. This includes the information related to case details and other case-related activities that are logged.
ARRFCOUNTRY	Stores the list of all countries and their ISO codes.
ARRFCOUNTRYLIST	Stores the list of all countries as listed in Quova data.
ARRFCURRCONVRATES	Stores the list of all supported currencies and their corresponding conversion rates.
ARRFCURRENCY	Stores the details of all currencies, their ISO codes, and exponents for each.
ARRFCURRENTCMSCHEDULE	Stores the case schedules created by Case Management Queuing Server.
ARRFCURRENTORGCONFIG	Stores the current configuration for all organizations in the system.

Table Name	Description
ARRFDATAVERSIONMAPPING	Stores all configured RiskMinder configuration information. The information in this table contains version information, and therefore can contain multiple entries per configuration.
ARRFDBERRORCODES	Contains all database error codes that indicate a possible communication failure. **Note:** It is recommended that you consult CA Support before editing this table.
ARRFDEVICECONTEXT	Stores the context information (such as device status, timestamp of the transaction, and the requested action) for each incoming transaction from a user device. **Note:** This information is used for Device Velocity checks.
ARRFDEVICEINFO	Stores detailed information for all devices used for user transactions.
ARRFDEVICEINFOHIST	Stores the history of all user devices registered with the system.
ARRFDEVICETYPE	Stores the master list of all supported desktop and hand-held devices.
ARRFDEVUSERASSO	Stores all information related to user-device mapping.
ARRFDEVUSERASSO_ARCHIVE	Stores all archived information related to user-device mapping.
ARRFDISPLAYNAMES	Stores all variable strings (for DISPLAYNAMEKEY) that are used by Administration Console labels (ARRFMESSAGES).
ARRFELEMENTSSUPPORTEDVALUES	Stores the Case Management layout details for viewing transaction details.
ARRFELEMOPREGIONMAP	Stores the detailed mapping for all elements-to-operations that you can use while using the Rules Builder to create custom rules. In other words, this table stores the metadata used for organizing the Rule Builder screen.
ARRFEXCEPTIONUSER	Stores the list of users marked as Exception Users.
ARRFEXCPUSERHIST	Stores the history of all users who were marked as exception users.

Table Name	Description
ARRFINSTANCEAUDITLOG	Stores all details related to all instances configured in the system along with all activities (such as restart, update, refresh, and shutdown) that were performed on the instance. In other words, this table stores the audit trail of all management activities for each instance in the system.
ARRFINSTANCES	Stores the details of all server instances configured in the system. These instances can either be RiskMinder Server instances or Case Management Queuing Server instances.
ARRFIPCONTEXT	Stores the IP context that is used by the IP velocity rule. **Note:** This table is for future use.
ARRFLIBRARYTOTYPEMAPPING	Stores the mapping of all supported custom rule types with the corresponding library name. **Note:** This table is for future use.
ARRFLOCALE	Stores information related to all supported locales.
ARRFMESSAGES	Stores the Response and Reason Codes messages.
ARRFNEGATIVECOUNTRYLIST	Stores the list of all negative countries.
ARRFOPERATORS	Stores the list of all operators (used for creating rules by using Rule Builder) supported by RiskMinder.
ARRFORGCHANNEL	Stores the list of all supported Channels for each organization.
ARRFORGQUEUES	Stores the list and basic details of all queues that belong to an organization and Channel.
ARRFOTHERELEMENTS	Stores detailed information for all non-channel elements (such as system time) using which you can write custom rules. In other words, this table stores the list of elements that are not passed during a transaction, but are used or displayed in the Rule Builder screen.
ARRFPROTOCOLREGISTRY	Stores configuration of each listener port of RiskMinder Server.
ARRFQUEUEADMIN	Stores the Queue-to-Administrator mapping details.
ARRFRULEDEPENDENCY	Stores the details of what other rules a rule is dependent on.

Table Name	Description
ARRFSERVERS	Stores the mapping of available RiskMinder Server instances.
ARRFSITES	Stores site details for each de-tokenization service. **Note:** Currently, this table is applicable only for TransFort-RiskMinder integration deployments. This table will soon be deprecated.
ARRFSYSAUDITLOG	Stores all details related to all transactions (risk evaluation and other activities) that are logged. If you configure additional Channels for your deployment, then corresponding tables are created and named with the Channel name appended to the default table name, for example, ARRFSYSAUDITLOG_<*channel_name*>.
ARRFSYSORGCONFIG	Stores all versions of configurations available for all organizations in the system. **Note:** This table stores both history and the changes made by the administrator.
ARRFSYSPARAMSCONFIG	Contains detailed information about all RiskMinder system parameters that are configurable by using Administration Console. **Note:** This table stores both history and the changes that are made by the administrator.
ARRFSYSRULEEXECCONFIG	Stores the configuration information for all rules. This information includes version and configuration for each rule. **Note:** This table stores both history and the changes that are made by the administrator.
ARRFSYSTERMRULESCORECONFIG	Stores configuration information for each rule and the corresponding result that impacts the risk score.
ARRFTRUSTEDIPLIST	Stores the information for all trusted aggregators, IP addresses, and ranges.
ARRFTXNTYPE	Stores the master list of all transaction types supported in the system.
ARRFUAOSLIST	Stores the master list of all User-Agent OS strings-to-actual-Operating System-and-Version mappings. This information is used for logical upgrades for Windows.
ARRFUNTRUSTEDIPLIST	Stores the details of all negative IP addresses.

Table Name	Description
ARRFUNTRUSTEDIPLIST_ ARCHIVE	Stores the archived information for the ARRFUNTRUSTEDIPLIST table. In other words, this table serves as an archive of details related to all deleted negative IP addresses.
ARRFUNTRUSTEDIPTYPE	Stores the mapping for all supported negative IP types.
ARRFUPLOADAUDITLOG	Stores the details of the operations performed on the GeoPoint and GeoAnonymizer tables.
ARRFUSERCONTEXT	Stores the context information (such as user status, timestamp of the transaction, and the requested action) for each incoming transaction from a user. **Note:** This information is used for User Velocity checks.
ARRFUSERCONTEXT_ARCHIVE	Stores the archived information for the ARRFUSERCONTEXT table. In other words, this table serves as an archive for user context information for deleted users.

Used by Administration Console

The following table lists all database tables that are used by Administration Console.

Table Name	Description
ARADMINAUDITTRAIL	Stores administrator activity audit.
ARADMINAUTHTOKEN	Stores the tokens that Administration Console uses for pluggable authentication. Every time you log in to Administration Console by using your password, a token is internally generated after password match and stored in this table.
ARADMINBASICAUTHPWDHISTORY	Stores the last *n* occurrences of password of all administrators in all organizations that use Basic Authentication (for administrators) to log in to 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 Administration Console.

Table Name	Description
ARADMINCONFIG	Stores Administration Console configurations.
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 RiskMinder 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 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 *all* the existing organizations in the system.
ARADMINSUPPORTEDAUTHMECH	Stores the information about all supported authentication mechanisms to log in to Administration Console.
ARADMINSUPPORTEDTIMEZONE	Stores the list of all available time zones that do not change after you install RiskMinder or any other dependent product. **Note:** This table is an internal table.
ARADMINTURNEDOFFPRIVILEGE	Stores the list of all privileges that are not available for the given custom role.
ARADMINTXID	Stores information required to generate a unique ID for each transaction.
ARADMINUITAB	Stores information about the tabs that are available and the order in which they are available in Administration Console.

Table Name	Description
ARADMINUITASK	Stores information about all the tasks that are available and the order in which they are available through Administration Console.
ARADMINUITASKATTRIBUTES	Stores details of the tasks that are displayed, when the first-level and the second-level tabs in Administration Console are clicked. These tasks are referred to as landing pages.
ARADMINUITASKCONTAINER	Stores information related to available *task containers*. A task container can either be a second-level tab ID or the task group in Administration Console.
ARADMINUSER	Stores detailed information (such as organization to which they belong, current status, timezone, locale, last login time) of all existing administrators.
ARADMINUSER_ARCHIVE	Stores information related to all deleted users.
ARADMINWIZARDTASK	Stores 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 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 RiskMinder-specific information or its dependent products' information that Administration Console uses for rendering the console pages.
ARPFCMNCACHEREFRESHEVENT	Stores details of all cache refresh events for all instances in the system.
ARPFCMNCACHEREFRESHSCOPE	Stores information about all organizations that will be affected if a server cache refresh event occurs.
ARPFCMNCACHEREFRESHSTATUS	Stores the status of each cache refresh event for every instance for which it was triggered.
ARPFCMNINSTANCE	Stores detailed information for all RiskMinder Server instances configured in the system. This also includes the last time the instance was refreshed.
ARPFCMNORGCONFIGDATA	Stores configuration details for each organization. This includes global configurations that can be, typically, overridden at the organization-level.
ARPFCMNORGCONFIGSTATE	Stores the status of each assigned configuration from the ARPFCMNORGCONFIGDATA table.
ARPFCMNPRIVILEGEMAPPING	Stores the details of each privilege available through Administration Console.
ARSEQUENCETABLE	Used only by MS SQL Server, this table simulates sequences using stored procedures.
ARREPORTTABLES	Contains the metadata of other Administration Console and UDS tables.

Used by User Data Service (UDS)

The following table explains the database tables that are used by UDS.

Table Name	Description
ARCMNKEY	Stores all global-level and organization-level key labels.
ARUDSACCOUNTTYPE	Stores 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 configuration information (SDK method to be invoked), for Callouts when a delete event with cascade effect is triggered or enabled.
ARUDSCALLOUTINTERNALPARAMS	Stores details, such as parameters and types specific to internal Callouts.
ARUDSCALLOUTPARAM	Stores details, such as parameters and types specific to external Callouts.
ARUDSCONFIG	Stores UDS configuration parameters and their values.
ARUDSCONFIGAUDITLOG	Stores audit log information for the User Data Source (UDS) operations and their return status.
ARUDSCONTACTTYPE	Stores additional contact information (such as secondary email and telephone number) that can be configured at the organization or global level.
ARUDSCUSTOMATTREXT	Stores 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 archived information related to user account custom attributes when a user account is deleted.
ARUDSLDAPREPOSITORYCONFIG	Stores 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 organization-specific repository mapping information. For example, if you are using LDAP as the user repository, then a RiskMinder 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 Personal Identification Information (PII) fields. **Note:** You can also configure these attributes by using Administration Console.
ARUDSREPOCLONESTATUS	Stores status of temporary cloning of user information from an external repository (such as LDAP) to the ARUDSREPOSITORYUSER table.
ARUDSREPOSITORYTYPES	Stores 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 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 resource-to-organization mapping. However, it is different from the ARUDSRESOURCESCOPE table, because it specifies which resource is applicable for *all* organizations.
ARUDSSECUREATTRIBUTES	Stores information related to all attributes (such as fields that store PII) that need to be encrypted. **Note:** You can also configure these attributes by using 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.

Table Name	Description
ARUDSUSERACCOUNT	Stores user account information for specific users.
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 secondary contact information (such as email or telephone numbers) for users.
ARUDSUSERCONTACT_ARCHIVE	Stores 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 database administrators to calculate the approximate size of the database that must be set up for RiskMinder.

Denotations Used in Sample Calculations

The following denotations are used in the sample calculation:

- Number of users = *N*
- Average number of devices per user = *O*
- Average number of user-device associations = *A*
- Average number of transactions per day = *T*
- Number of entries in the Quova Data Feed = *Q*
- Computation time frame (in days) = *D*

Value Assumptions Made

The following assumptions have been made for calculation purposes:

- Number of users (**N**) = 1,000,000 (one million)
- Average number of devices per user (**O**) = 2
- Average number of user-device associations (**A**) = 2
- Average number of transactions per day (**T**) = 24,000
- Number of entries in the Quova Data Feed (**Q**) = 10,000,000 (ten million)
- Computation time frames (**D**) = 90 days

Sample Calculations Based on Assumptions Made

Considering the figures that were assumed in the preceding section, the final requirement must be:

- Based on the **total number of users**, the database size = (10 * *N*) KB

  In this calculation, the value 10 KB per user has been arrived at as follows:

  - **ARRFUSERCONTEXT:** 3 KB per record
  - **ARUDSUSER:** 3.5 KB per record
  - **ARUDSAUDITLOG:** 3 KB per record
- Based on the **total number of devices**, the database size = (6 * *O* * *N*) KB

  In this calculation, the value 6 KB per user has been arrived at as follows:

  - **ARRFDEVICECONTEXT:** 2 KB per record
  - **ARRFDEVICEINFO:** 4 KB per record

  In this calculation, based on the assumption that were made in the previous section:

  - **O**: 2
- Based on the **total number of user-device associations**, the database size = (5 * *A* * *N*) KB

  In this calculation, the value 5 KB per user has been arrived at as follows:

  - **DEVICEUSERASSOCIATION:** 1 KB per record
  - **DEVICEINFO:** 4 KB per record

  In this calculation, based on the assumption that were made in the previous section:

  - **A**: 2
- Based on the **daily activity,** the database size = (*T* * *D* * 20) KB
- Based on the **size of Quova Data Feed,** the database size = (*Q* * 2) KB

Database Tables Replication Advice

This section provides information about how frequently the tables must be replicated between the primary and the backup databases. This section covers the following topics:

- Tables That Need Real-Time Synchronization (see page 237)
- Tables That Need Periodic Synchronization (see page 239)
- Tables That Do Not Need Synchronization (see page 241)

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. As this data is required for authentication, you must perform real-time synchronization of these tables.

Component	Table
Administration Console	ARADMINAUDITTRAIL
	ARADMINBASICAUTHUSER
	ARADMINSCOPE
	ARADMINSCOPEALL
	ARADMINUSER
	ARSEQUENCETABLE
	ARADMINTXID
UDS	ARCMNKEY
	ARUDSORGANIZATION
	ARUDSORGREPOATTRIBUTES
	ARUDSORGSECUREATTRIBUTES
	ARUDSLDAPREPOSITORYCONFIG
	ARUDSACCOUNTTYPE
	ARUDSRESOURCESCOPE
	ARUDSRESOURCESCOPEALL
	ARUDSATTRMAP
	ARUDSCONTACTTYPE
	ARUDSUSER
	ARUDSUSERACCOUNT
	ARUDSCUSTOMATTREXT
	ARUDSAUTHSESSION
	ARUDSUSERCONTACT
	ARUDSREPOSITORYUSER
	ARPFCMNINSTANCE

Component	Table
RiskFort	ARRF_CMA
	ARRF_IMA
	ARRF_CASE_TXN
	ARRFCURRENTCMSCHEDULE
	ARRFADDONRULELISTDATA
	ARRFADDONRULEMAPPINGDATA
	ARRFCASEAUDITLOG
	ARRFCLIENTSSLROOTCAS
	ARRFCURRENTORGCONFIG
	ARRFDATAVERSIONMAPPING
	ARRFDEVICECONTEXT
	ARRFDEVICEINFO
	ARRFDEVUSERASSO
	ARRFEXCEPTIONUSER
	ARRFINSTANCEAUDITLOG
	ARRFINSTANCES
	ARRFIPCONTEXT
	ARRFNEGATIVECOUNTRYLIST
RiskFort	ARRFSYSPARAMSCONFIG
	ARUDSAUDITLOG
	ARRFSYSAUDITLOG
	ARRFSYSORGCONFIG
	ARRFSYSRULEEXECCONFIG
	ARRFSYSTERMRULESCORECONFIG
	ARRFTRUSTEDIPLIST
	ARRFUNTRUSTEDIPLIST
	ARRFUSERCONTEXT
	ARRFORGQUEUES
	ARRFQUEUEADMIN

Component	Table
	ARRFUPLOADAUDITLOG
	ARRFCASEQUEUES

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
	ARADMINCUSTOMROLE
	ARADMINMAP
	ARADMINPAFCONFIG
	ARADMINPWDPOLICY
	ARADMINBASICAUTHPWDHISTORY
Administration Console	ARADMINTURNEDOFFPRIVILEGE
	ARADMINCACHEREFRESH
	ARADMINAUDITTRAIL
	ARADMINUSER_ARCHIVE
	ARADMINMANAGEROLE
	ARADMINROLEPRIVILEGE
	ARPFCMNORGCONFIGDATA
	ARPFCMNORGCONFIGSTATE
	ARPFCMNCACHEREFRESHSTATUS
	ARPFCMNCACHEREFRESHEVENT
	ARPFCMNCACHEREFRESHSCOPE
	ARUDSUSERAUDITLOG
	ARUDSORGANIZATIONAUDITLOG
	ARUDSCONFIGAUDITLOG
	ARUDSCONFIG

UDS	ARUDSREPOSITORYTYPES
	ARUDSUSERATTRIBUTE
	ARUDSUSERACCOUNT_ARCHIVE
	ARUDSCUSTOMATTREXT_ARCHIVE
	ARUDSUSER_ARCHIVE
	ARUDSUSERCONTACT_ARCHIVE
	ARCMNCONFIG
	ARUDSREPOCLONESTATUS
	ARUDSCALLOUTINTERNAL
	ARUDSCALLOUTINTERNALPARAMS
	ARUDSCALLOUT
	ARUDSCALLOUTPARAM
	ARCMNBULKTASKPARAM
	ARCMNBULKUPLOADTASK
	ARCMNBULKREQUEST
	ARCMNBULKOPERATIONATTRIBUTE
	ARCMNBULKOPERATION
	ARRFCHANNEL
	ARRFCHANNELDETAILCATEGORY
	ARRFCHANNELELEMENTS
	ARUDSUSERATTRIBUTE
	ARQGEOANONYMIZER1
	ARQGEOANONYMIZER2
	ARQGEOPOINT1
	ARQGEOPOINT2
	ARQUOVAVERSION
	ARRFADDONRULETYPE
	ARRFADVICECONFIG
	ARRFBASECHANNELELEMENTS
	ARRFBUCKETELEMENTCONFIG

