

# CA Risk Authentication

インストールガイド (Windows 用)

8.0



このドキュメント（組み込みヘルプシステムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。本ドキュメントは、CA が知的財産権を有する機密情報であり、CA の事前の書面による承諾を受けずに本書の全部または一部を複製、譲渡、変更、開示、修正、複製することはできません。

本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし、CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負いません。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本書の制作者は CA および CA Inc. です。

「制限された権利」のもとでの提供：アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2014 CA. All rights reserved. 本書に記載されたすべての商標、商号、サービス・マークおよびロゴは、それぞれの各社に帰属します。

## CA への連絡先

テクニカル サポートの詳細については、弊社テクニカル サポートの Web サイト (<http://www.ca.com/jp/support/>) をご覧ください。



# 目次

---

|  |           |
|--|-----------|
| <b>第 1 章: はじめに</b>                                   | <b>9</b>  |
| システム アーキテクチャ .....                                   | 10        |
| ネットワークまたはインターネット用の Web 層の使用 .....                    | 11        |
| アプリケーション サーバ用のアプリケーション層 .....                        | 12        |
| ストレージ用のデータ層 .....                                    | 13        |
| CA Risk Authentication コンポーネント間の通信 .....             | 14        |
| <b>第 2 章: 新規インストールを実行する方法</b>                        | <b>15</b> |
| 展開モデルの選択 .....                                       | 19        |
| 単一システムへの展開 .....                                     | 20        |
| 分散システムへの展開 .....                                     | 23        |
| 高可用性環境への展開 .....                                     | 27        |
| <b>第 3 章: インストール前のタスク</b>                            | <b>29</b> |
| データベース サーバの設定 .....                                  | 30        |
| Microsoft SQL Server の設定 .....                       | 31        |
| Oracle サーバの設定 .....                                  | 33        |
| MySQL サーバの設定 .....                                   | 35        |
| データ ストアおよびデータベース情報のセットアップ .....                      | 37        |
| クライアントシステムの UTF- サポートの設定 .....                       | 38        |
| HSM の要件 .....  | 38        |
| Java 依存コンポーネントの要件 .....                              | 39        |
| <b>第 4 章: 単一システムに CA Risk Authentication を展開する方法</b> | <b>41</b> |
| Complete インストールの実行 .....                             | 46        |
| データベース スクリプトの実行 .....                                | 54        |
| データベースのセットアップの確認 .....                               | 55        |
| アプリケーション サーバを準備する方法 .....                            | 56        |
| Java ホームの設定 .....                                    | 56        |
| アプリケーション サーバへのデータベース アクセス ファイルのコピー .....             | 57        |
| アプリケーション サーバへの JDBC JAR ファイルのコピー .....               | 61        |
| Enterprise Archive ファイルの作成 .....                     | 64        |
| 管理コンソールの展開 .....                                     | 66        |

---

|  |    |
|--|----|
| 管理コンソールへのログイン .....                      | 67 |
| システムのブートストラップ タスクの実行 .....               | 68 |
| CA Risk Authentication サーバ サービスの開始 ..... | 70 |
| CA Risk Authentication ケース管理サービス .....   | 71 |
| ユーザ データ サービス (UDS) の展開 .....             | 72 |
| ユーザ行動プロファイリング アプリケーションの展開 .....          | 74 |
| サンプル アプリケーションの展開 .....                   | 76 |
| インストールの確認 .....                          | 77 |
| サンプル アプリケーションをリスク評価に使用する方法 .....         | 78 |
| インストール後のチェックリストの適用 .....                 | 82 |

## 第 5 章: 分散システムに CA Risk Authentication を展開する方法 83

|  |     |
|--|-----|
| 1 つ目のシステムへのインストール .....                  | 88  |
| データベース スクリプトの実行 .....                    | 97  |
| アプリケーション サーバを準備する方法 .....                | 98  |
| Java ホームの設定 .....                        | 99  |
| アプリケーション サーバへのデータベース アクセス ファイルのコピー ..... | 99  |
| アプリケーション サーバへの JDBC JAR ファイルのコピー .....   | 103 |
| Enterprise Archive ファイルの作成 .....         | 106 |

## 第 6 章: 管理コンソールの展開 107

|                     |     |
|---------------------|-----|
| 管理コンソールへのログイン ..... | 109 |
|---------------------|-----|

## 第 7 章: ブートストラップ タスクの実行 109

|  |     |
|--|-----|
| CA Risk Authentication サーバ サービスの開始 .....   | 112 |
| CA Risk Authentication ケース管理サービスの開始 .....  | 112 |
| CA Risk Authentication サーバのインストールの確認 ..... | 113 |

## 第 8 章: ユーザ データ サービスの展開 113

## 第 9 章: ユーザ行動プロファイリング アプリケーションの展開 114

## 第 10 章: 2 つ目のシステムへの CA Risk Authentication のインストール 116

|                                   |     |
|-----------------------------------|-----|
| 2 つ目のシステムへのサンプル アプリケーションの展開 ..... | 117 |
|-----------------------------------|-----|

---

|   |            |
|---|------------|
| <b>第 11 章: CA Risk Authentication サーバと通信するためのサンプル アプリケーションの設定</b> | <b>117</b> |
| <b>第 12 章: サンプル アプリケーションをリスク評価操作に使用</b>                           | <b>119</b> |
| 初めてのユーザのリスク評価および後評価の実行 .....                                      | 120        |
| ユーザ アカウントの作成 .....  | 122        |
| 既知のユーザのリスク評価および後評価の実行 .....                                       | 123        |
| デフォルト プロファイルの編集およびリスク評価の実行 .....                                  | 124        |
| <b>第 13 章: インストール後のチェックリストの適用</b>                                 | <b>125</b> |
| <b>第 14 章: サイレントモード インストール</b>                                    | <b>126</b> |
| サイレントモードインストールのガイドライン .....                                       | 126        |
| デフォルト プロパティ ファイル .....  | 127        |
| プライマリ データベースの詳細 .....   | 129        |
| バックアップ データベースの詳細 .....  | 130        |
| 暗号化の詳細 .....  | 131        |
| サイレントインストールを実行する方法 .....  | 132        |
| <b>第 15 章: ユーザ行動プロファイリング モデルを展開する方法</b>                           | <b>133</b> |
| 前提条件の確認 .....   | 137        |
| データベースの設定 .....   | 139        |
| Microsoft SQL Server の設定 .....                                    | 140        |
| Oracle サーバの設定 .....   | 141        |
| MySQL サーバの設定 .....  | 143        |
| データベース スクリプトの実行 .....   | 144        |
| データベースのセットアップの確認 .....  | 145        |
| ユーザ行動プロファイリング ソフトウェアの展開 .....                                     | 146        |
| ユーザ行動プロファイリング モデル用の CA Advanced Authentication の設定 .....          | 148        |
| 新しいユーザ行動プロファイリング モデルを適用するルールの設定 .....                             | 149        |
| ユーザ行動プロファイリング モデルの確認 .....  | 150        |
| ユーザ行動プロファイリング モデルの削除 .....  | 150        |
| ユーザ行動プロファイリング モデルの無効化 .....                                       | 151        |
| ユーザ行動プロファイリングのアンインストール .....                                      | 152        |

---

|   |            |
|---|------------|
| <b>第 16 章: Oracle RAC 用の CA Risk Authentication の設定</b>   | <b>153</b> |
| arcot-db-config-for-common-2.0.sql スクリプトの更新 .....         | 153        |
| arcotcommon.ini ファイルの更新 .....                             | 155        |
| データベース接続の詳細の更新 .....                                      | 156        |
| <br>  |            |
| <b>第 17 章: データベース接続プールのためのアプリケーション サーバの設定</b>             | <b>157</b> |
| データベース接続プールの有効化 .....                                     | 157        |
| LDAP 接続プールの有効化 .....                                      | 166        |
| JBoss アプリケーションサーバ .....                                   | 170        |
| Apache Tomcat のセキュリティ マネージャの有効化 .....                     | 171        |
| <br>  |            |
| <b>付録 A: IBM WebSphere への管理コンソールの展開</b>                   | <b>173</b> |
| <br>  |            |
| <b>第 18 章: CA Risk Authentication SDK および Web サービスの設定</b> | <b>177</b> |
| CA Risk Authentication API の設定 .....                      | 178        |
| Java API の設定 .....  | 179        |
| CA Risk Authentication Web サービスの設定 .....                  | 181        |
| デバイス ID および DeviceDNA の設定 .....                           | 183        |
| <br>  |            |
| <b>第 19 章: カスタムアクションの追加</b>                               | <b>187</b> |
| <br>  |            |
| <b>付録 B: CA Risk Authentication のエラーのトラブルシューティング</b>      | <b>191</b> |
| インストールエラー .....   | 193        |
| データベース関連のエラー .....  | 197        |
| RiskMinder サーバのエラー .....                                  | 201        |
| SDK エラー .....   | 202        |
| <br>  |            |
| <b>第 20 章: CA Risk Authentication をアンインストールする方法</b>       | <b>202</b> |
| CA Risk Authentication スキーマの削除 .....                      | 204        |
| CA Risk Authentication サーバのアンインストール .....                 | 206        |
| アンインストール後のタスクの実行 .....                                    | 207        |



# 第 1 章: はじめに

---

CA Risk Authentication は順応性の高い認証ソリューションです。広範囲に収集されたデータを既定のルールで検査することによって、オンライントランザクションの 1 つ 1 つを評価します。評価後、CA Risk Authentication によって各トランザクションにリスク スコアとアドバイスが割り当てられます。リスク スコアが高いほど、不正行為である可能性が高くなります。このリスク スコアを利用して、トランザクションを承認または拒否したり、追加の認証を要求したり、テクニカル サポート担当者にアラートを発行したりすることができます。

CA Risk Authentication は設定可能で、ビジネス ポリシーやリスク緩和要件との整合性を保ちながら、任意のリスク評価ルールの設定パラメータを柔軟に変更することができます。また、個々のルールでデフォルトのリスク スコア、スコアリング設定、およびスコアリング優先度を変更したり、1 つ以上のルールに対して実行の有効化と無効化を選択的に指定したりすることもできます。

事前設定済みのすぐに使えるルールに加えて、ルール ビルダ機能によって、新しいルールを迅速に作成できます。

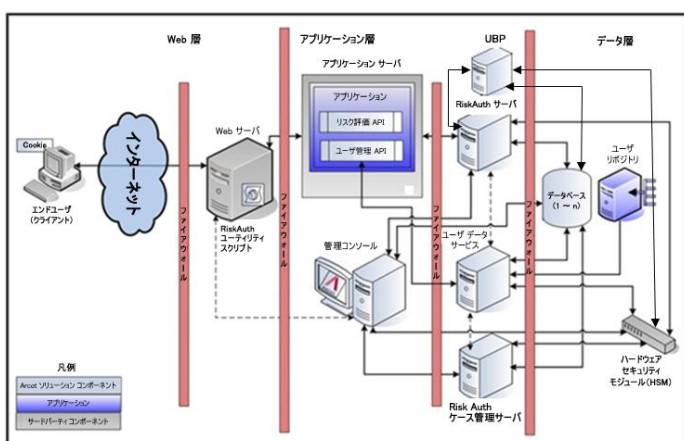
このガイドでは、さまざまなソリューション要件に基づく CA Risk Authentication の展開の計画について説明します。各シナリオは複数のコンポーネントで構成され、これらのコンポーネントが相互に、および企業内のほかのシステムや複数のネットワークで形成されるシステムと通信します。

**重要:** このガイドでは、コードオブジェクトやその他の製品の一部に Arcot、WebFort、RiskFort、WebFort、RiskMinder、AuthMinder という用語が使用されています。ArcotID は、現在、AuthID と呼ばれています。また、このガイドには標準的なフォーマットのガイドラインに従っていないトピックが一部あります。

## システム アーキテクチャ

CA Risk Authentication は、単一のシステムにインストールするか、そのコンポーネントを複数のシステムに分散してインストールできます。ただし、データとトランザクションのセキュリティと整合性を最大限に高めるためには、以下の図に示されている 3 層アーキテクチャを使用します。

- [Web 層](#) (P. 11)
- [アプリケーション層](#) (P. 12)
- [データ層](#) (P. 13)



## ネットワークまたはインターネット用の Web 層の使用

この層は HTML コンテンツで構成され、ネットワークまたはインターネットを介してユーザと直接対話します。

**CA AuthMinder** ユーティリティ スクリプト (`ArcotDeviceDNA.js`) は、ユーザのアプリケーションに含める必要があるクライアント側 JavaScript です。これは、この層に存在する Web サーバによってエンドユーザのブラウザに提供されます。このスクリプトによって、以下を実行できます。

- エンドユーザのデバイス上でデバイス ID を設定します。
- マシンフィンガープリント (MFP)、DeviceDNA およびデバイス ID 情報を収集します。

注: ユーティリティ スクリプトを使用する方法については、「**CA Risk Authentication 開発者ガイド**」の「デバイス ID および DeviceDNA の収集」を参照してください。

## アプリケーション サーバ用のアプリケーション層

この層は、システム内にあるすべてのアプリケーション サーバ コンポーネント（CA Risk Authentication サーバ、UDS、管理コンソール、CA Risk Authentication SDK など）で構成されます。以下のリストでは、各サーバ コンポーネントの機能について説明します。

**注:** この層のコンポーネントをすべて単一のシステムにインストールするか、または複数のシステムに分散できます。

- **CA Risk Authentication サーバ:** このサーバ コンポーネントは、CA Risk Authentication SDK を介したアプリケーションからのリスク評価リクエストを処理します。
- **ケース管理キュー サーバ:** このサーバ コンポーネントは、ケースをスケジュールしてテクニカル サポート担当者（CSR）に送信し、その後これらのケースのライフサイクルを管理します。
- **管理コンソール:** Web ベースのコンソールで、CA Risk Authentication コンポーネント間の通信モード、ビジネスルールとその対応データなどのサーバ インスタンスを設定したり、組織、管理者、およびユーザを管理したりするために使用します。
- **ユーザ データ サービス:** この抽象化層は、リレーショナル データベース（RDBMS）およびディレクトリ サーバ（LDAP）などの各種のユーザ リポジトリのユーザ関連データや組織関連データへのアクセスを提供します。
- **リスク評価 SDK:** このサーバ コンポーネントは、アプリケーションが CA Risk Authentication サーバに対してリスク分析リクエストを呼び出すことができる API と Web サービスを調査します。
- **リスク評価 Web サービス:** この Web ベース インターフェースは、CA Risk Authentication サーバとアプリケーション間のネットワークを介したやり取りを可能にします。リスク評価を実行するために Web アプリケーションから呼び出すことができる Web サービスで構成されます。
- **ユーザ管理 Web サービス:** この Web サービスは、CA Risk Authentication でのユーザの登録およびユーザ詳細の管理のために、ユーザ データ サービスにリクエストを転送するアプリケーションによって呼び出すことができます。

- **サンプルアプリケーション**：サンプルアプリケーションは、CA Risk Authentication Java API の使用方法およびアプリケーションと CA Risk Authentication の統合方法の例を示します。また、CA Risk Authentication が正常にインストールされているかどうかや、リスク評価操作を実行できるかどうかを確認する際にもサンプルアプリケーションを使用できます。
- **ユーザ行動プロファイリングアプリケーション**：ユーザ行動プロファイリングモデルは、データが不十分な場合に、同じユーザまたはそのピアグループのユーザによる以前のアクセスと現在のトランザクションとの類似点または相違点を測定します。

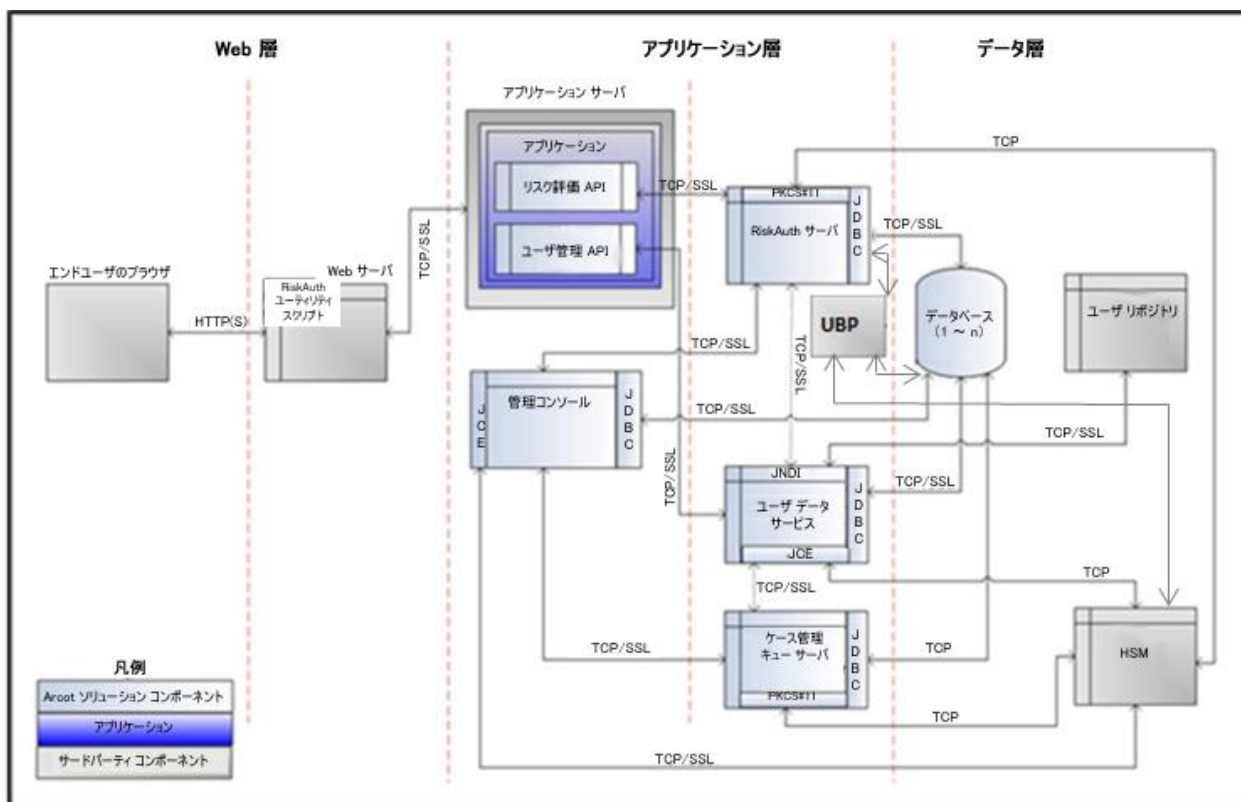
## ストレージ用のデータ層

この層は、各トランザクションを分析するために CA Risk Authentication で使用される設定、ユーザ、および履歴データを格納するリレーショナルデータベースのインスタンスで構成されます。また、この層には、ユーザの詳細の格納用に設定したすべてのディレクトリサーバ (LDAP) も含まれます。

機密ユーザデータの暗号化のためにハードウェアセキュリティモジュール (HSM) を使用する場合、HSM もこの層の一部となります。

## CA Risk Authentication コンポーネント間の通信

以下の図は、CA Risk Authentication とそのコンポーネントによってサポートされている通信モードを示しています。



コンポーネント間の通信のデフォルトモードは TCP です。CA Risk Authentication サーバは、トランザクション中に交換されるデータの整合性と機密性を確保するために、以下のコンポーネントとの SSL 通信（双方向および一方通信）をサポートしています。

- ケース管理キュー サーバ
- CA Risk Authentication データベース
- ユーザ データ サービス
- CA Risk Authentication SDK（リスク評価）
- サンプルアプリケーション
- 評価コールアウト
- スコアリング コールアウト

注: CA Risk Authentication では、ビジネス要件に基づいて評価ルールをカスタマイズすることができます。このカスタムルールは**評価コールアウト**と呼ばれます。CA Risk Authentication では、**スコアリング コールアウト**と呼ばれるスコアリング ロジックをカスタマイズすることもできます。詳細については、「CA Risk Authentication 管理ガイド」を参照してください。

## 第 2 章: 新規インストールを実行する方法

---

このシナリオは、展開モデルの選択と、各システムにインストールする **CA Risk Authentication** コンポーネントと事前インストールソフトウェアの判断について説明します。以下の図は、**CA Risk Authentication** をインストールするために実行する必要があるタスクを示しています。

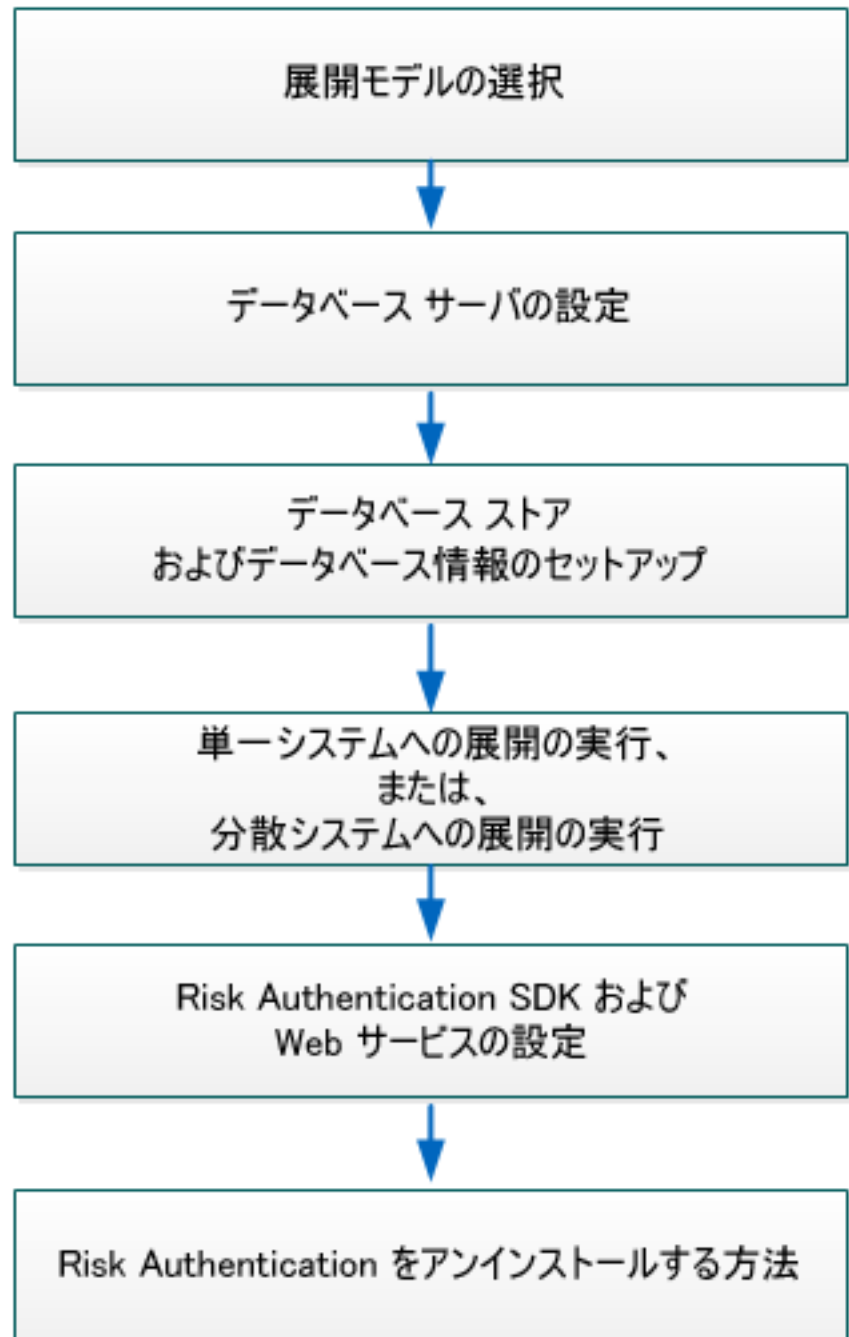
**注:** このガイドでは、システムは物理デバイスを指し、サーバはシステム上で実行されるソフトウェアを指します。



## 新規インストールを実行する方法



システム管理者



以下の手順に従います。

1. [展開モデルを選択します。](#) (P. 19)
2. [データベース サーバを設定します。](#) (P. 30)
3. [データベース ストアおよびデータベース情報をセットアップします。](#)  
(P. 37)
4. 単一システムへの展開を実行します。詳細については、「単一システムへの展開の実行」を参照してください。
5. 分散システムへの展開を実行します。詳細については、「分散システムへの展開の実行」を参照してください。
6. [CA Risk Authentication SDK および Web サービスを設定](#) (P. 177) します。

## 展開モデルの選択

CA Risk Authentication サーバはインストールする必要がある主要コンポーネントです。トランザクションリスク評価などのリスク評価サービスは、サーバによって提供されます。CA Risk Authentication サーバを使用する必要があるアプリケーションは、付属の Java SDK または Web サービスを使用して CA Risk Authentication サーバに統合できます。

CA Risk Authentication には、サーバ設定データ、ユーザ固有の基本設定、および使用データを格納するための SQL データベースも必要です。

通常、CA Risk Authentication のすべてのコンポーネントを単一のシステムにインストールします。ただし、運用展開およびステージング環境の場合は、CA Risk Authentication サーバを同じシステムにインストールします。付属の SDK または Web サービスは、ユーザがログインするアプリケーションが配置された別のシステムにインストールします。

CA Risk Authentication にはサンプルアプリケーションも付属しています。これらのアプリケーションは、CA Risk Authentication が正しくインストールされているかどうかを確認したり、リスク評価を実行したりするために使用できます。また、CA Risk Authentication を既存のアプリケーションに統合するためのサンプルコードとしても役立ちます。

CA Risk Authentication は以下の展開シナリオをサポートしています。

- **単一システム展開** - 開発環境またはテスト環境用
- **分散システム展開** - 運用環境またはステージング環境用
- **高可用性展開** - 可用性および拡張性の高い、運用環境またはステージング環境用

### 単一システムへの展開

単一システム展開では、**CA Risk Authentication** のすべてのコンポーネントとユーザがログインするアプリケーションを、単一のシステムにインストールします。データベースは、**CA Risk Authentication** がインストールされているのと同じシステム上、または異なるシステム上のどちらにあってもかまいません。

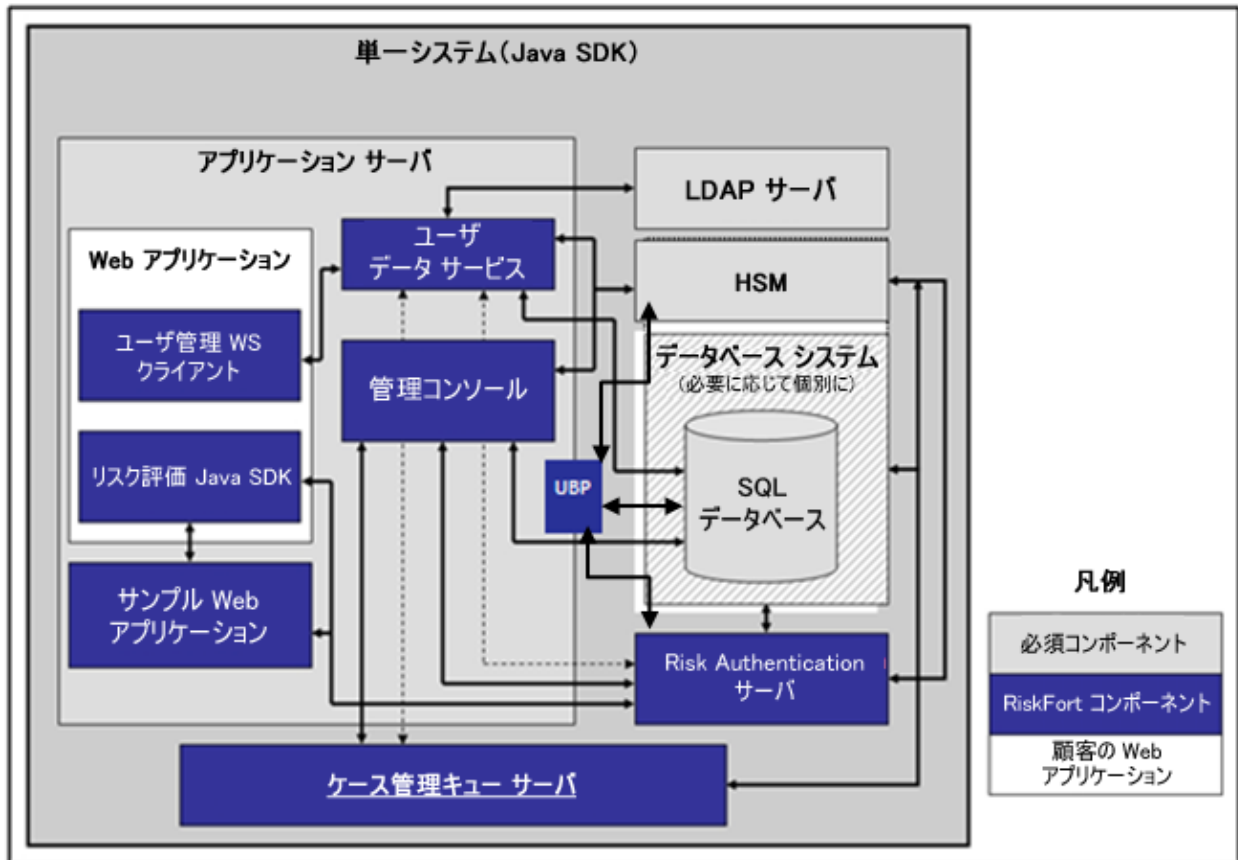
単一システム展開では **Java SDK** と **Web** サービスの両方を使用することができます。これらのコンポーネントの事前インストールソフトウェアは同じです。単一システム展開を実行する最も単純な方法は、**CA Risk Authentication** インストーラの実行中に **Complete** インストールオプションを選択することです。

単一システムへの展開を実行する場合は、以下の手順を実行します。

- a. **CA Risk Authentication** サーバが配置されているシステムにデータベースサーバをインストールします。  
別のシステム上にある既存のデータベースを使用できます。
- b. サンプルアプリケーションを使用するか、独自の **Web** アプリケーションを作成します。  
**重要:** サンプルアプリケーションを運用環境で使用しないでください。サンプルアプリケーションのコードを参考にして、独自の **Web** アプリケーションを作成してください。
- c. **Java SDK** または **Web** サービスを使用して、ご使用の **Web** アプリケーションと統合します。