RiskFort	ARRFBUCKETCONFIG
	ARRFCONFIGURATION
	ARRFCOUNTRY
	ARRFCOUNTRYLIST
	ARRFCHANNELMSGPROPERTIES
	ARRFCHANNELTXNTYPE
	ARRFCHANNELTXNTYPEELEMENTS
	ARRFCLIENTCERTSANDKEYS
	ARRFCONFIGURATION
	ARRFCURRCONVRATES
	ARRFDEVICEINFOHIST
	ARRFELEMOPREGIONMAP
	ARRFELEMENTSSUPPORTEDVALUES
	ARRFEXCPUSERHIST
	ARRFLIBRARYTOTYPEMAPPING
	ARRFOPERATORS
	ARRFOTHERELEMENTS
	ARRFORGCHANNEL
	ARRFPROTOCOLREGISTRY
	ARRFSERVERS
	ARRFSITES
	ARRFTXNTYPE
	ARRFUNTRUSTEDIPTYPE
	ARRFUSERCONTEXT_ARCHIVE

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**
	ARADMINAUTHTOKEN
	ARCMNDBERRORCODES

Component	Table
Administration Console	ARADMINPREDEFINEDROLE
	ARADMINSUPPORTEDAUTHMECH
Administration Console	ARADMINUITAB
	ARADMINUITASK
	ARADMINUITASKATTRIBUTES
	ARADMINUITASKCONTAINER
	ARADMINWIZARDTASK
	ARREPORTTABLES
	ARCMNMAPDATATYPE
	ARCMNCACHEREFRESH
	ARCMNMAPDATATYPE
	ARPFCMNPRIVILEGEMAPPING
	ARADMINSUPPORTEDTIMEZONE
UDS	ARUDSSECUREATTRIBUTES
RiskFort	ARRFADVICECODE
	ARRFADDONEXPOSEDPARAMS
	ARRFCOUNTRYLIST
	ARRFCURRENCY
	ARRFDBERRORCODES
	ARRFDISPLAYNAMES
	ARRFLOCALE
	ARRFMESSAGES

Database Tables Archival Recommendations

This section walks you through the recommendations for:

- Tables that Grow Rapidly (see page 244)
- Tables that Grow Moderately (see page 245)

Important! It is recommended that you only trim the tables that capture transaction details, such as audit log information. Do *not* trim tables that capture user information, which is necessary to assess the risk evaluation.

Tables that Grow Rapidly

The following table types grow rapidly with every transaction, and must be archived or purged according to the archival policy of your organization:

- Tables that store **audit data**, such as:
  - ARADMINAUDITLOG
  - ARADMINAUDITTRAIL
  - ARRFINSTANCEAUDITLOG
  - ARRFUPLOADAUDITLOG
  - ARUDSAUDITLOG
- Tables that store **transaction data**, such as:
  - ARRFCASEAUDITLOG
  - ARRFSYSAUDITLOG
  - ARRFSYSAUDITLOG_<*channel*>
  - ARRF_CASE_TXN
  - ARRF_CASE_TXN_<*channel*>
  - ARRFUSERCONTEXT
- Tables that store **reports data** and **device data**, such as:
  - ARREPORTS
  - ARRFDEVICECONTEXT
  - ARRFDEVICEINFO
  - ARRFDEVUSERASSO
  - ARRFUSERCONTEXT
  - ARRF_IMA
  - ARRF_CMA
- Tables that store **configuration data**, such as:
  - ARRFCURRENTCMSCHEDULE
  - ARRFADVICECONFIG
  - ARRFCURRENTORGCONFIG
  - ARRFSYSORGCONFIG

When you archive data in these categories of rapidly growing tables, the following procedure is recommended:

1. Archive from the backup database. This action does not affect transactions, but only reports.

2. Clean up the backup database.
3. To use the backup database, failover the servers.
4. Clean up the primary database.
5. Revert to the primary database.

Tables that Grow Moderately

The following tables grow moderately, depending on how the Case Management feature is used. Therefore, these tables can be archived or purged at a lower frequency according to the archival policy of your organization:

- ARRFCASES
- ARUDSUSER

Note: Each entry in ARUDSUSER represents an enrolled user. User record enters in to this table through the User Management Web service. However, in some cases, you can choose to archive user data in the ARUDSUSER table. For example, you may want to archive information for users who have not accessed the application for a specified duration. In such cases, you can treat the returning user as a new user and can provide risk scores consistent with that classification.
If your organization is interested in such optimizations, then it is recommended that you work with the CA Support team for the same.

Database Connection Tuning Parameters

The parameters that you can use to tune the connection between RiskMinder Server and the database are configured by using the Instance Management page in Administration Console. To access this console page, you must be logged in as Master Administrator (MA). The following table lists the (common) parameters that you can use to tune the connection between RiskMinder Server and the database.

Field	Description
Minimum Connections	The minimum number of connections to initially create between the RiskMinder Server and the database.
Maximum Connections	The maximum number of connections that can be created between the RiskMinder Server and the database. **Note:** Set this value depending on the maximum connections that the database supports, because this value overrides the MaxConnections parameter. See your database vendor documentation for more information.

Field	Description
Increment Connections By	The number of connections that will be added to the existing connections, when the need arises. The total number of connections cannot exceed the maximum number of connections.
Monitor Thread Sleep Time (in Seconds)	The amount of time the monitoring thread sleeps between heartbeat checks on all the databases.
Monitor Thread Sleep Time in Fault Conditions (in Seconds)	The interval at which the database monitor thread checks the health of the connection pool in case of faulty database connections.
Log Query Details	Enables you to log all the database queries.
Monitor Database Connectivity	The option to enable checking of the pools proactively in the database monitor thread.
Auto-Revert to Primary	Enables the server to switch from the backup to primary database when the primary database becomes functional.

Book: See "Managing RiskMinder Server Instances" in the *CA RiskMinder Administration Guide* for detailed information about configuring the database parameters.

Appendix E: Configuring CA RiskMinder for Oracle RAC

Perform the steps in this section if you want to use Oracle RAC with RiskMinder 3.1.01.

This section contains the following topics:

Updating the arcot-db-config-for-common-2.0.sql Script

You run database scripts as a post-installation task in the RiskMinder installation procedure. The arcot-db-config-for-common-2.0.sql script is one of the database scripts that you run. Before you run this script, modify it for Oracle RAC.

Follow these steps:

1. To determine the Oracle RAC shared datafile path, log in to the database and run the following command:

   ```
   SELECT file_name, tablespace_name FROM dba_data_files
   ```

   The following is sample output of this command:

   ```
   +DATA\qadb\datafile\users.259.797224649      USERS
   +DATA\qadb\datafile\undotbs1.258.797224649   UNDOTBS1
   +DATA\qadb\datafile\sysaux.257.797224647     SYSAUX
   ```

2. Open the arcot-db-config-for-common-2.0.sql file. This file is in the *install_location*\Arcot Systems\dbscripts\oracle\ directory.

3. Search for the following line in the file:

   ```
   filename varchar2(50) := 'tabspace_arreports_'|| to_char(current_timestamp,
   'YYYY-MM-DD-HH24-MI-SS') || '.dat';
   ```

4. Replace that line with the following line:

   ```
   filename varchar2(100) :=
   '+shared_location/service_name/datafile/tabspace_arreports_'||
   to_char(current_timestamp, 'YYYY-MM-DD-HH24-MI-SS') || '.dat';
   ```

   In the new line:

   - Replace shared_location with the shared datafile path that you determined by running the command given in the first step.
   - Replace service_name with the service name of the Oracle RAC installation.

   The following is a sample line:

   ```
   filename varchar2(100) := '+DATA/forwardinc/datafile/tabspace_arreports_'||
   to_char(current_timestamp, 'YYYY-MM-DD-HH24-MI-SS') || '.dat';
   ```

5. Save and close the script file, and then run it.

Updating the arcotcommon.ini File

The arcotcommon.ini file contains the parameters for database and instance settings. When you run the installer, database configuration data items that you enter on the installer screens are stored in this file. The JDBC URL of the database is one such data item. If you are using Oracle RAC, specify the JDBC URL in the format supported by Oracle RAC.

Follow these steps:

1. Open the arcotcommon.ini file in a text editor. This file is in the *install_location*\Arcot Systems\conf\ directory.
2. Specify a value for the URL parameter in the [arcot/db/primarydb] section and, if required, in the [arcot/db/backupdb] section of the INI file. Enter the URL in the following format:

   ```
   URL.1=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=host_name)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=service_name)(SERVER=DEDICATED)))
   ```

   For example:

   ```
   URL.1=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=172.30.250.18)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=forwardinc)(SERVER=DEDICATED)))
   ```

   Note: If Oracle RAC is client configured, then include all the nodes in this format.
3. If the database user that you specified while running the AuthMinder installer is different from the database user in Oracle RAC, then:
   a. Change the database user credentials in the arcotcommon.ini file.
   b. Use DBUtil to change the database user credentials in the securestore.enc file. DBUtil is available in the ARCOT_HOME\tools\win directory. Instructions on using DBUtil are given in Preparing for the Upgrade to Release 3.1.01 (see page 157).
4. Save and close the arcotcommon.ini file.

Updating the Database Connection Details

To establish a connection between RiskMinder and Oracle RAC, you must create an ORA file and define the address for connecting to the RAC.

Follow these steps:

1. Create a *.ora file on the system on which you have installed AuthMinder. For example, C:\Program Files (x86)\tns.ora.
2. Add the following lines in the new file:

   ```
   section_name =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = host_name_or_IP_address)(PORT = 1521))
       )
       (CONNECT_DATA =
         (SERVICE_NAME = service_name)
       )
     )
   ```

   For example:

   ```
   fwdincrac =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = 172.30.250.18)(PORT = 1521))
       )
       (CONNECT_DATA =
         (SERVICE_NAME = forwardinc)
       )
     )
   ```

   Note: If Oracle RAC is client configured, then include all the nodes in this format.
3. Save the file.
4. Modify the DSN that you created during the installation.
5. For the required DSN, clear all the parameters in the Standard Connection section.

   This makes the TNSNames Connection section editable.
6. Add the following parameters to this section:

   ```
   TNSNamesFile=ARCOT_HOME\ora_file_name
   ServerName=section_name
   ```

   For example:

   ```
   TNSNamesFile= C:\Program Files (x86)\tns.ora
   ServerName=fwdincrac
   ```

7. Save and close the file.

Appendix F: Default Port Numbers and URLs

This appendix lists the default port numbers and URLs that RiskMinder uses. It contains the following sections:

Default Port Numbers

During installation, the installer checks if the required default port number is in use. If the port number is not in use, then the installer assigns it to the RiskMinder component. However, if the default port number is already in use by a dependent product or by any other application, then you must specify the port number manually by using the RiskFort Protocol Setup screen in Administration Console.

The following table lists the default port numbers used by RiskMinder.

Protocol	Default Port Number	Description
RiskFort Server		
Native (TCP)	7680	This is a proprietary protocol used by RiskMinder for the purpose of risk evaluation. This port is used for communication between the RiskMinder Server instance and the RiskMinder Java SDKs (which include Risk Evaluation.)
Native (SSL)	7681	This is a proprietary protocol to enable SSL-based communication between the RiskMinder Server instance and the RiskMinder Java SDKs (which include Risk Evaluation.)

Protocol	Default Port Number	Description
Administration Web Service	7777	This protocol is used for communication between RiskMinder Server and Administration Web services, and is used to create and manage rule configurations. **Note:** These calls do *not* include the Risk Evaluation or User and Organization Management calls.
Transaction Web Service	7778	This protocol is used by the Risk Evaluation Web service to connect to the RiskMinder Server instance. **Note:** These calls do *not* include the Administration service calls.
Server Management	7980	This protocol is used by Administration Console to communicate with the RiskMinder Server instance for server management activities (such as, graceful shutdown, server cache refresh, instance management, and protocol management).
Case Management Queuing Server **Book:** See *CA RiskMinder Administration Guide* for detailed information on Case Management.		
Case Management Queueing Server	7779	This protocol is used by the Case Management Queuing Server module to listen to the Case Management requests (at the server end) on the specified port.
Case Management Queueing Administration	7780	This protocol is used by Administration Console to communicate with the Case Management Queuing Server instance for server management activities (such as, graceful shutdown, server cache refresh, instance management, and protocol management).

Important! If another service is already running on the default ports that RiskMinder uses, then you must set a new port for the protocols.
To set new port number for the protocols, use the **Protocol Configuration** page in Administration Console. See "Managing RiskMinder Server Instances" in *CA RiskMinder Administration Guide*.

URLs for RiskMinder Components

Use the URLs listed in the following table to access RiskMinder components after installation. The URLs in the table use the default ports.

Component or Service	URL
Administration Console (For Master Administrator (MA))	*http://<rf_hostname>:<rf_port>/arcotadmin/masteradminlogin.htm* **Note:** The port that you must specify here is your application server port.
Administration Console (For Other Administrators)	*http://<rf_hostname>:<rf_port>/arcotadmin/adminlogin.htm* **Note:** The port that you must specify here is your application server port.
Sample Application	*http://<rf_hostname>:<appserver_port>/riskfort-3.1.01-sample-application/index.jsp* **Note:** The port that you must specify here is your application server port.
Risk Evaluation Web Service	*http://<rf_hostname>:<rf_port>/services/RiskFortEvaluateRiskSvc* **Note:** The default port here is 7778.
RiskFort Administration Web Service	*http://<rf_hostname>:<rf_port>/services/ArcotRiskFortAdminSvc* **Note:** The default port here is 7777.
User Management Web Service	*http://<appserver_hostname>:<appserver_port>/arcotuds/services/ArcotUserRegistrySvc* **Note:** The port that you must specify here is your application server port where UDS has been deployed.

Component or Service	URL
Organization Management Web Service	*http://<appserver_hostname>:<appserver_port>/arcotuds/services/ArcotUserRegistryMgmtSvc* **Note:** The port that you must specify here is your application server port where UDS has been deployed.
Configuration Registry Web Service	*http://<appserver_hostname>:<appserver_port>/arcotuds/services/ArcotConfigRegistrySvc* **Note:** The port that you must specify here is your application server port where UDS has been deployed.

Appendix G: Configuring Application Server for 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 implementing database connection pooling, you can avoid the overhead of creating a new database connection every time a RiskMinder component deployed on your application server requires access to the database.

This appendix outlines the steps for:

- Enabling Database Connection Pooling (see page 257)
- Enabling LDAP Connection Pooling (see page 265)
- Enabling Apache Tomcat Security Manager (see page 270)

Enabling Database Connection Pooling

This section quickly walks you through the steps to set up database connection pooling on the application server, where you have deployed RiskMinder components. The configuration steps for the following supported application servers are covered:

- Apache Tomcat (see page 258)
- IBM WebSphere (see page 261)
- Oracle WebLogic (see page 263)
- JBoss Application Server (see page 264)

Apache Tomcat

This section provides the steps to enable Apache Tomcat for JNDI-based database operations. To create a JNDI connection in Apache Tomcat:

1. Install the Apache Tomcat application server and test the installation by using the following URL:
   http://localhost:8080/
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 RiskMinder.

     Important! This name *must* match with the AppServerConnection PoolName.*N* in arcotcommon.ini (see page 198) (*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 the Oracle driver, then URL will be:
     `jdbc:oracle:thin:<server>:<database_port>:<sid>`
4. Add the following entry to define the data source 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 available 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. To enable database connection pooling, download the following files from the corresponding third-party source. Then, copy these files to the *<TOMCAT_HOME>*\common\lib\ folder (on Apache Tomcat 5.x) or the *<TOMCAT_HOME>*\lib\ directory (on Apache Tomcat 6.x and 7.x).
   - commons-dbcp-1.2.2.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)
   - mysql-connector-java-5.1.22-bin.jar (for MySQL database)

Configuration changes for Tomcat8 (apache-tomcat-8.0.24) and JDK8 (1.8.0_51)

1. Create JNDI connection

   There are few configuration attribute names that have been updated in tomcat 8. Couple of them are listed in the sample here. Verify your attribute names from tomcat 8 documentation if you are using any others which are not shown in the sample.

   <Resource name="< data source_name >" auth="Container"

   type="javax.sql.DataSource" username="USER_ID" password="PASSWORD"

   driverClassName="JDBC_Driver_Class " url="JDBC_url" maxWaitMillis="30000"

   maxTotal="32" maxIdle="4" initialSize="4"

   timeBetweenEvictionRunsMillis="600000"

   minEvictableIdleTimeMillis="600000"/>

   Note :

   The **maxActive** configuration option has been renamed to **maxTotal**

   The **maxWait** configuration option has been renamed to **maxWaitMillis**
2. Make sure you use the latest database jar.
   - For sql server - sqljdbc4.jar
   - For oracle ojdbc6.jar
3. Enable SSLv3

   Java 8 disables SSLv3 by default. To enable it follow these steps:

   1. Go to “<JRE_HOME>\lib\security” folder used by tomcat.
   2. Open java.security file.

3. Check if the property “jdk.tls.disabledAlgorithms” has SSLv3, if yes remove the SSLv3 value.

IBM WebSphere

This section provides the steps to enable IBM WebSphere for JNDI-based database operations. To configure an IBM WebSphere instance for deploying Java-dependent components of RiskMinder:

1. Log in to WebSphere Administration Console.
2. Select Resources and expand the JDBC node.
3. Click JDBC Providers.

   The JDBC Providers page appears.
4. In the Preferences section, click New to create an appropriate JDBC provider based on the database that you are using.

   The Create a new JDBC Provider page appears.
5. Perform the following tasks to create the JDBC Provider.

   Note: For more information, refer to: *http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.base.iseries.doc/info/iseries/ae/tdat_ccrtprov.html*
6. Specify the Database Type and Provider Type.

   Select Connection pool data source from the Implementation Type drop-down list.

   a. Enter a Name for the JDBC provider. You can also enter a Description for the JDBC Provider.

   b. Click Next.

      The Enter database class path information screen appears.

   c. Enter the absolute path for the JAR file.

   d. Click Next.

      The Summary screen appears.

   e. After reviewing the summary of the information that you have entered, click Finish to complete the JDBC provider configuration.
7. 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.
8. 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.

Note: This name *must* match with the value of AppServerConnection PoolName.*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 MySQL:**
  `jdbc:mysql://<server>:<port-number>/<database>`

i. Click Next.

The Setup Security aliases screen appears.

j. Click Next to view the Summary screen, and then click Finish.

9. Click the data source created in Step 8.
10. In the Related Items section, click JAAS - J2C authentication data.
11. Click New to create a new credential.
12. Enter login credentials that are used to connect to the database and save the credential.
13. Click Apply, and then click OK to save the changes made.
14. Click Data Sources and select the data source that you created in Step 8.
15. Under Security Settings -> Component-managed authentication alias, select the JAAS credential that you created in Step 12 and click Apply, and then OK.
16. Click Data Sources and select the check box for the data source you created in Step 8.
17. 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 walks you through the steps to enable Oracle WebLogic for JNDI-based database operations. To create a data source for RiskMinder in Oracle WebLogic:

1. Log in to WebLogic Administration Console.
2. Click the Lock & Edit button in the Change Center, if it is not already done.
3. Navigate to Services, JDBC, and the Data Sources.
4. Under JDBC, open Data Sources and click New to open the Create a New JDBC Data Source page.
5. Set the following JNDI and the database information:
    a. Set Name = ArcotDB

       Note: This name *must* match with the value of AppServerConnection PoolName.*N* in arcotcommon.ini.

    b. Set JNDI Name = ArcotDB
    c. Select the required Database Type, for example Oracle.
    d. Select the required Database Driver, for example Oracle Thin Driver.
6. Click Next, retain the default values and click Next again.
7. In the Connection Properties page that appears, set the database connection details. For example, the values for Oracle can be:
    - **Database Name** = SID or service name of the database server
    - **Host Name** = Host name or the IP address of the database server
    - **Port** = 1521 or any other port the database server is running
    - **Database User Name** = Database account user name that can create the database connections
    - **Password** / **Confirm Password** = Password for the specified Database User Name
8. Click Next.
9. Click Test Configuration to verify the database information that you specified.
10. Click Next and set the preferred data source target server for the WebLogic server instance.
11. Click Finish to return to the data source list page.
12. Click the Activate button in the Change Center to enable the data source settings that you configured in the preceding steps.

JBoss Application Server

This section walks you through the steps to enable JBoss Application Server for JNDI-based database operations. To create a data source for RiskMinder in JBoss Application Server:

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 RiskMinder components. This name must match with the AppServerConnection PoolName.*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 the URL will be: jdbc:oracle:thin:*<server>*:*<database_port>*:*<sid>*.
   - **Exception Sorter Class:** The class for implementing 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

It covers the configuration steps for the following application servers:

- Apache Tomcat (see page 265)
- IBM WebSphere (see page 266)
- Oracle WebLogic (see page 267)
- JBoss Application Server (see page 269)

Apache Tomcat

To create a LDAP connection pool:

1. Install the Apache Tomcat application server and test the installation by 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.prefsize=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

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.cmd`
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.prefsize=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:

```
@REM START WEBLOGIC
echo starting weblogic with Java version:
%JAVA_HOME%\bin\java %JAVA_VM% -version
if "%WLS_REDIRECT_LOG%"=="" (
echo Starting WLS with line:
echo %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%
%JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%
-Dcom.sun.jndi.ldap.connect.pool.maxsize=64
-Dcom.sun.jndi.ldap.connect.pool.prefsize=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
-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
)
```

5. Save and 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 space-separated JVM options:
   - -Dcom.sun.jndi.ldap.connect.pool.maxsize=64
   - -Dcom.sun.jndi.ldap.connect.pool.prefsize=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 Application Server

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.prefsize=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 AS.

Enabling Apache Tomcat Security Manager

If you notice that RiskMinder does not work on Apache Tomcat if the Java **Security Manager** is enabled, then to enable Tomcat Security Manager to work with RiskMinder:

1. Navigate to the following Apache Tomcat installation location:
   <*Tomcat_Home*>\bin\
2. Double-click the **tomcat<version>w.exe** file.

   The Apache Tomcat Properties dialog box appears.
3. Activate the **Java** tab.
4. In the **Java Options** section, add the following entries:
   - -Djava.security.manager
   - -Djava.security.policy=<*Tomcat_Home*>\conf\catalina.policy
5. Click **Apply** to save the changes.
6. Click **OK** to close the Apache Tomcat Properties dialog box.
7. Navigate to the following Apache Tomcat location:
   <*Tomcat_Home*>\conf\
8. Open the catalina.policy file in a text editor of your choice.
9. 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";
};
```

10. 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;
};
```

11. Save and close the file.
12. Restart Apache Tomcat.

Appendix H: Deploying Administration Console on IBM WebSphere 7.0

If you plan to deploy Administration Console on IBM WebSphere 7.0, you might see an HTTPCLIENT error when you access some Administration Console pages, such as Instance Management. In such cases, you must perform the following steps:

1. Access the Administration Console WAR file from *<install_location>*\Arcot Systems\java\webapps\.
2. Copy arcotadmin.war to a temporary directory, say C:\Arcot_temp\.
3. Extract the arcotadmin.war file contents.

   Of the JARs that are extracted to the C:\Arcot_temp\arcotadmin\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
4. Log in to WebSphere Administration Console.

5. 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 or 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:
       `C:/Arcot_temp/arcotadmin/WEB-INF/lib/axiom-api-1.2.10.jar`
    e. Click Apply to save the changes made.
6. Navigate to the location (<*install_location*>\Arcot Systems\java\webapps\) where the Administration Console WAR file is located.
7. Deploy arcotadmin.war in the application server.
8. 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.
9. 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.
10. Ensure that the application is restarted.

Chapter 9: Adding Custom Actions

Each channel in RiskMinder has a set of actions associated with it. An action, in turn, has data elements associated with it. A rule in RiskMinder is a specific combination of the elements associated with an action for a channel or set of channels.

This section describes the procedure to add a custom action. While adding a custom action, you specify the channel with which the action must be associated. The elements that are associated with the other actions defined for that channel are automatically associated with the new action. You can use these elements to build rules for the new action.

Note: If you plan to build rules for actions that are available in all channels, then you must first add the action in each channel.

Follow these steps:

1. (Optional) Perform the following steps if you do not know the name of the channel with which you want to associate a new action:
   a. Log in to the Administration Console as a GA.
   b. Click the Services and Server Configurations tab.
   c. Under the Rules Management section on the side-bar menu, click the Rules and Scoring Management link.

      The Rules and Scoring Management page opens.
   d. Select any ruleset from the Select a Ruleset list.
   e. Click Add a New Rule.

      The RiskFort Rule Builder page opens.
   f. Note down the name of the channels for which you want to add new actions.
2. Ensure that you have the database privileges listed in Configuring the Database Server.
3. Log in to the database.
4. Run the following command to determine the ID of the channel to which you want to add the action:

   ```
   select channelid from arrfchannel where
   channelname='<channel-name>';
   ```

   In this command, replace *<channel-name>* with the name of the channel.

5. Run one of the following commands:

   Note: In the command that you run, replace *<channel-id>* with the channel ID that you determine by running the previous step. Similarly, replace *<action-name>* with the name of the channel. The action name can contain alphanumeric characters and the underscore character. No other character can be used in the action name.

   - For MS SQL Server:
     `EXEC ADD_CUSTOM_ACTION <channel-id>, '<action-name>'`
   - For Oracle Database:
     ```
     set serveroutput on;
     execute ADD_CUSTOM_ACTION(<channel-id>, '<action-name>');
     ```
   - For MySQL:
     `call ADD_CUSTOM_ACTION(<channel-id>, '<action-name>');`

   The following message appears:
   `"New action added successfully for the given channel."`

   The new action is added in the database.

6. Refresh cache. See the *CA RiskMinder Administration Guide* for instructions.
7. Verify that the action has been successfully added as follows:
   a. Log in to the administration console.
   b. Navigate to the Rules and Scoring Management screen.
   c. Click Add a new rule.
   d. Check whether the newly added action is displayed in the Actions list.

   After you verify that the action has been added, you can start using it to build new rules.

Appendix I: Troubleshooting RiskMinder Errors

This appendix describes the troubleshooting steps, which will help you resolve the errors that you might face while using RiskMinder. The troubleshooting topics are classified based on different RiskMinder components, as follows:

- Installation Errors (see page 279)
- Database-Related Errors (see page 283)
- RiskMinder Server Errors (see page 287)
- SDK Errors (see page 288)
- Upgrade Errors (see page 289)

Before you perform any troubleshooting tasks, check the RiskMinder log files to see if there were any errors. By default, all the log files are saved in the *<install_location>*\Arcot Systems\logs\ directory. The following table lists the default log file names of the RiskMinder components.

RiskFort Component	File Name	Description
RiskFort Server	arcotriskfortstartup.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 RiskMinder service does not start up.
	arcotriskfort.log	This file records all requests processed by the RiskMinder Server after its startup.
Case Management Server	arcotriskfortcasemgmtserver.log	This file records all the start-up (or boot) actions for Case Management. The information in this file is very useful in identifying the source of the problems if the Case Management service does not start up.
	arcotriskfortcasemgmtserverstartup.log	This file records all requests processed by the Case Management Server after its startup.
Administration Console	arcotadmin.log	This file records the Administration Console operations.

RiskFort Component	File Name	Description
User Data Service	arcotuds.log	This file records the User Data Service (UDS) operations.

Book: Refer to appendix, "RiskMinder Logging" in *CA RiskMinder Administration Guide* detailed information on these log files.

Installation Errors

Problem:

I cannot find arcotadmin.war in *<install_location>*\Arcot Systems\java\webapps directory.

Cause:

The installer failed to create the arcotadmin.war WAR file during installation.

Solution:

If the file was not automatically created, you must manually create it. To do so:

1. Open the command prompt window.
2. Ensure that the ARCOT_HOME environment variable is set.
3. Navigate to *<install_location>*\Arcot Systems\tools\common\bundlemanager directory.
4. Run bundlemanager as follows:
   ```
   java -jar bundle-manager.jar
   ```
   The preceding command generates the arcotadmin.war file in *<install_location>*\Arcot Systems\java\webapps directory.

Problem:

I cannot start the Risk Authentication Server (Risk Authentication **Service**). I see the following error in arcotriskfortstartup.log:

```
Failed DBPoolManager initialization
```

or

```
Datasource Name Not Found
```

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 arcotcommon.ini. 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. On Windows, you will find the 32-bit version at C:\Windows\SysWOW64.

Note: For detailed information on arcotcommon.ini and other configuration files, see Configuration Files and Options (see page 195).

Problem:

I cannot start the Risk Authentication Server (Risk 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 successfully created.

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.
   - If you found the entry, then clean up the database (See "Dropping Risk Authentication Schema") and reseed the database, as described in Section, "Running Database Scripts".
2. Restart the Risk Authentication Server.

Problem:

When I launch the Administration Console for the first time ("Logging In to Administration Console" (see page 84)) 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:

- ArcotAccessKeyProvider.dll
- arcot-crypto-util.jar

Solution:

Do the following:

1. Ensure that the PATH variable includes the absolute path to the following files:
   - ArcotAccessKeyProvider.dll
   - arcot-crypto-util.jar

   Depending on the type of deployment, see one of the following sections for information about the location of these files:
   - For a single system deployment, see Step 2: Copying Database Access Files to Your Application Server (see page 76).
   - For a distributed system deployment, see Step 2: Copying Database Access Files to Your Application Server (see page 115).
2. Restart the application server.

Problem:

I do not see the log file (arcotadmin.log, arcotuds.log, casemanagementserver.log, or riskfortserver.log) in the logs directory in ARCOT_HOME.

Cause:

Some of the probable causes for this issue might be:

- ARCOT_HOME might not be correctly set 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 Systems\.

  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 Systems\.

- Ensure that you copy the ArcotAccessKeyProvider.dll and arcot-crypto-util.jar files in the application server JAVA HOME location.

  Depending on the type of deployment, see one of the following sections for information about the location of these files:

  - For a single system deployment, see Step 2: Copying Database Access Files to Your Application Server (see page 76).
  - For a distributed system deployment, see Step 2: Copying Database Access Files to Your Application Server (see page 115).

Problem:

I deployed the UDS WAR (arcotuds.war), but the UDS 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:

To resolve this issue:

Ensure that you have copied the ArcotAccessKeyProvider.dll and arcot-crypto-util.jar files in the application server JAVA HOME location. Depending on the type of deployment, see one of the following sections for information about the location of these files:

- For a single system deployment, see Step 2: Copying Database Access Files to Your Application Server (see page 76).
- For a distributed system deployment, see Step 2: Copying Database Access Files to Your Application Server (see page 115).

Database-Related Errors

Problem:

I ran the Risk Authentication database scripts and I did not see any errors. However when i try to access a Risk Authentication table, I see an error message stating that the table does not exist.

or

I am using Oracle database and I see the following error when I try to access a table:

```
ERROR : common.database.DBFOManagerImpl(65) : Failed to retrieve
Database error codes for Datasource[1]. Error: ORA-00942: table or
view does not exist.
```

Cause:

- The database user that you used to run the scripts did not have the required file permissions.
- You did not run the database scripts in the specified order.

Solution:

Do the following:

1. Ensure that the user has the right file permissions.
2. Clean up the database.

   See section, "Dropping Risk Authentication Schema" for detailed instructions.
3. Run the database scripts again *in the right order*.

   See section, "Running Database Scripts" for more information in case of single-system installation.

   See section, "Running Database Scripts" for more information in case of distributed-system installation.
4. Run the following query to verify if the database was seeded correctly:

   ```
   SELECT SERVERNAME, VERSION FROM ARRFSERVERS;
   ```

Problem:

The connection to my Oracle database fails with the following entry in the RiskMinder 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 the following:

- Listener service on your database server.
- The TNSnames.ora file settings on the system where Risk Authentication Server is installed.

Problem:

Connection to the Oracle database fails with the following entry in the Risk Authentication 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 see 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. Typically, 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 know 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 Oracle database fails with the following entry in the Risk Authentication 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.

Problem:

Connection to the database fails with the following entry in the Risk Authentication 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.

Book: Refer to *Risk Authentication Administration Guide* for more information on how to use DButil.

Problem:

Connection to the MSSQL database fails with the following error:

```
java.sql.SQLException: No Datasource is set.
```

Cause:

The possible reason for this problem might be that the required JDBC JAR file might not be copied or might not be copied to the correct location on the application server you are using.

This is because the Administration Console, User Data Service (UDS), and Sample Application, which are Java-dependent components of Risk Authentication need Java Database Connectivity (JDBC) Java ARchive (JAR) files to connect to the database.

Solution:

Do the following:

1. If required, download the JDBC JAR file for the database you are using:
   - **For Oracle Databases:** ojdbc14.jar (version 10.2.0.1.0)
   - **For Microsoft SQL Server Databases:** sqljdbc.jar (version 1.2.2828)
   - **For MySQL**: mysql-connector-java-5.1.22-bin.jar (version 5.1.22)
2. Copy or deploy the JDBC JAR:

- If you are using Apache Tomcat, then refer to the "Apache Tomcat" subsection of the "Step 3: Copying JDBC JAR Files to Your Application Server" (see page 79).
- If you are using IBM WebSphere, then refer to the "IBM WebSphere" subsection of the "Step 3: Copying JDBC JAR Files to Your Application Server" (see page 79).
- If you are using Oracle WebLogic, then refer to the "Oracle WebLogic" subsection of the "Step 3: Copying JDBC JAR Files to Your Application Server" (see page 79).
- If you are using JBoss Application Server, then refer to "JBoss Application Server" subsection of the "Step 3: Copying JDBC JAR Files to Your Application Server" (see page 79).

Risk Authentication Server Errors

Problem:

I am trying to restart Risk Authentication Server, but it is not coming up. The last line in arcotriskfortstartup.log shows the following error:
`Cannot continue due to ARRF_LIB_init failure, SHUTTING DOWN`

Cause:

The possible cause might be that you have configured a rule that requires $$*rulelibname*$$.dll, but this DLL is not present in the %ARCOT_HOME%\plugins\rules\ directory.