#### Java SDK

以下の図は、単一システムに展開された **CA Risk Authentication** サーバおよび **Java SDK** を示しています。

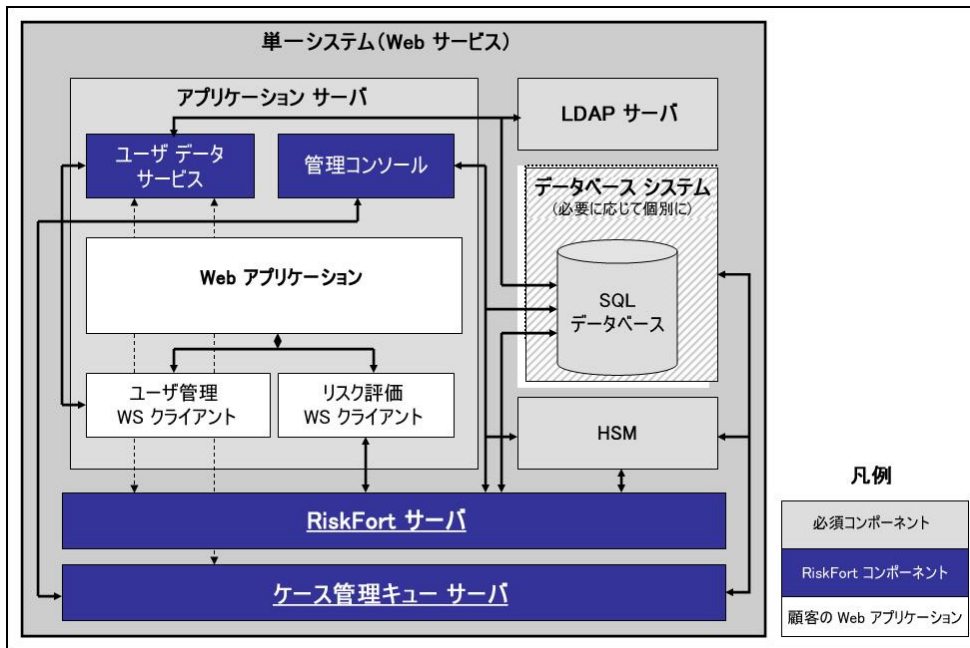


注: アプリケーションサーバの HTML ページを配信するための Web サーバの使用はオプションであり、CA Risk Authentication に対して透過的です。運用展開では、アプリケーションサーバのパフォーマンスとセキュリティを高めるため、通常はこの方法が使用されます。詳細については、アプリケーションサーバのドキュメントを参照してください。

Web サービス

以下の図は、単一システム上の CA Risk Authentication サーバおよび Web サービスを示しています。

注: すべての Web サービスは CA Risk Authentication サーバモジュール自体に組み込まれているので、CA Risk Authentication サーバをターゲットシステムにインストールし、必要なクライアントスタブを生成します。追加設定は必要ありません。



## 分散システムへの展開

分散型モデルは、コンポーネントが Web 層、アプリケーション層、およびデータ層にわたって分散された Web ベースのアプリケーションであり、Web サーバとアプリケーション サーバ間の安全なゾーンを必要とします。分散型モデルで CA Risk Authentication を展開する理由は、以下のとおりです。

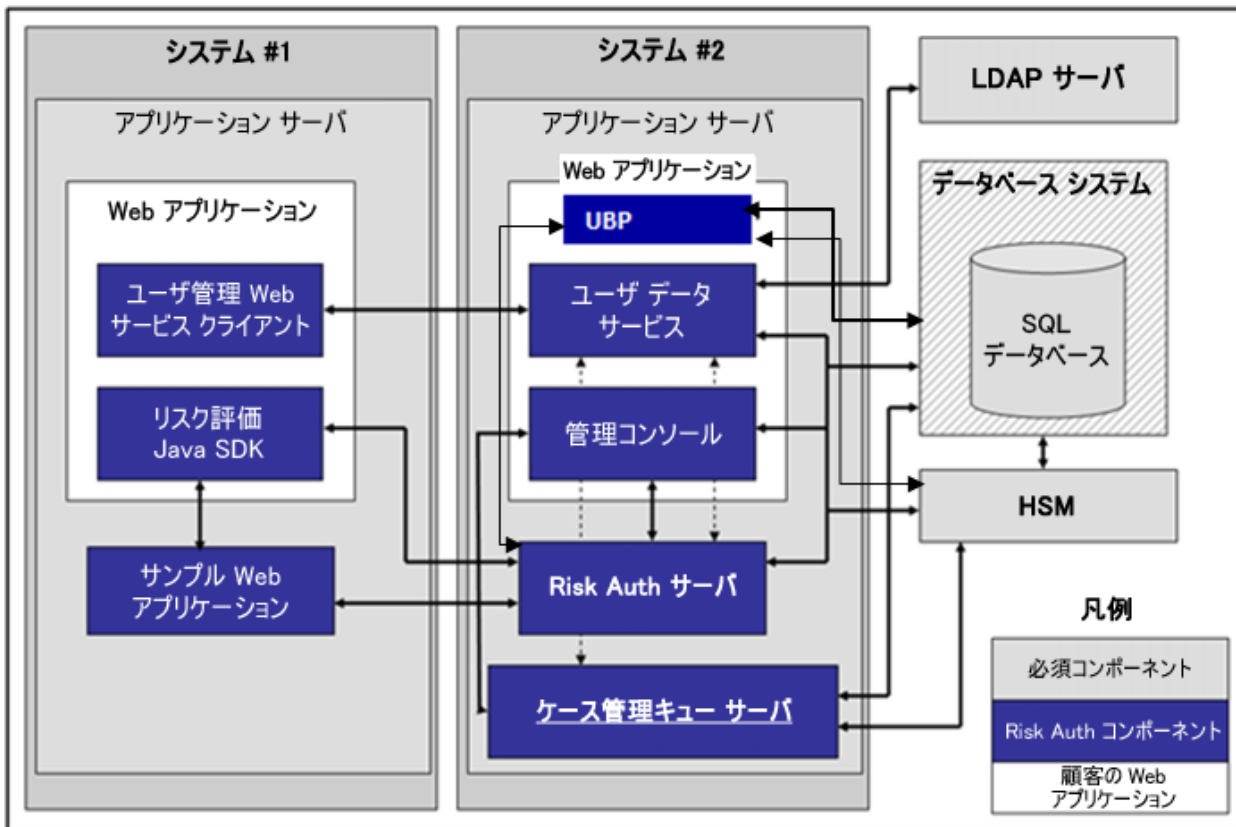
- 高可用性（フェールオーバーとロード バランシング）
- 高パフォーマンス
- スループットの増加

分散システム展開では、CA Risk Authentication コンポーネントをさまざまなサーバにインストールします。その目的は、セキュリティとパフォーマンスを高めることと、複数のアプリケーションがリスク評価機能を使用できるようにすることです。たとえば、最も一般的な展開では、1つのシステムに CA Risk Authentication サーバをインストールし、追加のシステムに 1つ以上の Web アプリケーションをインストールします。

分散システム展開を実行するには、CA Risk Authentication インストーラで [Custom] インストール オプションを選択する必要があります。

### Java SDK を使用した単一アプリケーションへの展開

以下の図は、Java SDK を使用した単一アプリケーションへの CA Risk Authentication の展開を示しています。

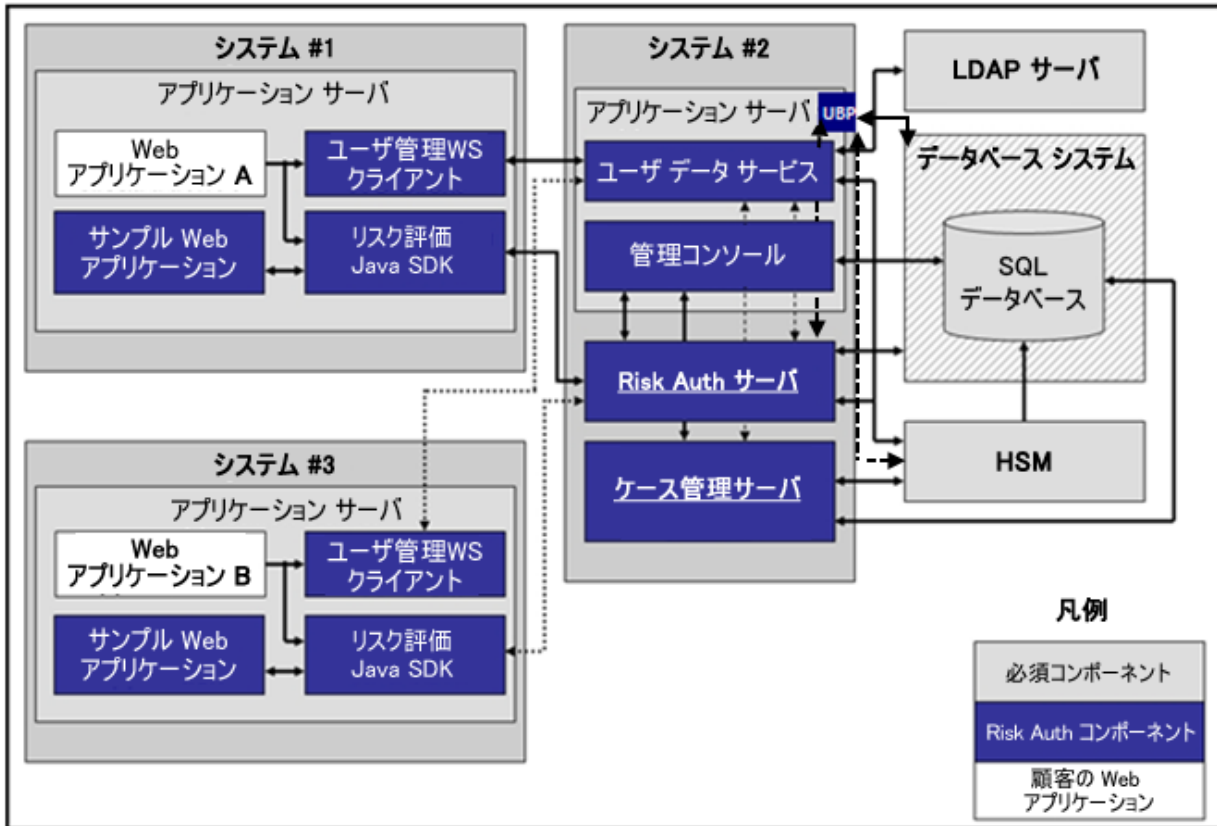




注: 管理コンソールと UBP は、任意の個別のシステム、すべてのシステム、または図に示されていないシステムにインストールできます。

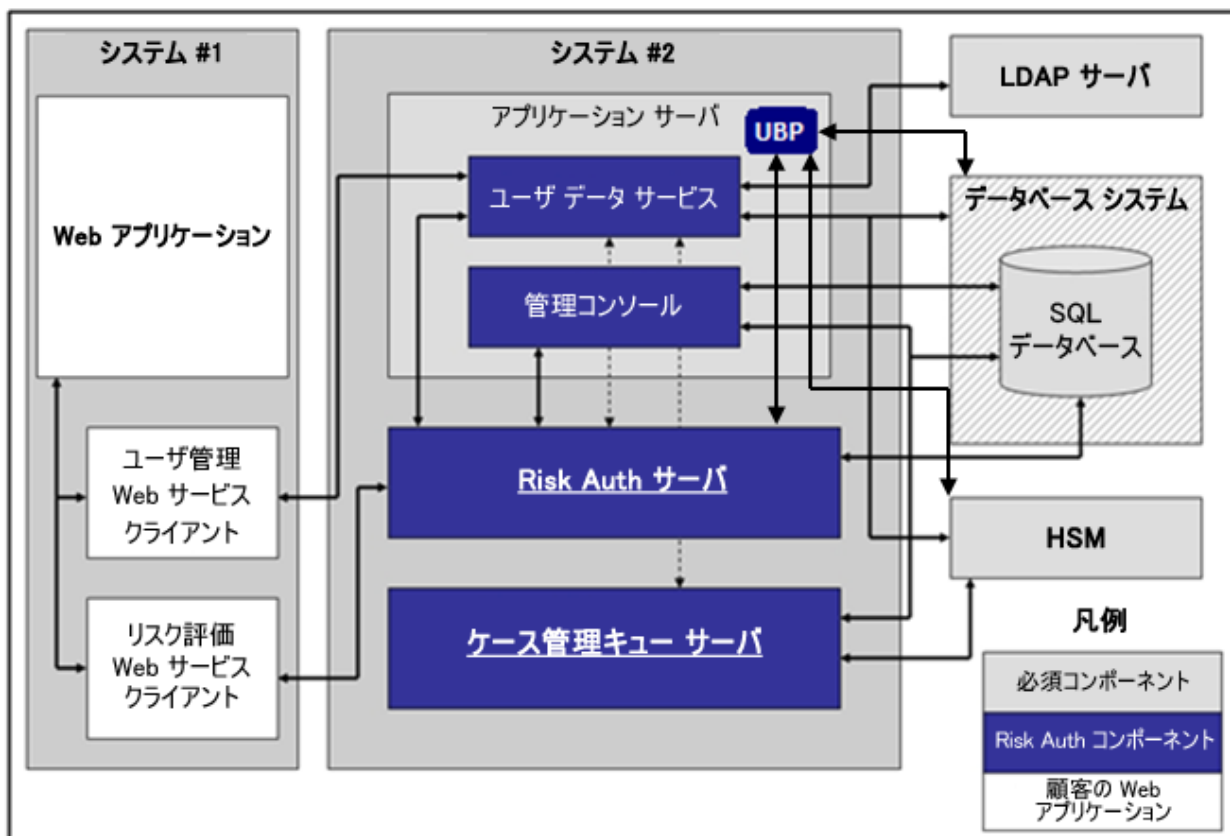
Java SDK を使用した複数アプリケーションの展開

以下の図は、Java SDK を使用した複数アプリケーションへの CA Risk Authentication の展開を示しています。



Web サービスを使用した単一アプリケーションの展開

以下の図は、Web サービスを使用した単一アプリケーションへの CA Risk Authentication の展開を示しています。



## 高可用性環境への展開

高可用性展開では、高可用性と拡張性を実現するため、**CA Risk Authentication** コンポーネントを 2 台以上のサーバにインストールします。図は、事前インストール コンポーネントと **CA Risk Authentication** コンポーネントを複数のシステムにインストールする場合のいくつかのオプションを示しています。

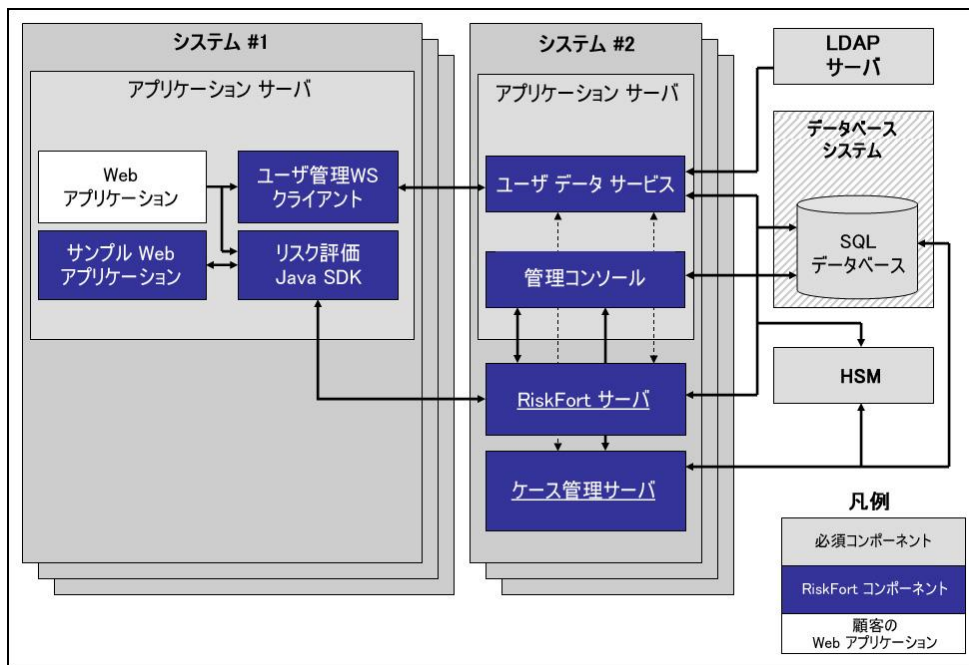
トランザクション レートが許容されるしきい値（組織のポリシーによって決定）を超えた場合、サーバインスタンスを追加する必要があります。以下の **CA Risk Authentication** コンポーネントにより、ほとんどの場合、複数のインスタンスが機能できるようになります。

- **CA Risk Authentication サーバ**：複数インスタンスがサポートされています。数は、目標のトランザクション レートによって異なります。
- **ケース管理キュー サーバ**：複数インスタンスがサポートされています。数は、目標のトランザクション レートによって異なります。
- **管理コンソール**：複数インスタンスがサポートされています。数は、管理コンソールに同時にログインするシステム内の管理者の数によって異なります。
- **UDS サーバ**：現在、1 つのみサポートされています。
- **SDK**：複数インスタンスがサポートされています。数は、サポートするアプリケーションインスタンスの数によって異なります。

以下の図は、展開を決定する方法を示しています。

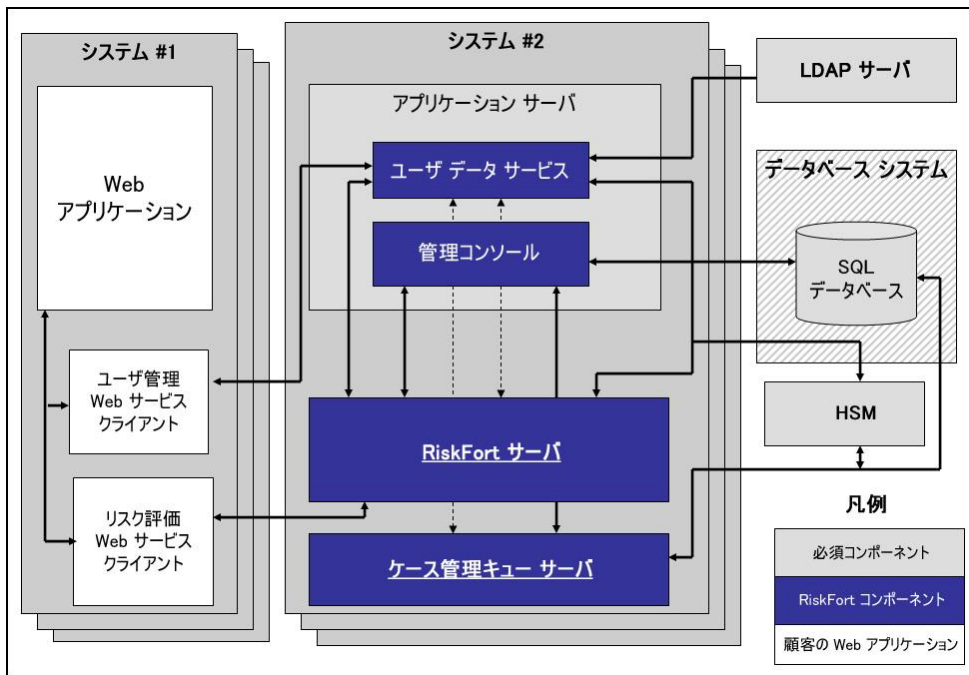
### Java SDK を使用した高可用性展開

以下の図は、Java SDK を使用した複数インスタンス展開を示しています。



Web サービスを使用した高可用性展開

以下の図は、Web サービスを使用した複数インスタンス展開を示しています。



---

## 第 3 章: インストール前のタスク

---

CA Risk Authentication とそのコンポーネントをインストールする前に、使用しているコンピュータがすべてのシステム要件を満たしていることを確認してください。ハードウェアおよびソフトウェア要件の詳細については、プラットフォーム サポート マトリックスを参照してください。

この章は、このインストールに必要な以下のセクションで構成されています。

- [データベース サーバの設定](#) (P. 30)
- [データ ストアおよびデータベース情報をセットアップする方法](#) (P. 37)

## データベース サーバの設定

インストールする前に、ユーザ情報、サーバ設定データ、監査ログ データ、およびその他の情報を格納するためのデータベースを設定します。

CA Risk Authentication では、プライマリ データベースと、高可用性展開でのフェールオーバー時とフェールバック時に使用できるバックアップ データベースを使用できます。以下の方法でデータベース接続を設定します。

- データベースは、CA Risk Authentication のインストール時に、ユーザが入力したデータベース情報を使用してインストーラが `arcotcommon.ini` ファイルを編集するときに自動的に設定されます。

サポートされるデータベース (Microsoft SQL Server、Oracle または MySQL) ごとに、特定の設定要件があります。

**注:** JBoss アプリケーションサーバでは、バックアップ データベースの設定時に以下の手順を実行します。

- a. `<JBOSS_HOME>%modules%system%layers%base%sun%jdk%main` フォルダ内の `module.xml` ファイルを編集して、以下のステートメントを記述します。

```
<path name="com/sun/rowset"/>
<path name="com/sun/rowset/internal"/>
<path name="com/sun/rowset/providers"/>
```

アプリケーション サーバを再起動します。

**重要:** データベース サーバを保護するには、ファイアウォールまたはその他のアクセス制御メカニズムを使用し、すべての依存製品と同じタイムゾーンに設定します。

## Microsoft SQL Server の設定

このセクションでは、SQL Server 用の以下の設定手順を示します。

**注:** このセクションに示すタスクの実行の詳細については、SQL Server のドキュメントを参照してください。

次の手順に従ってください：

1. SQL Server が *SQL Server 認証モード* と *Windows 認証モード* をサーバ認証に使用するように設定されていることを確認します。 [オブジェクト エクスプローラ] ウィンドウ内のサーバを右クリックし、 [セキュリティ] ページを選択します。

SQL Server が「*Windows 認証モード*」のみに設定されている場合、CA Risk Authentication はデータベースに接続できません。

2. 以下の条件でデータベースを作成します。
  - 推奨される名前は `arcotdb` です。
  - データベース サイズは自動的に拡大するように設定する必要があります。
3. 以下の手順に従って、DB ユーザ (`CH4_SQL`) を作成します。
  - a. SQL Server Management Studio で、`<SQL_Server_Name>` に移動し、 [セキュリティ] フォルダを展開して、 [ログイン] をクリックします。

**注:** `<SQL_Server_Name>` は、データベースを作成した SQL Server のホスト名または IP アドレスを指します。
  - b. [ログイン] フォルダを右クリックし、 [新しいログイン] をクリックします。
  - c. ログイン名を入力します (推奨される名前は `arcotuser`) 。
  - d. パラメータを *SQL Server 認証* に対する *認証* に設定します。
  - e. ログインの [パスワード] および [パスワードの確認入力] を指定します。
  - f. 組織のパスワード ポリシーに従い、このページのその他のパスワード設定を指定してください。
  - g. 作成したデータベース (`arcotd`) をデフォルトデータベースに設定します。
  - h. このログインセクションへのユーザのマッピングを実行します。

- i. デフォルト データベースのユーザ (SQL 2005) を `db_owner` にマップします ( [`<db_name>` のデータベース ロール メンバシップ] セクション) 。



## Oracle サーバの設定

このセクションでは、Oracle データベース サーバを作成するための設定情報を示します。

### 前提条件

- 2つのテーブルスペースを持った Oracle 上で CA Risk Authentication を実行します。2つのテーブルスペースが必要な理由を以下に示します。
  - 1つ目のテーブルスペースは、設定データ、監査ログ、およびユーザ情報の格納に使用されます。このテーブルスペースは、CA Risk Authentication データベース内でデフォルトのユーザ テーブルスペースにすることができます。
  - 2つ目のテーブルスペースでレポートを実行します。レポートを実行するために個別のテーブルスペースを使用することをお勧めします。
- CA Risk Authentication データベース設定スクリプトを使用します。このスクリプトは、このスクリプトを実行するデータベース ユーザがテーブルスペースを作成するための十分な権限を持っている場合、レポートのテーブルスペースを自動的に作成します。必要な権限がユーザにない場合、データベース管理者はこのテーブルスペースを手動で作成し、レポートを作成するセクションをスクリプトから削除する必要があります。

```
arcot-db-config-for-common-8.0.sql
```

**重要:** レポートのテーブルスペースを作成するための

`arcot-db-config-for-common-8.0.sql` データベース スクリプト内のパラメータは、データベース管理者の希望に応じて変更できます。ただし、レポートを正常に生成するには、テーブルスペース名を *ARReports* にする必要があります。

Oracle サーバを作成するには、以下の手順に従います。

- UTF-8 文字セットで情報を格納する新しいデータベースを作成します。この文字セットにより、CA Risk Authentication でダブルバイト言語を含む国際的な文字を使用できるようになります。Oracle データベースの UTF-8 サポートを有効にするには、以下の手順に従います。
  - SYS または SYSTEM として Oracle データベース サーバにログインします。
  - 以下のコマンドを実行します。

```
sys.props$ set value$='UTF8'
```

(where name='NLS\_NCHAR\_CHARACTERSET' Or name =  
'NLS\_CHARACTERSET')

- c. データベースを再起動し、文字セットが UTF-8 に設定されているかどうかを確認します。
2. データベース ユーザを作成します。
    - a. 新しいデータベース **arcotdb** のスキーマを使用して、ユーザを作成します（推奨される名前は **arcotuser**）。
    - b. 開発またはテスト用の展開では、ユーザのクォータを少なくとも 5 ～ 10 GB に設定します。

注: 運用環境、ステージング、またはその他の負荷の高いテスト用の展開の場合、ユーザに必要なクォータを決定する方法については、「データベース リファレンス」を参照してください。
  - c. ユーザに **DBA** ロールを付与します。

## MySQL サーバの設定

このセクションでは、MySQL 用の以下の設定情報を示します。

次の手順に従ってください：

1. InnoDB ストレージエンジンが MySQL のインストールでサポートされているかどうかを確認するには、SHOW ENGINES コマンドを使用します。

**注：**CA Risk Authentication は、MySQL の InnoDB ストレージエンジンを使用します。このコマンドの出力に InnoDB がサポートされていないことが示されている場合は、InnoDB のサポートを有効にします。InnoDB のサポートを有効にする方法については、MySQL のドキュメントを参照してください。

2. Windows 以外のプラットフォームで MySQL を実行している場合は、lower\_case\_table\_names 変数を 1 に設定します。

**注：**詳細については、MySQL のドキュメントを参照してください。

3. データベースを作成するには、以下の手順に従います。
  - a. MySQL コマンドウィンドウを開きます。
  - b. データベース スキーマを作成するには、以下のコマンドを実行します。

```
CREATE SCHEMA '<schema-name>' DEFAULT CHARACTER SET utf8;
```
  - c. データベース ユーザを作成するには、以下のコマンドを実行します。
4. 以下の条件に従ってユーザを作成します。
  - a. 新しいデータベース arcotdb にユーザを作成します（推奨される名前は arcotuser）。

```
CREATE USER '<user-name>' identified by '<user-password>';
```

- b. ユーザに以下の権限を付与します。

- オブジェクト権限

- SELECT
- INSERT
- UPDATE
- DELETE
- EXECUTE

- DDL 権限
  - CREATE
  - ALTER
  - CREATE ROUTINE
  - ALTER ROUTINE
  - DROP
- その他の権限
  - GRANT OPTION

## データストアおよびデータベース情報のセットアップ

CA Risk Authentication のインストールに進む前に、CA Risk Authentication データストア (データベースクライアント) を設定し、必要なデータベース情報を収集します。正しい JDK バージョンとアプリケーションサーバがインストールされていることを確認してください。

データベースサーバと通信する CA Risk Authentication コンポーネントをインストールするシステム (CA Risk Authentication サーバ、管理コンソール、およびユーザデータサービスなど) で UTF-8 サポートを有効にします。このセクションでは、その手順について説明します。

次の手順に従ってください：

1. 必要な言語パッケージをインストールします。この方法の詳細については、ベンダーのドキュメントを参照してください。
2. 以下の場所に移動します。  
[スタート] - [設定] - [コントロールパネル] - [地域と言語のオプション]  
[地域と言語のオプション] ダイアログボックスが表示されます。
3. [言語] タブをアクティブにします。
4. 以下のオプションを選択します。
  - 複合文字や右から左方向に書く言語 (タイ語を含む) のファイルをインストールする
  - 東アジア言語のファイルをインストールする
5. [Apply] をクリックして、変更を保存します。
6. [OK] をクリックしてダイアログボックスを閉じます。

## クライアント システムの UTF- サポートの設定

データベース サーバと通信するコンポーネントをインストールするシステム（CA Risk Authentication サーバ、管理コンソール、およびユーザ データ サービスなど）で UTF-8 サポートを有効にするには、以下の手順に従います。

次の手順に従ってください：

1. 必要な言語パッケージをインストールします。この方法の詳細については、ベンダーのドキュメントを参照してください。
2. 以下の場所に移動します。  
[スタート] - [設定] - [コントロール パネル] - [地域と言語のオプション]
3. [言語] タブをアクティブにします。
4. 以下のオプションを選択します。
  - 複合文字や右から左方向に書く言語（タイ語を含む）のファイルをインストールする
  - 東アジア言語のファイルをインストールする
5. [適用] をクリックします。
6. [OK] をクリックします。

## HSM の要件

このセクションは、HSM を使用する場合にのみ適用されます。HSM を使用して暗号化キーを格納する場合は、インストール前に以下のコンポーネントを設定します。

- HSM Server
- HSM クライアント
- HSM で作成された少なくとも 1 つの 3DES キー（この 3DES キーはデータベース内の情報の暗号化に必要なになります）。

**重要:** 3DES キーのラベルを安全に書き留めたことを確認します。これらは後でデータベース内の情報を暗号化するために必要になります。

詳細については、プラットフォーム ベンダーのマニュアルを参照してください。

## Java 依存コンポーネントの要件

管理コンソール、CA Risk Authentication Java SDK、および Web サービスによって必要とされる以下のコンポーネントをインストールします。

- JDK

**注:** JDK の新規インストールを実行する場合は、JAVA\_HOME 環境変数を設定する必要があります。PATH 変数は %JAVA\_HOME%\bin¥ を参照している必要があります。含めなかった場合、管理コンソールおよびその他の JDK 依存コンポーネントが起動しない可能性があります。

- Application Server





## 第 4 章：単一システムに CA Risk Authentication を展開する方法

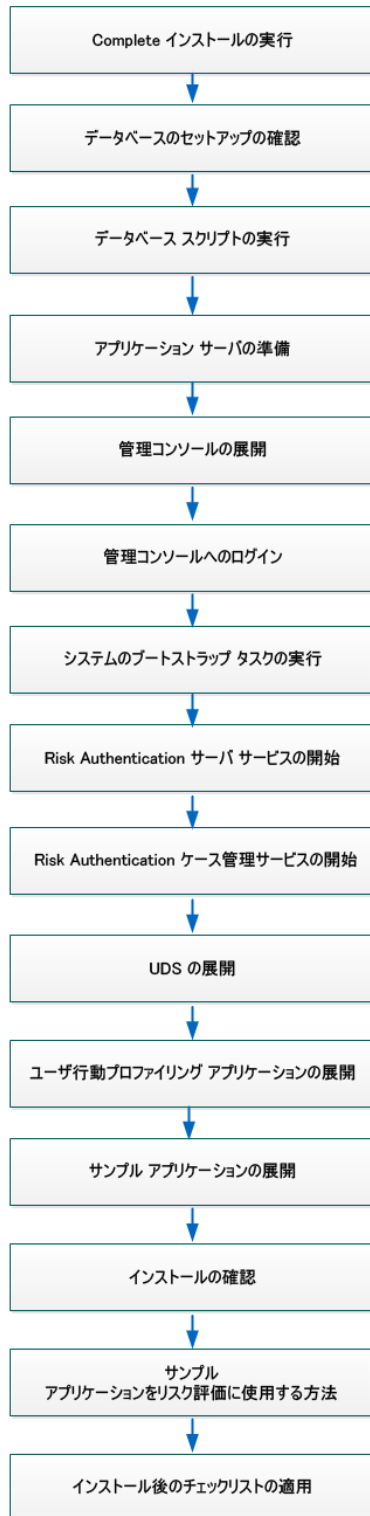
---

CA Risk Authentication コンポーネントをインストールするには、CA Risk Authentication 8.0 InstallAnywhere ウィザードを使用します。このウィザードでは *Complete* と *Custom* のインストールタイプをサポートしています。