Solution:

Do the following:

1. Search for the occurrences of the following string:
   `Couldn't find symbol [$$`*`RULENAME`*`$$] in library [$$`*`rulelibname`*`$$]`
2. If you find the preceding string, then copy the (corresponding) $$*rulelibname*$$.dll to the %ARCOT_HOME%\plugins\rules\ directory.
3. Search for the occurrences of the following string:
   `"Couldn't get function pointer for symbol [ARRF_AddOnRule] in lib [$$rulelibname$$]`
4. If you find the preceding string, then copy the (corresponding) $$*rulelibname*$$.dll to the %ARCOT_HOME%\plugins\rules\ directory.
5. If you do not see any of these log strings, then it is strongly recommended that you look for any ERROR or WARNING messages in the log file. It should provide you sufficient information to debug this issue.

Problem:

I am trying to restart Risk Authentication Server, but it is not coming up. The last lines in arcotriskfortstartup.log shows the following error:
`"Transport Exception on Admin channels: bind: Address already in use"`
`"Cannot continue due to loadAdminProtocolsAndAddTranports failure, SHUTTING DOWN"`

Cause:

The possible cause for this issue is that the Server Management Port (Default Port Number: 7980) is already open on the host by some other process. While, Risk Authentication 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%.
3. Start Risk Authentication Server in the debug mode, as follows:

   arrfserver.exe -debug -port <*new_port*>

After the Server Management port is open, the Master Administrator can log in to the Administration Console and configure the other ports.

SDK Errors

Problem:

I have set a new Risk Authentication configuration and I am trying to invoke Java APIs that uses this new configuration, but I see the following error:
`Configuration not Found`

Cause:

You might not have restarted the Risk Authentication Server.

Solution:

You must restart the Risk Authentication Server to use any new configurations.

Upgrade Errors

This section describes the troubleshooting steps, which will help you resolve the errors that you might face while upgrading RiskMinder.

Problem:

The upgrade tool fails with the following error:
`Error Occured: IO exception while parsing, %ARCOT_HOME%\tools\common\upgrade\xml\arcot-<product-name>-upgrade-meta-data.xml`

Cause:

The upgrade tool could not find the arcot-<*product-name*>-upgrade-meta-data.xml file. Here, *product-name* can be either common or riskfort.

Solution:

Check if the arcot-<*product-name*>-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 Step 6 in "Step 2: Migrating the Database for Common Components for RiskMinder 2.x or 3.0" (see page 158) ("Upgrading RiskMinder" (see page 147)). 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 directory.

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 re-import 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 &quot;/WEB-INF/jsp/dynamic/navbar_GA.jsp&quot;`

Cause:

The Work folder of the application server where Administration Console is deployed still contains the cache of the earlier Administration Console version.

Solution:

Clear the Work folder of the application server where 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 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 Administration Console.

Problem:

After the upgrade process, RiskMinder Server or Case Management Server do not start and the following lines appear in the log file:

```
ArDBM::Executing
Query[ArRFProtocolRegistryQuery_InsertProtocolDetailsForInstance]
ArDBConnection::GetDBDiagnosis: SQL State:23000, Native Code: 1, ODBC
code: [Arcot Systems][ODBC Oracle Wire Protocol
driver][Oracle]ORA-00001: unique constraint
(XXXXXXX.UN_ARPROTOCOLREGISTRY) violated
Dbm::SQL State:23000, Native Code: 1, ODBC code: [Arcot Systems][ODBC
Oracle Wire Protocol driver][Oracle]ORA-00001: unique constraint
(XXXXXXX.UN_ARPROTOCOLREGISTRY) violated
```

Cause:

The auto-generated indexes for the corresponding unique key in Oracle 11g database are not dropped. Even after you run the drop constraint on the unique key, the corresponding index is not dropped.

Solution:

To resolve this issue, do the following:

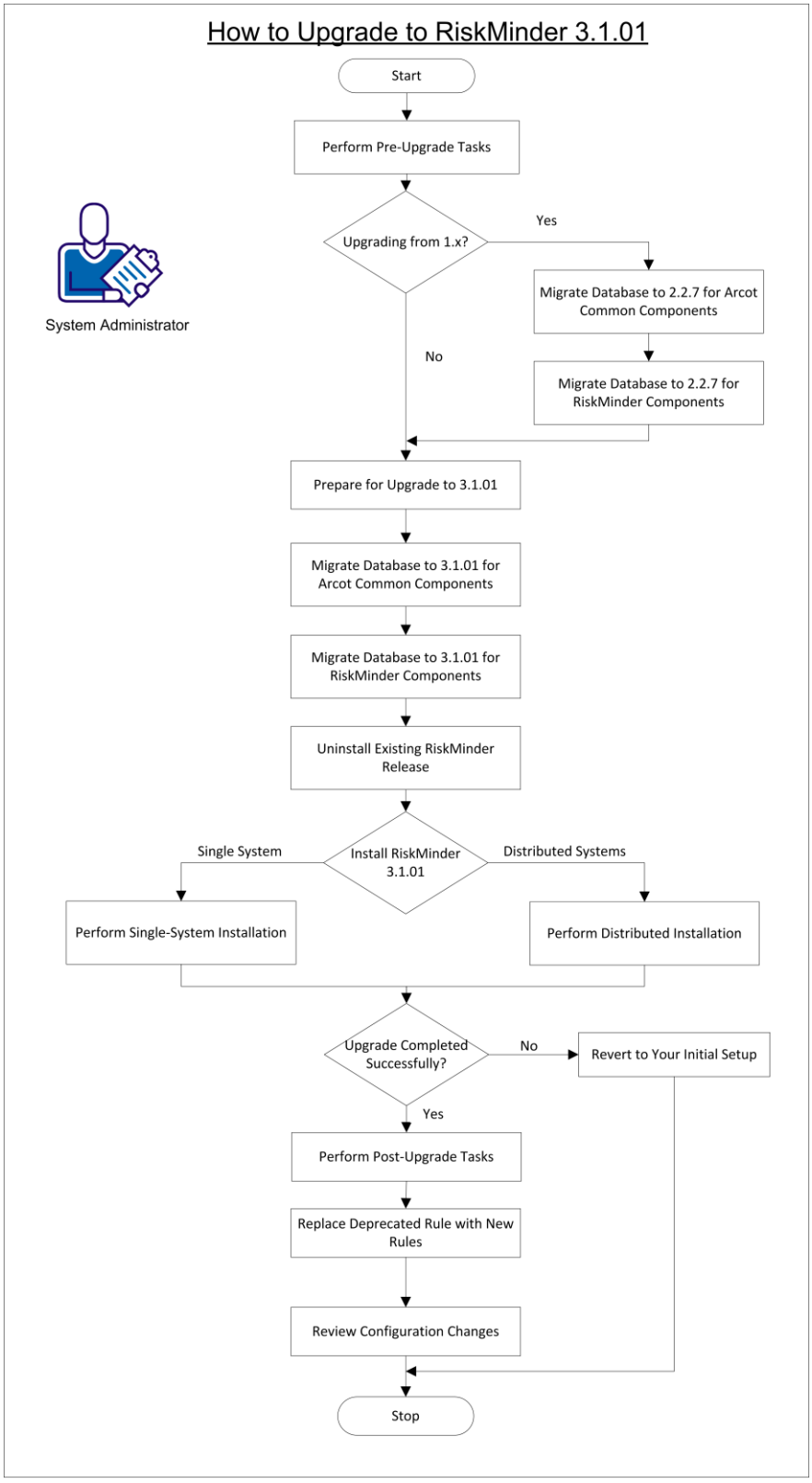
1. Run the following SQL commands:
   ```
   DELETE FROM ARRFINSTANCES;
   DELETE FROM ARRFPROTOCOLREGISTRY WHERE INSTANCEID <> 'DEFAULTID';
   ALTER TABLE ARRFPROTOCOLREGISTRY DROP CONSTRAINT
   UN_ARPROTOCOLREGISTRY CASCADE;
   COMMIT;
   ```
2. Verify whether the index has been dropped by running the following command:
   ```
   Select * from user_indexes where
   index_name='UN_ARPROTOCOLREGISTRY';
   ```
   You should not see any rows returned for the preceding query.

3. Start RiskMinder Server and Case Management Server.

Chapter 10: Upgrading to Release 3.1.01

Upgrading from 1.x to 3.1.01 is a two-stage procedure. You first upgrade from 1.x to 2.2.7 and then upgrade from 2.2.7 to 3.1.01. In contrast, if you are upgrading from 2.x or 3.x, you directly upgrade to 3.1.01.

The following flow diagram shows the steps to upgrade to release 3.1.01:


How to Upgrade to RiskMinder 3.1.01
Start
Perform Pre-Upgrade Tasks
Upgrading from 1.x?
Yes
No
Migrate Database to 2.2.7 for Arcot Common Components
Migrate Database to 2.2.7 for RiskMinder Components
System Administrator
Prepare for Upgrade to 3.1.01
Migrate Database to 3.1.01 for Arcot Common Components
Migrate Database to 3.1.01 for RiskMinder Components
Uninstall Existing RiskMinder Release
Install RiskMinder 3.1.01
Single System
Distributed Systems
Perform Single-System Installation
Perform Distributed Installation
Upgrade Completed Successfully?
No
Revert to Your Initial Setup
Yes
Perform Post-Upgrade Tasks
Replace Deprecated Rule with New Rules
Review Configuration Changes
Stop

To upgrade to RiskMinder 3.1.01, perform the following steps:

1. Performing Pre-Upgrade Tasks (see page 297)
2. If you are upgrading from 1.x, then perform the following steps. Do not perform these steps if you are upgrading from 2.x or 3.x.
   - Migrating the Database to Release 2.2.7 for Arcot Common Components (see page 155)
   - Migrating the Database to Release 2.2.7 for RiskMinder Components (see page 156)
3. Preparing for the Upgrade to Release 3.1.01 (see page 157)
4. Migrating the Database to Release 3.1.01 for Arcot Common Components (see page 158)
5. Migrating the Database to Release 3.1.01 for RiskMinder Components (see page 161)
6. Uninstalling the Existing Release of RiskMinder (see page 162)
7. Reinstalling RiskMinder (see page 163)
8. If you encounter any warnings during the Server startup and if your transactions fail, then perform the procedure described in (In Error Scenario Only) Reverting to Your Initial Setup (see page 361).
9. Performing Post-Upgrade Tasks (see page 166)
10. Replacing Deprecated Rules with New Rules (see page 166)
11. Review Configuration Changes After the Upgrade (see page 366)

Performing Pre-Upgrade Tasks

Important! Perform the upgrade procedure on the system where the Administration Console is installed.

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

- If you have earlier installed both CA AuthMinder and CA RiskMinder and you plan to upgrade both products, then ensure that you follow the guidelines that are given at various places in this document.
- Ensure that the account that you plan to use for the upgrade operation belongs to the Administrators group.
- If you are upgrading from RiskMinder 1.*x* to 3.1.01, migrate all your proposed configuration data to the production environment. Only active data is migrated and available after upgrade.

  Note: If you are upgrading from RiskMinder 2.*x* or 3.x to 3.1.01, both proposed and active configuration data get migrated.
- From release 3.1 onward, a rule with a score of 0 no longer carries the ALLOW advice. Instead, a score of 0 implies SILENT, which means that the rule is executed but is not used for scoring. In addition, if the default score was 0 before the upgrade, then the default score is changed to 1 during the upgrade.

  Note: For information about changing the score of a rule, see the administration guide for your RiskMinder release.
- Custom add-on rule types that you created in release 2.x or earlier releases are not migrated during the upgrade. The feature to create a custom add-on rule type by importing an XML file has been deprecated. If your RiskMinder deployment contains custom add-on rule types, then delete them before the upgrade.
- If the mnemonic or name of an existing rule is the same as the mnemonic or name of a rule that is newly introduced or modified by the upgrade, then the upgrade fails. The same issue is encountered if the name of an existing rule is the same as the name of a new rule. To avoid this issue:

  1. Use the administration console to compare the mnemonics of your existing rules with the mnemonics of the rules that are newly introduced or modified by the upgrade.

     The following table lists the rules that are newly introduced or modified by the upgrade:

Rule Name	Rule Mnemonic
Unknown DeviceID	UNKNOWNDEVICEID
Device MFP Not Match	MFPMISMATCH
User Not Associated with DeviceID	USERDEVICENOTASSOCIATED
Unknown User	UNKNOWNUSER

2. If the mnemonic of an existing rule matches the mnemonic of a new rule, delete the existing rule and then re-create it. While you re-create the rule, give it a different mnemonic. The system allows the rule name to be the same for two different rules but it is recommended that you change the name of the existing rule to avoid confusion.

   Note: For information about deleting and creating rules, see the administration guide for your current RiskMinder release.

- In release 2.x, you can have a rule, ruleset, or miscellaneous rule configuration refer to another rule, ruleset, or miscellaneous rule configuration. This feature is not available in release 3.1. Perform the following steps for each rule, ruleset, or miscellaneous rule configuration that refers to another rule, ruleset, or miscellaneous rule configuration:

  a. Log in to the Administration Console as a GA or OA.

  b. If you have logged in as the GA and you want to perform this procedure for a system ruleset, click the Services and Server Configurations tab.

  c. If you have logged in as the GA or OA to perform this procedure for a single organization:

     Activate the Organizations tab.

     Click the Search Organization link under Manage Organizations.

     Click the Search button on the Search Organization page to display the list of organizations.

     Click the name of the organization.

     Click the RiskFort Configuration tab.

  d. Under the Rules Management section on the side-bar menu, click the link for the rule, ruleset, or miscellaneous rule configuration that refers to another rule, ruleset, or miscellaneous rule configuration.

  e. Select Use Own.

  f. Select Copy from an Existing Ruleset.

  g. From the Ruleset Name list, select the ruleset to which this rule, ruleset, or miscellaneous rule configuration was referring.

  h. Click Save.

  i. Migrate the changes to the production environment.

     Note: For detailed information about migrating the changes to the production environment and refreshing the cache, see the *CA RiskMinder Administration Guide*.

- The upgrade process is supported only in the offline mode. Shut down the following gracefully:
  - RiskFort Server
  - Case Management Queuing Server
  - Any application servers where Administration Console and User Data Service are deployed

- If Administration Console is open, close it.
- 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 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.
- 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 plan to upgrade is available throughout the upgrade process.
- Ensure that the database on which you plan to upgrade is disabled for replication.
- Back up the database containing the RiskMinder schema.
- If you require multibyte character or internationalization support in RiskMinder and if your database does not currently support multibyte data, then migrate the database to a character set that supports multibyte data. For more information, see "Configuring Database Server" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.
- Consider the requirements such as rollback segment size, based on data volume, before running the upgrade tool.
- Ensure that you have the database privileges that are required to upgrade RiskMinder. For the complete list of privileges, see Prerequisites for Upgrading to RiskMinder 3.1.01.
- 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 RiskMinder is installed.
- 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 that was created by the existing RiskMinder installation. Typically, ARCOT_HOME is *install_location*/arcot/.

ARCOT_HOME_BACKUP refers to the backup directory into which you copy the contents of the existing the ARCOT_HOME directory. If you encounter any errors during upgrade, use the ARCOT_HOME_BACKUP directory to revert to the initial setup.

Migrating the Database to Release 2.2.7 for Arcot Common Components

Note: Perform the tasks in this section only if you are upgrading from release 1.x. If you are upgrading from release 2.x or 3.x, you can ignore this topic.

Important! If you installed CA AuthMinder with CA RiskMinder and you have completed the upgrade to CA AuthMinder release 7.1.01, then do not migrate the database for Arcot common components. This step has already been performed during the AuthMinder upgrade process.

Migrate the database to the release 2.2.7 state for Arcot common components.

Follow these steps:

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

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

   - arcot-common-upgrade-0.*x*-1.0.zip
   - arcot-riskfort-upgrade-1.*x*-2.2.7.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>`

   Here, *db_type* can be mssql or oracle.

5. Run the arcot-db-config-for-common-1.0.sql script.

   Note: In the case of 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.

6. Navigate to the following directory:
   `%ARCOT_HOME%\dbscripts\<db_type>\upgrade-scripts\`

   Here, *db_type* can be mssql or oracle.

7. Run the arcot-upgrade-for-common-1.0.sql script.

   Note: In the case of 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 your database to the following directory:
   `%ARCOT_HOME%\java\lib`

9. Back up the existing ArcotAccessKeyProvider.dll file if it is in <*JAVA_HOME used by APP_SERVER*>\jre\bin. Then, copy the %ARCOT_HOME%\native\win\<*32bit-or-64bit*>\ArcotAccessKeyProvider.dll file to <*JAVA_HOME used by APP_SERVER*>\jre**bin**.
10. Set the PATH variable to include the directory where ArcotAccessKeyProvider.dll is copied.
11. Copy the file %ARCOT_HOME%\java\lib\arcot-crypto-util.jar to <*JAVA_HOME used by APP_SERVER*>\jre\lib**ext**\.
12. Navigate to the %ARCOT_HOME%\tools\upgrade directory.
13. Run the upgrade-common.bat tool.
14. To ensure that the common database upgrade operation was run successfully, see the %ARCOT_HOME%\logs\upgrade-common.log file.

Migrating the Database to Release 2.2.7 for RiskMinder Components

Important! Perform the steps in this section only if you are upgrading from release 1.x. If you are upgrading from release 2.x or 3.x, you can ignore this procedure.

After you migrate the database for Arcot common components, migrate the database to the release 2.2.7 state for RiskMinder components.

Follow these steps:

1. Copy the arcot-riskfort-upgrade-1.*x*-2.2.7.zip file to the ARCOT_HOME directory.
2. Extract the contents of the arcot-riskfort-upgrade-1.*x*-2.2.7.zip file in this directory.
3. Navigate to the following directory:
   `%ARCOT_HOME%\dbscripts\<db_type>\upgrade-scripts`

   Here, *db_type* can be mssql or oracle.
4. Run the SQL script corresponding to your current release of RiskMinder, as listed in the following table.

Current RiskMinder Release	SQL Script to Run
1.5.1 or 1.5.1.*x*	arcot-riskfort-upgrade-1.5.1.8-2.2.7.sql
1.6 or 1.6.0.*x*	arcot-riskfort-upgrade-1.6.0.3-2.2.7.sql
1.7 or 1.7.0.*x*	arcot-riskfort-upgrade-1.7.0.3-2.2.7.sql

5. Navigate to the following directory:
   `%ARCOT_HOME%\dbscripts\<db_type>\upgrade-scripts\`

   Here, *db_type* can be mssql or oracle.

6. Run the arcot-post-upgrade-for-common-1.0.sql script.

   This script ensures the following configurations:

   - The user ID for Master Administrator is changed from MASTER_ADMIN to MASTERADMIN.
   - The password for the MASTERADMIN account is **master1234!**
   - The organization that MASTERADMIN belongs to is MASTERADMIN. This feature 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.

Preparing for the Upgrade to Release 3.1.01

This section describes the steps that you must perform to prepare your setup for upgrading to 3.1.01.

Follow these steps:

1. If application server connection pooling was being used in your existing RiskMinder deployment, navigate to the %ARCOT_HOME%\bin directory, and update the securestore.enc file by running the following command for the primary database:
   `DBUtil -pi <DB_username> <DB_password>`

   Note: To determine whether database connection pooling is being used, open the %ARCOT_HOME%\conf\arcotcommon.ini file. Check the value of the AppServerConnectionPoolName parameter.

2. If SSL has been configured for the connection with the database, navigate to the %ARCOT_HOME%\bin directory and set the TrustStore password using DBUtil, as follows:
   `DBUtil -pi TrustStorePath.1 <truststore-password>`

   Note: To determine whether SSL has been configured, check the value of the TrustStorePath parameter in the arcotcommon.ini file.

Migrating the Database to Release 3.1.01 for Arcot Common Components

Migrate the database to the release 3.1.01 state for Arcot common components.

Follow these steps:

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

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

   - arcot-common-upgrade-1.0.*x*-2.0.zip
   - arcot-riskfort-upgrade-2.x-3.x-3.1.01.zip

2. Copy the arcot-common-upgrade-1.0.*x*-2.0.zip file to the ARCOT_HOME directory.
3. Extract the contents of the arcot-common-upgrade-1.0.*x*-2.0.zip file in this directory.

   Note: Click **Yes** if you are prompted to overwrite any existing files.

4. Navigate to the following directory:
   `%ARCOT_HOME%\tools\common\upgrade\`
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 **exact** name, as follows:

   - ORACLE: ojdbc.jar
   - SQL Server: sqljdbc.jar

7. Locate the JAVA_HOME used by the existing installation and ensure that you use the same JAVA_HOME to run the upgrade tool.
8. Set the PATH variable to include the directory where ArcotAccessKeyProvider.dll is copied.

   Important! If you are upgrading from release 3.x to 3.1.01, do not perform the remaining steps of this procedure. Instead, directly proceed to the next section.

9. At the command prompt, change your working directory to:
   `%ARCOT_HOME%\tools\common\upgrade\`
10. 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	The following JVM options are required only if LDAP organizations are configured: ■ -Xmx<*heap_memory_size_in_MB*>M: Sets the maximum heap size to 1GB. If there are more than 1,00,000 users in the configured LDAP, then it is strongly recommended that you increase the heap size to 2048M (2GB). ■ -Dcom.arcot.ldap.migration.timeout=<*duration*>: The migration of an LDAP organization involves fetching all the users from the LDAP server and migrating the users to the RiskMinder database. This parameter sets the maximum time (in minutes) taken to fetch all users from the LDAP server, beyond 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, the timeout would depend on the type of hardware configuration being used. The default value of this parameter is 240 minutes. **Note:** Ensure that the java command executable belongs to JAVA_HOME identified in Step 7. If JAVA_HOME is not set, modify the PATH environment variable to include %*JAVA_HOME*%\bin.
log-file	Specifies the path to the log file: ■ 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%\logs\ 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.

Option	Description
product-name	Specifies the name of the product for which the upgrade is run. If you do not specify the product name, the product name is assumed to be common. Possible values are: ■ common: Indicates the Arcot common components. ■ riskfort: Indicates RiskMinder. **Note:** Upgrade the Arcot common components before you upgrade RiskMinder.
prompt	Prompts whether to proceed further after each phase of the upgrade process is completed successfully. The upgrade process happens in the following phases: ■ Pre-upgrade: Involves performing various DDL and DML operations to migrate the database schema. ■ Upgrade: Involves migrating the data to the new schema. ■ Post-upgrade: Involves cleanup or follow-up actions that are required to be performed after the upgrade. ■ Verification: Involves the verification of whether the upgrade is successful. This option You can choose to run the upgrade 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. If you are upgrading from release 1.0.x, then check for the following line in the %ARCOT_HOME%\logs\arcot_common_upgrade.log file:

   Upgrade for common from version 1.0.x to version 2.0 run successfully.

   The presence of this line in the log confirms that the database was upgraded successfully.

Migrating the Database to Release 3.1.01 for RiskMinder Components

After you migrate the database for Arcot common components, migrate the database to the release 3.1.01 state for RiskMinder components.

Follow these steps:

1. Extract the contents of the arcot-riskfort-upgrade-2.x-3.x-3.1.01.zip file in the ARCOT_HOME directory.
2. Navigate to the following directory:
   `%ARCOT_HOME%\tools\common\`**`upgrade`**`\`
3. Run the following command:
   `java -jar arcot-common-upgrade-framework.jar --product-name` **`riskfort`**

   See the table in Migrating the Database to Release 3.1.01 for Arcot Common Components (see page 158) for a description of the command options.
4. Depending on the release that you are upgrading from, locate one of the following lines in the arcot_common_upgrade.log file in the %ARCOT_HOME%\logs directory:
   `Upgrade for riskfort from version <`*`your-RiskMinder-release`*`> to version 3.1.01 run successfully.`

   For example, if you upgraded from release 3.0, then locate the following line:
   `Upgrade for riskfort from version 3.0 to version 3.1.01 run successfully.`

   The presence of this line in the log confirms that the database was upgraded successfully.

Uninstalling the Existing Release of RiskMinder

Uninstall the existing release of RiskMinder. Also uninstall the RiskMinder components 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 RiskMinder installation, follow the uninstallation instructions that are given in the installation guide for your existing release of RiskMinder.

Follow these steps:

1. Uninstall the existing release of RiskMinder as follows:
    a. Ensure that the following components have been shut down gracefully:
        - RiskFort Server
        - Case Management Queuing Server
        - Any application servers where other RiskFort components are deployed.
    b. Ensure that the Administration Console is not open.
    c. Ensure that all INI and other files that are related to the RiskMinder configuration are closed.
    d. On the desktop, click Start, Settings, Control Panel, Add/Remove Programs to open the Add or Remove Programs window.
    e. From the Currently installed programs list, select Arcot RiskFort, and click Change/Remove.

        The Uninstall Arcot RiskFort window appears.

        Note: You can also uninstall RiskMinder by running Uninstall Arcot RiskFort.exe available in the <*install_location*>\Arcot Systems\Uninstall Arcot RiskFort\ directory.
    f. Select Complete Uninstall.

        Note: You may have to wait for a few minutes for the uninstallation process to complete.

        After the software is uninstalled successfully, the Uninstallation Complete screen appears with a success message.
    g. Click Done to exit the wizard and complete the uninstallation process.
    h. If you have installed CA AuthMinder and CA RiskMinder and you are deleting both products, delete any files that are left over in the ARCOT_HOME directory.
2. Undeploy the Administration Console, User Data Service, and Sample Application Web applications from the application server. For detailed information, see the application server documentation.

Reinstalling RiskMinder

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

- Reinstalling RiskMinder on a Single System (scenario) (see page 310)
- Reinstalling RiskMinder on a Distributed System (scenario) (see page 333)

Reinstalling RiskMinder on a Single System

To reinstall RiskMinder on a single system, perform the tasks that are described in the following sections:

1. Performing Complete Installation (see page 311)

   Note: While you install RiskMinder 3.1.01, ensure that you specify the same primary and backup database details from arcotcommon.ini in the $ARCOT_HOME/conf/ directory.

2. Verifying the Database Setup (see page 74)
3. Preparing Your Application Server (see page 75)
4. Deploying Administration Console (see page 325)
5. Logging In to Administration Console (see page 84)

   Important! Ensure that you use the current MA password and *not* the default password, because the MA password has been reset during the bootstrap process that you performed during 2.x installation.

6. Starting RiskMinder Server (see page 87)
7. Starting the Case Management Queuing Server (see page 88)
8. Deploying User Data Service (see page 89)
9. Deploying Sample Application (see page 91)
10. Verifying the Installation (see page 91)

    Note: If there are any warnings during the Server startup and if your transactions fail, then the upgrade has not been performed successfully. You can revert to your initial setup by following the steps that are listed in (In Error Scenario Only) Reverting to Your Initial Setup (see page 361).

11. Using Sample Application (see page 332)
12. Applying the Post-Installation Checklist (see page 96)

Performing Complete Installation

To install (and later configure) RiskMinder on Microsoft Windows successfully, the user account that you plan to use for the installation *must* belong to the Administrators group. Otherwise, some critical steps in the installation, such as DSN creation and configuration, and RiskMinder service creation, will not complete successfully, though the installation may complete without any errors.

Complete installation allows you to install all components of the RiskMinder package. These components include RiskMinder Server and the scripts that are required for setting up the database that you intend to use for RiskMinder.

Note: Before you proceed with the installation, ensure that all prerequisite software components are installed and the database is set up, as described in "Preparing for Installation" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Follow these steps:

1. To run the installation wizard, navigate to the directory where the Arcot-RiskFort-3.1.01-Windows-Installer.exe file is located and double-click the file.

   The Welcome screen appears.

2. Click Next to continue.

   The License Agreement screen appears.

3. Carefully read the license agreement, select the I accept the terms of the License Agreement option, and click Next.

   The installer now checks if any other CA product is installed on the computer.

   If it does not find an existing CA product installation, then you are prompted for an installation directory. In this case, the Installation Location screen appears.

   If the installer detects an existing CA product installation (an existing ARCOT_HOME), then:

   - You are not prompted for an installation directory.
   - In addition, you are not prompted for the database and encryption setup. The installer uses the existing database and encryption settings. As a result, you see the screen in Step 6, though the configuration is disabled, and the screen corresponding to Step 10 is not displayed.

4. Click Next to install in the specified directory.

   The Installation Type screen appears.

5. Select Complete to install all components in one ARCOT_HOME and then click Next to continue.

   The Database Type screen appears.

6. Depending on the type of database you have, you can select Microsoft SQL Server, Oracle Database, or MySQL. Click Next to proceed.

   If you selected Microsoft SQL Server, then the SQL Server Database Details screen appears.

   Note: If you are using a SQL database, then ensure that the ODBC Driver version you are using is the same as the one mentioned in "Preparing for Installation" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

   If you selected Oracle on the Database Type screen, then the Oracle Database Details screen appears.

   Note: CA RiskMinder release 3.1.01 is now certified to work with Oracle Real Application Clusters (Oracle RAC). To use Oracle RAC with your RiskMinder Installation, select Oracle Database in this step, perform the next step (Step 7), and then perform the steps in Configuring CA RiskMinder for Oracle RAC (W) (see page 247).

   If you selected MySQL on the Database Type screen, then the MySQL Database Details screen appears.

7. Based on your database choice in the preceding screen:

   - If you selected Microsoft SQL Server, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.
Server	The host name or IP address of the RiskMinder datastore. Default Instance ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase Named Instance ■ **Syntax**: *<server_name>**<instance_name>* ■ **Example**: demodatabase\instance1

Parameter	Description
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.
Password	The password associated with the User Name you specified in the previous field and which is used by RiskMinder to access the database. This password is specified by the database administrator.
Database	The name of the MS SQL database instance.
Port Number	The port at which the database listens to the incoming requests. The default port at which an MS SQL database listens is 1433. However, if you would like to specify another port, enter the port value in this field.

- If you selected Oracle Database, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.

Parameter	Description
Password	The password associated with the **User Name** you specified in the previous field and which is used by RiskMinder 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. The default port at which an Oracle database listens is 1521. However, if you would like to specify another port, enter the port value in this field.
Host Name	The host name or IP address of the RiskMinder datastore. ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase

1. After you specify the database details, test if you can successfully connect to the database by clicking the **Test Data Source** button and verify the result of the same in the field below the button.

   Note: If the connection was not successful, ensure that you have specified the correct database details and click **Test Data Source** again. Do not proceed with the installation unless the database connectivity is successful.

2. Click **Next** to continue.

   The Encryption Configuration screen appears. Use this screen to select the encryption mode and configure the information that is used for encryption.

3. Specify the following information:

Field Name	Description
Master Key	Specify the password for the Master Key, which is stored at *<install_location>*\Arcot Systems\conf\securestore.enc and will be used to encrypt the data stored in the database. By default, this value is set to MasterKey. **Note:** If you want to change the value of Master Key *after* the installation, then regenerate securestore.enc with a new Master Key value. For more information, see appendix, "Changing Hardware Security Module Information After the Installation" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Field Name	Description
Configure HSM	Select this option only if you will use a Hardware Security Module (HSM) to encrypt the sensitive data. If you do not select this option, then, by default, the data is encrypted by using the Software Mode.
PIN	Enter the password to connect to the HSM.
Choose Hardware Module	Choose one of the following HSMs that you plan to use: ■ Luna HSM ■ nCipher netHSM
HSM Parameters **Tip:** The HSM parameter values are recorded in arcotcommon.ini, which is available in <*install_location*>\Arcot Systems**conf**\. To change these values *after* installation, edit this file, as discussed in the section titled, Configuration Files and Options" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.	Set the following HSM information: ■ **Shared Library:** The absolute path to the PKCS#11 shared library corresponding to the HSM. For Luna (cryptoki.dll) and for nCipher netHSM (cknfast.dll), enter the absolute path and name of the DLL. ■ **Storage Slot Number:** The HSM slot where the 3DES keys used for encrypting the data are available. – For **Luna**, the default value is 0. – For **nCipher netHSM**, the default value is 1.

The Pre-Installation Summary screen appears.

1. Review the information about this screen, and if you must change a previous selection, then click **Previous** to do so. After you change the required selection, click **Next** to go to the next screen.
2. Click **Install** to begin the installation process.

   The Microsoft Visual C++ 2010 x86 Redistributable Setup screen appears. *This screen appears only if the current system where you are installing RiskMinder does not have Microsoft Visual C++ 2010 x86.*
3. On the Microsoft Visual C++ 2010 x86 Redistributable Setup screen:
   a. Select the **I have read and accept the license terms** option, and click **Install**.

      The Installation Progress screen appears. This may take a few seconds. After some time the Installation Is Complete screen appears.

   b. Click **Finish** to close the Microsoft Visual C++ 2010 x86 Redistributable Setup dialog and continue with the RiskMinder installation.

   The Installing Arcot RiskFort screen appears. This may take several minutes.

   After some time the Installation Complete screen appears.
4. Click **Done** to complete the RiskMinder installation.

Installation Logs

After installation, you can access the installation log file (Arcot_RiskFort_Install_*<timestamp>*.log) in the *<install_location>* directory. For example, if you had specified the C:\Program Files directory as the installation directory, then the installation log file is created in the C:\Program Files directory.

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

Verifying the Database Setup

After you run the required database scripts, verify that the RiskMinder schemas were seeded correctly. To do so:

1. Log in to the RiskMinder database as the user who installed the database.

   Note: If you are following the upgrade path, then log in to the database as the user who upgraded the database.

2. Run the following query:
   ```
   SELECT SERVERNAME, VERSION FROM ARRFSERVERS;
   ```

   You must see the following output as a result of the preceding query:
   ```
   SERVERNAME                    VERSION
   ------------------------  ---------------
   RiskFort                      3.1.01
   RiskFortCaseManagement        3.1.01
   ```

3. Log out of the database console.

Preparing Your Application Server

Two components of RiskMinder, 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 Application Server

Before you deploy the WAR files for these web applications on the application server of your choice, copy the files that are 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 and to deploy the WAR files of these web applications:

- Step 1: Setting Java Home (see page 75)
- Step 2: Copying Database Access Files to Your Application Server (see page 76)
- Step 3: Copying JDBC JAR Files to Your Application Server (see page 79)
- Step 4: (Mandatory for Oracle WebLogic 10.1) Creating Enterprise Archive Files (see page 81)

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. This JAVA_HOME must be your application server JAVA_HOME.

In addition, %JAVA_HOME%**bin**\ must be added to the PATH variable. If you fail to do so, then Administration Console, UDS, 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 RiskMinder database securely:

- arcot-crypto-util.jar available at:
  <install_location>\Arcot Systems**java****lib**\
- ArcotAccessKeyProvider.dll available at:
  <install_location>\Arcot Systems**native****win**\<**32bit-or-64bit**>\

As a result, these files must be copied to the appropriate location on the application server where you have deployed these RiskMinder 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 files that are required for database access:

1. Copy arcot-crypto-util.jar *to <Tomcat_JAVA_HOME>*\jre\lib**ext**\.

   Here, *<Tomcat_JAVA_HOME>* represents the JAVA_HOME used by your Apache Tomcat instance.