**注:** 単一のコンピュータ上に CA Risk Authentication をインストールして設定する場合、インストーラを実行する際に [Complete] オプションを使用します。

以下の図は、CA Risk Authentication 8.0 をインストールするために実行するタスクを示しています。

## 単一システムに Risk Authentication をインストールする方法



以下のタスクを実行します。

1. [Complete インストールの実行](#) (P. 46)
2. [データベースのセットアップの確認](#) (P. 55)
3. [データベース スクリプトの実行](#) (P. 54)
4. [アプリケーション サーバの準備](#) (P. 56)
5. [管理コンソールの展開](#) (P. 66)
6. [管理コンソールへのログイン](#) (P. 67)
7. [システムのブートストラップ タスクの実行](#) (P. 68)
8. [CA Risk Authentication サーバ サービスの開始](#) (P. 70)
9. [CA Risk Authentication ケース管理サービスの開始](#) (P. 71)
10. [UDS の展開](#) (P. 72)
11. [ユーザ行動プロファイリング アプリケーションの展開](#) (P. 74)
12. [サンプルアプリケーションの展開](#) (P. 76)
13. [\(オプション\) ユーザ行動プロファイリング アプリケーションの展開](#) (P. 74)
14. [インストールの確認](#) (P. 77)
15. [サンプルアプリケーションをリスク評価に使用する方法](#) (P. 78)
16. [インストール後のチェックリストの適用](#) (P. 82)

**重要:**

CA Risk Authentication を単一システムにインストールする場合は、以下の点に注意してください。

- `<install_location>` に特殊文字が含まれていないことを確認してください ( ~ ! @ # \$ % ^ & \* ( ) \_ + = { } [ ] " ' など ) 。
- MySQL データベース名にドット ( . ) 文字を含めることはできません。
- 現時点では、インストーラを使用して CA Risk Authentication コンポーネントを変更または修復することはできません。必ずコンポーネントをアンインストールしてから、再インストールしてください。

- インストールの実行中は、インストーラ ウィンドウを閉じないでください。インストール中 (特に最後の段階) に **[Cancel]** ボタンをクリックしてインストールを中止した場合、それまでに作成されたディレクトリはすべてが削除されるとは限りません。インストールディレクトリ、`<install_location>%Arcot Systems%`、およびそのサブディレクトリは手動でクリーンアップする必要があります。
- 既存の `ARCOT_HOME` のインスタンスがすでに含まれているシステム上でインストーラを実行する場合。
  - インストールディレクトリを要求されません。
  - データベースのセットアップを要求されません。インストーラは既存のデータベースを使用します。
  - 暗号化をセットアップするように要求されません。
  - **CA Strong Authentication** は、**CA Risk Authentication** と共にインストールして使用することができます。両方の製品は、特定の共通コンポーネントを使用します。これらは、各製品のインストール中にコピーされます。**CA Strong Authentication** をすでにインストールしており、**CA Risk Authentication** インストール手順を開始しようとしている場合、**CA Risk Authentication** インストーラは **CA Strong Authentication** のインストール時にコピーされた共通のコンポーネントの存在を検出できます。検出すると、**CA Risk Authentication** インストーラは **Custom** インストールを実行するための画面を表示します。

このセクションには、以下のトピックが含まれています。

[Complete インストールの実行 \(P. 46\)](#)

[アプリケーションサーバを準備する方法 \(P. 56\)](#)

[管理コンソールの展開 \(P. 66\)](#)

[システムのブートストラップタスクの実行 \(P. 68\)](#)

[ユーザデータ サービス \(UDS\) の展開 \(P. 72\)](#)

[ユーザ行動プロファイリングアプリケーションの展開 \(P. 74\)](#)

[サンプルアプリケーションの展開 \(P. 76\)](#)

[サンプルアプリケーションをリスク評価に使用する方法 \(P. 78\)](#)

[インストール後のチェックリストの適用 \(P. 82\)](#)

## Complete インストールの実行

CA Risk Authentication をインストールおよび設定するには、Administrators グループの単一のユーザアカウントを使用します。そうしないと、インストールがエラーなしで完了した場合でも、インストールの重要な手順が正常に完了しません。

CA Risk Authentication パッケージのすべてのコンポーネントをインストールするには、Complete インストールを実行します。これらのコンポーネントには、CA Risk Authentication サーバ、およびデータベースの設定に必要なスクリプトが含まれます。

次の手順に従ってください：

1. CA Risk Authentication-8.0-Windows-Installer.exe ファイルのあるディレクトリに移動し、このファイルをダブルクリックします。
2. [次へ] をクリックします。
3. 使用許諾契約書を読み、[I accept the terms of the License Agreement] オプションを選択し、[Next] をクリックします。

インストーラはこの時点で、その他の CA 製品がシステムに存在するかどうかを確認します。

インストーラが既存の CA 製品インストール（既存の ARCOT\_HOME）を検出した場合

- インストールディレクトリを要求されません。
  - データベースおよび暗号化のセットアップを要求されません。インストーラは既存のデータベースおよび暗号化設定を使用します。そのため、設定は無効になっていますが、手順 6 に移動できます。手順 10 の画面は表示されないため、その手順を実行する必要はありません。
4. [次へ] をクリックします。
  5. [Complete to Install] を選択し、[Next] をクリックします。
  6. データベースのタイプに応じて、以下のいずれかを選択し、[Next] をクリックします。

- Microsoft SQL Server

注: SQL データベースを使用している場合、使用している ODBC ドライバのバージョンが「インストールの準備」に記載されているバージョンと同じであることを確認してください。

- Oracle データベース

注: CA Risk Authentication は Oracle Real Application Clusters (Oracle RAC) で動作することが確認されています。CA Risk Authentication インストール環境で Oracle RAC を使用するには、この手順で Oracle データベースを選択し、次の手順(手順 7)を実行してから、「Oracle RAC 用の CA Risk Authentication の設定」の手順を実行します。

- MySQL

選択したデータベースに応じて、以下の画面が表示されます。

7. 以下の情報を入力します。

- Microsoft SQL Server

ODBC DSN

インストーラが DSN の作成に使用する値を定義します。CA Risk Authentication サーバは、この DSN を使用して CA Risk Authentication データベースに接続します。推奨される入力値は *arcotdsn* です。

サーバ

CA Risk Authentication データストアのホスト名または IP アドレスを指定します。

デフォルト インスタンス

構文: <server\_name>

例: demodatabase

名前付きインスタンス

構文: <server\_name>¥<instance\_name>

例: demodatabase¥instance1

User Name

データベース ユーザ名を指定します。ユーザは CREATE SESSION 権限および DBA 権限を持っている必要があります。

注: ユーザ名はプライマリ DSN とバックアップ用 DSN とで異なる必要があります。

Password

ユーザ名に関連付けられているパスワードを指定します。このパスワードはデータベース管理者によって指定されます。

データベース

MS SQL データベース インスタンスの名前を指定します。

#### Port Number

データベースが受信リクエストをリスンするポート番号を指定します。

デフォルトポート：1433

### ■ Oracle Server

#### ODBC DSN

インストーラが DSN の作成に使用する値を指定します。CA Risk Authentication サーバは、この DSN を使用して CA Risk Authentication データベースに接続します。推奨される入力値は *arcotdsn* です。

#### User Name

CA Risk Authentication がデータベースにアクセスする際のデータベース ユーザ名を指定します。この名前は、データベース管理者によって指定されます。

ユーザは CREATE SESSION 権限および DBA 権限を持っている必要があります。

注：ユーザ名はプライマリ DSNs とバックアップ用 DSN とで異なる必要があります。

#### Password

上記のフィールドで指定したユーザ名に関連付けられているパスワードを指定します。このパスワードはデータベース管理者によって指定されます。

#### Service ID

サーバ上で実行される Oracle データベースのインスタンスを表す Oracle システム識別子 (SID) を指定します。

#### Port Number

データベースが受信リクエストをリスンするポート番号を指定します。

デフォルト：1521

#### ホスト名

CA Risk Authentication データストアのホスト名または IP アドレスを指定します。



構文 : <server\_name>

例 : demodatabase

- MySQL サーバ

**DBC DSN**

インストーラが DSN の作成に使用する値を指定します。CA Risk Authentication サーバは、この DSN を使用して CA Risk Authentication データベースに接続します。推奨される入力値は *arcotdsn* です。

**サーバ**

CA Risk Authentication データストアのホスト名または IP アドレスを指定します。

**デフォルト インスタンス**

構文 : <server\_name>

例 : demodatabase

**名前付きインスタンス**

構文 : <server\_name>¥<instance\_name>

例 : demodatabase¥instance1

**User Name**

CA Risk Authentication がデータベースにアクセスする際のデータベース ユーザ名を指定します。この名前は、データベース管理者によって指定されます。

ユーザは CREATE SESSION 権限および DBA 権限を持っている必要があります。

**注:** ユーザ名はプライマリ DSNs とバックアップ用 DSN とで異なる必要があります。

**Password**

上記のフィールドで指定したユーザ名に関連付けられているパスワードを指定します。このパスワードはデータベース管理者によって指定されます。

**データベース**

MySQL データベース インスタンスの名前を指定します。

**Port Number**

データベースが受信リクエストをリスンするポート番号を指定します。

**デフォルト :** 3306

8. データベースの詳細を指定した後、[**Test Data Source**] ボタンをクリックして、データベースに正常に接続できるかテストし、ボタンの下のフィールド内と同じか結果を確認します。
9. [次へ] をクリックします。
10. 暗号化モードを選択し、暗号化に使用される情報を入力します。

#### マスタ キー

データベースに保存されるデータを暗号化するために使用されるマスタ キー用のパスワードを指定します。

**デフォルト値** : MasterKey

**注**: インストール後にマスタ キーの値を変更する場合は、新しいマスタ キーの値を使用して `securestore.enc` を再生成します。詳細については、「インストール後のハードウェアセキュリティモジュール情報の変更」を参照してください。

#### HSM の設定

(オプション) ハードウェアセキュリティモジュール (HSM) を使用して機密データを暗号化する場合に指定します。このオプションを選択しない場合、デフォルトでは、ソフトウェアモードを使用してデータが暗号化されます。

#### PIN

HSM に接続するパスワードを入力します。

#### Choose Hardware Module

以下のいずれかの HSM を指定します。

- Luna HSM
- nCipher netHSM

#### HSM パラメータ

以下の HSM 情報を設定します。

**Shared Library** : HSM に対応する PKCS#11 共有ライブラリへの絶対パス。

Luna (`cryptoki.dll`) および nCipher netHSM (`cknfast.dll`) の場合は、ファイルの絶対パスと名前を指定します。

**Storage Slot Number** : データの暗号化に使用される 3DES キーが使用可能な HSM スロット。

- Luna の場合、デフォルト値は 0 です。

- nCipher netHSM の場合、デフォルト値は 1 です。

注: HSM のパラメータ値は、<install\_location>%Arcot Systems¥conf¥にある arcotcommon.ini ファイルに記録されます。インストール後にこれらの値を変更する場合は、「設定ファイルおよびオプション」の説明に従って、このファイルを編集します。

[次へ] をクリックします。

11. [Pre-Installation Summary] 画面内の情報を確認し、[Next] をクリックします。

12. [Install] をクリックし、インストールプロセスを開始します。

[Microsoft Visual C++ 2010 x86 Redistributable Setup] 画面が表示されます。この画面は、CA Risk Authentication をインストールしている現在のシステムに Microsoft Visual C++ 2010 x86 がインストールされていない場合にのみ表示されます。

13. [Microsoft Visual C++ 2010 x86 Redistributable Setup] 画面で、以下の手順を実行します。

- a. [I have read and accept the license terms] を選択して、[Install] をクリックします。

[Installation Progress] 画面が表示されます。数秒間表示される場合があります。しばらくすると、[Installation Is Complete] 画面が表示されます。

- b. [Finish] をクリックして [Microsoft Visual C++ 2010 x86 Redistributable Setup] ダイアログ ボックスを閉じ、インストールを続けます。

[Installing CA Risk Authentication] 画面が表示されます。

しばらくすると、[Installation Complete] 画面が表示されます。

14. [Done] をクリックして、CA Risk Authentication のインストールを完了します。

注: インストールが完了したら、「インストール後の作業の実行」の説明に従ってインストール後のタスクを実行してください。

## インストール ログ

インストール後、<install\_location> ディレクトリのインストール ログ ファイル (CA\_CA Risk Authentication\_Install\_<timestamp>.log) にアクセスできます。たとえば、インストールディレクトリとして C:¥Program Files ディレクトリを指定した場合、インストール ログ ファイルは C:¥Program Files ディレクトリに作成されます。

インストールが何らかの理由で失敗した場合、エラー メッセージはこの ログ ファイルに記録されます。

## データベース スクリプトの実行

データベース テーブルを作成するには、必要なデータベース スクリプトを実行します。

次の手順に従ってください:

**重要:** スクリプトを実行する前に、「データベース サーバの設定」セクションで作成したときと同じデータベース ユーザとしてログインしていることを確認してください。

1. 以下のディレクトリに移動します。  
`<install_location>%Arcot Systems%dbscripts%`
2. 使用しているデータベースに基づいて以下のいずれかのサブディレクトリに移動します。
  - Oracle の場合 : Oracle%
  - Microsoft SQL の場合 : mssql%
  - MySQL の場合 : mysql%
3. スクリプトを次に示す順序で実行します。

a. `arcot-db-config-for-common-8.0.sql`

**重要:** CA Strong Authentication をインストール済みの場合は、CA Strong Authentication のインストール時にすでに実行しているため、`arcot-db-config-for-common-8.0.sql` を実行しないでください。

b. `arcot-db-config-for-riskfort-8.0.sql`

c. (3D セキュア チャネルを作成する必要がある場合にのみ、オプション) `arcot-db-config-for-3dsecure-8.0.sql`

d. (オプション) ユーザ行動プロファイリングを使用する場合にのみ、以下のコマンドを実行します。

`arcot-db-config-for-userprofiling-2.0.sql`

**注:** スクリプトの実行は一度だけのジョブです。毎回スクリプトを実行すると、レコードの重複により、既存テーブルや挿入の失敗などを示すエラーが表示される可能性があります。

## データベースのセットアップの確認

必要なデータベース スクリプトを実行した後、CA Risk Authentication スキーマを確認します。

次の手順に従ってください:

1. データベースをインストールしたユーザとして CA Risk Authentication データベースにログインします。

注: アップグレードパスに従っている場合は、データベースをアップグレードしたユーザとしてデータベースにログインします。

2. 以下のクエリを実行します。

```
SELECT SERVERNAME, VERSION FROM ARRFSEVERERS;
```

上記のクエリの結果、以下の出力が表示されます。

| SERVERNAME                            | VERSION |
|---------------------------------------|---------|
| CA Risk Authentication                | 8.0     |
| CA Risk Authentication CaseManagement | 8.0     |

3. データベース コンソールからログアウトします。

## アプリケーション サーバを準備する方法

ユーザ データ サービス (UDS) および管理コンソールは Risk Authentication の Web ベースのコンポーネントであり、以下のサポート対象アプリケーション サーバのいずれかに展開する必要があります。

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic
- JBoss アプリケーション サーバ

アプリケーション サーバにこれらの Web アプリケーションの WAR ファイルを展開する前に、UDS および管理コンソールに必要なファイルをアプリケーション サーバの適切な場所にコピーします。このセクションでは、アプリケーション サーバに必要な暗号化ファイルをコピーし、これらの Web アプリケーションの WAR ファイルを展開する手順について説明します。

1. Java ホームの設定
2. アプリケーション サーバへのデータベース アクセス ファイルのコピー
3. アプリケーション サーバへの JDBC JAR ファイルのコピー
4. Enterprise Archive ファイルの作成

### Java ホームの設定

このセクションでは、Java ホーム環境のセットアップについて説明します。

次の手順に従ってください：

1. JAVA\_HOME 環境変数を設定していることを確認します。JAVA\_HOME は、アプリケーション サーバの JAVA\_HOME である必要があります。
2. %JAVA\_HOME%\bin を PATH 変数に追加します。



## アプリケーション サーバへのデータベース アクセス ファイルのコピー

UDS および管理コンソールでは、CA Risk Authentication データベースに安全にアクセスするために以下のファイルを使用します。

- arcot-crypto-util.jar。以下の場所にあります。  
<install\_location>%Arcot Systems%java%lib%
- ArcotAccessKeyProvider.dll。以下の場所にあります。  
<install\_location>%Arcot Systems%native%win%<32bit-or-64bit>%

そのため、CA Risk Authentication コンポーネントを展開したアプリケーション サーバ上の適切な場所にこれらのファイルをコピーします。以下のサブセクションで、以下のサーバ用ファイルのコピーについて説明します。

### Apache Tomcat

次の手順に従ってください：

1. arcot-crypto-util.jar を <Tomcat\_JAVA\_HOME>%jre%lib%ext% にコピーします。

<Tomcat\_JAVA\_HOME>

Apache Tomcat インスタンスによって使用される JAVA\_HOME を指定します。

2. ArcotAccessKeyProvider.dll を <Tomcat\_JAVA\_HOME>%jre%bin% にコピーします。
3. アプリケーション サーバを再起動します。

### IBM WebSphere

次の手順に従ってください：

1. WebSphere Administration Console にログインします。
2. [Environment] - [Shared Libraries] をクリックします。
  - a. [Scope] ドロップダウンから、有効な可視性範囲を選択します。アプリケーションを展開するターゲット サーバまたはノードを含めます。
  - b. [新規] をクリックします。
  - c. 名前を入力します。

例：ArcotJNI

- d. クラスパスを入力します。

このパスは、arcot-crypto-util.jar ファイルが存在し、ファイル名も含まれる場所を指している必要があります。

例：C:\Program Files\Arcot Systems\java\lib\arcot-crypto-util.jar

- e. JNI ライブラリ パスを入力します。

このパスは、ArcotAccessKeyProvider.dll ファイルが存在する場所を指している必要があります。

3. [適用] をクリックします。
4. サーバレベルのクラス ロードを設定します。
  - a. [Servers] - [Server Types] - [WebSphere Application Servers] に移動します。
  - b. [Application Servers] で、サーバの設定ページにアクセスします。
  - c. [Java and Process Management] をクリックします。[Class Loader] をクリックします。
  - d. [新規] をクリックします。
  - e. デフォルトの [Classes loaded with parent class loader first] を選択して、[OK] をクリックします。
  - f. 自動生成されたクラス ロード ID をクリックします。
  - g. [Shared Library References] をクリックします。
  - h. [Add] をクリックし、[ArcotJNI] を選択します。[適用] をクリックします。
  - i. 変更を保存します。
5. ArcotAccessKeyProvider.dll を <WebSphere\_JAVA\_HOME>\jre\bin\ にコピーします。

ここで、<WebSphere\_JAVA\_HOME> は、IBM WebSphere インスタンスによって使用される JAVA\_HOME を表します。
6. アプリケーション サーバを再起動します。

### Oracle WebLogic

次の手順に従ってください：

1. ArcotAccessKeyProvider.dll を <WebLogic\_JAVA\_HOME>\jre\bin\ にコピーします。

ここで、<Weblogic\_JAVA\_HOME> は、Oracle WebLogic インスタンスによって使用される JAVA\_HOME を表します。

2. arcot-crypto-util.jar を <WebLogic\_JAVA\_HOME>%jre%lib%ext% にコピーします。

注: 必ず WebLogic によって使用される適切な <JAVA\_HOME> を使用してください。

3. WebLogic Administration Console にログインします。
4. [Deployments] に移動します。
5. [Lock and Edit] オプションを有効にします。
6. [インストール] をクリックします。arcot-crypto-util.jar ファイルがあるディレクトリに移動します。
7. [次へ] をクリックします。
8. [Next] をクリックして、[Summary] ページを表示します。
9. [完了] をクリックします。
10. 変更を有効にします。
11. アプリケーション サーバを再起動します。

## JBoss アプリケーション サーバ

次の手順に従ってください:

1. ArcotAccessKeyProvider.dll を <JBoss\_JAVA\_HOME>%jre%bin% にコピーします。

ここで、<JBoss\_JAVA\_HOME> は、JBoss アプリケーション サーバ インスタンスによって使用される JAVA\_HOME を表します。

2. <JBoss\_HOME>%modules%advauth-admin-libs%main% というフォルダ構造を作成し、<ARCOT\_HOME>%java%lib から以下の JAR をこのフォルダにコピーします。

- arcot-crypto-util.jar
- bcprov-jdk15-146.jar

3. 同じフォルダ (<JBoss\_HOME>%modules%advauth-admin-libs%main) 内に *module.xml* という名前で、以下のコードを持つファイルを作成します。

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="advauth-admin-libs">
<resources>
<resource-root path="arcot-crypto-util.jar"/>
```

```
<resource-root path="bcprov-jdk15-146.jar"/>
</resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
```

4. アプリケーションサーバを再起動します。

## アプリケーション サーバへの JDBC JAR ファイルのコピー

CA Risk Authentication は、以下の JDBC JAR ファイルをサポート対象のデータベースに必要とします。

- **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)

以下のセクションでは、データベースに必要な JDBC JAR をコピーするための手順について説明します。

### Apache Tomcat

次の手順に従ってください：

1. <Database\_JAR> ファイルをダウンロードした場所に移動します。
2. <Database\_JAR> ファイルを以下のディレクトリにコピーします。
  - **Apache Tomcat 5.5.x の場合** : <TOMCAT\_HOME>\%common\lib¥
  - **Apache Tomcat 6.x および 7.x の場合** : <TOMCAT\_HOME>\lib¥
3. サーバを再起動します。

### IBM WebSphere

次の手順に従ってください：

1. WebSphere Administration Console にログインします。
2. [Environment] - [Shared Libraries] をクリックします。以下の手順を実行します。
  - a. [Scope] リストから、有効な可視性範囲を選択します。アプリケーションを展開するターゲット サーバまたはノードを含めます。
  - b. [新規] をクリックします。
  - c. 名前を入力します。  
例：JDBCJAR
  - d. クラスパスを指定します。

**重要**：このパスは、<Database\_JAR> ファイルが存在し、ファイル名が含まれる場所を指している必要があります。

- e. [適用] をクリックします。
3. サーバレベルのクラスローダを設定し、以下の手順に従います。
  - a. [Servers] - [Server Types] - [WebSphere Application Servers] に移動します。
  - b. [Application Servers] で、設定ページにアクセスします。
  - c. [Java and Process Management] をクリックします。 [Class Loader] をクリックします。
  - d. [新規] をクリックします。
  - e. デフォルトの [Classes loaded with parent class loader first] を選択します。 [OK] をクリックします。
  - f. 自動生成されたクラスローダ ID をクリックします。
  - g. [Shared Library References] をクリックします。
  - h. [Add] をクリックし、 [JDBCJAR] を選択します。 [適用] をクリックします。
  - i. 変更を保存します。
4. アプリケーションサーバを再起動します。

### Oracle WebLogic

次の手順に従ってください：

注: Oracle データベースを使用している場合、WebLogic はデフォルトで Oracle データベースをサポートしているため、このセクションで説明されている設定を行わないでください。

1. <Database\_JAR> ファイルを <Weblogic\_JAVA\_HOME>\lib\ext にコピーします。  
ここで、<WebLogic\_JAVA\_HOME> は、Oracle WebLogic インスタンスによって使用される JAVA\_HOME を表します。
2. WebLogic Administration Console にログインします。
3. [Deployments] に移動します。
4. [Lock and Edit] オプションを有効にします。
5. [Install] をクリックして、必要な <Database\_JAR> ファイルが含まれるディレクトリに移動します。
6. [次へ] をクリックします。

7. [Next] をクリックして、[Summary] ページを表示します。
8. [完了] をクリックします。
9. 変更を有効にします。
10. アプリケーション サーバを再起動します。

## JBoss アプリケーション サーバ

次の手順に従ってください:

1. このフォルダに <JBASS\_HOME>%modules%advauth-jdbc-driver%main% というフォルダ構造を作成し、そのフォルダに JDBC Jar ファイルをコピーします。
2. <JBASS\_HOME>%modules%advauth-jdbc-driver%main% に *module.xml* という名前でファイルを作成します。
3. ファイルに、以下のコードを追加します。

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="advauth-jdbc-driver">
  <resources>
    <resource-root path="<JDBC Jar Name>" />
  </resources>
  <dependencies>
    <module name="javax.api" />
    <module name="javax.transaction.api" />
  </dependencies>
</module>
```

4. JDBC Jar ファイル名を指定しているタグで「<JDBC Jar Name>」を編集します。

例: sqljdbc.jar

5. アプリケーション サーバを再起動します。

### Enterprise Archive ファイルの作成

#### Oracle WebLogic 10.1 で有効

ほとんどのエンタープライズアプリケーションサーバでは、単一のエンタープライズアプリケーション（またはアーカイブ）に1つのベンダー（例：CA）から関連するJARまたはWARファイルをバンドルすることをサポートしています。

その結果、関連するすべてのJARまたはWARを一緒に展開して、クラスローダでロードできます。また、このアーカイブにはapplication.xmlファイルが含まれます。このファイルは自動的に生成され、バンドルされた各モジュールの展開方法が記載されています。

UDSと管理コンソールを展開するためのデフォルトのWARファイルが付属しています。ただし、必要に応じて、これらのファイルの形式をエンタープライズアーカイブ(EAR)に変更し、EARファイルを展開できます。

以下のサブセクションの1つでは、UDSと管理コンソールの両方のEARファイルを個別に生成できます。または、両方のWebアーカイブを含む単一のEARファイルを生成することもできます。

UDSおよび管理コンソールに対して個別のEARファイルを作成するには、以下の手順に従います。

1. コマンドプロンプトウィンドウを開きます。
2. `<install_location>%Arcot Systems%tools%common%bundlemanager%` ディレクトリに移動します。
3. EARファイルを作成するには、以下のコマンドを実行します。

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

上記のコマンドによって、以下の場所に個別のEARファイルが生成されます。

```
<install_location>%Arcot Systems%java%webapps%
```



UDS と管理コンソールの Web アーカイブを含んだ単一の EAR ファイルを作成するには、以下の手順に従います。

1. コマンドプロンプト ウィンドウを開きます。
2. `<install_location>%Arcot Systems%tools%common%bundlemanager%` ディレクトリに移動します。
3. EAR ファイルを作成するには、以下のコマンドを実行します。  

```
java -jar bundle-manager.jar -ear <filename.ear> -warList  
arcotadmin.war arcotuds.war
```

上記のコマンドによって、以下の場所に単一の EAR ファイルが生成されます。

`<install_location>%Arcot Systems%java%webapps%`

## 管理コンソールの展開

管理コンソールは、サーバ設定のカスタマイズや展開したシステムの管理を実行できるブラウザ ベースのインターフェースです。

**注:** IBM WebSphere 7.0、8.0、または 8.5 に管理コンソールを展開する場合は、付録「IBM WebSphere への管理コンソールの展開」に記載されている手順を参照してください。

管理コンソールを使用して CA Risk Authentication を管理するためには、CA Risk Authentication サーバがインストールされているシステムに管理コンソールがホスト名でアクセスできることを確認します。

次の手順に従ってください：

1. アプリケーション サーバの適切なディレクトリに `arcotadmin.war` を展開します。

**注:** 展開手順は、使用しているアプリケーション サーバによって異なります。詳細な手順については、アプリケーション サーバベンダーのドキュメントを参照してください。

**例:** Apache Tomcat の場合は、`<APP_SERVER_HOME>%webapps%` に `WAR` ファイルを展開する必要があります。

2. (32 ビットの WebSphere の場合のみ)アプリケーション ファイルが更新されると、Admin クラスを再ロードするように設定します。以下の手順に従います。
  - a. [Application] - [Enterprise Applications] に移動し、[Admin settings] ページにアクセスします。
  - b. [Class loader order] で、[Classes loaded with local class loader first (parent last)] オプションを選択します。
  - c. [WAR class loader policy] で、[Single class loader for application] を選択します。
  - d. [適用] をクリックします。
  - e. Admin アプリケーションを再起動します。
3. アプリケーション サーバを再起動します。
4. コンソールが正常に展開されていることを確認するには、以下の手順に従います。
  - a. 以下の場所に移動します。  
`<install_location>%Arcot Systems%logs%`

- b. 任意のエディタで `arcotadmin.log` ファイルを開き、以下の行を見つけます。
  - 2.0.3
  - Arcot Administration Console Configured Successfully.

注: これらの行は、管理コンソールが正常に展開されていることを示しています。
- c. また、ログ ファイルに **FATAL** および **WARNING** のメッセージが含まれていないことも確認します。
- d. ファイルを閉じます。

## 管理コンソールへのログイン

初めて管理コンソールにログインするときは、展開時にデータベースに自動的に設定される MA（マスタ管理者）認証情報を使用します。

次の手順に従ってください：

1. Web ブラウザ ウィンドウで、管理コンソールを起動します。管理コンソールのデフォルト URL は以下のとおりです。

*http://<host>:<appserver\_port>/arcotadmin/masteradminlogin.htm*

例：Apache Tomcat の場合は、デフォルト ホストは `localhost` であり、ポートは `8080` です。

2. 以下のように、デフォルトのマスタ管理者アカウントの認証情報を使用してログインします。
  - ユーザ名： `masteradmin`
  - パスワード： `master1234!`

## システムのブートストラップ タスクの実行

ブートストラップは、これらのセットアップ タスクについて説明するウィザード主導のプロセスです。ほかの管理リンクは、これらのタスクを実行した後で有効になります。管理コンソールを使用して **CA Risk Authentication** の管理を始めるには、以下の必須の手順を実行してシステムを初期化する必要があります。

- デフォルトのマスタ管理者パスワードの変更
- グローバル キー ラベルの設定
- デフォルトの組織の設定を指定

管理コンソールを展開すると、組織が 1 つ自動的に作成されます。この組織はデフォルトの組織 (DEFAULTORG) と呼ばれます。単一の組織システムとして、デフォルトの組織は、何らかの組織を作成せずにそれ自身で使用できます。

MA (マスタ管理者) として初めて管理コンソールにログインすると、[ブートストラップ] ウィザード画面の [サマリ] 画面が表示されます。

次の手順に従ってください：

1. [開始] をクリックすると、プロセスが起動します。  
[パスワードの変更] 画面が表示されます。
2. [現在のパスワード]、[新規パスワード]、[パスワードの確認] を指定し、[次へ] をクリックします。
3. [グローバル キー ラベルの設定] ページで、以下の手順に従います。
  - グローバル キー ラベルを入力して、[次へ] をクリックします。

**CA Risk Authentication** では、ハードウェアまたはソフトウェアベースの機密データの暗号化を使用できます。(デフォルトではソフトウェアベースの暗号化が有効ですが、*arcotcommon.ini* ファイルを使用してハードウェアベースの暗号化を有効にできます。)ハードウェアの暗号化かソフトウェアの暗号化かに関係なく、ユーザおよび組織データの暗号化にグローバル キー ラベルが使用されます。

ハードウェアの暗号化を使用している場合、このラベルは、HSM デバイスに格納されている実際の 3DES キーへの参照（ポインタ）としてのみ機能します。そのため、HSM キー ラベルと一致する必要があります。ただし、ソフトウェア ベースの暗号化の場合、このラベルはキーとして機能します。

**重要:** ブートストラップ プロセスの完了後に、このキー ラベルを更新することはできません。

- [暗号化ストレージタイプ] に、暗号化キーがデータベース（ソフトウェア）に格納されているか、または HSM（ハードウェア）に格納されているかを入力します。
4. [次へ] をクリックして続行します。
  5. [デフォルト組織設定] セクションで、以下のパラメータを入力します。

#### 表示名

組織のわかりやすい名前を指定します。この名前は、管理コンソールの他のすべてのページおよびレポート上に表示されます。

#### 管理者認証メカニズム

デフォルトの組織に属する管理者を認証するために使用されるメカニズムのいずれかを指定します。管理コンソールは、管理者に対して以下の 3 種類の認証方式をサポートしています。

**LDAP ユーザパスワード:** このオプションを選択すると、管理者はディレクトリ サービスに格納されているそれぞれの認証情報を使用して認証されます。

**注:** このメカニズムを管理者の認証に使用する場合、「*ユーザデータ サービス (UDS) の展開*」の説明に従い、UDS を展開します。

**基本:** このオプションを選択すると、管理コンソールで提供される組み込みの認証方式が管理者の認証に使用されます。

**CA Strong Authentication パスワード:** ユーザがここで [CA Strong Authentication パスワード] オプションを選択すると、AuthMinder サーバによって認証情報が発行されて認証されます。この場合、CA AuthMinder サーバがインストールされている必要があります。

**注:** CA Strong Authentication のインストールと設定の詳細については、「CA Strong Authentication インストールおよび展開ガイド」を参照してください。

6. [キー ラベル設定] セクションで、以下の値を指定します。

### グローバル キーの使用

デフォルトのグローバル キーを指定します。上記の手順で指定したグローバル キー ラベルを無効にして、新たに暗号化ラベルを指定する場合は、このオプションを選択解除します。

### キー ラベル

[グローバル キーの使用] オプションをオフにした場合に、新しいキー ラベルを指定します。

### 暗号化ストレージタイプ

暗号化キーがデータベース（ソフトウェア）に格納されるか HSM（ハードウェア）に格納されるかを示します。

7. [終了] をクリックして、ブートストラップ プロセスを完了します。
8. [続行] をクリックして、管理コンソールを使用するほかの設定に進みます。

## CA Risk Authentication サーバ サービスの開始

次の手順に従ってください：

1. デスクトップ ウィンドウ上の [スタート] ボタンをクリックします。
2. [設定] - [コントロール パネル] - [管理ツール] - [サービス] に移動します。
3. [CA Risk Authentication Service] を見つけてダブルクリックします。
4. サービスのウィンドウで、[開始] をクリックします。

注：CA Risk Authentication サーバを停止する場合は、手順 1～3 に従い、サービス ウィンドウで [停止] をクリックします。

## CA Risk Authentication ケース管理サービス

次の手順に従ってください：

1. [スタート] - [設定] - [コントロールパネル] - [管理ツール] - [サービス] をクリックします。
2. [CA Risk Authentication Case Management Service] をダブルクリックします。
3. サービスのウィンドウで、[開始] をクリックします。

注：CA Risk Authentication ケース管理サービスを停止する場合は、手順 1 ～ 3 に従い、サービス ウィンドウで [停止] をクリックします。

## ユーザ データ サービス(UDS)の展開

CA Risk Authentication は、RDBMS から、またはユーザ データ サービス (UDS) を使用して LDAP サーバから直接ユーザ データにアクセスできません。

USD は、CA Risk Authentication に対して、組織によって展開されたサードパーティ データ リポジトリへのシームレスなアクセスを提供する抽象化層です。

次の手順に従ってください：

1. アプリケーション サーバに *arcotuds.war* を展開します。このファイルは以下から入手できます。

`<install_location>%Arcot Systems%java%webapps%`

例：Apache Tomcat の場合は、`<APP_SERVER_HOME>%webapps%` に WAR ファイルを展開します。

注：展開手順は、使用しているアプリケーション サーバによって異なります。詳細な手順については、アプリケーション サーバベンダーのドキュメントを参照してください。

2. (*WebSphere* のみ)アプリケーション ファイルが更新されると、UDS クラスを再ロードするように設定します。次の手順に従ってください：
  - a. [Applications] - [Application Types] - [WebSphere Enterprise Applications] に移動し、[UDS settings] 画面にアクセスします。
  - b. [Class loader order] で、[Classes loaded with local class loader first (parent last)] オプションを選択します。
  - c. [WAR class loader policy] で、[Single class loader for application] を選択します。
  - d. [適用] をクリックします。
3. アプリケーション サーバを再起動します。
4. UDS が正常に展開されていることを確認します。次の手順に従ってください：

注：UDS 関連情報のログ記録には *arcotuds.log* ファイルが使用されません。

- a. 以下の場所に移動します。  
`<install_location>%Arcot Systems%logs%`



- b. 任意のエディタで `arcotuds.log` ファイルを開き、以下の行を見つけます。
  - `User Data Service (Version: 2.0.3) initialized successfully.`  
この行は、UDS が正常に展開されたことを示しています。
- c. ログ ファイルに **FATAL** および **WARNING** のメッセージが含まれていないことを確認します。
- d. ファイルを閉じます。

## ユーザ行動プロファイリング アプリケーションの展開

ユーザ行動プロファイリング (UBP) モデルは、データが不十分な場合に、同じユーザまたはそのピアグループのユーザによる以前のアクセスと現在のトランザクションとの類似点または相違点を測定します。CA Risk Authentication は UBP アプリケーションと通信して類似性スコアを取得し、それをリスク評価スコアに含めます。

UBP を展開するには、ca-userprofiling-2.0-application.war ファイルが必要です。

次の手順に従ってください：

1. アプリケーション サーバに ca-userprofiling-2.0-application.war を展開します。このファイルは以下の場所にあります。

```
<install_location>%Arcot Systems%java%webapps%
```

例：Apache Tomcat の場合は、<APP\_SERVER\_HOME>%webapps% に WAR ファイルを展開します。

注：展開手順は、使用しているアプリケーションサーバによって異なります。詳細については、アプリケーションサーバベンダーのドキュメントを参照してください。

2. (WebSphere の場合) アプリケーションファイルが更新されると、UDS クラスを再ロードするように設定します。

- a. [Application] - [Enterprise Applications] - [UDS Settings] に移動します。

- b. [Class loader order] で、[Classes loaded with local class loader first (parent last)] オプションを選択します。

- c. [WAR class loader policy] で、[Single class loader] を選択します。

- d. bcprov-jdk15-146 jar ファイルを

<ARCOT\_HOME>/sdk/java/lib/external から以下の場所にコピーします。

```
<JRE_HOME>/lib/ext フォルダ
```

注：ここで、JRE\_HOME は WebSphere アプリケーションサーバによって使用される jre インストールです。

- e. [適用] をクリックします。

(WebLogic の場合：サードパーティの JDBC ドライバを使用する方法については、WebLogic のドキュメントを参照してください)

3. アプリケーション サーバを再起動します。
4. UDS が正常に展開されていることを確認します。

注: UDS 関連情報のログ記録には `arcotuds.log` ファイルが使用されません。

- a. 以下の場所に移動します。

`<install_location>%Arcot Systems%logs%`

- b. 任意のエディタで `ubp_logfile.log` ファイルを開き、以下のステートメントを見つけます。
- c. ログ ファイルに **FATAL** および **WARNING** のメッセージが含まれていないことを確認します。
- d. ファイルを閉じます。

## サンプル アプリケーションの展開

サンプルアプリケーションを使用して、CA Risk Authentication が正常にインストールおよび設定されていることを確認します。

また、以下の操作の例を示します。

- 一般的な CA Risk Authentication のワークフロー
- CA Risk Authentication API の基本操作（呼び出しと後処理）
- CA Risk Authentication とアプリケーションの統合

**重要:** サンプルアプリケーションを運用環境で使用しないでください。サンプルアプリケーションのコードを参考にして、独自の Web アプリケーションを作成することをお勧めします。

サンプルアプリケーションは、CA Risk Authentication の Complete インストールの一部として自動的にインストールされます。

次の手順に従ってください：

1. 以下の場所から `ca-riskauth-8.0-sample-application.war` ファイルを展開します。  
`<install_location>%Arcot Systems%samples%java%`
2. 必要に応じて、アプリケーションサーバを再起動します。
3. Web ブラウザでサンプルアプリケーションにアクセスします。以下の URL がサンプルアプリケーションのデフォルトの URL です。  
`http://<host>:<appserver_port>/ca-riskauth-8.0-sample-application/index.jsp`

## インストールの確認

次の手順に従ってください：

1. 以下の場所に移動します。  
<install\_location>¥Arcot Systems¥Logs¥
2. 任意のエディタで arcotwebfortstartup.log ファイルを開き、以下の行を見つけます。
  - STARTING CA Risk Authentication 8.0
  - CA Risk Authentication Service READY
3. 任意のエディタで arcotriskfortcasemgmtserverstartup.log ファイルを開き、以下の行を見つけます。
  - STARTING CA Risk Authentication Case Management Service 8.0
  - CA Risk Authentication Case Management Service READY

**注：**ログファイルに FATAL および WARNING のメッセージが含まれていないことを確認します。

## サンプル アプリケーションをリスク評価に使用する方法

このセクションでは、サンプルアプリケーションをリスク評価操作に使用する方法について説明します。サンプルアプリケーションでの各操作は、CA Risk Authentication がインストールされ、機能していれば、エラーなく実行されるように設計されています。

サンプルアプリケーションでは、CA Risk Authentication サーバが実行できる以下の操作について、その例を示します。

- 初めてのユーザのリスク評価および後評価の実行
- ユーザの作成
- 既知のユーザのリスク評価および後評価の実行
- デフォルト プロファイルの編集およびリスク評価の実行

### 初めてのユーザのリスク評価および後評価の実行

次の手順に従ってください：

1. サンプルアプリケーションが (Web ブラウザで) 開いていることを確認します。以下の URL がサンプルアプリケーションのデフォルトの URL です。

```
http://<host>:<appserver_port>/CA Risk  
Authentication-8.0-sample-application/index.jsp
```

2. [Evaluate Risk] をクリックします。
3. [User Name] フィールドにユーザ (評価対象) の名前を入力します。
4. 必要に応じて、ユーザが所属する組織の名前を [User Organization] フィールドに入力します。
5. 必要に応じて、トランザクションが発生したチャネルを入力します。
6. [Evaluate Risk] をクリックし、[Risk Evaluation Results] ページを開きます。

このページには、リスク スコアおよび関連付けられているリスク アドバイスが表示され、指定した組織用に設定されたルールがリスト表示されます。初めてのユーザの場合、結果は **ALERT** になります。

7. [Next Step] をクリックし、[Post Evaluation] ページを開いて、指定したユーザ プロファイルに対して後評価を実行します。

アプリケーションは後評価を通じて、現在のユーザやユーザが使用しているデバイスに関するフィードバックを CA Risk Authentication サーバに提供します。CA Risk Authentication では、このフィードバックに基づいて、ユーザ属性やデバイス属性、ユーザとデバイスの関連付けを更新し、その後ユーザのトランザクションに伴うリスクを適宜評価します。

8. [Result of Secondary Authentication] リストから適切なオプションを選択して、2 次認証の結果を選択します。
9. ユーザ名とデバイスの関連付けの名前を [Association Name] に入力します。
10. [Post Evaluate] をクリックしてプロセスを完了すると、[Post Evaluation Results] セクションに結果が表示されます。

### ユーザの作成

次の手順に従ってください：

1. GA アカウントを作成するには、以下の手順に従います。
  - a. MA として管理コンソールにログインします。
  - b. [ユーザと管理者] タブがアクティブであることを確認します。
  - c. 左側のメニューで、[管理者の作成] リンクをクリックします。
  - d. 必要な情報を指定し、[次へ] をクリックします。
  - e. [管理者の作成] ページで、[グローバル管理者] を選択します。
  - f. [パスワード] と [パスワードの確認] に入力します。
  - g. [管理する] セクションで [全組織] オプションを選択します。
  - h. [作成] をクリックします。
  - i. ページの右上隅の [ログアウト] をクリックして、MA としてログアウトします。
2. GA (グローバル管理者) または OA (組織管理者) として管理コンソールにログインします。URL は以下のとおりです。  
`http://<host>:<appserver_port>/arcotadmin/adminlogin.htm`
3. パスワードを変更するために表示される手順に従います。
4. [ユーザと管理者] タブの [ユーザと管理者の管理] サブタブをアクティブにします。
5. [ユーザと管理者の管理] (左側のメニュー) に移動し、[ユーザの作成] をクリックします。

6. [ユーザの作成] ページで、以下の手順に従います。
  - a. [ユーザ詳細] セクションに一意のユーザ名、それらの組織名、および必要に応じてその他のユーザ情報を入力します。
  - b. 必要に応じて、対応するフィールドにその他のユーザ情報を入力します。
  - c. 必要なユーザ ステータスを選択します。
  - d. [ユーザの作成] をクリックします。

指定したユーザがデータベースに追加されると、「ユーザを正常に作成しました」というメッセージが表示されます。
7. サンプルアプリケーション ページに戻ります。

### 既知のユーザのリスク評価および後評価の実行

次の手順に従ってください：

1. サンプルアプリケーションのメイン ページで [Evaluate Risk] をクリックします。
2. 「ユーザの作成」 セクションで作成したユーザの名前を入力します。
3. ユーザの組織を入力します。
4. 必要に応じて、トランザクションが発生したチャネルを入力します。
5. [Evaluate Risk] をクリックします。

リスク アドバイスは通常 INCREASEAUTH です。
6. [Store DeviceID] をクリックして、エンドユーザのデバイスにデバイス ID 情報の指定されたタイプを保存します。
7. [Next Step] をクリックして、以下のように後評価を実行します。
  - リストから [Result of Secondary Authentication] を選択します。
  - 必要に応じて [Association Name] を編集します。
8. [Post Evaluate] をクリックして、最終的なアドバイスを表示します。

手順 1～手順 5 を繰り返せば、[Risk Evaluation Results] ページのリスク アドバイスは **ALLOW** に変わります。

### デフォルト プロファイルの編集およびリスク評価の実行



サンプルアプリケーションを使用して、使用しているコンピュータの DeviceDNA、IP アドレス、およびデバイス ID を変更して、さまざまな状況をシミュレートできます。ユーザのデフォルトプロファイルを編集するには、次の手順に従ってください：

1. サンプルアプリケーションのメイン ページで [Evaluate Risk] をクリックします。
2. [User Name] フィールドにプロファイルを編集するユーザ名を入力します。
3. [User Organization] フィールドにユーザの組織を入力します。
4. [Edit Inputs] をクリックします。
5. 生成されたリストから必要に応じて、1 つ以上のフィールドの値を変更します。
6. [Evaluate Risk] をクリックします。
7. [Next Step] をクリックし、[Post Evaluation] ページを開いて、指定したユーザプロファイルに対して後評価を実行します。
8. [Result of Secondary Authentication] リストから適切なオプションを選択して、2 次認証の結果を選択します。
9. [Post Evaluate] をクリックし、後評価プロセスを完了すると、同じ後評価プロセスの結果が表示されます。

**注：**コンポーネント間の安全な通信を確保するために、SSL (Secure Socket Layer) トランスポート モードをサポートするよう設定できます。詳細については、「CA Risk Authentication 管理ガイド」の「SSL の設定」を参照してください。

**重要：**これらのインストール後のタスクを完了したら、「CA Risk Authentication SDK および Web サービスの設定」の説明に従って、SDK および Web サービスの設定を行います。

## インストール後のチェックリストの適用

CA Risk Authentication のインストールおよびセットアップ情報を使用して以下のチェックリストに記入します。各種管理タスクを実行する際に、これらの情報が役立ちます。

| 情報                     | 入力例  | 記入欄 |
|------------------------|--|-----|
| ARCOT_HOME             | C:¥Program Files¥Arcot Systems                       |     |
| <b>システム情報</b>          |  |     |
| ホスト名                   | my-bank  |     |
| User Name              | 管理者  |     |
| Password               | password1234!  |     |
| 設定済みのコンポーネント           | CA Risk Authentication サーバ<br>管理コンソール<br>ユーザデータ サービス |     |
| <b>管理コンソール情報</b>       |  |     |
| ホスト名                   | localhost  |     |
| ポート                    | 8080   |     |
| マスタ管理者パスワード            | mypassword1234!                                      |     |
| <b>ユーザデータ サービス情報</b>   |  |     |
| ホスト名                   | localhost  |     |
| ポート                    | 8080   |     |
| アプリケーション コンテキスト<br>ルート | arcotuds   |     |

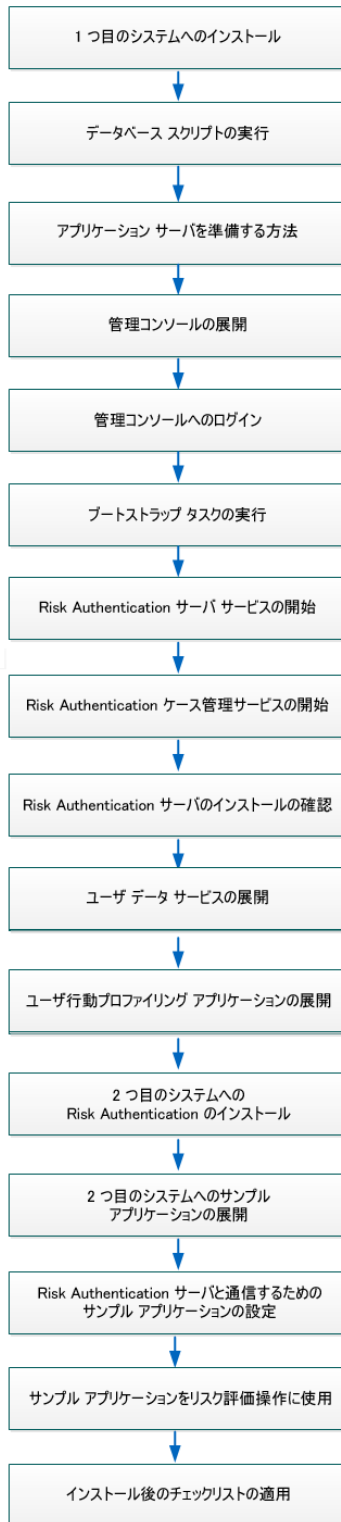
# 第 5 章: 分散システムに CA Risk Authentication を展開する方法

---

CA Risk Authentication コンポーネントのインストールは、CA Risk Authentication 8.0 InstallAnywhere ウィザードを使用して実行します。このウィザードでは Complete と Custom のインストールタイプをサポートしています。分散環境に CA Risk Authentication をインストールして設定する場合、インストーラを実行する際に [Custom] オプションを使用します。

以下の図は、CA Risk Authentication 8.0 をインストールするために実行するタスクを示しています。

Risk Authentication を分散システムに展開する方法



以下のタスクを実行します。

1. [1つ目のシステムへのインストール](#) (P. 88)
2. [データベース スクリプトの実行](#) (P. 97)
3. [アプリケーション サーバを準備する方法](#) (P. 98)
4. [管理コンソールの展開](#) (P. 107)
5. [管理コンソールへのログイン](#) (P. 109)
6. [ブートストラップ タスクの実行](#) (P. 109)
7. [CA Risk Authentication サーバサービスの開始](#) (P. 112)
8. [CA Risk Authentication ケース管理サービスの開始](#) (P. 112)
9. [CA Risk Authentication サーバのインストールの確認](#) (P. 113)
10. [ユーザ データ サービスの展開](#) (P. 113)
11. [ユーザ行動プロファイリング アプリケーションの展開](#) (P. 74)
12. [2つ目のシステムへの CA Risk Authentication のインストール](#) (P. 116)
13. [2つ目のシステムへのサンプルアプリケーションの展開](#) (P. 117)
14. [CA Risk Authentication サーバと通信するためのサンプルアプリケーションの設定](#) (P. 117)
15. [サンプルアプリケーションをリスク評価操作に使用](#) (P. 119)
16. [インストール後のチェックリストの適用](#) (P. 82)

### 重要:

単一のシステムまたは分散環境に **CA Risk Authentication** をインストールする際は、以下の点に注意してください。

- `<install_location>` には特殊文字が含まれないようにしてください (~ ! @ # \$ % ^ & \* ( ) \_ + = { } [ ] " ' など)。
- MySQL データベース名にドット (.) 文字を含めることはできません。
- 現時点では、インストーラを使用して **CA Risk Authentication** コンポーネントを変更または修復することはできません。必ずコンポーネントをアンインストールしてから、再インストールしてください。

- インストールの実行中は、インストーラ ウィンドウを閉じないでください。インストール中（特に最後の段階）に [Cancel] ボタンをクリックしてインストールを中止した場合、それまでに作成されたディレクトリはすべてが削除されるとは限りません。インストールディレクトリ、<install\_location>¥Arcot Systems¥、およびそのサブディレクトリは手動でクリーンアップする必要があります。
- 既存の ARCOT\_HOME のインスタンスがすでに含まれているシステム上でインストーラを実行する場合。
  - インストールディレクトリを要求されません。
  - データベースのセットアップを要求されません。インストーラは既存のデータベースを使用します。
  - 暗号化をセットアップするように要求されません。

このセクションには、以下のトピックが含まれています。

[1つ目のシステムへのインストール \(P. 88\)](#)

[データベース スクリプトの実行 \(P. 97\)](#)

[アプリケーション サーバを準備する方法 \(P. 98\)](#)

## 1 つ目のシステムへのインストール

分散システム インストールでは、1 つ目のシステムに CA Risk Authentication サーバをインストールします。

上級ユーザには、コンポーネントを選択してインストールできる *Custom* インストールをお勧めします。

インストールを正常に実行するには、インストールに使用するユーザアカウントが Administrators グループに属している必要があります。

**注:** 「インストールの準備」の説明に従って、事前インストールソフトウェアコンポーネントがすべてインストールされ、データベースがセットアップされていることを確認してください。

次の手順に従ってください：

1. CA Risk Authentication-8.0-Windows-Installer.exe ファイルのあるディレクトリに移動し、このファイルをダブルクリックします。
2. [次へ] をクリックして続行します。
3. [I accept the terms of the License Agreement] オプションを選択します。  
[次へ] をクリックします。

**注:** インストーラによって、ほかの CA 製品がコンピュータにインストールされているかどうかを確認されます。インストールされた既存の CA 製品が見つからなければ、インストールディレクトリの入力を求めるプロンプトが表示されます。

インストーラが既存の CA 製品インストール（既存の ARCOT\_HOME）を検出した場合

- インストールディレクトリを要求されません。
  - データベースおよび暗号化のセットアップを要求されない場合、インストーラは既存のデータベースおよび暗号化設定を使用します。そのため、手順 8 の画面が表示されます。
4. インストールディレクトリの場所を入力し、[Next] をクリックします。
  5. [Custom] を選択して、[Next] をクリックします。
  6. 以下のコンポーネントのうち、必要のないものを選択解除します。デフォルトでは、すべてのコンポーネントが選択されています。



例：CA Risk Authentication サーバ、ケース管理キュー サーバ、管理コンソール（SDK およびサンプルアプリケーションなし）、およびユーザ行動プロファイリング アプリケーションをインストールする場合は、以下のオプションを選択します。

- a. リスク評価サーバ
- b. ケース管理キュー サーバ
- c. 管理コンソール
- d. ユーザ データ サービス
- e. ユーザ行動プロファイリング アプリケーション

注：サンプルアプリケーションをインストールするには、[CA Risk Authentication SDKs and Sample Application] オプションを選択して、インストールを続行します。

コンポーネントに関する情報を以下の表に示します。

| [Component]  | Description   |
|--------------|---|
| リスク評価サーバ     | <p>管理コンソールからの以下のリクエストを処理するコア処理エンジン（CA Risk Authentication サーバ）がインストールされます。</p> <ul style="list-style-type: none"> <li>■ リスク評価</li> <li>■ Configuration</li> </ul> <p>また、このコンポーネントでは、サーバに組み込まれている以下の Web サービスもインストールされます。</p> <ul style="list-style-type: none"> <li>■ <b>リスク評価 Web サービス</b>：CA Risk Authentication サーバによるリスク評価用の Web ベースのプログラミング インターフェースを提供します。</li> <li>■ <b>ユーザ管理 Web サービス</b>：ユーザの作成と管理用の Web ベースプログラミング インターフェースを提供します。</li> <li>■ <b>管理 Web サービス</b>：管理コンソールで使用される Web ベースのプログラミング インターフェースを提供します。</li> </ul> |
| ケース管理キュー サーバ | <p>ケースに対応するテクニカル サポート担当者（CSR）にケースを割り当てるコア キュー エンジン（ケース管理キュー サーバ）をインストールします。</p> <p>注：管理コンソールのすべてのインスタンスは、ある一時点では、ケース管理キュー サーバの単一のインスタンスにのみ接続できます。</p>   |

| [Component]                                       | Description  |
|---|--|
| <p>CA Risk Authentication SDK およびサンプルアプリケーション</p> | <p>CA Risk Authentication サーバにリスク評価リクエストを転送するためにアプリケーションから呼び出すことができるプログラミング インターフェースを（API および Web サービスの形式で）提供します。このパッケージは、以下のサブコンポーネントで構成されます。</p> <ul style="list-style-type: none"> <li>■ <b>リスク評価 SDK</b>： CA Risk Authentication サーバによるリスク評価用の Java プログラミング インターフェースを提供します。</li> <li>■ <b>サンプルアプリケーション</b>： CA Risk Authentication Java API の使用方法の例を示します。 CA Risk Authentication が正常にインストールされているかどうかの確認、およびリスク評価リクエストを実行できるかどうかの確認にも使用できます。</li> </ul> <p>詳細については、「CA Risk Authentication SDK および Web サービスの設定」を参照してください。</p> |
| <p>管理コンソール</p>                                    | <p>CA Risk Authentication サーバおよびリスク評価関連の設定を管理するための Web ベースのインターフェースを提供します。</p>   |
| <p>ユーザデータ サービス</p>                                | <p>リレーショナルデータベース（RDBMS）やディレクトリサーバ（LDAP）など、各種ユーザリポジトリにアクセスするための抽象化層として機能する UDS をインストールします。</p>  |
| <p>ユーザ行動プロファイリング</p>                              | <p>データが不十分な場合に、同じユーザまたはそのピアグループのユーザによる以前のアクセスと現在のトランザクションとの類似点または相違点を測定します。</p>  |

注: この画面で [Evaluation Server] オプションを選択しなかった場合、手順 7 から手順 9 の画面は表示されません。

[Next] を選択して続行します。

1. データベース タイプ (Microsoft SQL Server、Oracle データベース、または MySQL) を選択します。

[次へ] をクリックします。

注: Microsoft SQL Server データベースを使用している場合、ODBC ドライバのバージョンが「データベース サーバの設定」の章に記載されているものと同じであることを確認してください。

2. 選択したデータベースに応じて、データベースの詳細を入力します。
  - Microsoft SQL Server を選択した場合は、以下の表に示されている詳細を入力します。

| パラメータ    | Description   |
|----------|---|
| ODBC DSN | <p>インストーラはこの値を使用して DSN を作成します。CA Risk Authentication サーバは、この DSN を使用してデータベースに接続します。推奨される入力値は arcotdsn です。</p> <p>注: データベース ソース名 (DSN) によって、ODBC ドライバを使用してデータベースに接続する際に必要な情報が指定されます。この情報にはデータベース名、ディレクトリ、データベース ドライバ、ユーザ ID、およびパスワードが含まれます。</p>  |
| サーバ      | <p>CA Risk Authentication データストアのホスト名または IP アドレス。</p> <p>デフォルト インスタンス</p> <ul style="list-style-type: none"> <li>■ 構文: &lt;server_name&gt;</li> <li>■ 例: demodatabase</li> </ul> <p>名前付きインスタンス</p> <ul style="list-style-type: none"> <li>■ 構文: &lt;server_name&gt;¥&lt;instance_name&gt;</li> <li>■ 例: demodatabase¥instance1</li> </ul> |

| パラメータ       | Description  |
|-------------|--|
| User Name   | <p>CA Risk Authentication がデータベースにアクセスする際のデータベース ユーザ名。この名前は、データベース管理者によって指定されます。(MS SQL Server では一般的に、このユーザ名を「ログイン」と呼びます)。</p> <p>このユーザには、セッションの作成権限と DBA 権限が付与されている必要があります。</p> <p>注: ユーザ名はプライマリ DSN とバックアップ用 DSN とで異なる必要があります。</p> |
| Password    | <p>上記のフィールドで指定したユーザ名に関連付けられているパスワード。CA Risk Authentication がデータベースにアクセスする際に使用されます。このパスワードはデータベース管理者によって指定されます。</p>   |
| データベース      | <p>MS SQL データベース インスタンスの名前。</p>  |
| Port Number | <p>データベースが受信リクエストをリスンするポート。デフォルトのポートは 1433 です。ただし、別のポートを指定する場合は、このフィールドにポート値を入力します。</p>  |

- Oracle データベースを選択した場合は、フィールドに以下の情報を入力します。

| パラメータ     | Description  |
|-----------|--|
| ODBC DSN  | <p>インストーラはこの値を使用して DSN を作成します。CA Risk Authentication サーバは、この DSN を使用して CA Risk Authentication データベースに接続します。推奨される入力値は arcotdsn です。</p> <p>注: データベース ソース名 (DSN) によって、ODBC ドライバを使用してデータベースに接続する際に必要な情報が指定されます。この情報にはデータベース名、ディレクトリ、データベース ドライバ、ユーザ ID、およびパスワードが含まれます。</p> |
| User Name | <p>CA Risk Authentication がデータベースにアクセスする際のデータベース ユーザ名。この名前は、データベース管理者によって指定されます。(MS SQL Server では一般的に、このユーザ名を「ログイン」と呼びます)。</p> <p>このユーザには、セッションの作成権限と DBA 権限が付与されている必要があります。</p> <p>注: ユーザ名はプライマリ DSN とバックアップ用 DSN とで異なる必要があります。</p>                                   |

| パラメータ       | Description  |
|-------------|--|
| Password    | 上記のフィールドで指定したユーザ名に関連付けられているパスワード。CA Risk Authentication がデータベースにアクセスする際に使用されます。このパスワードはデータベース管理者によって指定されます。  |
| Service ID  | サーバ上で実行される Oracle データベースのインスタンスを表す Oracle システム識別子 (SID)  |
| Port Number | データベースが受信リクエストをリスンするポート。Oracle データベースがリスンするデフォルトポートは 1521 です。ただし、別のポートを指定する場合は、このフィールドにポート値を入力します。   |
| ホスト名        | CA Risk Authentication データストアのホスト名または IP アドレス。<br><ul style="list-style-type: none"> <li>■ 構文 : &lt;server_name&gt;</li> <li>■ 例 : demodatabase</li> </ul> |