2. Copy ArcotAccessKeyProvider.dll *to <Tomcat_JAVA_HOME>*\jre**bin**\.
3. Restart the application server.

IBM WebSphere

To copy the files that are required for database access:

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 or 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, C:\Program Files\Arcot Systems\java\lib**arcot-crypto-util.jar**.

   e. Enter the JNI Library path.

This path must point to the location where the ArcotAccessKeyProvider.dll file is present.

3. Click **Apply** to save the changes.
4. Configure the 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 must be performed.
   c. Click **Java and Process Management** and then click **Class Loader**.
   d. Click **New**.
   e. Select default **Classes loaded with parent class loader first** and click **OK.**
   f. Click the auto-generated **Class Loader ID**.
   g. On the class loader **Configuration** page, click **Shared Library References**.
   h. Click **Add**, select **ArcotJNI,** and then click **Apply**.
   i. Save the changes.
5. Copy ArcotAccessKeyProvider.dll to <*WebSphere_JAVA_HOME*>\jre**bin**\.

   Here, <*WebSphere_JAVA_HOME*> represents the JAVA_HOME used by your IBM WebSphere instance.
6. Restart the application server.

Oracle WebLogic

To copy the files that are required for database access:

1. Copy ArcotAccessKeyProvider.dll to <*WebLogic_JAVA_HOME*>\jre**bin**\.

   Here, <*Weblogic_JAVA_HOME*> represents the JAVA_HOME used by your Oracle WebLogic instance.
2. Copy arcot-crypto-util.jar to <*WebLogic_JAVA_HOME*>\jre\lib**ext**\.

   Note: Ensure that you use the appropriate <*JAVA_HOME*> used by WebLogic.
3. Log in to WebLogic Administration Console.
4. Navigate to **Deployments**.
5. Enable the **Lock and Edit** option.
6. Click **Install** and navigate to the directory that contains the arcot-crypto-util.jar file.
7. Click **Next** to open the Application Installation Assistant.
8. Click **Next** to display the Summary page.
9. Click **Finish**.
10. Activate the changes.

11. Restart the application server.

JBoss Application Server

To copy the files that are required for database access:

1. Copy ArcotAccessKeyProvider.dll *to* <*JBoss_JAVA_HOME*>\jre**bin**\.

   Here, <*JBoss_JAVA_HOME*> represents the JAVA_HOME used by your JBoss Application Server instance.

2. Copy arcot-crypto-util.jar *to* <*JBoss_JAVA_HOME*>\jre\lib**ext**\.

3. Restart the application server.

Step 3: Copying JDBC JAR Files to Your Application Server

RiskMinder requires the following JDBC JAR files for the supported databases:

- **Oracle 10g:** Oracle JDBC Driver (10.2.0.1.0)
- **Oracle 11g:** Oracle JDBC Driver (11.2.0.2.0)
- **Microsoft SQL Server:** MSSQL JDBC Driver (1.2.2828)
- **MySQL:** MySQL JDBC Driver (5.1.22)

The following subsections 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 required JDBC JAR file:

1. Navigate to the location where you have downloaded the *<Database_JAR>* file.
2. Copy the *<Database_JAR>* file to the following directory:
   - **On Apache Tomcat 5.5.x:** *<TOMCAT_HOME>*\common\lib\
   - **On Apache Tomcat 6.x and 7.x:** *<TOMCAT_HOME>*\lib\
3. Restart the server.

IBM WebSphere

To copy the required JDBC JAR file:

1. Log in to the WebSphere Administration Console.
2. Click Environment, and then click Shared Libraries.
   a. From the Scope list, select a valid visibility scope. The scope must include the target server or node on which the application is deployed.
   b. Click New.
   c. Enter the Name, say, JDBCJAR.
   d. Specify the Classpath.

      Important! This path *must* point to the location where the *<Database_JAR>* file is present and *must* include the file name.

   e. Click Apply to save the changes that were made.
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 is performed.

c. Click **Java and Process Management**, and then click **Class Loader**.

d. Click **New**.

e. Select default **Classes loaded with parent class loader first** and click **OK.**

f. Click the auto-generated **Class Loader ID**.

g. In the class loader **Configuration** page, click **Shared Library References**.

h. Click **Add**, select **JDBCJAR**, and then click **Apply**.

i. Save the changes that were made.

4. Restart the application server.

Oracle WebLogic

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

To copy the required JDBC JAR file in the case of Microsoft SQL Server:

1. Copy the <*Database_JAR*> file to <*Weblogic_JAVA_HOME*>\lib**ext**\.

   Here, <*WebLogic_JAVA_HOME*> represents the JAVA_HOME used by your Oracle WebLogic instance.

2. Log in to the 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 required <*Database_JAR*> file.
6. Click **Next** to display the Application Installation Assistant page.
7. Click **Next** to display the Summary page.
8. Click **Finish**.
9. Activate the changes.
10. Restart the application server.

JBoss Application Server

To copy the required JDBC JAR file:

1. Copy the JDBC JAR file to the following location on the JBOSS installation directory: <*JBOSS_HOME*>\server\default\lib\
2. Restart the application server.

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

Most enterprise Application Servers (such as WebSphere and Weblogic) enable you to bundle the related Java ARchive (JAR) or Web ARchive (WAR) files from one vendor (say, CA) to 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, WAR files are provided to deploy UDS and Administration Console. However if necessary, you can also change the format of these files to Enterprise ARchive (EAR) and then deploy the EAR files.

As discussed in the following subsections, you can either generate separate EAR files for both UDS and Administration Console, or you can generate a single EAR file that contains both web archives.

Generating Separate EAR Files

To create a separate EAR file each for UDS and Administration Console:

1. Open the Command Prompt window.
2. Navigate to the <*install_location*>\Arcot Systems\tools\common**bundlemanager**\ directory.
3. To create the EAR file, run the following command:
   `java -jar bundle-manager.jar -ear <filename.ear> -warList <filename.war>`

   The preceding command generates individual EAR files that are available at:
   `<install_location>\Arcot Systems\java\webapps\`

Generating a Single EAR File

To create a single EAR file that contains UDS and Administration Console Web archives:

1. Open the Command Prompt window.
2. Navigate to the <*install_location*>\Arcot Systems\tools\common**bundlemanager**\ directory.
3. To create the EAR file, run the following command:
   `java -jar bundle-manager.jar -ear <filename.ear> -warList arcotadmin.war arcotuds.war`

   The preceding command generates a single EAR file that is available at:
   `<install_location>\Arcot Systems\java\webapps\`

Deploying Administration Console

Note: If you are deploying the Administration Console on IBM WebSphere 7.0, then instead of the following instructions, refer to the instructions in the topic that is titled "Deploying Administration Console on IBM WebSphere 7.0" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Administration Console is a browser-based interface to RiskMinder that enables you to customize the server configurations and manage the deployed system.

You need the **arcotadmin.war** file to deploy the RiskMinder Administration Console. All Administration Console information is logged in the arcotadmin.log file. After you deploy arcotadmin.war, you can verify if it was correctly deployed by using this log file (arcotadmin.log).

Note: To manage RiskMinder by using Administration Console, ensure that Administration Console can access the system where RiskMinder Server is installed by its host name.

To deploy the Administration Console WAR file on your application server, and to verify if it was successfully deployed, perform the following steps:

1. Deploy arcotadmin.war in the appropriate directory on the application server.

   Note: The deployment procedure depends on the application server that you are using. See 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***.

2. **(For 32-bit WebSphere Only)** Configure reload of the Admin class when the application files are updated.

   a. Navigate to Application > Enterprise Applications and access the Admin 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.

   e. Restart the Admin application.