- MySQL を選択した場合は、以下の情報を入力します。

| パラメータ    | Description   |
|----------|---|
| ODBC DSN | インストーラはこの値を使用して DSN を作成します。CA Risk Authentication サーバは、この DSN を使用して CA Risk Authentication データベースに接続します。推奨される入力値は arcotdsn です。<br>注: データベース ソース名 (DSN) によって、ODBC ドライバを使用してデータベースに接続する際に必要な情報が指定されます。この情報にはデータベース名、ディレクトリ、データベース ドライバ、ユーザ ID、およびパスワードが含まれます。   |
| サーバ      | CA Risk Authentication データストアのホスト名または IP アドレス。<br>デフォルト インスタンス<br><ul style="list-style-type: none"> <li>■ 構文 : &lt;server_name&gt;</li> <li>■ 例 : demodatabase</li> </ul> 名前付きインスタンス<br><ul style="list-style-type: none"> <li>■ 構文 : &lt;server_name&gt;¥&lt;instance_name&gt;</li> <li>■ 例 : demodatabase¥instance1</li> </ul> |

| パラメータ       | Description   |
|-------------|---|
| User Name   | <p>CA Risk Authentication がデータベースにアクセスする際のデータベース ユーザ名。この名前は、データベース管理者によって指定されます。</p> <p>このユーザには、セッションの作成権限と DBA 権限が付与されている必要があります。</p> <p><b>注:</b> ユーザ名はプライマリ DSN とバックアップ用 DSN とで異なる必要があります。</p> |
| Password    | <p>上記のフィールドで指定したユーザ名に関連付けられているパスワード。CA Risk Authentication がデータベースにアクセスする際に使用されます。このパスワードはデータベース管理者によって指定されます。</p>  |
| データベース      | <p>MySQL データベース インスタンスの名前。</p>  |
| Port Number | <p>データベースが受信リクエストをリスンするポート。MySQL データベースがリスンするデフォルトポートは 3306 です。ただし、別のポートを指定する場合は、このフィールドにポート値を入力します。</p>  |

1. データベースに正常に接続できるかどうかテストするには、[Test Data Source] ボタンをクリックして結果を確認します。  
[次へ] をクリックして続行します。
2. 暗号化のセットアップで、以下の情報を指定します。

#### マスタ キー

<install\_location>%Arcot Systems%conf%securestore.enc に格納され、データベースに格納されたデータを暗号化するために使用されるマスタ キーのパスワードを指定します。デフォルトでは、この値は MasterKey に設定されています。

注: インストール後にマスタ キーの値を変更する場合は、新しいマスタ キーの値を使用して securestore.enc を再生成する必要があります。詳細については、「インストール後のハードウェア セキュリティ モジュール情報の変更」を参照してください。

#### HSM の設定

機密データの暗号化にハードウェア セキュリティ モジュール (HSM) を使用するかどうかを指定します。

このオプションを選択しない場合、デフォルトでは、ソフトウェア モードを使用してデータが暗号化されます。

#### PIN

HSM に接続するためのパスワードを指定します。

#### Choose Hardware Module

Luna HSM と nCipher netHSM の 2 つのオプションから、使用する HSM を指定します。

#### HSM パラメータ

以下の HSM 情報を指定します。

- **Shared Library** : HSM に対応する PKCS#11 共有ライブラリの絶対パス。

Luna (cryptoki.dll) および nCipher netHSM (cknfast.dll) の場合は、ファイルの絶対パスと名前を指定します。

- **Storage Slot Number** : データの暗号化に使用される 3DES キーが使用可能な HSM スロット。

Luna の場合、デフォルト値は 0 です。

nCipher netHSM の場合、デフォルト値は 1 です。

3. [次へ] をクリックします。
4. [Install] をクリックし、インストールプロセスを開始します。

**注:** CA Risk Authentication をインストールしている現在のシステムに Microsoft Visual C++ 2010 x86 がない場合は、[Microsoft Visual C++ 2010 x86 Redistributable Setup] 画面が表示されます。

[Microsoft Visual C++ 2010 x86 Redistributable Setup] 画面で、以下の手順を実行します。

- a. [I have read and accept the license terms] オプションを選択して、[Install] をクリックします。
- b. [完了] をクリックします。

CA Risk Authentication のインストールを続行します。

5. [終了] をクリックします。

**注:** インストールが完了したら、以降のセクションの説明に従ってインストール後のタスクを実行してください。

### インストール ログ

インストールの完了後、<install\_location> ディレクトリのインストール ログ ファイル (Arcot\_RiskFort\_Install\_<timestamp>.log) にアクセスできます。

**例:** インストールディレクトリとして C:¥Program Files ディレクトリを指定した場合、インストール ログ ファイルは C:¥Program Files ディレクトリに作成されます。

インストールが何らかの理由で失敗した場合、エラー メッセージはこのログ ファイルに記録されます。



## データベース スクリプトの実行

データベース スクリプトを実行するには、以下の手順に従います。

**重要:** 実行する前に、「データベース サーバの設定」セクションで作成したときと同じデータベース ユーザとしてログインしていることを確認してください。

次の手順に従ってください：

1. 以下のディレクトリに移動します。  
`<install_location>\Arcot Systems\dbscripts\`
2. 使用しているデータベースに応じて、以下のサブディレクトリに移動します。
  - Oracle の場合：Oracle\
  - Microsoft SQL Server の場合：mssql\
  - MySQL の場合：mysql\
3. スクリプトを次に示す順序で実行します。
  - a. `arcot-db-config-for common-2.0.sql`  
**重要:** CA Strong Authentication 8.0 をインストール済みの場合は、`arcot-db-config-for-common-2.0.sql` を実行しないでください。
  - b. `arcot-db-config-for-riskfort-8.0.sql`
  - c. (3D セキュア チャネルを作成する必要がある場合にのみ、オプション) `arcot-db-config-for-3dsecure-8.0.sql`
  - d. (オプション) ユーザ行動プロファイリングを使用する場合にのみ、以下のコマンドを実行します。  
`arcot-db-config-for-userprofiling-2.0.sql`

### データベースのセットアップの確認

必要なデータベース スクリプトを実行した後、CA Risk Authentication スキーマが正しく機能していることを確認します。

次の手順に従ってください：

1. データベースをインストールしたユーザとして CA Risk Authentication データベースにログインします。

注: アップグレードパスに従っている場合は、データベースをアップグレードしたユーザとしてデータベースにログインします。

2. 以下のクエリを実行します。

```
SELECT SERVERNAME, VERSION FROM ARRFSEVERERS;
```

結果として、以下の出力が表示されます。

| SERVERNAME             | VERSION |
|------------------------|---------|
| -----                  | -----   |
| RiskFort               | 8.0     |
| RiskFortCaseManagement | 8.0     |

3. データベース コンソールからログアウトします。

## アプリケーション サーバを準備する方法

CA Risk Authentication のコンポーネントであるユーザデータ サービス (UDS) および管理コンソールは、Web ベースのコンポーネントであり、以下のサポート対象アプリケーション サーバのいずれかに展開します。

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic
- JBoss アプリケーション サーバ

選択したアプリケーション サーバにこれらの Web アプリケーションの WAR ファイルを展開する前に、UDS および管理コンソールに必要なファイルをアプリケーション サーバの適切な場所にコピーします。このセクションでは、アプリケーション サーバに必要な暗号化ファイルをコピーし、以下の Web アプリケーションの WAR ファイルを展開する手順について説明します。

1. [Java ホームの設定](#) (P. 99)
2. [アプリケーションへのデータベース アクセス ファイルのコピー](#) (P. 99)
3. [アプリケーション サーバへの JDBC JAR ファイルのコピー](#) (P. 103)
4. [Enterprise Archive ファイルの作成](#) (P. 106)

## Java ホームの設定

次の手順に従ってください：

1. JAVA\_HOME 環境変数を設定していることを確認します。この JAVA\_HOME は、ユーザのアプリケーションサーバ JAVA\_HOME である必要があります。
2. %JAVA\_HOME%\bin を PATH 変数に追加します。含めなかった場合、管理コンソール、UDS、およびその他の JDK 依存コンポーネントが起動しない可能性があります。

## アプリケーション サーバへのデータベース アクセス ファイルのコピー

UDS および管理コンソールでは、データベースに安全にアクセスするために以下のファイルを使用します。

- arcot-crypto-util.jar。以下の場所にあります。  
<install\_location>%Arcot Systems%java%lib%
- ArcotAccessKeyProvider.dll。以下の場所にあります。  
<install\_location>%Arcot Systems%native%win%<32bit-or-64bit>%

CA Risk Authentication コンポーネントを展開したアプリケーションサーバ上の適切な場所にこれらのファイルをコピーします。

### Apache Tomcat

ファイルをコピーするには、以下の手順に従います。

1. arcot-crypto-util.jar を <Tomcat\_JAVA\_HOME>%jre%lib%ext% にコピーします。  
**注:** ここで、<Tomcat\_JAVA\_HOME> は、Apache Tomcat インスタンスによって使用される JAVA\_HOME を表します。
2. ArcotAccessKeyProvider.dll を <Tomcat\_JAVA\_HOME>%jre%bin% にコピーします。
3. アプリケーションサーバを再起動します。

### IBM WebSphere

ファイルをコピーするには、以下の手順に従います。

1. WebSphere Administration Console にログインします。
2. [Environment] - [Shared Libraries] をクリックします。
  - a. 有効な可視性範囲を選択します。範囲には、アプリケーションを展開するターゲット サーバまたはノードを含める必要があります。
  - b. [新規] をクリックします。
  - c. 名前を入力します。

例：ArcotJNI。
  - d. クラスパスを指定します。

このパスは、arcot-crypto-util.jar ファイルが存在し、ファイル名が含まれる場所を指している必要があります。

例：C:\Program Files\Arcot Systems\java\lib\arcot-crypto-util.jar
  - e. JNI ライブラリ パスを入力します。

このパスは、ArcotAccessKeyProvider.dll ファイルが存在する場所を指している必要があります。
3. [Apply] をクリックして、変更を保存します。
4. サーバレベルのクラス ローダを設定します。
5. [Servers] - [Server Types] - [WebSphere Application Servers] に移動します。
  - a. [Application Servers] で、サーバの設定ページにアクセスします。
  - b. [Java and Process Management]-[Class Loader]をクリックします。
  - c. [新規] をクリックします。
  - d. デフォルトの [Classes loaded with parent class loader] を選択して、[OK] をクリックします。
  - e. 自動生成されたクラス ローダ ID をクリックします。
  - f. クラス ローダの [Configuration] ページで、[Shared Library References] をクリックします。
  - g. [Add] - [ArcotJNI] - [Apply] をクリックします。
  - h. 変更を保存します。

6. ArcotAccessKeyProvider.dll を <WebSphere\_JAVA\_HOME>%jre%bin% にコピーします。

ここで、<WebSphere\_JAVA\_HOME> は、IBM WebSphere インスタンスによって使用される JAVA\_HOME を表します。

7. WebSphere を再起動します。

## Oracle WebLogic

ファイルをコピーするには、以下の手順に従います。

1. ArcotAccessKeyProvider.dll を <WebLogic\_JAVA\_HOME>%jre%bin% にコピーします。

ここで、<WebLogic\_JAVA\_HOME> は、Oracle WebLogic インスタンスによって使用される JAVA\_HOME を表します。

2. arcot-crypto-util.jar を <WebLogic\_JAVA\_HOME>%jre%lib%ext% にコピーします。

注: 必ず WebLogic によって使用される適切な <JAVA\_HOME> を使用してください。

3. WebLogic Administration Console にログインします。
4. [Deployments] に移動します。
5. [Lock and Edit] オプションを有効にします。
6. [Install] をクリックして、arcot-crypto-util.jar ファイルが含まれるディレクトリに移動します。
7. [Next] をクリックし、アプリケーションインストールアシスタントを開きます。
8. [Next] をクリックして、[Summary] ページを開きます。
9. [完了] をクリックします。
10. 変更を有効にします。
11. サーバを再起動します。

### JBoss アプリケーション サーバ

ファイルをコピーするには、以下の手順に従います。

1. ArcotAccessKeyProvider.dll を <JBoss\_JAVA\_HOME>%jre%bin% にコピーします。

**注:** ここで、<JBoss\_JAVA\_HOME> は、JBoss アプリケーション サーバインスタンスによって使用される JAVA\_HOME を表します。

2. <JBOSSE\_HOME>%modules%advauth-admin-libs%main% というフォルダ構造を作成し、<ARCOT\_HOME>%java%lib から以下の JAR をこのフォルダにコピーします。

- arcot-crypto-util.jar
- bcprov-jdk15-146.jar

3. 同じフォルダ (<JBOSSE\_HOME>%modules%advauth-admin-libs%main%) 内に module.xml という名前で、以下のコードを持つファイルを作成します。

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="advauth-admin-libs">
  <resources>
    <resource-root path="arcot-crypto-util.jar"/>
    <resource-root path="bcprov-jdk15-146.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
```

4. アプリケーション サーバを再起動します。

## アプリケーション サーバへの JDBC JAR ファイルのコピー

CA Risk Authentication は、以下の JDBC JAR ファイルをサポート対象のデータベースに必要とします。

- **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)

以下のセクションでは、データベースに必要な JDBC JAR を以下のアプリケーション サーバにコピーするための手順について説明します。

### Apache Tomcat

必要な JDBC JAR ファイルをコピーするには、以下の手順に従います。

1. <Database\_JAR> ファイルをダウンロードした場所に移動します。
2. <Database\_JAR> ファイルを以下のディレクトリにコピーします。
  - **Apache Tomcat 5.5.x の場合** : <TOMCAT\_HOME>¥common¥lib¥
  - **Apache Tomcat 6.x および 7.x の場合** : <TOMCAT\_HOME>¥lib¥
3. アプリケーション サーバを再起動します。

### IBM WebSphere

必要な JDBC JAR ファイルをコピーするには、以下の手順に従います。

1. WebSphere Administration Console にログインします。
2. [Environment] - [Shared Libraries] をクリックします。以下の手順に従います。
  - a. [Scope] リストから、有効な可視性範囲を選択します。範囲には、アプリケーションを展開するターゲットサーバまたはノードを含める必要があります。
  - b. [新規] をクリックします。
  - c. 名前を入力します (例 : JDBCJAR) 。
  - d. クラスパスを指定します。

**重要:** このパスは、<Database\_JAR> ファイルが存在し、ファイル名が含まれる場所を指している必要があります。

- e. [適用] をクリックします。
3. サーバレベルのクラスローダを設定するには、以下の手順に従います。
  - a. [Servers] - [Server Types] - [WebSphere Application Servers] に移動します。
  - b. [Application Servers] で、設定を行うサーバの設定ページにアクセスします。
  - c. [Java and Process Management] をクリックし、[Class Loader] をクリックします。
  - d. [新規] をクリックします。
  - e. デフォルトの [Classes loaded with parent class loader first] を選択します。  
[OK] をクリックします。
  - f. 自動生成されたクラスローダ ID をクリックします。
  - g. [Shared Library References] をクリックします。
  - h. [Add] をクリックして [JDBCJAR] を選択し、[Apply] をクリックします。
  - i. 変更を保存します。
4. アプリケーションサーバを再起動します。

### Oracle WebLogic

必要な JDBC JAR ファイルをコピーするには、以下の手順に従います。

**注:** Oracle データベースを使用している場合、WebLogic はデフォルトで Oracle データベースをサポートしているため、このセクションで説明されている設定を行わないでください。

1. <Database\_JAR> ファイルを <Weblogic\_JAVA\_HOME>%lib%ext% にコピーします。  
ここで、<WebLogic\_JAVA\_HOME> は、Oracle WebLogic インスタンスによって使用される JAVA\_HOME を表します。
2. WebLogic Administration Console にログインします。
3. [Deployments] に移動します。
4. [Lock and Edit] オプションを有効にします。



5. [Install] をクリックして、必要な <Database\_JAR> ファイルが含まれるディレクトリに移動します。
6. [Next] をクリックし、[Application Installation Assistant] ページを開きます。
7. [Next] をクリックして、[Summary] ページを表示します。
8. [完了] をクリックします。
9. 変更を有効にします。
10. アプリケーション サーバを再起動します。

## JBoss アプリケーション サーバ

必要な JDBC JAR ファイルをコピーするには、以下の手順に従います。

次の手順に従ってください：

1. このフォルダに <JBASS\_HOME>%modules%advauth-jdbc-driver%main% というフォルダ構造を作成し、そのフォルダに JDBC Jar ファイルをコピーします。
2. <JBASS\_HOME>%modules%advauth-jdbc-driver%main% に *module.xml* という名前でファイルを作成します。
3. ファイルに、以下のコードを追加します。

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="advauth-jdbc-driver">
  <resources>
    <resource-root path="<JDBC Jar Name>" />
  </resources>
  <dependencies>
    <module name="javax.api" />
    <module name="javax.transaction.api" />
  </dependencies>
</module>
```

4. JDBC Jar ファイル名を指定しているタグで「<JDBC Jar Name>」を編集します。

例：sqljdbc.jar

アプリケーション サーバを再起動します。

### Enterprise Archive ファイルの作成

#### Oracle WebLogic 10.1 で有効

デフォルトで、UDS と管理コンソールを展開するための WAR ファイルが提供されます。必要に応じて、これらのファイルの形式を Enterprise ARchive (EAR) に変更し、EAR ファイルを展開できます。

UDS と管理コンソールの両方の EAR ファイルを個別に生成できます。または、両方の Web アーカイブを含む単一の EAR ファイルを生成することもできます。

#### 個別の EAR ファイルの生成

以下の手順に従います。

1. コマンドプロンプト ウィンドウを開きます。
2. `<install_location>%Arcot Systems%tools%common%bundlemanager%` ディレクトリに移動します。

3. EAR ファイルを作成するには、以下のコマンドを実行します。

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

このコマンドによって、以下の場所に個別の EAR ファイルが生成されます。

```
<install_location>%Arcot Systems%java%webapps%
```

#### 単一の EAR ファイルの生成

以下の手順を実行します。

1. コマンドプロンプト ウィンドウを開きます。
2. `<install_location>%Arcot Systems%tools%common%bundlemanager%` ディレクトリに移動します。

3. EAR ファイルを作成するには、以下のコマンドを実行します。

```
java -jar bundle-manager.jar -ear <filename.ear> -warList arcotadmin.war arcotuds.war
```

このコマンドによって、以下の場所に単一の EAR ファイルが生成されます。

```
<install_location>%Arcot Systems%java%webapps%
```

## 第 6 章：管理コンソールの展開

---

管理コンソールは、サーバ設定のカスタマイズや展開したシステムの管理を実行できる CA Risk Authentication へのブラウザ ベースのインターフェースです。

CA Risk Authentication を管理するためには、CA Risk Authentication サーバがインストールされているシステムに、管理コンソールがホスト名でアクセスできることを確認します。

**注：** IBM WebSphere 7.0、8.0、または 8.5 に管理コンソールを展開する場合は、付録「IBM WebSphere への管理コンソールの展開」の説明を参照してください。

次の手順に従ってください：

1. アプリケーション サーバの適切なディレクトリに `arcotadmin.war` を展開します。

**注：** 展開手順は、使用しているアプリケーション サーバによって異なります。詳細な手順については、アプリケーション サーバベンダーのドキュメントを参照してください。

**例：** Apache Tomcat の場合は、`<APP_SERVER_HOME>\webapps\` に WAR ファイルを展開する必要があります。

2. (32 ビットの *WebSphere*) : アプリケーションファイルが更新されると、Admin クラスを再ロードするように設定します。
  - a. [Application] - [Enterprise Applications] に移動し、[Admin settings] ページにアクセスします。
  - b. [Class loader order] で、[Classes loaded with local class loader first (parent last)] オプションを選択します。
  - c. [WAR class loader policy] で、[Single class loader for application] を選択します。
  - d. [適用] をクリックします。
  - e. Admin アプリケーションを再起動します。
3. アプリケーション サーバを再起動します。
4. コンソールが正常に展開されていることを確認するには、以下の手順に従います。
  - a. 以下の場所に移動します。  
`<install_location>%Arcot Systems%logs%`
  - b. 任意のエディタで `arcotadmin.log` ファイルを開き、以下の行を見つけます。
    - 2.0.3
    - Administration Console Configured Successfully.

注: これらの行は、管理コンソールが正常に展開されていることを示しています。

  - c. ログファイルに **FATAL** および **WARNING** のメッセージが含まれていないことを確認します。
  - d. ファイルを閉じます。

## 管理コンソールへのログイン

初めて管理コンソールにログインするときは、展開時にデータベースに自動的に設定される MA（マスタ管理者）認証情報を使用します。

次の手順に従ってください：

1. Web ブラウザ ウィンドウで、管理コンソールを起動します。管理コンソールのデフォルト URL は以下のとおりです。

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

注：上記の URI で指定するホストとポートの情報は、管理コンソールを展開したアプリケーション サーバのものである必要があります。

例：Apache Tomcat の場合は、デフォルト ホストは localhost であり、ポートは 8080 です。

2. デフォルトの Master Administrator アカウント クレデンシャルを使用してログインします。認証情報は以下のとおりです。
  - ユーザ名：`masteradmin`
  - パスワード：`master1234!`

## 第 7 章：ブートストラップ タスクの実行

---

ブートストラップは、これらのセットアップタスクについて説明するウィザード主導のプロセスです。ほかの管理リンクは、ブートストラップタスクを実行した後で有効になります。

「ブートストラップタスクの実行」に進む前に、デフォルトの組織に関する概念を理解しておく必要があります。

### デフォルトの組織

Administration Console を展開すると、組織が 1 つ自動的に作成されます。この組織はデフォルトの組織 (DEFAULTORG) と呼ばれます。単一の組織システムとして、デフォルトの組織は、ほかの組織を作成せずにそれ自身で使用できます。

管理コンソールを使用して CA Risk Authentication の管理を始めるには、以下のタスクを実行して、システムのブートストラップ設定を初期化する必要があります。

- デフォルトのマスタ管理者パスワードの変更
- グローバルキーラベルの設定
- デフォルトの組織の設定を指定

次の手順に従ってください：

1. [開始] をクリックします。
2. [現在のパスワード]、[新規パスワード]、[パスワードの確認] を入力し、[次へ] をクリックします。
3. 以下のフィールドに入力します。

#### グローバルキーラベル

ハードウェアまたはソフトウェア暗号化に関係なく、ユーザおよび組織のデータを暗号化するために使用される暗号化キーを指定します。CA Risk Authentication では、ハードウェアまたはソフトウェアベースの機密データの暗号化を使用できます。デフォルトではソフトウェアベースの暗号化が有効ですが、*arcotcommon.ini* ファイルを使用してハードウェアベースの暗号化を有効にできます。ハードウェアの暗号化を使用している場合、このラベルは、HSM デバイスに格納されている実際の 3DES キーへの参照 (ポイント) としてのみ機能します。そのため、HSM キーラベルと一致する必要があります。ソフトウェアベースの暗号化の場合、このラベルはキーとして機能します。

**注意：** ブートストラッププロセスの完了後に、このキー ラベルを更新することはできません。

### 暗号化ストレージタイプ

このオプションを指定して、暗号化キーがデータベース（ソフトウェア）に格納されているか、HSM（ハードウェア）に格納されているかを示します。

4. [次へ] をクリックして続行します。
5. デフォルトの組織に以下のパラメータを入力し、[次へ] をクリックします。

### 表示名

組織のわかりやすい名前を指定します。この名前は、管理コンソールの他のすべてのページおよびレポート上に表示されます。

### 管理者認証メカニズム

デフォルトの組織に属する管理者の認証に使用されるメカニズムを指定します。管理コンソールは、管理者がログインするための以下の3種類の認証方式をサポートしています。

#### LDAP ユーザ パスワード

管理者がディレクトリ サービスに格納されているそれぞれの認証情報を使用して認証されることを指定します。

このメカニズムを管理者の認証に使用する場合、「ユーザデータ サービス (UDS) の展開」の説明に従い、UDS を展開します。

#### 基本

管理コンソールで提供される組み込みの認証方式が管理者の認証に使用されることを指定します。

#### WebFort パスワード

認証情報が CA Strong Authentication サーバによって発行および認証されることを指定します。このオプションを使用するには、CA Strong Authentication をインストールします。

CA Strong Authentication のインストールおよび設定の詳細については、「CA Strong Authentication インストールおよび展開ガイド」を参照してください。

6. 以下の情報を入力し、[次へ] をクリックします。

### グローバル キーの使用

デフォルトでは、選択したオプションが指定されます。上記の手順で指定したグローバルキーラベルを無効にして、新たに暗号化用のラベルを指定する場合は、このオプションを選択解除します。

### キーラベル

[グローバルキーの使用] オプションを選択解除した場合は、デフォルトの組織に対して使用する新しいキーラベルを指定します。

### 暗号化ストレージタイプ

暗号化キーがデータベース（ソフトウェア）に格納されるか HSM（ハードウェア）に格納されるかを示します。

7. [完了] をクリックします。
8. (オプション) [続行] をクリックして、管理コンソールを使用するほかの設定に進みます。

## CA Risk Authentication サーバサービスの開始

CA Risk Authentication サーバを起動するには、以下の手順に従います。

1. デスクトップウィンドウ上の [スタート] ボタンをクリックします。
2. [設定] - [コントロールパネル] - [管理ツール] - [サービス] にナビゲートします。
3. [CA Risk Authentication Service] を見つけてダブルクリックします。
4. サービスのウィンドウで、[開始] をクリックします。

## CA Risk Authentication ケース管理サービスの開始

CA Risk Authentication ケース管理サービスを開始するには、以下の手順に従います。

1. デスクトップウィンドウ上の [スタート] ボタンをクリックします。
2. [設定] - [コントロールパネル] - [管理ツール] - [サービス] に移動します。
3. [CA Risk Authentication Case Management Service] を見つけてダブルクリックします。
4. サービスのウィンドウで、[開始] をクリックします。



## CA Risk Authentication サーバのインストールの確認

サーバが正常に起動したことを確認するには、以下の手順に従います。

1. 以下の場所に移動します。  
`<install_location>%Arcot Systems%Logs%`
2. 任意のエディタで `arcotriskfortstartup.log` ファイルを開き、以下の行を見つけます。
  - STARTING CA Risk Authentication 8.0
  - CA Risk Authentication Service READY
3. 任意のエディタで `arcotriskfortcasemgmtserverstartup.log` ファイルを開き、以下の行を見つけます。
  - STARTING CA Risk Authentication Case Management Service 8.0
  - CA Risk Authentication Case Management Service READY

注: ログファイルに **FATAL** および **WARNING** のメッセージが含まれていないことを確認します。

## 第 8 章: ユーザ データ サービスの展開

CA RiskMinder は、リレーショナル データベース (RDBMS) から、または UDS を使用して LDAP サーバから直接ユーザ データにアクセスできます。UDS は、RiskMinder に対して、組織によって展開されたサードパーティ データ リポジトリへのシームレスなアクセスを提供する抽象化層です。

次の手順に従ってください：

1. アプリケーション サーバに `arcotuds.war` を展開します。このファイルは以下から入手できます。

```
<install_location>%Arcot Systems%java%webapps%
```

たとえば、Apache Tomcat の場合は、`<APP_SERVER_HOME>/webapps/` に `WAR` ファイルを展開する必要があります。

注: 展開手順は、使用しているアプリケーション サーバによって異なります。詳細な手順については、アプリケーション サーバベンダーのドキュメントを参照してください。

2. (WebSphere のみ) アプリケーションファイルが更新されると、UDS クラスを再ロードするように設定します。
  - a. [Applications] - [Application Types] - [WebSphere Enterprise Applications] に移動し、[UDS settings] 画面にアクセスします。
  - b. [Class loader order] で、[Classes loaded with local class loader first (parent last)] オプションを選択します。
  - c. [WAR class loader policy] で、[Single class loader for application] を選択します。
  - d. [Apply] をクリックして、変更を保存します。
3. アプリケーションサーバを再起動します。
4. UDS が正常に展開されたかどうかを確認する方法

注: UDS 関連情報のログ記録には `arcotuds.log` ファイルが使用されます。

- a. 以下の場所に移動します。  
`<install_location>%Arcot Systems%logs%`
- b. 任意のエディタで `arcotuds.log` ファイルを開き、以下の行を見つけます。
  - User Data Service (Version: 2.0.3) initialized successfully.この行は、UDS が正常に展開されたことを示しています。
- c. また、ログファイルに **FATAL** および **WARNING** のメッセージが含まれていないことも確認します。
- d. ファイルを閉じます。

## 第 9 章: ユーザ行動プロファイリング アプリケーションの展開

---

ユーザ行動プロファイリング (UBP) モデルは、データが不十分な場合に、同じユーザまたはそのピアグループのユーザによる以前のアクセスと現在のトランザクションとの類似点または相違点を測定します。CA Risk Authentication は UBP アプリケーションと通信して類似性スコアを取得し、それをリスク評価スコアに含めます。

UBP を展開するには、ca-userprofiling-2.0-application.war ファイルが必要です。

次の手順に従ってください：

1. アプリケーションサーバに ca-userprofiling-2.0-application.war を展開します。このファイルは以下の場所にあります。

```
<install_location>%Arcot Systems%java%webapps%
```

例：Apache Tomcat の場合は、<APP\_SERVER\_HOME>%webapps% に WAR ファイルを展開します。

注：展開手順は、使用しているアプリケーションサーバによって異なります。詳細については、アプリケーションサーバベンダーのドキュメントを参照してください。

2. (WebSphere の場合) アプリケーションファイルが更新されると、UDS クラスを再ロードするように設定します。

- a. [Application] - [Enterprise Applications] - [UDS Settings] に移動します。

- b. [Class loader order] で、[Classes loaded with local class loader first (parent last)] オプションを選択します。

- c. [WAR class loader policy] で、[Single class loader] を選択します。

- d. bcprov-jdk15-146 jar ファイルを

<ARCOT\_HOME>/sdk/java/lib/external から以下の場所にコピーします。

```
<JRE_HOME>/lib/ext フォルダ
```

注：ここで、JRE\_HOME は WebSphere アプリケーションサーバによって使用される jre インストールです。

- e. [適用] をクリックします。

(WebLogic の場合：サードパーティの JDBC ドライバを使用する方法については、WebLogic のドキュメントを参照してください)

3. アプリケーションサーバを再起動します。
4. UDS が正常に展開されていることを確認します。

注: UDS 関連情報のログ記録には `arcotuds.log` ファイルが使用されま  
す。

- a. 以下の場所に移動します。

`<install_location>%Arcot Systems%logs%`

- b. 任意のエディタで `ubp_logfile.log` ファイルを開き、以下のステート  
メントを見つけます。
- c. ログ ファイルに **FATAL** および **WARNING** のメッセージが含まれて  
いないことを確認します。
- d. ファイルを閉じます。

## 第 10 章: 2 つ目のシステムへの CA Risk Authentication のインストール

---

CA Risk Authentication サーバおよび管理コンソールをインストールした後  
に、その他の残りのコンポーネントを 2 つ目のシステムにインストールし  
ます。インストールするコンポーネントは、「[展開の計画](#)」の章で説明  
している計画の実行の際に決定されています。

次の手順に従ってください:

1. インストーラ ファイル `CA Risk Authentication-8.0-Windows-Installer.exe`  
を 2 つ目のシステムにコピーします。
2. インストーラをダブルクリックして実行します。
3. [Choose Install Set] 画面が表示されるまで、「1 つ目のシステムへの  
インストール」の手順 2 以降のインストーラ手順に従います。
4. コンポーネントを選択します。