3. **(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 Systems\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>
```

Add the following category for cryptographic operations:

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

f. Save and close the file.

g. Take a backup of the existing JBoss logging libraries. These library files are available at:
<*JBOSS_HOME*>\lib\

h. 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/*

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

4. Restart the application server.
5. Verify that the console was successfully deployed:
   a. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`
   b. Open the arcotadmin.log file in any editor and locate the following lines:
      - 2.0.3
      - Arcot Administration Console Configured Successfully.

      These lines indicate that your Administration Console was deployed successfully.
   c. Also ensure that the log files *do not* contain any FATAL *and* WARNING messages.
   d. Close the file.

Logging In to Administration Console

When you log in to Administration Console for the first time, use the Master Administrator (MA) credentials that are configured automatically in the database during the deployment.

To log in to Administration Console as MA:

1. Launch the Administration Console in a Web browser window. The default URL for Administration Console is:
   `http://<host>:<appserver_port>/arcotadmin/masteradminlogin.htm`

   Note: The *host* and *port* information that you specify in the preceding URL must be of the application server where you deployed Administration Console.
   For example, in case of Apache Tomcat, the default *host* is localhost and *port* is 8080.
2. Log in by using the default Master Administrator account credentials. The credentials are:
   - **User Name: masteradmin**
   - **Password: master1234!**

Starting RiskMinder Server

To start RiskMinder Server:

1. Click the **Start** button on your desktop window.
2. Navigate to **Settings, Control Panel, Administrative Tools,** and then **Services.**
3. Locate and double-click **Arcot RiskFort Service**.
4. Click **Start** in the service window.

Note: If you want to stop RiskMinder Server, then follow the Steps 1 through 3, and click the **Stop** button in the service window.

Starting the Case Management Queuing Server

To start Case Management Queuing Server:

1. Click the **Start** button on your desktop window.
2. Navigate to **Settings, Control Panel, Administrative Tools,** and then **Services.**
3. Locate and double-click the **Arcot RiskFort Case Management Queuing Service**.
4. Click **Start** in the service window.

Note: If you want to stop the Case Management Queuing Server, then follow the Steps 1 through 3, and click the **Stop** button in the service window.

Deploying User Data Service (UDS)

RiskMinder can access user data either from a relational database (RDBMS) or directly from an LDAP server by using UDS, which is an abstraction layer that provides RiskMinder seamless access to the third-party data repositories deployed by your organization.

You need the arcotuds.war file to deploy UDS, as follows:

1. Deploy arcotuds.war on the application server. This file is available at: <*install_location*>\Arcot Systems\java\webapps\

   For example, in the case of Apache Tomcat, deploy the WAR file at <*APP_SERVER*_HOME>\webapps\.

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

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 to save the changes.

3. **(For JBoss Only)** Perform the following steps, if you have deployed UDS on a JBoss application server:

   a. Copy the Bouncy Castle JAR file (bcprov-jdk15-146.jar) from <*install_location*>\Arcot Systems\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 while Deploying Administration Console (see page 325):

<appender-ref ref="arcotudslog"></appender-ref>

f. Add the following line in the cryptographic category that you created while Deploying Administration Console (see page 325):

<appender-ref ref="arcotudslog"></appender-ref>

g. Save and close the file.

4. Restart the application server.
5. Verify if UDS was deployed successfully:

Note: The arcotuds.log file is used for logging UDS-related information.

a. Navigate to the following location:
`<install_location>\Arcot Systems\logs\`

b. Open the arcotuds.log file in any editor and locate the following line:

- User Data Service (Version: 2.0.3) initialized successfully.

This line indicates that UDS was deployed successfully.

c. Also ensure that the log files *do not* contain any FATAL *and* WARNING messages.

d. Close the file.

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 RiskMinder was installed and configured properly. In addition, it demonstrates:

- The typical RiskMinder workflows
- The basic operations (invocation and post-processing) of RiskMinder APIs
- Integration of your application with RiskMinder

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

1. Deploy the riskfort-3.1.01-sample-application.war file from the following location:
   `<install_location>\Arcot Systems\samples\java\`
2. If necessary, restart the application server.
3. Access Sample Application in a Web browser window. The default URL for Sample Application is:
   `http://<host>:<appserver_port>/riskfort-3.1.01-sample-application/index.jsp`

Verifying the Installation

After you have seeded the database schema, deployed UDS and Administration Console, and bootstrapped the system, and started the Server, ensure that all these components have started correctly. The log files that you must verify for this purpose is arcotriskfort.log.

To verify if the server started correctly:

1. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`
2. Open the arcotriskfortstartup.log file in any editor and locate the following lines:
   - STARTING Arcot RiskFort 3.1.01_w
   - Arcot RiskFort Service READY
3. Open the arcotriskfortcasemgmtserverstartup.log file in any editor and locate the following lines:
   - STARTING Arcot RiskFort Case Management 3.1.01_w
   - Arcot RiskFort Case Management Service READY

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

Using Sample Application

The following risk-evaluation operations can be performed by using Sample Application. Each of these operations is designed to run without error when RiskMinder is installed and functional.

- Performing Risk Evaluation and Post Evaluation for a First-Time User
- Creating Users
- Performing Risk Evaluation and Post Evaluation for a Known User
- Editing the Default Profile and Performing Risk Evaluation

Note: For information about running these operations, see the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Applying the Post-Installation Checklist

It is recommended that you fill the following checklist with the installation and setup information for RiskMinder. This information is useful when you perform various administrative tasks.

Your Information	Example Entry	Your Entry
ARCOT_HOME	C:\Program Files\Arcot Systems	
SYSTEM INFORMATION		
Host Name	my-bank	
User Name	administrator	
Password	password1234!	
Configured Components	RiskFort 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	

Your Information	Example Entry	Your Entry
Application Context Root	arcotuds	

Reinstalling RiskMinder on a Distributed System

To reinstall RiskMinder on a distributed system, perform the tasks that are described in the following sections:

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

1. Installing on the First System (see page 334)

   Note: While you install RiskMinder 3.1.01, ensure that you specify the same primary and backup database details from arcotcommon.ini in the $ARCOT_HOME/conf/ directory.

2. Verifying the Database Setup (see page 113)
3. Preparing Your Application Server (see page 114)
4. Deploying Administration Console (see page 350)
5. Logging In to Administration Console (see page 123)

   Important! Ensure that you use the current MA password and *not* the default password, because the MA password has been reset during the bootstrap process that you performed during 2.x installation.

6. Starting RiskMinder Server (see page 126)
7. Starting the Case Management Queuing Server (see page 127)
8. Deploying User Data Service (UDS) (see page 354)
9. Installing on the Second System (see page 356)
10. Deploying Sample Application on the Second System (see page 131)
11. Configuring Sample Application for Communication with RiskMinder Server (see page 358)
12. Verifying the Installation (see page 127)

    Note: If there are any warnings during the Server startup, and if your transactions fail, then the upgrade has not been performed successfully. You can revert to your initial setup by following the steps that are listed in (In Error Scenario Only) Reverting to Your Initial Setup (see page 361).

13. Using Sample Application (see page 359)
14. Applying the Post-Installation Checklist (see page 138)

Installing on the First System

To install (and later configure) RiskMinder on Microsoft Windows successfully, the user account that you plan to use for the installation *must* belong to the Administrators group. Otherwise, some critical steps in the installation, such as DSN creation and configuration, and the RiskMinder service creation, do not complete successfully, though the installation may complete without any errors.

In a distributed scenario, irrespective of how many systems you are distributing RiskMinder, Administration Console, Java SDKs, and Web services across, you typically install RiskMinder 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 you proceed with the installation, ensure that all prerequisite software components are installed and the database is set up, as described in "Preparing for Installation" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

To install the RiskMinder components, perform the following tasks:

1. Navigate to the directory where the Arcot-RiskFort-3.1.01-Windows-Installer.exe file is located and double-click the file to run the installation wizard.

   The Welcome screen appears.

2. Click Next to continue.

   The License Agreement screen appears.

3. Carefully read the license agreement, select the I accept the terms of the License Agreement option, and click Next.

   The installer now checks if any other CA product is installed on the computer.

   If it does not find an existing CA product installation, then you are prompted for an installation directory. In this case, the Installation Location screen appears.

   If the installer detects an existing CA product installation (an existing ARCOT_HOME), then:

   - You are not prompted for an installation directory.
   - In addition, you are not prompted for the database and encryption setup. The installer uses the existing database and encryption settings. As a result, you see the screen in Step 8, though the configuration is disabled, and the screen corresponding to Step 12 is not displayed at all.

4. Click Next to install in the specified directory.

   The Installation Type screen appears.

5. Select Custom to install the selected components in one ARCOT_HOME.

   The Component Selection screen appears.

6. Deselect the components that are not required. By default, all components are selected for the installation.

   For example, to install thev RiskMinder Server, Case Management Queuing Server, and Administration Console (*without* the SDKs and Sample Application) on the current system, select *only* the following options:

   a. Arcot Risk Evaluation Server

   b. Arcot Case Management Queuing Server

   c. Arcot Administration Console

   d. Arcot User Data Service

   Note: To install Sample Application *only*, select the Arcot RiskFort SDKs and Sample Application option and then proceed with the installation.

   The following table describes all components that are installed by the RiskMinder installer.

Component	Description
Arcot Risk Evaluation Server	This option installs the core Processing engine (RiskMinder Server) that serves the following requests from Administration Console: ■ Risk Evaluation ■ Configuration In addition, this component also installs the following Web services that have been built into the server: ■ **Risk Evaluation Web Service:** Provides the web-based programming interface for risk evaluation with RiskMinder Server. ■ **User Management Web Service**: Provides the web-based programming interface for the creation and management of users. ■ **Administration Web Service**: Provides the web-based programming interface that is used by the RiskMinder Administration Console.
Arcot Case Management Queuing Server	This option installs the core Queuing engine (Case Management Queuing Server) that allocates cases to the Customer Support Representatives (CSRs) who work on these cases. **Note:** At any given point in time, *all* instances of Administration Console can only connect to this single instance of Case Management Queuing Server.

Component	Description
Arcot RiskFort SDKs and Sample Application	This option provides programming interfaces (in form of APIs and Web services) that can be invoked by your application to forward risk evaluation requests to RiskMinder Server. This package comprises the following sub-components: ■ **Risk Evaluation SDK**: Provides the Java programming interface for risk evaluation with RiskMinder Server. ■ **Sample Application**: Demonstrates the usage of RiskMinder Java APIs. In addition, it can also be used to verify if RiskMinder was installed successfully, and if it is able to handle the risk evaluation requests. For more information on configuring these components, see "Configuring RiskMinder SDKs and Web Services" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.
Arcot Administration Console	This option provides the Web-based interface for managing RiskMinder Server and risk evaluation-related configurations.
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.)

Note: If you did not select the Arcot Risk Evaluation Server option on this screen, then screens in Step 7 through Step 9 do not appear.

1. Select Next to continue.

   The Database Type screen appears.

2. Depending on the type of database you have, you can select Microsoft SQL Server, Oracle Database, or MySQL. Click Next to proceed.

   If you selected Microsoft SQL Server on the Database Type screen, then the SQL Server Database Details screen appears.

   Note: If you are using a SQL database, then ensure that the ODBC Driver version you are using is the same as the one mentioned in the "Configuring Database Server" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

   If you selected Oracle Database on the Database Type screen, then the Oracle Database Details screen appears.

   Note: CA RiskMinder release 3.1.01 is now certified to work with Oracle Real Application Clusters (Oracle RAC). To use Oracle RAC with your RiskMinder Installation, select Oracle Database in this step, perform the next step (Step 9), and then perform the steps in Configuring CA RiskMinder for Oracle RAC (W) (see page 247).

3. Based on your database choice in the preceding screen:

   - If you selected Microsoft SQL Server, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.
Server	The host name or IP address of the RiskMinder datastore. Default Instance ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase Named Instance ■ **Syntax**: *<server_name>**<instance_name>* ■ **Example**: demodatabase\instance1

Parameter	Description
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.
Password	The password associated with the User Name you specified in the previous field and which is used by RiskMinder to access the database. This password is specified by the database administrator.
Database	The name of the MS SQL database instance.
Port Number	The port at which the database listens to incoming requests. The default port at which an MS SQL database listens is 1433. However, if you would like to specify another port, enter the port value in this field.

- If you selected Oracle Database, then fill in the following information in the fields.

Parameter	Description
ODBC DSN	The installer creates the DSN by using this value. RiskMinder Server then uses this DSN to connect to the RiskMinder database. The recommended value to enter is arcotdsn. **Note:** Database Source Name (DSN) specifies the information that is required to connect to a database by using an ODBC driver. This information includes database name, directory, database driver, User ID, and password.
User Name	The database user name for RiskMinder to access the database. This name is specified by the database administrator. (MS SQL Server, typically, refers to this as login.) This user *must* have the create session and DBA rights. **Note:** The User Name for the Primary and Backup DSNs *must* be different.

Parameter	Description
Password	The password associated with the User Name you specified in the previous field and which is used by RiskMinder 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. The default port at which an Oracle database listens is 1521. However, if you would like to specify another port, enter the port value in this field.
Host Name	The host name or IP address of the RiskMinder datastore. ■ **Syntax**: *<server_name>* ■ **Example**: demodatabase

1. After you specify the database details, test if you can successfully connect to the database by clicking the Test Data Source button and verify the result of the same in the field below the button.

   Note: If the connection was not successful, ensure that you have specified the correct database details and click Test Data Source again. Proceed only if the database connectivity is successful.

2. Click Next to continue.

   The Encryption Setup screen appears. Use this screen to select the encryption mode and configure the information that was used for encryption.

3. Specify the following information:

Field Name	Description
Master Key	Specify the password for the Master Key, which is stored at *<install_location>*\Arcot Systems\conf\securestore.enc and will be used to encrypt the data stored in the database. By default, this value is set to MasterKey. **Note:** If you want to change the value of Master Key *after* the installation, then regenerate securestore.enc with a new Master Key value. For more information, see appendix, "Changing Hardware Security Module Information After the Installation" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Field Name	Description
Configure HSM	Enter **y** if you want to use a Hardware Security Module (HSM) to encrypt the sensitive data. Alternatively, enter n to use the software encryption. If you do not select this option, then, by default, the data is encrypted by using the Software Mode.
PIN	Enter the password to connect to the HSM.
Choose Hardware Module	Choose one of the following HSMs that you plan to use: ■ Luna HSM ■ nCipher netHSM
HSM Parameters **Note:** The HSM parameter values are recorded in arcotcommon.ini, which is available in <install_location>\Arcot Systems\conf\. To change these values *after* installation, edit this file, as discussed in "Configuration Files and Options" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.	Set the following HSM information: ■ **Shared Library:** The absolute path to the PKCS#11 shared library corresponding to the HSM. For Luna (cryptoki.dll) and for Cipher netHSM (cknfast.dll), enter the absolute path and name of the DLL. ■ **Storage Slot Number:** The HSM slot where the 3DES keys used for encrypting the data are available. – For Luna, the default value is 0. – For nCipher netHSM, the default value is 1.

The Pre-Installation Summary screen appears.

1. Review the information on this screen, and if you must change a previous selection, then click Previous to do so. After you change the required selection, click Next to go to the next screen.
2. Click Install to begin the installation process.

   The Microsoft Visual C++ 2010 x86 Redistributable Setup screen appears. This screen appears only if the current system where you are installing RiskMinder does not have Microsoft Visual C++ 2010 x86.
3. On the Microsoft Visual C++ 2010 x86 Redistributable Setup screen:

   a. Select the I have read and accept the license terms option, and click Install.

      The Installation Progress screen appears. This may take a few seconds. After some time, the Installation Is Complete screen appears.

   b. Click Finish to close the Microsoft Visual C++ 2010 x86 Redistributable Setup dialog and continue with the RiskMinder installation.

   The Installing Arcot RiskFort screen appears. This may take several minutes. After some time the Install Complete screen appears.
4. Click Done to complete the installation.

Installation Logs

After installation, you can access the installation log file (Arcot_RiskFort_Install_<*timestamp*>.log) in the <*install_location*> directory. For example, if you had specified the C:\Program Files directory as the installation directory, then the installation log file is created in the C:\Program Files directory.

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

Verifying the Database Setup

After you run the required database scripts, verify that the RiskMinder schemas were seeded correctly. To do so:

1. Log in to the RiskMinder database as the user who installed the database.

   Note: If you are following the upgrade path, then log in to the database as the user who upgraded the database.

2. Run the following query:
   ```
   SELECT SERVERNAME, VERSION FROM ARRFSERVERS;
   ```

   You must see the following output as a result of the preceding query:
   ```
   SERVERNAME                    VERSION
   ------------------------  ---------------
   RiskFort                       3.1.01
   RiskFortCaseManagement         3.1.01
   ```

3. Log out of the database console.

Preparing Your Application Server

Two components of RiskMinder, 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 Application Server

Before you deploy the WAR files for these web applications on the application server of your choice, copy the files that UDS and Administration Console require 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 and to deploy the WAR files of these web applications:

- Step 1: Setting Java Home (see page 114)
- Step 2: Copying Database Access Files to Your Application Server (see page 115)
- Step 3: Copying JDBC JAR Files to Your Application Server (see page 118)
- Step 4: (Mandatory for Oracle WebLogic 10.1) Creating Enterprise Archive Files (see page 120)

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. This JAVA_HOME must be your application server JAVA_HOME.

In addition, %JAVA_HOME%**bin**\ must be added to the PATH variable. If you fail to do so, then Administration Console, UDS, 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 RiskMinder database securely:

- arcot-crypto-util.jar available at:
  <install_location>\Arcot Systems**java****lib**\
- ArcotAccessKeyProvider.dll available at:
  <install_location>\Arcot Systems**native****win**\<**32bit-or-64bit**>\

As a result, these files must be copied to the appropriate location on the application server where you have deployed these RiskMinder 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 files:

1. Copy arcot-crypto-util.jar *to <Tomcat_JAVA_HOME>*\jre\lib**ext**\.

   Here, *<Tomcat_JAVA_HOME>* represents the JAVA_HOME used by your Apache Tomcat instance.

2. Copy ArcotAccessKeyProvider.dll *to <Tomcat_JAVA_HOME>*\jre**bin**\.
3. Restart the application server.

IBM WebSphere

To copy the files:

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 or 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, C:\Program Files\Arcot Systems\java\lib**arcot-crypto-util.jar**.

   e. Enter the JNI Library path.

This path must point to the location where the ArcotAccessKeyProvider.dll file is present.

3. Click **Apply** to save the changes.
4. Configure the 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 must be performed.
   c. Click **Java and Process Management** and then click **Class Loader**.
   d. Click **New**.
   e. Select default **Classes loaded with parent class loader first** and click **OK.**
   f. Click the auto-generated **Class Loader ID**.
   g. On the class loader **Configuration** page, click **Shared Library References**.
   h. Click **Add**, select **ArcotJNI,** and then click **Apply**.
   i. Save the changes.
5. Copy ArcotAccessKeyProvider.dll to <*WebSphere_JAVA_HOME*>\jre**bin**\.

   Here, <*WebSphere_JAVA_HOME*> represents the JAVA_HOME used by your IBM WebSphere instance.
6. Restart WebSphere.

Oracle WebLogic

To copy the files:

1. Copy ArcotAccessKeyProvider.dll to <*WebLogic_JAVA_HOME*>\jre**bin**\.

   Here, <*WebLogic_JAVA_HOME*> represents the JAVA_HOME used by your Oracle WebLogic instance.
2. Copy arcot-crypto-util.jar to <*WebLogic_JAVA_HOME*>\jre\lib**ext**\.

   Note: Ensure that you use the appropriate <*JAVA_HOME*> used by WebLogic.
3. Log in to WebLogic Administration Console.
4. Navigate to **Deployments**.
5. Enable the **Lock and Edit** option.
6. Click **Install** and navigate to the directory that contains the arcot-crypto-util.jar file.
7. Click **Next** to open the Application Installation Assistant.
8. Click **Next** to display the Summary page.
9. Click **Finish**.
10. Activate the changes.

11. Restart the server.

JBoss Application Server

To copy the files:

1. Copy ArcotAccessKeyProvider.dll *to* <*JBoss_JAVA_HOME*>\jre**bin**\.

   Here, <*JBoss_JAVA_HOME*> represents the JAVA_HOME used by your JBoss Application Server instance.

2. Copy arcot-crypto-util.jar *to* <*JBoss_JAVA_HOME*>\jre\lib**ext**\.

3. Restart the application server.

Step 3: Copying JDBC JAR Files to Your Application Server

RiskMinder requires the following JDBC JAR files for the supported databases:

- **Oracle 10g:** Oracle JDBC Driver (10.2.0.1.0)
- **Oracle 11g:** Oracle JDBC Driver (11.2.0.2.0)
- **Microsoft SQL Server:** MSSQL JDBC Driver (1.2.2828)
- **MySQL:** MySQL JDBC Driver (5.1.22)

The following subsections 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 required JDBC JAR file:

1. Navigate to the location where you have downloaded the *<Database_JAR>* file.
2. Copy the *<Database_JAR>* file to the following directory:
   - **On Apache Tomcat 5.5.x:** *<TOMCAT_HOME>*\common\lib\
   - **On Apache Tomcat 6.x and 7.x:** *<TOMCAT_HOME>*\lib\
3. Restart the application server.

IBM WebSphere

To copy the required JDBC JAR file:

1. Log in to the WebSphere Administration Console.
2. Click Environment, and then click Shared Libraries.
   a. From the Scope list, select a valid visibility scope. The scope must include the target server or node on which the application is deployed.
   b. Click New.
   c. Enter the Name, say, JDBCJAR.
   d. Specify the Classpath.

      Important! This path must point to the location where the <Database_JAR> file is present and must include the file name.

   e. Click Apply to save the changes that were made.
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 is performed.
c. Click Java and Process Management, and then click Class Loader.
d. Click New.
e. Select default Classes loaded with parent class loader first and click OK.
f. Click the auto-generated Class Loader ID.
g. In the class loader Configuration page, click Shared Library References.
h. Click Add, select JDBCJAR, and then click Apply.
i. Save the changes that were made.

4. Restart the application server.

Oracle WebLogic

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

To copy the required JDBC JAR file in case of Microsoft SQL Server:

1. Copy the <*Database_JAR*> file to <*Weblogic_JAVA_HO*ME>\lib\ext\.

   Here, <WebLogic_JAVA_HOME> represents the JAVA_HOME used by your Oracle WebLogic instance.

2. Log in to the 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 required <Database_JAR> file.
6. Click Next to display the Application Installation Assistant page.
7. Click Next to display the Summary page.
8. Click Finish.
9. Activate the changes.
10. Restart the application server.

JBoss Application Server

To copy the required JDBC JAR file:

1. Copy the JDBC JAR file to the following location on the JBOSS installation directory: <*JBOSS_HOME*>\server\default\lib\
2. Restart the application server.

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

Most enterprise Application Servers (such as WebSphere and WebLogic) enable you to bundle the related Java ARchive (JAR) or Web ARchive (WAR) files from one vendor (say, CA) to 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, WAR files are provided to deploy UDS and Administration Console. However if necessary, you can also change the format of these files to Enterprise ARchive (EAR) and then deploy the EAR files.

As discussed in the following subsections, you can either generate separate EAR files for both UDS and Administration Console, or you can generate a single EAR file that contains both Web archives.

Generating Separate EAR Files

To create a separate EAR file each for UDS and Administration Console, follow these steps:

1. Open the Command Prompt window.
2. Navigate to the <*install_location*>\Arcot Systems\tools\common**bundlemanager**\ directory.
3. To create the EAR file, run the following command:
   `java -jar bundle-manager.jar -ear <filename.ear> -warList <filename.war>`

   The preceding command generates individual EAR files that are available at:
   `<install_location>\Arcot Systems\java\webapps\`

Generating a Single EAR File

To create a single EAR file that contains UDS and Administration Console Web archives:

1. Open the Command Prompt window.
2. Navigate to the <*install_location*>\Arcot Systems\tools\common**bundlemanager**\ directory.
3. To create the EAR file, run the following command:
   `java -jar bundle-manager.jar -ear <filename.ear> -warList arcotadmin.war arcotuds.war`

   The preceding command generates a single EAR file that is available at:
   `<install_location>\Arcot Systems\java\webapps\`

Deploying Administration Console

Note: If you are deploying the Administration Console on IBM WebSphere 7.0, then instead of the following instructions, see the instructions in "Deploying Administration Console on IBM WebSphere 7.0" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Administration Console is a browser-based interface to RiskMinder that enables you to customize the server configurations and manage the deployed system.

You need the **arcotadmin.war** file to deploy the RiskMinder Administration Console. All Administration Console information is logged in the arcotadmin.log file. After you deploy arcotadmin.war, you can verify if it was correctly deployed by using this log file (arcotadmin.log). This log file is in the %ARCOT_HOME%\Arcot Systems\logs directory.

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

To deploy the Administration Console WAR file on your application server, and to verify if it was successfully deployed, follow these steps:

1. Deploy arcotadmin.war in the appropriate directory on the application server.

   Note: The deployment procedure depends on the application server that you are using. See 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**\.

2. **(For 32-bit WebSphere Only)** Configure reload of the Admin class when the application files are updated.

   a. Navigate to Application > Enterprise Applications and access the Admin 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.

   e. Restart the Admin application.

3. **(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 Systems\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>
Add the following category for cryptographic operations:
<category name="com.arcot.crypto.impl.NCipherCrypter">
<priority value="FATAL" />
<appender-ref ref="arcotadminlog"></appender-ref>
</category>
```

f. Save and close the file.

g. Take a backup of the existing JBoss logging libraries. These library files are available at:
<*JBOSS_HOME*>\lib\

h. 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/*

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

4. Restart the application server.
5. Verify that the console was successfully deployed:
   a. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`
   b. Open the arcotadmin.log file in any editor and locate the following lines:
      - 2.0.3
      - Arcot Administration Console Configured Successfully.

      These lines indicate that your Administration Console was deployed successfully.
   c. Also ensure that the log files *do not* contain any FATAL *and* WARNING messages.
   d. Close the file.

Logging In to Administration Console

When you log in to Administration Console for the first time, use the Master Administrator (MA) credentials that are configured automatically in the database during the deployment.

To log in to Administration Console as MA:

1. Launch the Administration Console in a Web browser window. The default URL for Administration Console is:
   `http://<host>:<appserver_port>/arcotadmin/masteradminlogin.htm`

   Note: The *host* and *port* information that you specify in the preceding URL must be of the application server where you deployed Administration Console.
   For example, in case of Apache Tomcat, the default *host* is localhost and *port* is 8080.
2. Log in by using the default Master Administrator account credentials. The credentials are:
   - **User Name: masteradmin**
   - **Password: master1234!**

Starting RiskMinder Server

To start RiskMinder Server:

1. Click the **Start** button on your desktop window.
2. Navigate to **Settings, Control Panel, Administrative Tools,** and then **Services.**
3. Locate and double-click **Arcot RiskFort Service**.
4. Click **Start** in the service window.

Note: If you want to stop RiskMinder Server, then follow the Steps 1 through 3, and click the **Stop** button in the service window.

Starting the Case Management Queuing Server

To start Case Management Queuing Server:

1. Click the **Start** button on your desktop window.
2. Navigate to **Settings, Control Panel, Administrative Tools,** and then **Services.**
3. Locate and double-click the **Arcot RiskFort Case Management Queuing Service**.
4. Click **Start** in the service window.

Note: If you want to stop the Case Management Queuing Server, then follow the Steps 1 through 3, and click the **Stop** button in the service window.

Deploying User Data Service (UDS)

RiskMinder can access user data either from a relational database (RDBMS) or directly from an LDAP server by using UDS, which is an abstraction layer that provides RiskMinder seamless access to the third-party data repositories deployed by your organization.

You need the arcotuds.war file to deploy UDS, as follows:

1. Deploy arcotuds.war on the application server. This file is available at: <*install_location*>\Arcot Systems\java\webapps\

   For example, in the case of Apache Tomcat, deploy the WAR file at <*APP_SERVER_HOME*>\webapps\.

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

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 to save the changes.

3. (For JBoss Only) Perform the following steps if you have deployed UDS on a JBoss application server:

   a. Copy the Bouncy Castle JAR file (bcprov-jdk15-146.jar) from <*install_location*>\Arcot Systems\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>
```

Add the following line in the com.arcot category that you created in Deploying Administration Console (see page 350):

```
<appender-ref ref="arcotudslog"></appender-ref>
```

e. Add the following line in the cryptographic category that you created in Deploying Administration Console (see page 350):

```
<appender-ref ref="arcotudslog"></appender-ref>
```

f. Save and close the file.

4. Restart the application server.
5. Verify if UDS was deployed successfully:

   Note: The arcotuds.log file is used for logging UDS-related information.

   a. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`

   b. Open the arcotuds.log file in any editor and locate the following line:

      - User Data Service (Version: 2.0.3) initialized successfully.

      This line indicates that UDS was deployed successfully.

   c. Also ensure that the log files *do not* contain any FATAL *and* WARNING messages.

   d. Close the file.

RM_3.1--Installing on the Second System (scenario) (distributed)

After you install the RiskMinder Server and Administration Console, install the other remaining components on the second system in this distributed environment. The specific components to install must have been determined when you performed your planning in "Planning the Deployment" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Note: Before you proceed with the installation, ensure that all prerequisite software components are installed on this system as described in "Preparing for Installation" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

To install the RiskMinder components on the subsequent system:

1. Copy the installer file Arcot-RiskFort-3.1.01-Windows-Installer.exe on the target (second) system.
2. Double-click the installer to run it.
3. Follow the installer instructions from Step 2 in Installing on the First System (see page 334) until you reach the Choose Install Set screen.
4. Select the components that you want to install.

   Typically, you install the Java SDKs for Risk Evaluation and Sample Application.
5. After you have selected all the components, follow the steps from Step 7 through Step 16 in Installing on the First System (see page 334) to complete the installation process.

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 RiskMinder was installed and configured properly. In addition, it demonstrates:

- The typical RiskMinder workflows
- The basic operations (invocation and post-processing) of RiskMinder APIs
- Integration of your application with RiskMinder

Note: If you did not install Sample Application during the installation, then you can install *only* Sample Application by running the installer again and by selecting the SDKs and Sample Application options and proceed with the installation.

To deploy Sample Application on your application server:

1. Deploy the riskfort-3.1.01-sample-application.war file from the following location: `<install_location>\Arcot Systems\samples\java\`
2. If necessary, restart the application server.
3. Access Sample Application in a Web browser window. The default URL for Sample Application is: `http://<host>:<appserver_port>/riskfort-3.1.01-sample-application/index.jsp`

Configuring Sample Application for Communication with RiskFort Server

The riskfort.risk-evaluation.properties file provides the parameters for the Java SDK and Sample Application to read RiskMinder Server information. Therefore, after deploying Sample Application, configure it to communicate with RiskMinder Server. This file is only available *after* you deploy the RiskFort Sample Application WAR file, riskfort-3.1.01-sample-application.war.

To configure the riskfort.risk-evaluation.properties file:

1. Navigate to the riskfort.risk-evaluation.properties file on your application server.

   In case of Apache Tomcat, this file is available at:
   `<App_Home\riskfort-3.1.01-sample-application>\WEB-INF\classes\properties\`

   Here, <App_Home\riskfort-3.1.01-sample-application\> represents the directory path where RiskMinder application WAR files are deployed.

2. Open the riskfort.risk-evaluation.properties file in an editor window and set the value for the following parameters:

   - HOST.1
   - PORT.1

   A default value is specified for the remaining parameters in the file. You can change these values, if necessary. For more information about configuration parameters, see "riskfort.risk-evaluation.properties" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

3. **Optional:** Perform this step only if you configured SSL-based communication.

   Set the following parameters:

   - TRANSPORT_TYPE=SSL (By default, this parameter is set to TCP.)
   - CA_CERT_FILE=<*absolute_path_of_Root_Certificate_in_PEM_FORMAT*>

     For example, you can specify one of the following values:

     - CA_CERT_FILE=<*install_location*>/certs/<ca_cert>.pem
     - CA_CERT_FILE=<install_location>\\certs\\<ca_cert>.pem

   Important! In the absolute path that you specify, ensure that you use \\ or / instead of \. This is because the change may not work, if you use the conventional \ that is used in Microsoft Windows for specifying paths.

4. Save the changes and close the file.
5. To ensure that these changes are reflected, restart the application server.

Verifying the Installation

To verify if the server started correctly:

1. Navigate to the following location:
   `<install_location>\Arcot Systems\logs\`
2. Open the arcotriskfortstartup.log file in any editor and locate the following lines:
   - STARTING Arcot RiskFort 3.1.01_w
   - Arcot RiskFort Service READY
3. Open the arcotriskfortcasemgmtserverstartup.log file in any editor and locate the following lines:
   - STARTING Arcot RiskFort Case Management 3.1.01_w
   - Arcot RiskFort Case Management Service READY

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

Using Sample Application

The following risk-evaluation operations can be performed by using Sample Application. Each of these operations is designed to run without error when RiskMinder is installed and functional.

- Performing Risk Evaluation and Post Evaluation for a First-Time User
- Creating Users
- Performing Risk Evaluation and Post Evaluation for a Known User
- Editing the Default Profile and Performing Risk Evaluation

Note: For information about running these operations, see the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Applying the Post-Installation Checklist

It is recommended that you fill the following checklist with the installation and setup information for RiskMinder. This information is useful when you perform various administrative tasks.

Your Information	**Example Entry**	**Your Entry**
ARCOT_HOME	C:\Program Files\Arcot Systems	
SYSTEM INFORMATION		
Host Name	my-bank	
User Name	administrator	

Your Information	Example Entry	Your Entry
Password	password1234!	
Configured Components	RiskFort 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	

(In Error Scenario Only) Reverting to Your Initial Setup

During upgrade, if there are any warnings during the Server startup and if your transactions fail, then you may want to revert to your initial setup.

To revert to the initial setup:

1. Uninstall RiskMinder 3.1.01.

   Note: For information about the procedure to uninstall RiskMinder, see "Uninstalling RiskMinder" in the *CA RiskMinder Installation and Deployment Guide for UNIX Platforms*.

2. Install the RiskMinder release to which you want to revert. For example, 1.*x* or 2.*x*.

   Note: For installation instructions, see the *CA RiskMinder Installation and Deployment Guide* that is shipped with the corresponding release.

3. Navigate to the location where the ARCOT_HOME_BACKUP directory is available.
4. Copy the contents of ARCOT_HOME_BACKUP to your current ARCOT_HOME.
5. Replace the ArcotAccessKeyProvider.dll file in *<JAVA_HOME used by Application Server>*\jre\bin with the backup that you created while performing the procedure described in Migrating the Database to Release 2.2.7 for Arcot Common Components (see page 155).
6. Deploy the web components, such as the Administration Console and UDS.
7. Restore the database from the backup that you had taken before you began the upgrade procedure.
8. Start RiskMinder Server and Case Management Queuing Server.
9. Test the installation.

Performing Post-Upgrade Tasks

This section describes the tasks that you must perform after upgrading to release 3.1.01.

Follow these steps:

1. If you disabled database replication before upgrade, then after you upgrade to RiskMinder 3.1.01 enable replication for the backup database.
2. If you configured SSL for the following ports in RiskMinder 2.2.7, then reconfigure SSL.
   - Port 7980 for Server Management protocol of the RiskMinder Server instance
   - Port 7780 for Case Management Queuing Administration protocol of the Case Management Queuing Server instance

   Reconfigure SSL as follows:
   - Between Administration Console and RiskMinder Server: Port 7980
   - Between Administration Console and Case Management Queuing Server: Port 7780

   This configuration is required because most administrative tasks, such as instance management and protocol configuration, are done using these ports in Administration Console in release 3.1.01.

   Note: For instructions on setting up SSL between Administration Console and RiskMinder Server or Case Management Queuing Server, see "Configuring SSL" in the *CA RiskMinder Administration Guide*.
3. Set the Base Currency Code for your organization from the Miscellaneous Configurations screen.

   Note: For more information about setting the organization-specific base currency code, see Managing Global Configurations" in the *CA RiskMinder Administration Guide*.
4. If there are any rules with a score of 0 and you want to use these rules for scoring, then change the score to a nonzero value, like 1 or 2.

Replacing Deprecated Rules with New Rules

Four of the predefined rules have been deprecated in release 3.1. Alternative rules have been introduced for these deprecated rules. The following table lists the deprecated and new rules and rule mnemonics:

Deprecated Rule Name and Rule Mnemonic	New Rule Name and Rule Mnemonic
DeviceID Known (DEVICEIDCHECK)	Unknown DeviceID (UNKNOWNDEVICEID)
Device MFP Match (SIGMATCH)	Device MFP Not Match (MFPMISMATCH)
User Associated with DeviceID (USERDEVICEASSOCIATED)	User Not Associated with DeviceID (USERDEVICENOTASSOCIATED)
User Known (USERKNOWN)	Unknown User (UNKNOWNUSER)

Important! Although these rules have been deprecated, they are still available and can be used after the upgrade. However, it is recommended that you replace each deprecated rule with the corresponding new rule by making the required changes in the rule expression.

For any of the four deprecated rules, if the rule evaluates to No, then the rule is considered to have matched and it is used for scoring. In contrast, each of the other predefined rules is considered to have matched when they evaluate to Yes.

In each of the four new rules that is introduced in release 3.1, if the rule evaluates to Yes, then the rule is considered to have matched. In this way, the four new rules are consistent with the other predefined rules.

The following table lists examples that highlight the difference between the deprecated rules and new rules:

Sample Use Case	Deprecated Rule	Deprecated Rule Result	New Rule	New Rule Result
User does not exist in the RiskMinder database.	USERKNOWN	No	UNKNOWNUSER	Yes
DeviceID does not exist in the RiskMinder database.	DEVICEIDCHECK	No	UNKNOWNDEVICEID	Yes

MFP does not exist in the RiskMinder database.	SIGMATCH	No	MFPMISMATCH	Yes
User is not associated with the DeviceID.	USERDEVICEASSOCIATED	No	USERDEVICENOTASSOCIATED	Yes

Follow these steps:

1. Log in to the administration console.
2. In the Rule Configurations Report for all organizations and rulesets, verify whether any of the mnemonics that are listed in the Rule expression column of the report belong to the list of deprecated mnemonics.
3. If a rule uses a deprecated mnemonic and if you do not want to use the deprecated mnemonic, use the corresponding new mnemonic.

   To modify a rule expression:

   a. Log in to the administration console as the GA or OA.

   b. If you have logged in as the GA and you want to perform this procedure for a system ruleset, click the Services and Server Configurations tab.

   c. If you have logged in as the GA or OA to perform this procedure for a single organization:

      Activate the Organizations tab.

      Click the Search Organization link under Manage Organizations.

      Click the Search button on the Search Organization page to display the list of organizations.

      Click the name of the organization.

      Click the RiskFort Configuration tab.

   d. Under the Rules Management section on the side-bar menu, click the Rules and Scoring Management link.

      The Rules and Scoring Management page appears.

   e. From the Select a Ruleset list, select the ruleset for which this configuration is applicable.

      The configuration information for the specified ruleset appears.

   f. Click the rule that you want to modify.

      The Rule Builder page opens.

   g. Make the required changes in the Rule being developed text field.

   h. Save the changes and close the Rule Builder page.
4. Migrate the modified rule to the production environment, and then refresh the cache.

   Note: For detailed information about migrating a rule to the production environment and refreshing the cache, see the *CA RiskMinder Administration Guide*.

Reviewing Configuration Changes After the Upgrade

For information about the configuration changes made by the upgrade process, see "Reviewing Configuration Changes After Upgrade" in the *CA RiskMinder Installation and Deployment Guide for Microsoft Windows*.

Appendix J: Configuring CA RiskMinder for Oracle RAC

Perform the steps in this section if you want to use Oracle RAC with RiskMinder 3.1.01.

This section contains the following topics:

Updating the arcot-db-config-for-common-2.0.sql Script

You run database scripts as a post-installation task in the RiskMinder installation procedure. The arcot-db-config-for-common-2.0.sql script is one of the database scripts that you run. Before you run this script, modify it for Oracle RAC.

Follow these steps:

1. To determine the Oracle RAC shared datafile path, log in to the database and run the following command:

   ```
   SELECT file_name, tablespace_name FROM dba_data_files
   ```

   The following is sample output of this command:

   ```
   +DATA\qadb\datafile\users.259.797224649      USERS
   +DATA\qadb\datafile\undotbs1.258.797224649   UNDOTBS1
   +DATA\qadb\datafile\sysaux.257.797224647     SYSAUX
   ```

2. Open the arcot-db-config-for-common-2.0.sql file. This file is in the *install_location*\Arcot Systems\dbscripts\oracle\ directory.

3. Search for the following line in the file:

   ```
   filename varchar2(50) := 'tabspace_arreports_'|| to_char(current_timestamp,
   'YYYY-MM-DD-HH24-MI-SS') || '.dat';
   ```

4. Replace that line with the following line:

   ```
   filename varchar2(100) :=
   '+shared_location/service_name/datafile/tabspace_arreports_'||
   to_char(current_timestamp, 'YYYY-MM-DD-HH24-MI-SS') || '.dat';
   ```

   In the new line:

   - Replace shared_location with the shared datafile path that you determined by running the command given in the first step.
   - Replace service_name with the service name of the Oracle RAC installation.

   The following is a sample line:

   ```
   filename varchar2(100) := '+DATA/forwardinc/datafile/tabspace_arreports_'||
   to_char(current_timestamp, 'YYYY-MM-DD-HH24-MI-SS') || '.dat';
   ```

5. Save and close the script file, and then run it.

Updating the arcotcommon.ini File

The arcotcommon.ini file contains the parameters for database and instance settings. When you run the installer, database configuration data items that you enter on the installer screens are stored in this file. The JDBC URL of the database is one such data item. If you are using Oracle RAC, specify the JDBC URL in the format supported by Oracle RAC.

Follow these steps:

1. Open the arcotcommon.ini file in a text editor. This file is in the *install_location*\Arcot Systems\conf\ directory.
2. Specify a value for the URL parameter in the [arcot/db/primarydb] section and, if required, in the [arcot/db/backupdb] section of the INI file. Enter the URL in the following format:

   ```
   URL.1=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=host_name)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=service_name)(SERVER=DEDICATED)))
   ```

   For example:

   ```
   URL.1=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=172.30.250.18)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=forwardinc)(SERVER=DEDICATED)))
   ```

   Note: If Oracle RAC is client configured, then include all the nodes in this format.
3. If the database user that you specified while running the AuthMinder installer is different from the database user in Oracle RAC, then:
   a. Change the database user credentials in the arcotcommon.ini file.
   b. Use DBUtil to change the database user credentials in the securestore.enc file. DBUtil is available in the ARCOT_HOME\tools\win directory. Instructions on using DBUtil are given in Preparing for the Upgrade to Release 3.1.01 (see page 157).
4. Save and close the arcotcommon.ini file.

Updating the Database Connection Details

To establish a connection between RiskMinder and Oracle RAC, you must create an ORA file and define the address for connecting to the RAC.

Follow these steps:

1. Create a *.ora file on the system on which you have installed AuthMinder. For example, C:\Program Files (x86)\tns.ora.
2. Add the following lines in the new file:

   ```
   section_name =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = host_name_or_IP_address)(PORT = 1521))
       )
       (CONNECT_DATA =
         (SERVICE_NAME = service_name)
       )
     )
   ```

   For example:

   ```
   fwdincrac =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = 172.30.250.18)(PORT = 1521))
       )
       (CONNECT_DATA =
         (SERVICE_NAME = forwardinc)
       )
     )
   ```

   Note: If Oracle RAC is client configured, then include all the nodes in this format.
3. Save the file.
4. Modify the DSN that you created during the installation.
5. For the required DSN, clear all the parameters in the Standard Connection section.

   This makes the TNSNames Connection section editable.
6. Add the following parameters to this section:

   ```
   TNSNamesFile=ARCOT_HOME\ora_file_name
   ServerName=section_name
   ```

   For example:

   ```
   TNSNamesFile= C:\Program Files (x86)\tns.ora
   ServerName=fwdincrac
   ```

7. Save and close the file.