注: 通常は、リスク評価とサンプル アプリケーション用の Java SDK を  
インストールします。

5. 「[1 つ目のシステムへのインストール \(P. 88\)](#)」の手順 7 から手順 13 の  
手順に従って、インストールを実行します。

## 2 つ目のシステムへのサンプル アプリケーションの展開

2 つ目のシステムにサンプル アプリケーションを展開する手順を実行します。これは、Java SDK および Web サービスをインストールしたインストール後のタスクです。

**重要:** サンプル アプリケーションを運用環境で使用しないでください。サンプル アプリケーションのコードを参考にして、独自の Web アプリケーションを作成することをお勧めします。

サンプル アプリケーションを使用して、CA Risk Authentication が正常にインストールおよび設定されていることを確認できます。また、サンプル アプリケーションは以下についての例を提供します。

- 一般的な CA Risk Authentication のワークフロー
- CA Risk Authentication API の基本操作（呼び出しと後処理）
- CA Risk Authentication とアプリケーションの統合

**注:** サンプル アプリケーションを製品インストール時にインストールしなかった場合は、インストーラを再度実行し、[SDKs and Sample Application] オプションを選択してインストールを続行すれば、サンプル アプリケーションのみをインストールできます。

次の手順に従ってください：

1. 以下の場所から CA Risk Authentication-8.0-sample-application.war ファイルを展開します。  
`<install_location>%Arcot Systems%samples%java%`
2. 必要に応じて、アプリケーション サーバを再起動します。
3. Web ブラウザ ウィンドウでサンプル アプリケーションにアクセスします。サンプル アプリケーションのデフォルト URL は次のとおりです。  
`http://<host>:<appserver_port>/ca-riskauth-8.0-sample-application/index.jsp`

## 第 11 章：CA Risk Authentication サーバと通信するためのサンプル アプリケーションの設定

CA Risk Authentication.risk-evaluation.properties ファイルには、CA Risk Authentication サーバ情報を読み取るための Java SDK とサンプルアプリケーションのパラメータが含まれています。サンプルアプリケーションの展開後、CA Risk Authentication サーバと通信できるようファイルを設定する必要があります。このファイルは、CA Risk Authentication サンプルアプリケーション WAR ファイル (CA Risk Authentication-8.0-sample-application.war) を展開した後でのみ利用できます。

次の手順に従ってください：

1. アプリケーションサーバの CA Risk Authentication.risk-evaluation.properties ファイルに移動します。  
Apache Tomcat の場合、このファイルは以下の場所にあります。  
<App\_Home¥CA Risk Authentication-8.0-sample-application>¥WEB-INF¥classes¥properties¥  
  
<App\_Home¥CA Risk Authentication-8.0-sample-application¥>  
CA Risk Authentication アプリケーション WAR ファイルを展開したディレクトリパスを示します。
2. エディタ ウィンドウで riskfort.risk-evaluation.properties ファイルを開き、以下のパラメータの値を設定します。
  - HOST.1
  - PORT.1ファイル内の残りのパラメータには、デフォルト値が指定されています。必要に応じて、これらの値を変更できます。
3. (オプション)「SSL の設定」で SSL ベースの通信を設定した場合のみ、この手順を実行してください。  
以下のパラメータを設定します。
  - TRANSPORT\_TYPE=SSL (デフォルトで、このパラメータは TCP に設定されます)。
  - CA\_CERT\_FILE=<PEM 形式のルート証明書の絶対パス>  
たとえば、以下のいずれかの値を指定します。
    - CA\_CERT\_FILE=<install\_location>/certs/<ca\_cert>.pem
    - CA\_CERT\_FILE=<install\_location>¥¥certs¥¥<ca\_cert>.pem

**重要:** 絶対パスを指定する際、必ず ¥ の代わりに ¥¥ または / を使用してください。これは、Microsoft Windows でパスの指定に使用される従来の ¥ を使用すると変更が機能しない場合があるからです。

4. 変更を保存して、ファイルを閉じます。
5. これらの変更を確実に反映するため、アプリケーション サーバを再起動します。

## 第 12 章: サンプル アプリケーションをリスク評価操作に使用

---

このセクションでは、サンプル アプリケーションを使用して実行できるリスク評価操作について説明します。サンプル アプリケーションでの各操作は、CA Risk Authentication がインストールされ、機能していれば、エラーなく実行されるように設計されています。

サンプル アプリケーションでは、CA Risk Authentication サーバが実行できる以下の操作について、その例を示します。

- [初めてのユーザのリスク評価および後評価の実行](#) (P. 120)
- [ユーザの作成](#) (P. 122)
- [既知のユーザのリスク評価および後評価の実行](#) (P. 123)
- [デフォルト プロファイルの編集およびリスク評価の実行](#) (P. 124)

## 初めてのユーザのリスク評価および後評価の実行

ユーザのデフォルト プロファイルでリスク評価を実行するには、以下の手順に従います。

1. サンプルアプリケーションが Web ブラウザ ウィンドウで開いていることを確認します。以下の URL がサンプルアプリケーションのデフォルトの URL です。

`http://<host>:<appserver_port>/ca-riskauth-8.0-sample-application/index.jsp`

2. [Evaluate Risk] をクリックします。
3. 以下の情報を入力します。

### User Name

評価対象のユーザの名前を指定します。

### User Organization

ユーザが所属する組織を指定します。

### チャネル

トランザクションが発生したチャネルを指定します。これはオプションフィールドです。

4. [Evaluate Risk] をクリックします。

**注:** [Evaluate Risk Result] ページには、リスク スコアおよび関連付けられているリスク アドバイスが表示され、指定した組織用に設定されたルールがリスト表示されます。初めてのユーザの場合、結果は **ALERT** になります。

5. [Next Step] をクリックして、指定したユーザ プロファイルの後評価を実行します。

アプリケーションは後評価を通じて、現在のユーザやユーザが使用しているデバイスに関するフィードバックを CA Risk Authentication サーバに提供します。CA Risk Authentication では、このフィードバックに基づいて、ユーザ属性やデバイス属性、ユーザとデバイスの関連付けを更新し、その後ユーザのトランザクションに伴うリスクを評価します。

6. [Result of Secondary Authentication] リストからセカンダリ認証の結果を選択します。
7. ユーザ名とデバイスの関連付けの名前を [Association Name] に入力します。



8. [Post Evaluate] をクリックし、後評価プロセスを完了すると、[Post Evaluation Results] セクションに同じ後評価プロセスの結果が表示されます。

## ユーザアカウントの作成

ユーザを作成するには、以下の手順に従います。

1. 以下の手順に従うことにより、GA アカウントを作成します。
  - a. MA として管理コンソールにログインします。
  - b. [ユーザと管理者] タブがアクティブであることを確認します。
  - c. 左側のメニューから、[管理者の作成] リンクをクリックします。
  - d. 詳細を入力し、[次へ] をクリックします。
  - e. [ロール] リストから [グローバル管理者] を選択します。
  - f. [パスワード] と [パスワードの確認] に入力します。
  - g. [管理する] セクションで [全組織] オプションを選択します。
  - h. [作成] をクリックします。
  - i. ページの右上隅からの [ログアウト] をクリックします。
2. GA (グローバル管理者) または OA (組織管理者) として管理コンソールにログインします。以下の URL が管理コンソールページの URL です。  
`http://<host>:<appserver_port>/arcotadmin/adminlogin.htm`
3. パスワードを変更するために表示される手順に従います。
4. [ユーザと管理者] タブの [ユーザと管理者の管理] をアクティブにします。
5. 左ペイン ( [ユーザと管理者の管理] ) から、[ユーザの作成] をクリックします。
6. [ユーザの作成] ページに以下の詳細を入力します。
  - a. [ユーザ詳細] セクションに一意のユーザ名、それらの組織名、および必要に応じてその他のユーザ情報を入力します。
  - b. (オプション) ページ上の対応するフィールドでその他のユーザ情報を入力します。
  - c. ユーザステータスを選択します。
  - d. [ユーザの作成] をクリックします。
7. CA Risk Authentication サンプルアプリケーション ページに戻ります。

## 既知のユーザのリスク評価および後評価の実行

既知のユーザのリスク評価および後評価を実行するには、以下の手順に従います。

1. サンプルアプリケーションのメイン ページで [Evaluate Risk] をクリックします。
2. 以下の詳細を入力します。

### User Name

「ユーザの作成」で作成したユーザの名前を指定します。

### User Organization

ユーザが所属する組織を指定します。

### チャンネル

トランザクションが発生したチャンネルを指定します。これはオプションフィールドです。

3. [Evaluate Risk] をクリックします。  
リスク アドバイスは通常 INCREASEAUTH です。
4. [Store DeviceID] をクリックして、エンドユーザのデバイスにデバイス ID 情報の指定されたタイプを保存します。
5. [Next Step] をクリックして、後評価を実行します。
  - リストから [Result of Secondary Authentication] を選択します。
  - 必要に応じて [Association Name] を編集します。
6. [Post Evaluate] をクリックして、最終的なアドバイスを表示します。

注: 手順 1 ~ 手順 5 を繰り返せば、[Risk Evaluation Results] ページのリスク アドバイスは ALLOW に変わります。

## デフォルト プロファイルの編集およびリスク評価の実行

サンプルアプリケーションを使用して、使用しているコンピュータの DeviceDNA、IP アドレス、およびデバイス ID を変更して、さまざまな状況をシミュレートします。

次の手順に従ってください：

1. サンプルアプリケーションのメイン ページで [Evaluate Risk] をクリックします。
2. 以下の情報を入力します。

### User Name

「ユーザの作成」で作成したユーザの名前を指定します。

### User Organization

ユーザが所属する組織を指定します。

### チャンネル

トランザクションが発生したチャンネルを指定します。これはオプションフィールドです。

3. [Edit Input] をクリックします。
4. 1 つ以上の必要なフィールドの値を変更します。
  - 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
5. [Evaluate Risk] をクリックします。
6. [Next Step] をクリックして、指定したユーザ プロファイルの後評価を実行します。
7. [Result of Secondary Authentication] リストからセカンダリ認証オプションの結果を選択します。
8. [Post Evaluate] をクリックし、後評価プロセスを完了すると、同じ後評価プロセスの結果が表示されます。

注: CA Risk Authentication コンポーネント間の安全な通信を確保するために、SSL (Secure Socket Layer) トランスポート モードをサポートするよう設定できます。詳細については、「CA Risk Authentication 8.0 管理ガイド」の「SSL の設定」を参照してください。

## 第 13 章: インストール後のチェックリストの適用

CA Risk Authentication のインストールおよびセットアップ情報を使用して以下のチェックリストに記入します。各種管理タスクを実行する際に、これらの情報が役立ちます。

| 情報                     | 入力例  | 記入欄 |
|------------------------|--|-----|
| ARCOT_HOME             | C:¥Program Files¥Arcot Systems                       |     |
| <b>システム情報</b>          |  |     |
| ホスト名                   | my-bank  |     |
| User Name              | 管理者  |     |
| Password               | password1234!  |     |
| 設定済みのコンポーネント           | CA Risk Authentication サーバ<br>管理コンソール<br>ユーザデータ サービス |     |
| <b>管理コンソール情報</b>       |  |     |
| ホスト名                   | localhost  |     |
| ポート                    | 8080   |     |
| マスタ管理者パスワード            | mypassword1234!                                      |     |
| <b>ユーザデータ サービス情報</b>   |  |     |
| ホスト名                   | localhost  |     |
| ポート                    | 8080   |     |
| アプリケーション コンテキスト<br>ルート | arcotuds   |     |

## 第 14 章: サイレント モード インストール

---

CA Risk Authentication をインストールした後に、サイレント モードのインストールを使用して、コンポーネントを再度インストールできます。サイレントインストールでは、ユーザによる操作なしでインストールが完了します。

### サイレント モード インストールのガイドライン

サイレント インストールを開始する前に、以下のガイドラインを確認します。

- デフォルト プロパティ ファイルを変更する前に、バックアップします。
- パラメータ名、等号 (=) およびパラメータの値の間に、決して余分なスペースを追加しないでください。
- 変更後に、ファイルを保存します。

**重要:** サイレント インストールで使用される応答ファイルを生成するために、「-r」オプションを使用してインストーラの実行可能ファイルを実行しないでください。最初のインストール時に作成されるデフォルト プロパティ ファイルのみを使用する必要があります。

## デフォルトプロパティファイル

デフォルトプロパティファイル内のパラメータを変更するには、テキストエディタを使用します。デフォルトパラメータは、最初のインストール中に入力された情報を反映します。デフォルトプロパティファイルには、機密情報と関連付けられているパラメータがあります。たとえば、データベースパスワード、マスタキー、およびHSMのPINに関連するパラメータなどです。それらに適切な値を指定します。

### CA Risk Authentication プロパティファイル

CA Risk Authentication プロパティファイルのデフォルトの名前および場所は以下のとおりです。

#### Name

installer.properties

#### 場所

risk\_auth\_home¥

risk\_auth\_home

CA Risk Authentication のインストールパスを指定します。

### CA Risk Authentication インストーラのプロパティファイルの変更

インストール変数を定義するには、CA Risk Authentication インストーラのプロパティファイルを変更します。

以下のデフォルトパラメータでは、CA Risk Authentication の最初のインストール時にユーザが入力した情報が指定されています。

#### CHOSEN\_FEATURE\_LIST

インストールされる機能のカンマ区切りリストを指定します。

有効な値は以下のとおりです。

#### RFSRV - CA Risk Authentication サーバ

認証、プロビジョニング、設定およびサーバインスタンスの管理、を行うサーバ

#### RFCASE - ケース管理キュー サーバ

送信呼び出し元に対して動作するケースを処理します。管理コンソールのすべてのインスタンスは、常にケース管理キューサーバの1つの共通のインスタンスだけに接続します。

#### RFSDK - CA Risk Authentication の Java SDK および WS

CA Risk Authentication サーバへの発行、認証、および設定のリクエストを可能にする Java SDK および Web サービス。

#### ADMIN - 管理コンソール

サーバ設定を管理するための Web ベースのコンソール。

#### UDS - ユーザ データ サービス

リレーショナルデータベース (RDBMS) やディレクトリサーバ (LDAP) などの、さまざまなタイプのユーザリポジトリにアクセスするための抽象化層。

#### UBP - ユーザ行動プロファイリング

ユーザの行動を予測してセキュリティを向上するモデル。

#### USER\_INSTALL\_DIR\_SILENT

CA Risk Authentication のインストール場所を指定します

#### ARCOT\_DBTYPE\_SILENT

設定されているデータベースのタイプを指定します。



有効な値 : oracle、mssqlserver、mysql

## プライマリ データベースの詳細

プライマリ データベースには、以下のデータベース関連の詳細があります。

### ARCOT\_CONFIG\_PRIMARY\_DB\_SILENT

プライマリ データベースが設定されているかどうかを指定します。

有効な値 : true、false

### ARCOT\_PRIMARY\_DSN\_NAME\_SILENT=

データベースのデータ ソース名を指定します。

### ARCOT\_PRIMARY\_DATABASE\_SILENT

データベース インスタンスの名前を指定します。

### ARCOT\_PRIMARY\_SID\_SILENT

Oracle データベースの SID を指定します。その他のデータベース タイプでは、空白のままにします。

### ARCOT\_PRIMARY\_TNS\_SERVICE\_NAME\_SILENT

Oracle データベースの TNS サービス名を指定します。その他のデータベース タイプでは、空白のままにします。

### ARCOT\_PRIMARY\_HOST\_NAME\_SILENT

データベース サーバのホスト名を指定します。

### ARCOT\_PRIMARY\_PORT\_SILENT

指定したデータベース インスタンスのポート番号を指定します。

### ARCOT\_PRIMARY\_USER\_NAME\_SILENT

データベース ユーザ名を指定します。

### ARCOT\_PRIMARY\_PASSWORD\_SILENT

指定したデータベース ユーザ名のパスワードを指定します。

### ARCOT\_CONFIG\_BACKUP\_DB\_SILENT

バックアップ データベースが設定されているかどうかを指定します。

有効な値 : true、false

## バックアップ データベースの詳細

バックアップ データベースには、以下のデータベース関連の詳細があります。

### ARCOT\_BACKUP\_DSN\_NAME\_SILENT

データベースのデータ ソース名を指定します。

### ARCOT\_BACKUP\_DATABASE\_SILENT

データベース インスタンスの名前を指定します。

### ARCOT\_BACKUP\_SID\_SILENT

Oracle データベースの SID を指定します。その他のデータベース タイプでは、空白のままにします。

### ARCOT\_BACKUP\_TNS\_SERVICE\_NAME\_SILENT

Oracle データベースの TNS サービス名を指定します。その他のデータベース タイプでは、空白のままにします。

### ARCOT\_BACKUP\_HOST\_NAME\_SILENT

データベース サーバのホスト名を指定します。

### ARCOT\_BACKUP\_PORT\_SILENT

指定したデータベース インスタンスのポート番号を指定します。

### ARCOT\_BACKUP\_USER\_NAME\_SILENT

データベース ユーザ名を指定します。

### ARCOT\_BACKUP\_PASSWORD\_SILENT

指定したデータベース ユーザ名のパスワードを指定します。

## 暗号化の詳細

データベースの暗号化の詳細を以下に示します。

暗号化方式：ソフトウェア/ハードウェア

ARCOT\_ENC\_TYPE\_SILENT

暗号化の方式を指定します。

有効な値：software、nfast、chrysalis

ARCOT\_ENC\_DEVICE\_NAME\_SILENT

ハードウェア暗号化用のデバイス名を指定します。

ARCOT\_KEY\_LABEL\_SILENT

マスタキーラベルを指定します。

ARCOT\_HSM\_PIN\_SILENT

HSMのピン番号を指定します。

ARCOT\_HSM\_SHARED\_LIBRARY\_SILENT

HSM共有ライブラリの完全パスを指定します。

ARCOT\_HSM\_STORAGE\_SLOT\_SILENT

HSMの「Storage Slot Number」を指定します。

## サイレント インストールを実行する方法

無人インストールを実行し、ユーザによる操作なしで CA Risk Authentication をインストールします。

次の手順に従ってください：

1. 無人インストールのガイドラインを確認します。
2. CA Risk Authentication ホストシステムから CA Risk Authentication プロパティ ファイルをコピーします。
3. CA Risk Authentication のインストール メディアをプロパティ ファイルと同じ場所にコピーします。
4. CA Risk Authentication インストーラのプロパティ ファイルを変更します。
5. CA Risk Authentication インストーラを実行します。

CA Risk Authentication のインストール実行可能ファイルおよびプロパティ ファイルをコピーしたディレクトリで、以下のコマンドを実行します。

```
installation_media -f installer.properties -i silent
```

### Installation\_media

CA Risk Authentication のインストール実行可能ファイルを指定します。

**注：**プロパティ ファイルがインストールメディアと同じディレクトリ内に存在しない場合は、その場所を指定します。引数にスペースが含まれている場合は、二重引用符を使用します。

### -i silent

インストーラがサイレントで実行されるように指定します。

**例：**

```
installation_media -f "C:¥Program Files¥CA¥Arcot Systems  
¥installer.properties" -i silent
```

インストールが始まります。インストーラは、ユーザがプロパティ ファイルで指定したパラメータを使用して CA Risk Authentication をインストールします。

6. CA Risk Authentication のインストールを確認します。

# 第 15 章: ユーザ行動プロファイリング モデルを展開する方法

---

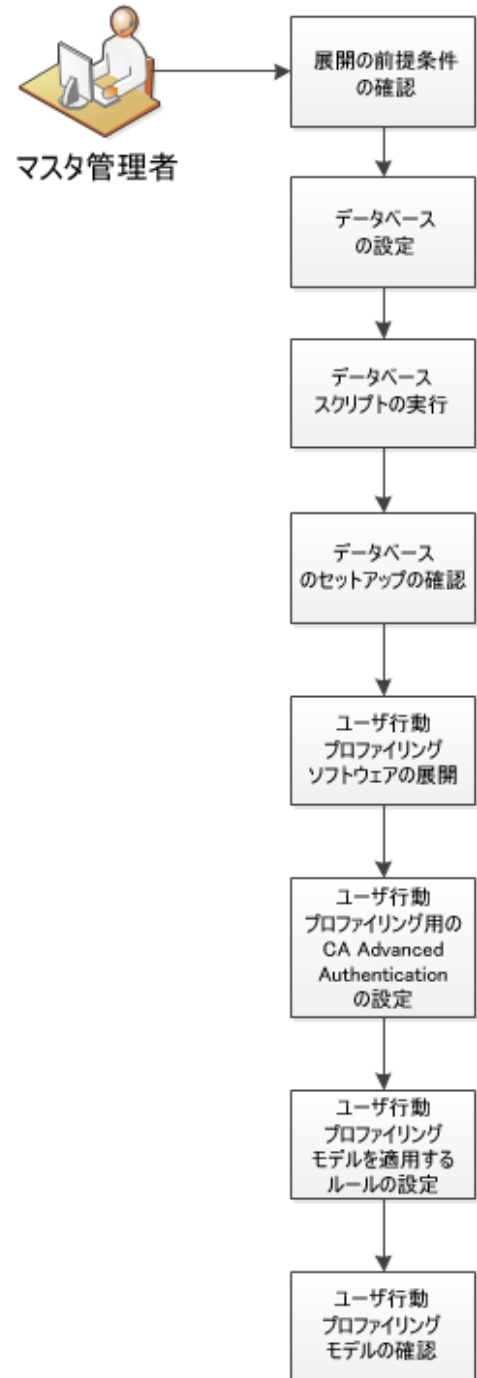
このセクションでは、マスタ管理者がユーザ行動プロファイリングをインストール、設定、展開する方法について説明します。

CA Risk Authentication は、より強力な認証が必要なケースを検出し、現在のトランザクションのパラメータを顧客の事前設定ルールに照らして評価します。呼び出し元のソフトウェアは、評価によって提供されるリスクスコアを使用して、ユーザに続行を許可する前に追加の認証が必要かどうかを判断します。

ユーザ行動プロファイリングは、データが不十分な場合に、同じユーザまたはそのピアグループのユーザによる以前のアクセスと現在のトランザクションとの類似点または相違点を測定します。

CA Advanced Authentication へのユーザ行動プロファイリングのインストールおよび展開を以下の図に示します。

## ユーザ行動プロファイリング を展開する方法



ユーザ行動プロファイリングをインストールするおよび展開するには、以下の手順に従います。

1. [展開の前提条件の確認](#) (P. 139)
2. [データベースの設定](#) (P. 139)
3. [データベース スクリプトの実行](#) (P. 144)
4. [データベースのセットアップの確認](#) (P. 145)
5. ユーザ行動プロファイリング ソフトウェアの展開
6. [ユーザ行動プロファイリング用の CA Advanced Authentication の設定](#) (P. 148)
7. [ユーザ行動プロファイリング モデルを適用するルールの設定](#) (P. 149)
8. [ユーザ行動プロファイリング モデルの確認](#) (P. 150)



## 前提条件の確認

ユーザ行動プロファイリングを設定する前に、以下の前提条件を確認します。

### ハードウェアおよびソフトウェアの要件

ユーザ行動プロファイリングの実装には、4つのサーバが必要です。ハードウェアおよびソフトウェア要件は以下のとおりです。

#### CA Advanced Authentication ユーザ行動プロファイリング サーバ

- CPU – 2.0 GHz AMD Opteron 6128 × 2
- メモリ – 4 GB
- HDD1 – 40 GB
- Microsoft Windows 2008 R2 SP1

#### CA CA Strong Authentication/CA CA Risk Authentication サーバ

- CPU – 2.0 GHz AMD Opteron 6128 × 2
- メモリ – 4 GB
- HDD1 – 40 GB
- Microsoft Windows 2008 R2 SP1

#### Active Directory サーバ

- CPU – 2.0 GHz AMD Opteron 6128 × 2
- メモリ – 3 GB
- HDD1 – 40 GB
- Microsoft Windows 2008 R2 SP1
  - Active Directory ドメイン サービス
  - DNS

#### データベース サーバ

- CPU - 2.0 GHz AMD Opteron 6128 × 4
- メモリ – 6 GB
- HDD1 – 32 GB (OS)
- HDD2 – 16 GB (スワップ)

- HDD3 – 40 GB
- Microsoft SQL Server 2008 R2 64 ビット版

### Third-Party のコンポーネント

#### Java SDK

CA Advanced Authentication ユーザ行動プロファイリングで使用する Java の現在のバージョンが 1.6 以上であることを確認します。

バージョン 1.7 へのアップグレードまたは Java コンポーネントをインストールするには、Oracle の Java ダウンロード サイトで提供されている手順に従ってください。

#### 環境変数

Java の場所でアプリケーションを実行するには、環境コンポーネントを設定します。

次の手順に従ってください：

1. コンピュータのシステム プロパティに移動します。
2. [詳細設定] タブで [環境設定] をクリックし、システム変数を設定します。
3. 新しいシステム変数を設定するには、JAVA\_HOME システム変数の値を ¥Program Files (x86)¥Java¥jdk1.7.0\_51 に設定します。

注: 別の JDK リリースをダウンロードした場合は、フォルダ名 (この例では、jdk1.7.0\_51) は異なります。変数の値が正しいフォルダを表していることを確認します。

4. システム変数を編集するには、パスの末尾に「¥%JAVA\_HOME%¥jre¥bin」を追加して PATH システム変数の値を更新します。
5. コマンドプロンプトウィンドウで「Java」と入力し、Enter キーを押します。
6. Java が PATH システム変数に正しく追加されている場合、java に関する使用情報が表示されます。

注: システム変数を更新するには、サーバの再起動が必要です。

## データベースの設定

インストールする前に、ユーザ情報、サーバ設定データ、監査ログデータ、およびその他の情報を格納するためのデータベースを設定します。

**CA Risk Authentication** では、プライマリ データベースと、高可用性展開でのフェールオーバー時とフェールバック時に使用できるバックアップデータベースを使用できます。以下の方法でデータベース接続を設定します。

データベースは、**CA Risk Authentication** のインストール時に、ユーザが入力したデータベース情報を使用してインストーラが `arcotcommon.ini` ファイルを編集するときに自動的に設定されます。

サポートされるデータベース (Microsoft SQL Server、Oracle または MySQL) ごとに、特定の設定要件があります。

**重要:** データベース サーバを保護するには、ファイアウォールまたはその他のアクセス制御メカニズムを使用し、すべての依存製品と同じタイムゾーンに設定します。

## Microsoft SQL Server の設定

このセクションでは、SQL Server 用の以下の設定手順を示します。

**注:** このセクションに示すタスクの実行の詳細については、SQL Server のドキュメントを参照してください。

次の手順に従ってください：

1. SQL Server が *SQL Server 認証モード* と *Windows 認証モード* をサーバ認証に使用するように設定されていることを確認します。 [オブジェクトエクスプローラ] ウィンドウ内のサーバを右クリックし、 [セキュリティ] ページを選択します。

SQL Server が「*Windows 認証モード*」のみに設定されている場合、CA Risk Authentication はデータベースに接続できません。

2. 以下の条件でデータベースを作成します。
  - 推奨される名前は `arcotdb` です。
  - データベース サイズは自動的に拡大するように設定する必要があります。
3. 以下の手順に従って、DB ユーザ (`CH4_SQL`) を作成します。
  - a. SQL Server Management Studio で、`<SQL_Server_Name>` に移動し、 [セキュリティ] フォルダを展開して、 [ログイン] をクリックします。

**注:** `<SQL_Server_Name>` は、データベースを作成した SQL Server のホスト名または IP アドレスを指します。
  - b. [ログイン] フォルダを右クリックし、 [新しいログイン] をクリックします。
  - c. ログイン名を入力します (推奨される名前は `arcotuser`) 。
  - d. パラメータを *SQL Server 認証* に対する *認証* に設定します。
  - e. ログインの [パスワード] および [パスワードの確認入力] を指定します。
  - f. 組織のパスワード ポリシーに従い、このページのその他のパスワード設定を指定してください。
  - g. 作成したデータベース (`arcotd`) をデフォルトデータベースに設定します。
  - h. このログインセクションへのユーザのマッピングを実行します。

- i. デフォルト データベースのユーザ (SQL 2005) を `db_owner` にマップします ( [ <db\_name> のデータベース ロール メンバシップ ] セクション ) 。

## Oracle サーバの設定

このセクションでは、Oracle データベース サーバを作成するための設定情報を示します。

### 前提条件

1. 2つのテーブルスペースを持った Oracle 上で CA Risk Authentication を実行します。2つのテーブルスペースが必要な理由を以下に示します。
  - 1つ目のテーブルスペースは、設定データ、監査ログ、およびユーザ情報の格納に使用されます。このテーブルスペースは、CA Risk Authentication データベース内でデフォルトのユーザ テーブルスペースにすることができます。
  - 2つ目のテーブルスペースでレポートを実行します。レポートを実行するために個別のテーブルスペースを使用することをお勧めします。
2. CA Risk Authentication データベース設定スクリプトを使用します。このスクリプトは、このスクリプトを実行するデータベース ユーザがテーブルスペースを作成するための十分な権限を持っている場合、レポートのテーブルスペースを自動的に作成します。必要な権限がユーザにない場合、データベース管理者はこのテーブルスペースを手動で作成し、レポートを作成するセクションをスクリプトから削除する必要があります。

`arcot-db-config-for-common-8.0.sql`

**重要:** レポートのテーブルスペースを作成するための `arcot-db-config-for-common-8.0.sql` データベース スクリプト内のパラメータは、データベース管理者の希望に応じて変更できます。ただし、レポートを正常に生成するには、テーブルスペース名を *ARReports* にする必要があります。

次の手順に従ってください:

1. UTF-8 文字セットで情報を格納する新しいデータベースを作成します。この文字セットにより、CA Risk Authentication でダブルバイト言語を含む国際的な文字を使用できるようになります。Oracle データベースの UTF-8 サポートを有効にするには、以下の手順に従います。
  - a. SYS または SYSTEM として Oracle データベース サーバにログインします。
  - b. 以下のコマンドを実行します。

```
sys.props$ set value$='UTF8'
```

(where name='NLS\_NCHAR\_CHARACTERSET' Or name = 'NLS\_CHARACTERSET')
  - c. データベースを再起動し、文字セットが UTF-8 に設定されているかどうかを確認します。
2. データベース ユーザを作成します。
  - a. 新しいデータベース `arcotdb` のスキーマを使用して、ユーザを作成します (推奨される名前は `arcotuser`) 。
  - b. 開発またはテスト用の展開では、ユーザのクォータを少なくとも 5 ~ 10 GB に設定します。

注: 運用環境、ステージング、またはその他の負荷の高いテスト用の展開の場合、ユーザに必要なクォータを決定する方法については、「データベース リファレンス」を参照してください。
  - c. ユーザに DBA ロールを付与します。

## MySQL サーバの設定

このセクションでは、MySQL 用の以下の設定情報を示します。

次の手順に従ってください：

1. InnoDB ストレージエンジンが MySQL のインストールでサポートされているかどうかを確認するには、SHOW ENGINES コマンドを使用します。

**注：** CA Risk Authentication は、MySQL の InnoDB ストレージエンジンを使用します。このコマンドの出力に InnoDB がサポートされていないことが示されている場合は、InnoDB のサポートを有効にします。InnoDB のサポートを有効にする方法については、MySQL のドキュメントを参照してください。

2. Windows 以外のプラットフォームで MySQL を実行している場合は、lower\_case\_table\_names 変数を 1 に設定します。

**注：** 詳細については、MySQL のドキュメントを参照してください。

3. データベースを作成するには、以下の手順に従います。
  - a. MySQL コマンドウィンドウを開きます。
  - b. データベース スキーマを作成するには、以下のコマンドを実行します。

```
CREATE SCHEMA '<schema-name>' DEFAULT CHARACTER SET utf8;
```
  - c. データベース ユーザを作成するには、以下のコマンドを実行します。
4. 以下の条件に従ってユーザを作成します。
  - a. 新しいデータベース arcotdb にユーザを作成します（推奨される名前は arcotuser）。

```
CREATE USER '<user-name>' identified by '<user-password>';
```

- b. ユーザに以下の権限を付与します。

- オブジェクト権限

- SELECT
- INSERT
- UPDATE
- DELETE
- EXECUTE

- DDL 権限
  - CREATE
  - ALTER
  - CREATE ROUTINE
  - ALTER ROUTINE
  - DROP
- その他の権限
  - GRANT OPTION

## データベース スクリプトの実行

データベース テーブルを作成するには、CA Risk Authentication に付属している必要なデータベース スクリプトを実行します。

**重要:** スクリプトを実行する前に、「データベース サーバの設定」セクションで作成したときと同じデータベース ユーザとしてログインしていることを確認してください。

次の手順に従ってください:

1. 以下のディレクトリに移動します。  
`<install_location>%Arcot Systems%dbscripts%`
2. 使用しているデータベースに基づいて以下のいずれかのサブディレクトリに移動します。
  - Oracle の場合 : Oracle%
  - Microsoft SQL の場合 : mssql%
  - MySQL の場合 : mysql%
3. スクリプトを実行します。  
`arcot-db-config-for-userprofiling-2.0.sql`



## データベースのセットアップの確認

必要なデータベース スクリプトを実行した後、CA Risk Authentication スキーマを確認します。

次の手順に従ってください:

1. SYSDBA 権限を持つユーザで CA Risk Authentication データベースにログインします。

2. 以下のクエリを実行します。  
`SELECT * from dbo.XUBPData`

上記のクエリの結果、以下の出力が表示されます。

```
USERNAME ORGNAME PARAMNAME DATA  
-----
```

3. データベース コンソールからログアウトします。

## ユーザ行動プロファイリング ソフトウェアの展開

データベース上で実行するためにユーザ行動プロファイリング ソフトウェアを展開します。ユーザ行動プロファイリングは、単一システムまたは分散システムに展開できます。

次の手順に従ってください：

1. CA Risk Authentication サービスを停止します。
2. アプリケーション サーバを停止します。

注：分散システムにインストールする場合は、インストール時に [Custom] オプションを選択し、[User Behavior Profiling] を選択します。

3. 管理コンソールを展開します。

管理コンソールは、サーバ設定のカスタマイズや展開したシステムの管理を実行できるブラウザ ベースのインターフェースです。

アプリケーション サーバに管理コンソールの WAR ファイルを展開し、正常に展開されたことを確認するには、以下の手順に従います。

1. アプリケーション サーバの適切なディレクトリに `arcotadmin.war` を展開します。

注：展開手順は、使用しているアプリケーション サーバによって異なります。詳細な手順については、アプリケーション サーバ ベンダーのドキュメントを参照してください。

例：Apache Tomcat の場合は、`<APP_SERVER_HOME>%java%webapps%` に WAR ファイルを展開する必要があります。

2. (32 ビットの *WebSphere* の場合のみ)アプリケーションファイルが更新されると、Admin クラスを再ロードするように設定します。以下の手順に従います。
  - a. [Application] - [Enterprise Applications] に移動し、[Admin settings] ページにアクセスします。
  - b. [Class loader order] で、[Classes loaded with local class loader first (parent last)] オプションを選択します。
  - c. [WAR class loader policy] で、[Single class loader for application] を選択します。
  - d. [適用] をクリックします。

- e. Admin アプリケーションを再起動します。
3. 以下のいずれかの環境から、アプリケーション サーバに管理コンソールを展開します。  
`http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-jdk/2.1.1.GA/`  
`http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-spi/2.1.1.GA/`  
`http://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-log4j/2.1.1.GA/`
4. アプリケーション サーバを再起動します。
5. コンソールが正常に展開されていることを確認するには、以下の手順に従います。
  - a. 以下の場所に移動します。  
`<install_location>%Arcot Systems%logs%`
  - b. 任意のエディタで `arcotadmin.log` ファイルを開き、以下の行を見つけます。
    - 2.0.3
    - CA Advanced Authentication Configured Successfully.

注: これらの行は、管理コンソールが正常に展開されていることを示しています。
  - c. また、ログ ファイルに **FATAL** および **WARNING** のメッセージが含まれていないことも確認します。
  - d. ファイルを閉じます。

## ユーザ行動プロファイリング モデル用の CA Advanced Authentication の設定

ユーザ行動プロファイリング モデルを使用するには、CA Advanced Authentication を設定します。

次の手順に従ってください：

1. マスタ管理者として CA Advanced Authentication UI にログインします。
2. [サービスおよびサーバの設定] タブをクリックします。
3. [モデル設定] をクリックします。
4. 予測モデル URL (プライマリ :  
http://<appserver\_hostname>:<appserver\_port>/ca-userprofiling-2.0-application/UBPServlet) をユーザ行動プロファイリングを実行するプライマリ サーバに更新します。
5. 予測モデル URL (バックアップ) をユーザ行動プロファイリングを実行するバックアップ サーバに更新します (存在する場合)。  
**注:** ユーザ行動プロファイリングのバックアップ インスタンスを実行しない場合は、両方の URL を同じに設定します。
6. [モデル設定のアップロード] をクリックします。
7. 組織レベルでユーザ行動プロファイリングを有効にするには、マスタ管理者として Administrative UI からログアウトし、グローバル管理者としてログインします。
8. [組織] タブに移動します。
9. [検索] をクリックし、ユーザ行動プロファイリングを実装する組織を選択します。
10. [CA Risk Authentication] タブを選択し、[モデル設定] を選択します。
11. この組織に対して定義されているルールセットを選択し、[モデルの有効化] オプションを選択します。
12. [保存] をクリックします。
13. 運用環境にこれらの変更を移動し、サーバキャッシュをリフレッシュします。

これで、ユーザ行動プロファイリングを組織で有効にするための変更は完了です。

## 新しいユーザ行動プロファイリング モデルを適用するルールの設定

ルールを設定して、各トランザクションがユーザ行動プロファイリングモデルによって調査されたことを確認します。ルールが定義されていない場合、トランザクションはユーザ行動プロファイリングモデルを通過しますが、応答は表示されません。新しいユーザ行動モデルを適用するルールを設定します。

次の手順に従ってください：

1. グローバル管理者または組織管理者として、CA Advanced Authentication UI にログインします。
2. [組織] タブをクリックします。
3. [検索] をクリックします。
4. ユーザ行動プロファイリング モデルを実装した組織をクリックします。
5. [リスク ベース認証] タブに移動し、[ルールおよびスコアリング管理] を選択します。
6. 組織のルールセットを選択します。
7. [新しいルールの追加] をクリックします。
8. ルールに名前を付け、ニックネーム（ルールの短縮名）および説明を入力します。
9. MODEL\_SCORE データ エレメントを選択し、必要な演算子をクリックします。

注：モデル スコアによってルールをトリガしてトランザクションをセカンダリ認証に移動する場合は、値を設定します。モデル スコアには、定義された値に対して GREATER\_THAN、LESS\_THAN、GREATER\_OR\_EQUAL、LESS\_OR\_EQUAL、EQUAL\_TO、NOT\_EQUAL\_TO、IN\_LIST、IN\_CATEGORY のいずれかの演算子を使用できます。設定されたルールは、選択した演算子に基づいてトリガされます。

10. [作成中のルール] フィールドに入力するために [追加] をクリックします。
11. ルールが入力された後、[作成] をクリックします。

新しいルールが正常に作成されたことを示すメッセージが表示されます。

12. 新しいルール横にある「有効化」をクリックして、リスクスコアの値を設定します。
13. このルールセット内のその他のルールに対してこの新しいルールの優先順位を付けるには、適切な優先順位を設定します。
14. 「保存」をクリックします。
15. 「実稼働にマイグレート」メニューをクリックし、組織に適切なルールセットを選択して「マイグレート」をクリックします。
16. キャッシュをリフレッシュします。

これで、新しく追加したユーザ行動プロファイリングモデルを組織に導入するためルールがサブミットされます。

CA Advanced Authentication へのユーザ行動プロファイリングの実装はこれで完了です。

## ユーザ行動プロファイリング モデルの確認

この手順では、ユーザ行動プロファイリングモデルが正常に機能していることを確認します。

次の手順に従ってください:

1. グローバル管理者としてログインします。
2. 「レポート」に移動し、「トランザクションの分析レポート」をクリックします。
3. 条件を入力し、「サブミット」をクリックします。
4. 「モデルスコア」属性にスコアが表示されます。

このユーザに対してより多くのデータを生成すると、モデルスコアはそれに応じて調整されます。モデルスコアの増加は、ユーザ行動プロファイリングモデルが正常に動作していることを示します。

## ユーザ行動プロファイリング モデルの削除

ユーザ行動プロファイリングを削除する場合は、要件に応じてモデルを無効化またはアンインストールします。

## ユーザ行動プロファイリング モデルの無効化

環境からユーザ行動プロファイリング モデルを削除する前に、それを無効にする必要があります。

次の手順に従ってください：

1. グローバル管理者として CA Advanced Authentication コンソールにログインします。
2. [組織] タブをクリックします。
3. [検索] をクリックし、ユーザ行動プロファイリング モデルが実装されている組織を選択します。
4. [CA Risk Authentication] タブに移動し、[モデル設定] を選択します。
5. この組織に対して定義されているルールセットを選択します。
6. [モデルの有効化] チェック ボックスをオンにします。
7. [保存] をクリックします。
8. [実稼働にマイグレート] メニューをクリックし、組織に適切なルールセットを選択します。
9. [マイグレート] をクリックして、この新しい変更を運用環境にマイグレートします。
10. [組織の検索] メニューに移動し、[検索] をクリックします。
11. 対象の組織を選択し、[キャッシュのリフレッシュ] をクリックします。
12. [OK] をクリックします。

これで、組織のルールセットからユーザ行動プロファイリング モデルが削除されます。

### ユーザ行動プロファイリングのアンインストール

ユーザ行動プロファイリング モデルのアンインストールでは、ユーザ行動プロファイリング モデル スキーマを削除してから、ユーザ行動プロファイリング モデルをアンインストールします。

次の手順に従ってください：

1. 以下のディレクトリに移動します。  
<install\_location>%Arcot Systems%dbscripts¥
2. 使用しているデータベースに応じて、以下のいずれかのサブディレクトリに移動します。
  - Oracle の場合  
<install\_location>%Arcot Systems%dbscripts¥oracle¥
  - Microsoft SQL Server の場合  
<install\_location>%Arcot Systems%dbscripts¥mssql¥
  - MySQL の場合  
<install\_location>%Arcot Systems%dbscripts¥mysql¥
3. CA Risk Authentication と関連コンポーネントのすべてのデータベース テーブルを削除するには、以下の順序でスクリプトを実行します。
  - a. drop-arcot-db-config-for-userprofiling-2.0.sql を実行します。
4. 以下のサーバをシャットダウンします。
  - a. CA Risk Authentication サーバ
  - b. CA Risk Authentication ケース管理サービス
  - c. CA Risk Authentication のその他のコンポーネントが展開されているすべてのアプリケーション サーバ
5. 管理コンソールを閉じます。
6. INI ファイルおよびその他の CA Risk Authentication 設定ファイルがすべて閉じられていることを確認します。
7. [スタート] - [設定] - [コントロールパネル] - [プログラムの追加と削除] の順にクリックして、[プログラムの追加と削除] ウィンドウを開きます。
8. CA Risk Authentication を選択して [変更と削除] をクリックします。
9. [Uninstall CA Risk Authentication] ウィンドウが表示されます。
10. Uninstall CA Risk Authentication.exe を選択します。



11. アンインストール ウィザードで、[Uninstall Specific features] を選択します。
12. [Next] をクリックし、[Only User Behavioral Profiling] を選択します。
13. [Uninstall] を選択します。
14. [Done] をクリックして、処理を完了します。

ユーザ行動プロファイリング モデルがデータベースから正常に削除されます。

## 第 16 章: Oracle RAC 用の CA Risk Authentication の設定

このセクションの手順は、CA Risk Authentication 8.0 で Oracle RAC を使用する場合に実行します。

### arcot-db-config-for-common-2.0.sql スクリプトの更新

データベース スクリプト (arcot-db-config-for-common-2.0.sql スクリプト) をインストール後のタスクとして実行します。このスクリプトを実行する前に、Oracle RAC に対して変更します。

以下の手順に従います。

1. Oracle RAC の共有データ ファイルパスを確認するには、データベースにログインし、以下のコマンドを実行します。

```
SELECT file_name, tablespace_name FROM dba_data_files
```

このコマンドのサンプル出力を以下に示します。

```
+DATA%qadb%datafile%users.259.797224649    USERS
+DATA%qadb%datafile%undotbs1.258.797224649  UNDOTBS1
+DATA%qadb%datafile%sysaux.257.797224647    SYSAUX
```

2. arcot-db-config-for-common-2.0.sql ファイルを開きます。このファイルは、*install\_location*%Arcot Systems%dbscripts%oracle% ディレクトリにあります。
3. ファイル内で以下の行を見つけます。

```
filename varchar2(50) := 'tabspace_arreports_' || to_char(current_timestamp,
'YYYY-MM-DD-HH24-MI-SS') || '.dat';
```

4. その行を以下の行に置き換えます。

```
filename varchar2(100) :=  
'+shared_location/service_name/datafile/tabspace_arreports_'||  
to_char(current_timestamp, 'YYYY-MM-DD-HH24-MI-SS') || '.dat';
```

新しい行で以下を行います。

- a. `shared_location` を、最初の手順で指定されたコマンドの実行により確認した共有データ ファイルパスに置き換えます。
- b. `service_name` を、Oracle RAC インストールのサービス名に置き換えます。

以下は変更後の行の例です。

```
filename varchar2(100) := '+DATA/forwardinc/datafile/tabspace_arreports_'||  
to_char(current_timestamp, 'YYYY-MM-DD-HH24-MI-SS') || '.dat';
```

5. スクリプト ファイルを保存して閉じ、実行します。

## arcotcommon.ini ファイルの更新

arcotcommon.ini ファイルには、データベースとインスタンスの設定用のパラメータが含まれます。Oracle RAC を使用するには、arcotcommon.ini ファイルで Oracle RAC によってサポートされている形式で JDBC URL を指定します。

以下の手順に従います。

1. テキストエディタで arcotcommon.ini ファイルを開きます。このファイルは install\_location¥Arcot Systems¥conf¥ディレクトリにあります。
2. URL パラメータの値を、INI ファイルの [arcot/db/primarydb] セクションに指定し、必要に応じて [arcot/db/backupdb] セクションにも指定します。URL を以下の形式で入力します。

```
URL.1=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=host_name)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=service_name)(SERVER=DEDICATED)))
```

例：

```
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)))
```

注：Oracle RAC が設定されたクライアントである場合は、すべてのノードをこの形式で含めます。

3. CA Strong Authentication インストーラの実行中に指定したデータベースユーザが、Oracle RAC のデータベースユーザとは異なる場合は、arcotcommon.ini ファイル内のデータベースユーザ認証情報を変更します。
4. DBUtil ユーティリティを使用して、securestore.enc ファイル内のデータベースユーザ認証情報を変更します。DBUtil は、ARCOT\_HOME¥tools¥win ディレクトリ内にあります。
5. arcotcommon.ini ファイルを保存して閉じます。

## データベース接続の詳細の更新

CA Risk Authentication と Oracle RAC の間の接続を確立するには、ORA ファイルを作成し、RAC に接続するためのアドレスを定義する必要があります。

以下の手順に従います。

1. CA Strong Authentication をインストールしたシステムで \*.ora ファイルを作成します。例：C:\Program Files (x86)\tns.ora
2. 作成したファイルに以下の行を追加します。

```
section_name =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = host_name_or_IP_address)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = service_name)
    )
  )
```

例：

```
fwdincrac =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 172.30.250.18)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = forwardinc)
    )
  )
```

**注：**Oracle RAC が設定されたクライアントである場合は、すべてのノードをこの形式で含めます。

3. ファイルを保存します。
4. インストール中に作成した DSN を変更します。
5. 必要な DSN に対して、[Standard Connection] セクション内のパラメータをすべてクリアします。

これにより、[TNSNames Connection] セクションが編集可能になります。

6. このセクションに以下のパラメータを追加します。

```
TNSNamesFile=ARCOT_HOME\ora_file_name
ServerName=section_name
```

例：

```
TNSNamesFile= C:\Program Files (x86)\tns.ora  
ServerName=fwdincrac
```

7. ファイルを保存して閉じます。

## 第 17 章：データベース接続プールのためのアプリケーション サーバの設定

リクエストごとに新しい接続をセットアップするとオーバーヘッドとなり、システムのパフォーマンスを低下させることがあります。データベース接続プールの実装によって、アプリケーションサーバに展開されている **CA Risk Authentication** コンポーネントがデータベースへのアクセスを要求するたびに、新しいデータベース接続を作成するオーバーヘッドを回避できます。

### データベース接続プールの有効化

このセクションでは、**CA Risk Authentication** コンポーネントを展開したアプリケーションサーバに、データベース接続プールをセットアップする手順について説明します。

#### Apache Tomcat

ここでは、JNDI ベースのデータベース操作に Apache Tomcat を有効にする手順について説明します。

次の手順に従ってください：

1. Apache Tomcat アプリケーションサーバをインストールし、以下の URL を使用してインストールをテストします。  
`http://localhost:8080/`
2. <TOMCAT\_HOME>/conf/ ディレクトリにある server.xml ファイルを開きます。

3. データソースの定義に必要な以下の情報を収集します。

#### JNDI 名

CA Risk Authentication によって使用される JNDI 名を指定します。

**重要:** この名前は、arcotcommon.ini (java:comp/env/プレフィックスなし) の AppServerConnection PoolName.N と一致する必要があります。

#### ユーザ ID

データベース ユーザ ID を指定します。

#### Password

データベース パスワードを指定します。

#### JDBC ドライバ クラス

JDBC ドライバクラス名を指定します。例：  
oracle.jdbc.driver.OracleDriver

#### JDBC URL

データベース サーバ用の JDBC URL を指定します。たとえば、Oracle ドライバを使用している場合、URL は以下のとおりです。

jdbc:oracle:thin:<server>:<database\_port>:<sid>

4. 次のエントリを追加して、<GlobalNamingResources> タグ内にデータソースを定義します。

```
<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. <TOMCAT\_HOME>%conf% ディレクトリにある context.xml ファイルを開きます。
6. <Context> タグ内に以下のエントリを追加してデータソースを定義します。  
<ResourceLink global="SampleDS" name="SampleDS"  
type="javax.sql.DataSource"/>
7. データベース接続プールを有効にするには、対応するサードパーティソースから以下のファイルをダウンロードします。次に、それらのファイルを以下のディレクトリにコピーします。  
<TOMCAT\_HOME>%common%Lib% フォルダ (Apache Tomcat 5.x)、または  
<TOMCAT\_HOME>%Lib% ディレクトリ (Apache Tomcat 6.x および 7.x)
  - commons-dbcp-1.2.2.jar
  - ojdbc14-10.2.0.1.0.jar (Oracle データベースの場合)
  - sqljdbc.jar (MS SQL Server 2005 用 Microsoft JDBC ドライババージョン 1.2.2828)
  - mysql-connector-java-5.1.22-bin.jar (MySQL データベースの場合)

#### IBM WebSphere

ここでは、JNDI ベースのデータベース操作に IBM WebSphere を有効にする手順について説明します。

次の手順に従ってください：

1. WebSphere Administration Console にログインします。
2. [Resources] を選択し、[JDBC] ノードを展開します。
3. [JDBC Providers] をクリックします。
4. [Preferences] セクションで [New] をクリックし、使用しているデータベースに基づいて適切な JDBC プロバイダを作成します。
5. 以下のタスクを実行して、JDBC プロバイダを作成します。

注: 詳細については、以下のリンクを参照してください。

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.webSphere.base.iseries.doc/info/series/ae/tdat\\_ccrtprov.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.webSphere.base.iseries.doc/info/series/ae/tdat_ccrtprov.html)

6. [Database Type] および [Provider Type] を指定します。
  - a. [Implementation Type] ドロップダウンリストから [Connection pool data source] を選択します。
  - b. JDBC プロバイダの名前を入力します。JDBC プロバイダの説明を入力することもできます。
  - c. [次へ] をクリックします。
  - d. JAR ファイルの絶対パスを入力します。
  - e. [次へ] をクリックします。
  - f. 入力した情報のサマリを確認したら、[Finish] をクリックして JDBC プロバイダの設定を完了します。
7. 手順 5 で作成した JDBC プロバイダの CLASSPATH を設定します。
  - a. [Resources] を選択し、[JDBC] ノードを展開します。
  - b. [JDBC Providers] をクリックします。
  - c. 手順 5 で作成した JDBC プロバイダをクリックします。
  - d. JDBC JAR のクラスパスを設定します。
  - e. [Apply] をクリックして、変更を保存します。
8. 以下の手順に従って、データソースを作成します。
  - a. [Resources] に移動し、[JDBC] をクリックします。
  - b. [JDBC] の [Data Sources] を開き、[New] をクリックします。データソースを作成するには、以下の手順に従います。
  - c. データソース名を指定します。
  - d. JNDI 名を指定します。

注: この名前は `arcotcommon.ini` の `AppServerConnection PoolName.N` の値と一致する必要があります。



- e. [次へ] をクリックします。
- f. 手順 3 で作成した既存の JDBC プロバイダを選択します。
- g. [次へ] をクリックします。
- h. データベースの種類に応じて、以下の情報を入力します。

- **Oracle の場合**

JDBC URL の値を指定します。この URL は以下の形式になります。

```
jdbc:oracle:thin:@<server>:<oracle_port>:<sid>
```

[Data store helper class name] を選択します。

- **MS SQL Server の場合**

```
jdbc:sqlserver://<server>:<sql_port>;databaseName=<database name>;selectMethod=cursor
```

- **MySQL の場合**

```
jdbc:mysql://<server>:<port-number>/<database>
```

- i. [次へ] をクリックします。
  - j. [Next] をクリックして [Summary] 画面を確認し、[Finish] をクリックします。
9. 手順 8 で作成したデータ ソースをクリックします。
  10. [Related Items] セクションで、[JAAS - J2C authentication data] をクリックします。
  11. [New] をクリックして新しい認証情報を作成します。
  12. データベースへの接続に使用されるログイン クレデンシャルを入力し、クレデンシャルを保存します。
  13. [Apply] をクリックし、[OK] をクリックして変更内容を保存します。
  14. [Data Sources] をクリックし、手順 8 で作成したデータ ソースを選択します。

15. [Security Settings] - [Component-managed authentication alias] で、手順 12 で作成した JAAS 認証情報を選択し、[Apply] - [OK] の順にクリックします。
16. [Data Sources] をクリックし、手順 8 で作成したデータソースのチェックボックスをオンにします。
17. [Test connection] をクリックし、接続が正しく指定されているかどうかを検証します。

**注:** このテストでは、データベースサーバへの接続のみが確認され、データソースの定義が正しいかどうかは必ずしも確認されません。

### Oracle WebLogic

このセクションでは、Oracle WebLogic で JNDI ベースのデータベース操作を行うための手順を説明します。

次の手順に従ってください：

1. WebLogic Administration Console にログインします。
2. まだ [Change Center] の [Lock & Edit] ボタンをクリックしていない場合は、このボタンをクリックします。
3. [Services] - [JDBC] - [Data Sources] の順に移動します。
4. [JDBC] の [Data Sources] を開き、[New] をクリックします。
5. 以下の JNDI 情報とデータベース情報を設定します。
  - a. [Name] を「ArcotDB」に設定します。

**注:** この名前は arcotcommon.ini の AppServerConnection PoolName.N の値と一致する必要があります。
  - b. [JNDI Name] を「ArcotDB」に設定します。
  - c. 必要な [Database Type] を選択 (Oracle など)。
  - d. 必要な [Database Driver] を選択 (Oracle Thin Driver など)。

6. [Next] をクリックし、デフォルト値をそのまま使用して再度 [Next] をクリックします。
7. 表示される接続プロパティ ページで、データベース接続の詳細を設定します。たとえば、Oracle には以下の値を使用できます。
  - [Database Name] = データベース サーバの SID またはサービス名
  - [Host Name] = データベースのホスト名または IP アドレス
  - [Port] = 1521、またはデータベース サーバを実行している任意の他のポート
  - [Database User Name] = データベース接続を作成できるデータベース アカウント ユーザ名
  - [Password] / [Confirm Password] = 指定されたデータベース ユーザ名のパスワード
8. [次へ] をクリックします。
9. [Test Configuration] をクリックして、指定したデータベース情報を確認します。
10. [Next] をクリックし、WebLogic サーバインスタンス用の優先データソースのターゲット サーバを設定します。
11. [完了] をクリックします。
12. [Change Center] の [Activate] ボタンをクリックして、前の手順で設定したデータソースの設定を有効にします。

### JBoss アプリケーション サーバ

このセクションでは、JBoss アプリケーション サーバで JNDI ベースのデータベース操作を行うための手順を説明します。

次の手順に従ってください：

1. WAR ファイルを展開した場所に移動します。以下に例を示します。  
`<JBASS_HOME>%server%default%deploy%`
2. `arcotdatabase-ds.xml` というデータ ソース記述子ファイルを作成します。
3. `arcotdatabase-ds.xml` ファイルにデータ ソースを定義するために必要な以下の情報を収集します。
  - **JNDI 名**：CA Risk Authentication コンポーネントによって使用される JNDI 名。この名前は、`arcotcommon.ini` (`java:comp/env/` プレフィックスなし) の `AppServerConnection PoolName.N` と一致する必要があります。
  - **ユーザ ID**：データベース ユーザ ID。
  - **パスワード**：データベース パスワード。
  - **JDBC ドライバクラス**：JDBC ドライバクラス名。たとえば、`oracle.jdbc.driver.OracleDriver`。
  - **JDBC URL**：データベース サーバ用の JDBC URL。  
たとえば、Oracle ドライバを使用している場合、この URL は `jdbc:oracle:thin:<server>:<database_port>:<sid>` となります。
  - **Exception Sorter クラス**：例外が接続エラーを示すかどうかを判断する `org.jboss.resource.adapter.jdbc.ExceptionSorter` インターフェースを実装するクラス。  
このパラメータは、Oracle データベースにのみ使用します。  
`org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter` に設定します。

4. テキストエディタで `arcotdatabase-ds.xml` を開きます。
5. 以下の内容を追加します。

```
<?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. ファイルを保存して閉じます。

## LDAP 接続プールの有効化

以下のアプリケーション サーバの設定手順について説明します。

### Apache Tomcat

LDAP 接続プールを作成するには、以下の手順に従います。

1. Apache Tomcat アプリケーション サーバをインストールし、以下の URL を使用してインストールをテストします。  
`http://localhost:8080/`
2. 以下の場所に移動します。  
`<TOMCAT-HOME>%conf%`
3. テキスト エディタで `catalina.properties` ファイルを開きます。
4. ファイルに、以下のエントリを追加します。
  - `com.sun.jndi.ldap.connect.pool.protocol=plain ssl`
  - `com.sun.jndi.ldap.connect.pool.authentication=simple`
  - `com.sun.jndi.ldap.connect.pool.maxsize=64`
  - `com.sun.jndi.ldap.connect.pool.prefsiz=32`
  - `com.sun.jndi.ldap.connect.pool.timeout=240000`
  - `com.sun.jndi.ldap.connect.pool.initsize=8`
5. ファイルを保存して閉じます。
6. アプリケーション サーバを再起動します。

### IBM WebSphere

LDAP 接続プールを作成するには、以下の手順に従います。

1. WebSphere Administration Console にログインします。
2. [Servers] - [Server Types] - [WebSphere application servers] に移動します。
3. 設定するサーバをクリックします。
4. [Server Infrastructure] セクションで、[Java and Process Management] をクリックします。
5. [Process Definition] リンクをクリックします。
6. [Additional Properties] セクションで、[Java Virtual Machine] をクリックします。

7. [Additional Properties] セクションで、[Custom Properties] をクリックします。
8. [New] をクリックして、カスタムプロパティを追加します。
9. 以下の表にリストされている設定を、名前と値をペアにして [General Properties] セクションに追加します。名前と値のペアごとに処理を繰り返す必要があります。

| Name  | 値         |
|---|-----------|
| com.sun.jndi.ldap.connect.pool.maxsize        | 64        |
| com.sun.jndi.ldap.connect.pool.prefsiz        | 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    |

10. [適用] をクリックします。
11. WebSphere を再起動します。

Oracle WebLogic

### 起動スクリプトへの LDAP オプションの追加

このセクションでは、WebLogic サーバの起動スクリプトに LDAP 接続プールパラメータを含める手順について説明します。

1. システムにログインします。
2. WebLogic サーバの起動スクリプトのバックアップ コピーを作成します。このスクリプトは以下の場所にあります。  
domain-name¥bin¥startWebLogic.cmd
3. テキスト エディタでスクリプトを開きます。
4. WebLogic サーバの起動に使用されるセクションに以下のエントリを追加します。

- -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

以下のコード スニペットは、LDAP 接続プールパラメータが設定されているサンプルスクリプトを示しています。

```
@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.prefsiz=32
-Dcom.sun.jndi.ldap.connect.pool.initsize=8
-Dcom.sun.jndi.ldap.connect.pool.timeout=240000
-Dcom.sun.jndi.ldap.connect.pool.protocol="plain ssl"
-Dcom.sun.jndi.ldap.connect.pool.authentication=simple
-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. ファイルを保存して閉じます。
6. WebLogic サーバを再起動します。

### 管理対象サーバを使用した LDAP プール オプションの指定

1. WebLogic Administration Console にログインします。
2. ロックと編集が終わっていない場合は、[Lock & Edit] ボタンをクリックします。
3. [Domain Structure] ペインで、[Environment] - [Servers] に移動します。
4. 設定するサーバをクリックします。
5. 右側のペインで、[Server Start] をクリックします。
6. [Arguments] フィールドに、スペースで区切って以下の JVM オプションを含めます。
  - -Dcom.sun.jndi.ldap.connect.pool.maxsize=64
  - -Dcom.sun.jndi.ldap.connect.pool.prefsiz=32
  - -Dcom.sun.jndi.ldap.connect.pool.initsize=8
  - -Dcom.sun.jndi.ldap.connect.pool.timeout=240000
  - -Dcom.sun.jndi.ldap.connect.pool.protocol=plain ssl
  - -Dcom.sun.jndi.ldap.connect.pool.authentication=simple
7. [Save] をクリックして、[Activate Changes] をクリックします。
8. WebLogic サーバを再起動します。

## JBoss アプリケーション サーバ

LDAP 接続プールを作成するには、以下の手順に従います。

1. 以下の場所に移動します。  
<JBASS\_HOME>%standalone%configuration
2. テキストエディタで standalone.xml ファイルを開きます。
3. 以下のプロパティを追加します。

```
<system-properties>
<property name="com.sun.jndi.ldap.connect.pool.protocol"
value="plain ssl"/>
<property name="com.sun.jndi.ldap.connect.pool.authentication"
value="simple"/>
<property name="com.sun.jndi.ldap.connect.pool.maxsize"
value="64"/>
<property name="com.sun.jndi.ldap.connect.pool.prefsize"
value="32"/>
<property name="com.sun.jndi.ldap.connect.pool.timeout"
value="240000"/>
<property name="com.sun.jndi.ldap.connect.pool.initsize"
value="8"/>
</system-properties>
```

4. ファイルを保存して閉じます。
5. JBoss AS を再起動します。

## Apache Tomcat のセキュリティマネージャの有効化

Java セキュリティ マネージャが有効な場合に、CA Risk Authentication が Apache Tomcat 上で動作していない場合は、Tomcat のセキュリティ マネージャを有効にして CA Risk Authentication で動作するようにするために、以下の手順に従います。

1. 以下の Apache Tomcat のインストール場所に移動します。  
`<Tomcat_Home>%bin%`
2. **tomcat<version>w.exe** ファイルをダブルクリックします。  
[Apache Tomcat Properties] ダイアログ ボックスが表示されます。
3. [Java] タブをアクティブにします。
4. [Java Options] セクションで、以下のエントリを追加します。
  - -Djava.security.manager
  - -Djava.security.policy=<Tomcat\_Home>%conf%catalina.policy
5. [Apply] をクリックして、変更を保存します。
6. [OK] をクリックして、[Apache Tomcat Properties] ダイアログ ボックスを閉じます。
7. 以下の Apache Tomcat の場所に移動します。  
`<Tomcat_Home>%conf%`
8. 任意のテキスト エディタで `catalina.properties` ファイルを開きます。
9. 以下のコードを WEB APPLICATION PERMISSIONS セクションに追加します。

```
grant {
permission java.io.FilePermission
"${catalina.base}${file.separator}webapps${file.separator}arcotuds${file.separator}-", "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. 管理コンソール (arcotadmin) およびユーザ データ サービス (arcotuds) に対する権限を付与するために、以下のセクションを追加します。

```
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. ファイルを保存して閉じます。
12. Apache Tomcat を再起動します。

# 付録 A: IBM WebSphere への管理コンソールの展開

---

IBM WebSphere 7.0、8.0、または 8.5 に管理コンソールを展開する場合、[インスタンス管理] などの一部の管理コンソールページへのアクセスで、HTTPCLIENT エラーが表示される場合があります。そのような場合、以下の手順に従う必要があります。

1. `<install_location>%Arcot Systems%java%webapps%` から管理コンソール WAR ファイルにアクセスします。
2. 一時ディレクトリ（例： `C:%Arcot_temp%`）に `arcotadmin.war` をコピーします。
3. `arcotadmin.war` ファイルの内容を抽出します。

`C:%Arcot_temp%arcotadmin%WEB-INF%lib%` ディレクトリに抽出される JAR のうち、以下の JAR が IBM WebSphere で共有ライブラリを作成するために使用されます。

- `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. WebSphere Administration Console にログインします。

5. [Environment] をクリックしてから、[Shared Libraries] をクリックします。
  - a. [Scope] ドロップダウンから、有効な可視性範囲を選択します。範囲には、アプリケーションを展開するターゲット サーバまたはノードを含める必要があります。
  - b. [新規] をクリックします。
  - c. 名前を入力します。例：ArcotAdminSharedLibrary。
  - d. クラスパスを指定します。手順 3 で抽出したすべての JAR ファイルのパスとファイル名を入力します。  
例：  
C:/Arcot\_temp/arcotadmin/WEB-INF/lib/axiom-api-1.2.10.jar
  - e. [Apply] をクリックして、変更を保存します。
6. 管理コンソール WAR ファイルがある場所 (<install\_location>%Arcot Systems%java%webapps%) に移動します。
7. アプリケーション サーバに arcotadmin.war を展開します。
8. 以下の手順に従って、共有ライブラリを設定します。
  - a. [Applications] をクリックして、[WebSphere enterprise applications] をクリックします。
  - b. [arcotadmin\_war] をクリックします。
  - c. [References] セクションで、[Shared library references] をクリックします。
  - d. [arcotadmin\_war] を選択し、[Reference shared libraries] をクリックします。
  - e. [Available] リストから [ArcotAdminSharedLibrary] を選択し、[Selected] リストに移動させます。
  - f. [OK] をクリックして設定を保存します。

9. 以下の手順に従って、クラスローダの順序およびポリシーを設定します。
  - a. [Applications] - [Application Types] - [WebSphere enterprise applications] をクリックします。
  - b. [arcotadmin\_war] をクリックします。
  - c. [Class loading and update detection] リンクをクリックします。
  - d. [Class loader order] セクションで、[Classes loaded with local class loader first (parent last)] オプションを選択します。
  - e. [WAR class loader policy] セクションで、[Single class loader for application] オプションを選択します。
  - f. [OK] をクリックして設定を保存します。
10. アプリケーションが再起動されたことを確認します。

IBM WebSphere 8.0 および 8.5 に管理コンソールを展開する場合、[インスタンス管理] などの一部の管理コンソールページへのアクセスで、HTTPCLIENT エラーが表示される場合があります。そのような場合、以下の手順に従う必要があります。

1. <install\_location>%Arcot Systems%java%webapps% から管理コンソール WAR ファイルにアクセスします。
2. 一時ディレクトリ（例：C:%Arcot\_temp%）に arcotadmin.war をコピーします。
3. arcotadmin.war ファイルの内容を抽出します。

C:%Arcot\_temp%arcotadmin%WEB-INF%lib% ディレクトリに抽出される JAR のうち、以下の JAR が IBM WebSphere で共有ライブラリを作成するために使用されます。

- 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. WebSphere Administration Console にログインします。
  5. [Environment] をクリックしてから、[Shared Libraries] をクリックします。
    - a. [Scope] ドロップダウンから、有効な可視性範囲を選択します。範囲には、アプリケーションを展開するターゲット サーバまたはノードを含める必要があります。
    - b. [新規] をクリックします。
    - c. 名前を入力します（たとえば、ArcotAdminSharedLibrary）。
    - d. クラスパスを指定します。手順 3 で抽出したすべての JAR ファイルのパスとファイル名を入力します。

例：  
C:/Arcot\_temp/arcotadmin/WEB-INF/lib/axiom-api-1.2.10.jar
    - e. [Apply] をクリックして、変更を保存します。



6. 管理コンソール WAR ファイルがある場所 (<install\_location>%Arcot Systems%java%webapps%) に移動します。
7. アプリケーション サーバに arcotadmin.war を展開します。
8. 以下の手順に従って、共有ライブラリを設定します。
  - a. [Applications] をクリックして、[WebSphere enterprise applications] をクリックします。
  - b. [arcotadmin\_war] をクリックします。
  - c. [References] セクションで、[Shared library references] をクリックします。
  - d. [arcotadmin\_war] を選択し、[Reference shared libraries] をクリックします。
  - e. [Available] リストから [ArcotAdminSharedLibrary] を選択し、[Selected] リストに移動させます。
  - f. [OK] をクリックして設定を保存します。
9. 以下の手順に従って、クラス ロードの順序およびポリシーを設定します。
  - a. [Applications] - [Application Types] - [WebSphere enterprise applications] をクリックします。
  - b. [arcotadmin\_war] をクリックします。
  - c. [Class loading and update detection] リンクをクリックします。
  - d. [Class loader order] セクションで、[Classes loaded with local class loader first (parent last)] オプションを選択します。
  - e. [WAR class loader policy] セクションで、[Single class loader for application] オプションを選択します。
  - f. [OK] をクリックして設定を保存します。
10. アプリケーションが再起動されたことを確認します。

## 第 18 章: CA Risk Authentication SDK および Web サービスの設定

このセクションでは、CA Risk Authentication が提供するアプリケーションプログラミングインターフェース (API) と Web サービスを設定する手順について説明します。

## CA Risk Authentication API の設定

CA Risk Authentication には、Java API のセットが付属しています。CA Risk Authentication API パッケージを設定して、以下の操作を有効にすることができます。

- リスクの評価
- アドバイスの生成
- ユーザとデバイスの関連付けの表示
- 関連付けの削除

次の手順に従ってください：

1. 以下の場所に移動します。

```
<install_location>%Arcot Systems%sdk%java%lib%arcot%
```

2. コア JAR (リスク評価 SDK : *arcot-riskfort-evaluaterisk.jar*) を実装します。

さらに、コア JAR が依存する以下の JAR が提供されています。

- *arcot\_core.jar*
  - *arcot-pool.jar*
  - *arcot-riskfort-mfp.jar*
3. (オプション) 同じ場所から発行 SDK の JAR (*arcot-riskfort-issuance.jar*) を実装できます。

ただし、この API は、このリリースでは廃止されており、単に下位互換性のためにのみ含まれています。

**注:** この API の代わりに、*ユーザ管理 Web サービス*を使用できます。詳細については、「*CA Risk Authentication Web サービス開発者ガイド*」を参照してください。

## Java API の設定

このセクションでは、Java API をアプリケーションで使用できるように設定する手順について説明します。

**重要:** 先に進む前に、Java API を実装するために必要な JAR ファイルが、`<install_location>%Arcot Systems%sdk%java%lib%` にインストールされていることを確認してください。

Java API を設定するには、以下の手順に従います。

**注:** 以下の手順は Apache Tomcat サーバをベースとしています。設定プロセスは、使用しているアプリケーションサーバによって変わる可能性があります。これらの手順の詳細については、アプリケーションサーバのドキュメントを参照してください。

1. 以下の場所から、以下のリストの JAR ファイルをコピーします。  
`<install_location>%Arcot Systems%`  
`<APP_SERVER_HOME>` ディレクトリ内の適切な場所にそれらを配置します。

**例:** Apache Tomcat の場合、この場所は `<Application_Home>%WEB-INF%lib%` です。

- `/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`

**例:** Apache Tomcat 5.5.x では、これらのファイルを `C:%Program Files%Apache Software Foundation%Tomcat 5.5.31%webapps%<Your_Application>%WEB-INF%lib%` にコピーする必要があります。

2. 以下の手順に従って、`log4j.properties.risk-evaluation` および `riskfort.risk-evaluation.properties` ファイルを設定します。
  - アプリケーションに設定済みの `log4j.properties.risk-evaluation` ファイルがすでにある場合は、以下のログ設定ファイルとマージします。

```
<install_location>%Arcot  
Systems%sdk%java%properties%log4j.properties.risk-evaluatio  
n
```

および

```
<install_location>%Arcot  
Systems%sdk%java%properties%riskfort.risk-evaluation.proper  
ties
```

- アプリケーションに `log4j.properties` ファイルが設定されていない場合、以下の手順に従います。
  - a. `log4j.properties.risk-evaluation` の名前を `log4j.properties` に変更します。
  - b. `riskfort.risk-evaluation.properties` と `log4j.properties` をマージします。
  - c. `log4j.properties` ファイルを以下の場所にコピーします。  
<Application\_Home>%WEB-INF%classes%properties%

例：Apache Tomcat 5.5.x では、`log4j.properties` を `C:%Program Files%Apache Software Foundation%Tomcat 5.5.31%webapps%<Your_Application>%WEB-INF%classes%` にコピーする必要があります。

注：API とその初期化の詳細については、<install\_location>%Arcot Systems%docs%riskfort% Arcot-RiskFort-8.0-issuance-sdk-javadocs.zip にある CA Risk Authentication Javadoc を参照してください。

## CA Risk Authentication Web サービスの設定

CA Risk Authentication Web サービスを使用するには、*arcotuds.war* ファイルを展開します。

CA Risk Authentication は、ユーザ、組織、システムの管理を実行するため、およびリスク評価を実行するための Web サービスを提供します。これらの Web サービス用の WSDL は、以下の場所にあります。

`<install_location>%Arcot Systems%wsdls%`

### WSDL を使用したクライアントコードの生成

CA Risk Authentication パッケージのインストール後、CA Risk Authentication に付属する WSDL ファイルを使用して、コード化する言語でクライアントスタブを生成します。これらの WSDL により、Web サービスクライアントは、CA Risk Authentication サーバと通信可能になります。

**重要:** クライアントコードの生成を実行する前に、CA Risk Authentication パッケージが正常にインストールされ、サーバが稼働中であることを確認します。

次の手順に従ってください：

1. アプリケーションサーバを停止します。
2. 以下の場所に移動します。  
`<install_location>%Arcot Systems%wsdls%<required_folder>`
3. 必要な WSDL ファイル(以下の表を参照)を使用してクライアントコードを生成します。

| WSDL ファイル   | Description                                      |
|---|--|
| <code>admin/ArcotRiskFortAdminWebService.wsdl</code>        | ルールの作成および管理に対して使用されます<br>管理コンソールの使用により通常実行される設定。 |
| <code>riskfort/ArcotRiskFortEvaluateRiskService.wsdl</code> | リスク評価の実行に使用します。                                  |
| <code>uds/ArcotUserRegistryMgmtSvc.wsdl</code>              | セットアップ内の組織の作成と管理のために使用します。                       |
| <code>uds/ArcotConfigRegistrySvc.wsdl</code>                | ユーザアカウントタイプの作成と管理に使用します。                         |
| <code>uds/ArcotUserRegistrySvc.wsdl</code>                  | ユーザおよびユーザアカウントの作成と管理に使用します。                      |

4. アプリケーションサーバを再起動します。
5. ブラウザ ウィンドウで、エンドポイント URL（以下の表を参照）にアクセスして、クライアントが Web サービスにアクセスできるかどうかを確認します。

| Web サービス                         | URL   |
|----------------------------------|---|
| ArcotRiskFortAdminWebService     | <i>http://&lt;rf_hostname&gt;:&lt;rf_port&gt;/services/ArcotRiskFortAdminSvc</i><br>ここで指定するデフォルト ポートは <b>7777</b> です。   |
| ArcotRiskFortEvaluateRiskService | <i>http://&lt;rf_hostname&gt;:&lt;rf_port&gt;/services/RiskFortEvaluateRiskSvc</i><br>ここで指定するデフォルト ポートは <b>7778</b> です。 |
| ArcotUserRegistryMgmtSvc         | <i>http://&lt;app_server_hostname&gt;:&lt;appserver_port&gt;/arcotuds/services/ArcotUserRegistrySvc</i>                 |
| ArcotConfigRegistrySvc           | <i>http://&lt;app_server_hostname&gt;:&lt;appserver_port&gt;/arcotuds/services/ArcotConfigRegistrySvc</i>               |
| ArcotUserRegistrySvc             | <i>http://&lt;app_server_hostname&gt;:&lt;appserver_port&gt;/arcotuds/services/ArcotUserRegistryMgmtSvc</i>             |

注: Java クライアントの生成の詳細については、「CA Risk Authentication Web サービス開発者ガイド」を参照してください。

## デバイス ID および DeviceDNA の設定

CA Risk Authentication は、デバイス ID と DeviceDNA を使用して、トランザクション中にユーザが使用するデバイスを登録し、識別します。デバイス ID は、エンドユーザのデバイス上に格納されます。デバイス ID 情報は暗号化されています。

以下に、エンドユーザのデバイス上に Device ID を格納するオプションを示します。プラグインストアは、最も永続的なストレージオプションです。

- **プラグインストア**：プラグインストアは、エンドユーザのデバイス上の永続ストアです。プラグインストアに配置されるデバイス ID は、ブラウザ キャッシュのクリアやブラウザ Cookie の削除など、一般的なエンドユーザアクションでは削除できません。プラグインストアは、CA Risk Authentication クライアント リリース 2.1 以降でサポートされています。
- HTML5 で提供されるローカルストレージ
- **UserData ストア**：このストアは Microsoft Internet Explorer でのみ使用できます
- **Cookie ストア**：通常、Microsoft Windows では、デバイス ID は以下のいずれかのフォルダに格納されます。
  - **Microsoft Windows 7 または 2008 上の Internet Explorer**  
C:\Documents and Settings\<user\_profile>\Application Data\Microsoft\Windows\Cookies\
  - **Microsoft Windows 2003 または XP 上の Internet Explorer**  
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

**重要**：CA Risk Authentication クライアントバージョン 2.0 以降、デバイス ID は Flash cookie としては格納されません。以前のリリースからの既存の Flash cookie が存在する場合、それらの Cookie は、このセクションで前に示したストアの 1 つに自動的に移行されます。

### デバイス ID および DeviceDNA の収集に必要なファイル

Complete インストールを実行するか、または [Choose Install Set] 画面で CA Risk Authentication 評価 SDK または Web サービスをインストールすることを選択すると、以下のファイルが自動的にインストールされます。

```
<install_location>%Arcot  
Systems%sdk%devicedna%riskminder-client.js
```

このファイルは、デバイス ID および DeviceDNA を取得および設定するための関数を提供します。

### デバイス ID および DeviceDNA の収集の有効化

Cookie をエンドユーザ コンピュータに設定するには、Cookie を取得または設定するアプリケーション ページに riskminder-client.js を追加する必要があります。

次の手順に従ってください：

1. devicedna ディレクトリ全体を <install\_location>%Arcot Systems%sdk% から、適切な Web アプリケーションディレクトリにコピーします。通常、Web アプリケーションフォルダは以下の場所にあります。  
<APP\_SERVER\_HOME>%<Your\_Application\_Home>
2. 必要なアプリケーション ページに riskminder-client.js ファイルを追加します。これらのファイルが、index.jsp が含まれるフォルダから相対的な位置にあるフォルダに存在することを前提とします。  
<script type="text/javascript"  
src="devicedna/riskminder-client.js"></script>

### 以前のリリースからの Flash Cookie の移行

Flash Cookie は、デバイス ID の格納に対してサポートされなくなりました。ただし、以前のリリースからの既存の Flash Cookie が存在する場合、それらの Cookie は、以下のいずれかのガイドの「デバイス ID および DeviceDNA の収集」で説明されているタスクを完了したときに、エンドユーザのデバイス上でサポートされるストアの 1 つに自動的に移行されます。

- CA Risk Authentication Java 開発者ガイド
- CA Risk Authentication Web サービス開発者ガイド



CA Risk Authentication は、データを保護するためにハードウェアセキュリティモジュール (HSM) をサポートしています。HSM を使用してデータを暗号化する場合、データベースに保存されているデータは HSM にあるキーを使用して暗号化されます。

**注:** このセクションで説明されている設定を行う前に、HSM サーバおよびクライアントをセットアップしていて、HSM 内に 3DES キーを生成していることを確認します。「(オプション、HSM を使用している場合のみ) HSM の要件」を参照してください。

CA Risk Authentication は、ソフトウェア (S/W) モードを使用してデータを暗号化します。そのため、モードをハードウェア (**chrysalis** または **nfast**) に変更する必要があります。arcotcommon.ini ファイルの [arcot/crypto/device] セクションを使用して、これを実行します。

また、このファイルには、必要な HSM を設定するための個別のセクションがあります。現在のリリースでは以下のとおりです。

- Luna HSM ([crypto/pkcs11modules/chrysalis])
- nCipher netHSM ([crypto/pkcs11modules/nfast])

設定している HSM に基づいて、対応するセクションで sharedLibrary パラメータを指定します。HSM 情報を指定したら、HSM キー ラベルを使用して securestore.enc ファイルを再作成し、HSM を初期化して、HSM キーを使用するように CA Risk Authentication を初期化します。

CA Risk Authentication インストーラは、インストール時に、この HSM 関連情報を指定するよう要求します。ただし、データ暗号化モードの変更や、CA Risk Authentication で必要なその他の HSM 情報の設定など、後から HSM 設定を変更する場合には、以下の手順に従います。

次の手順に従ってください：

1. 以下の場所に移動します。  
<install\_location>%Arcot System%conf%
2. securestore.enc のバックアップをとります。
3. <install\_location>%Arcot System%conf% から既存の securestore.enc ファイルを削除します。
4. ソフトウェア (S/W) からハードウェア (chrysalis または nfast) にデータ暗号化モードを変更して、CA Risk Authentication が必要とする HSM 情報を設定するには、以下の手順に従います。
  - a. 以下の場所に移動します。

<install\_location>%Arcot System%conf%

- b. テキストエディタで arcotcommon.ini を開きます。
  - c. [arcot/crypto/device] セクションで、以下の操作を実行します。
    - Luna HSM に対して、HSMDevice パラメータを chrysalis に設定します。または
    - nCipher netHSM に対して、HSMDevice パラメータを nfast に設定します。
  - d. 設定する HSM に応じて、sharedLibrary パラメータを HSM ライブラリファイルがある場所に設定します。
    - Luna HSM ライブラリのデフォルトの場所は、  
<SYSTEM\_DRIVE>:%Program Files%LunaSA%cryptoki.dll です。または
    - nCipher netHSM のデフォルトの場所は、  
<SYSTEM\_DRIVE>:%nfast%bin%cknfast.dll です。
- 注: このセクションで使用可能なその他の HSM 設定パラメータの詳細については、「arcotcommon.ini」を参照してください。
- e. arcotcommon.ini ファイルを保存して閉じます。

5. DBUtil ツールがある以下の場所に移動します。  
<install\_location>%Arcot System%tools%platform%
6. 以下のコマンドを使用して DBUtil ツールを実行します。

注: 以下のコマンドで指定するデータベース ユーザ (<Database\_Username>) では、大文字と小文字が区別されます。

- a. `dbutil -init <HSM_Key_Label>`

注: <HSM\_Key\_Label> は、HSM に存在する 3DES キーに対応します。

上記のコマンドは指定したキー ラベルで `securestore.enc` ファイルを作成します。生成されたファイルは、<install\_location>%Arcot System%conf% に保存されます。

- b. `dbutil -i <HSM_Module_Name> <HSM_Password>`

注: <HSM\_Module\_Name> は、Luna HSM の場合は `chrysalis`、nCIPHER netHSM の場合は `nfast` です。

上記のコマンドは HSM を初期化します。

- c. `dbutil -pi <DSN_Name> <Database_Password> -h <HSM_Password> -d <HSM_Module_Name>`

注: <DSN\_NAME> は、CA Risk Authentication データベースに接続するために CA Risk Authentication サーバが使用する ODBC DSN を指します。<Database\_Password> は、データベースに接続するために使用されるパスワードを指します。

上記のコマンドは、HSM を使用して暗号化されるように CA Risk Authentication サーバデータを初期化します。

- d. `dbutil -pi <Database_Username> <Database_Password> -h <HSM_Password> -d <HSM_Module_Name>`

注: <Database\_Username> は、CA Risk Authentication データベースに接続するために使用されるユーザ名を指します。

<Database\_Password> は、データベースに接続するために使用されるパスワードを指します。

上記のコマンドは、HSM を使用して暗号化されるように、管理コンソールおよびユーザ データ サービス データを初期化します。

## 第 19 章: カスタム アクションの追加

CA Risk Authentication の各チャンネルには、一連のアクションが関連付けられています。また、アクションにはデータ エレメントが関連付けられています。CA Risk Authentication のルールは、チャンネルまたはチャンネルのセットに対するアクションに関連付けられたエレメントの特定の組み合わせです。

このセクションでは、カスタム アクションを追加する手順について説明します。カスタム アクションの追加では、アクションを関連付ける必要があるチャンネルを指定します。そのチャンネルに対して定義された別のアクションに関連付けられたエレメントは、新しいアクションと自動的に関連付けられます。これらのエレメントを使用して、新しいアクションのルールを構築できます。

**注:** すべてのチャンネルで利用可能なアクション用のルールを構築する場合、まず各チャンネルにアクションを追加する必要があります。

以下の手順に従います。

1. (オプション) 新しいアクションを関連付けるチャンネルの名前が不明な場合は、以下の手順に従います。
  - a. GA として管理コンソールにログインします。
  - b. [サービスおよびサーバの設定] タブをクリックします。
  - c. サイドバーメニューの [ルール管理] セクションで、[ルールおよびスコアリング管理] リンクをクリックします。
  - d. [選択ルールセット] リストからルールセットを選択します。
  - e. [新しいルールの追加] をクリックします。
  - f. 新しいアクションを追加するチャンネルの名前を書き留めます。

2. 「データベース サーバの設定」にリストされているデータベース権限があることを確認します。

3. データベースにログインします。

4. 以下のコマンドを実行して、アクションを追加するチャンネルの ID を確認します。

```
select channelid from arrfchannel where  
channelname='<channel-name>;
```

このコマンドでは、<channel-name> をチャンネルの名前に置き換えます。

5. 以下のいずれかのコマンドを実行します。

**注:** 実行するコマンドで、`<channel-id>` を、前の手順で確認したチャンネル ID に置き換えます。同様に、`<action-name>` をチャンネルの名前に置き換えます。アクション名は、英数字およびアンダースコア文字を含むことができます。アクション名では、その他の文字は使用できません。

- MS SQL Server の場合  
`EXEC ADD_CUSTOM_ACTION <channel-id>, '<action-name>'`
- Oracle データベースの場合  
`set serveroutput on;`  
`execute ADD_CUSTOM_ACTION(<channel-id>, '<action-name>');`
- MySQL の場合  
`call ADD_CUSTOM_ACTION(<channel-id>, '<action-name>');`

6. キャッシュをリフレッシュします。手順については、「CA Risk Authentication 管理ガイド」を参照してください。
7. 以下を実行して、アクションが正常に追加されたことを確認します。
  - a. 管理コンソールにログインします。
  - b. [ルールおよびスコアリング管理] 画面に移動します。
  - c. [新しいルールの追加] をクリックします。
  - d. 新しく追加したアクションが [アクション] リストに表示されていることを確認します。

アクションが追加されたことを確認した後、それを使用して新しいルールの構築を開始できます。



# 付録 B: CA Risk Authentication のエラーのトラブルシューティング

この付録では、CA Risk Authentication の使用時に発生する可能性があるエラーを解決するのに役立つトラブルシューティング手順について説明します。トラブルシューティングトピックは、CA Risk Authentication の各コンポーネントに基づいて以下のように分類されています

- インストールエラー
- Database-Related エラー
- CA Risk Authentication サーバエラー
- SDK エラー
- アップグレードエラー

トラブルシューティングタスクを実行する前に、CA Risk Authentication のログファイルでエラーがあるかどうかを確認してください。デフォルトでは、ログファイルはすべて `<install_location>\Arcot Systems\logs\` ディレクトリに保存されます。以下の表に、CA Risk Authentication コンポーネントのデフォルトログファイル名を示します。

| CA Risk Authentication コンポーネント | ファイル名                           | Description  |
|--------------------------------|---------------------------------|--|
| CA Risk Authentication サーバ     | arcotriskfortstartup.log        | このファイルには、すべての起動（ブート）アクションが記録されます。CA Risk Authentication サービスが起動しない場合、問題の原因を特定するためにこのファイルの情報が役立ちます。 |
|                                | arcotriskfort.log               | このファイルには、CA Risk Authentication サーバによって、再起動後に処理されたすべてのリクエストが記録されます。                                |
| ケース管理サーバ                       | arcotriskfortcasemgmtserver.log | このファイルには、ケース管理に関するすべての起動（ブート）アクションが記録されます。ケース管理サービスが起動しない場合、問題の原因を特定するためにこのファイルの情報が非常に役立ちます。       |

| CA Risk Authentication コンポーネント | ファイル名                                  | Description                                       |
|--------------------------------|--|---|
|                                | arcotriskfortcasemgmtserverstartup.log | このファイルには、ケース管理サーバによって、再起動後に処理されたすべてのリクエストが記録されます。 |
| 管理コンソール                        | arcotadmin.log                         | このファイルには、管理コンソールの操作が記録されます。                       |
| ユーザ データ サービス                   | arcotuds.log                           | このファイルには、ユーザ データ サービス (UDS) の操作が記録されます。           |

注: これらのログ ファイルの詳細については、「CA Risk Authentication 管理ガイド」の付録「CA Risk Authentication のログ」を参照してください。



## インストール エラー

### 問題:

<install\_location>%Arcot Systems%java%webapps ディレクトリに arcotadmin.war がありません。

### 原因:

インストーラが、インストール中に arcotadmin.war WAR ファイルの作成に失敗しました。

### 解決方法:

ファイルが自動的に作成されなかった場合、手動で作成する必要があります。以下の手順を実行します。

1. コマンドプロンプトウィンドウを開きます。
2. ARCOT\_HOME 環境変数が設定されていることを確認します。
3. <install\_location>%Arcot Systems%tools%common%bundlemanager ディレクトリに移動します。
4. 以下のように bundlemanager を実行します。

```
java -jar bundle-manager.jar
```

上記のコマンドにより、<install\_location>%Arcot Systems%java%webapps ディレクトリに arcotadmin.war ファイルが生成されます。

### 問題:

RiskMinder サーバ (**Arcot RiskFort サービス**) を起動できません。  
arcotriskfortstartup.log に以下のエラーが表示されます。  
Failed DBPoolManager initialization

または  
Datasource Name Not Found

### 原因:

この問題の考えられる原因は以下のとおりです。

- データベースの DSN がシステム DSN として作成されていない。
- 64 ビットのプラットフォームを使用している。その結果、DSN が 64 ビットの ODBC Manager を使用して作成されている。

### 解決方法:

arcotcommon.ini で DSN 関連の問題を確認できます。問題が DSN 関連の場合、以下の手順に従います。

1. 最初の原因を解決するには、DSN がシステム DSN であることを確認する必要があります。以下の手順を実行します。
  - a. [コントロールパネル] を開き、[管理ツール] - [データ ソース (ODBC)] に移動します。
  - b. [システム DSN] タブをアクティブにし、該当 DSN が存在することを確認します。存在しない場合は、以前と同じ名前で DSN を再作成する必要があります。
  - c. サービスを再起動します。
2. 2 番目の原因 (64 ビットのプラットフォームを使用している場合) を解決するには、ODBC Manager の 32 ビットのバージョンを使用する必要があります。Windows の場合、C:¥Windows¥SysWOW64 に 32 ビットのバージョンがあります。

注: arcotcommon.ini およびその他の設定ファイルの詳細については、「CA RiskMinder インストールおよび展開ガイド」の「設定ファイルおよびオプション」を参照してください。

### 問題:

RiskMinder サーバ (**Arcot RiskFort** サービス) を起動できません。エラーメッセージは、サービスが起動し、自動的に停止していることを示しています。

### 原因:

この問題の考えられる原因としては、インストール時にデータベースの詳細を指定したが、データソースが正常に作成されなかった可能性があります。

### 解決方法:

この問題を解決するには、以下の手順に従います。

1. DSN の対応するエントリが arcotcommon.ini にあるかどうかを確認します。
  - エントリがない場合は、手動で DSN を作成します。

- エントリがある場合は、「データベース スクリプトの実行」の説明に従って、データベースをクリーンアップし（「RiskFort スキーマの削除」を参照）、データベースを再シードします。

2. RiskMinder サーバを再起動します。

#### 問題:

マスタ管理者として管理コンソールを初めて起動した際（「管理コンソールへのログイン方法」）に、以下のメッセージが表示されます。

```
The server encountered an internal error that prevented it from fulfilling this request.
```

arcotadmin.log ファイルに以下のエラーが記録されています。

```
adminLog: java.lang.UnsatisfiedLinkError: no ArcotAccessKeyProvider in java.library.path
```

#### 原因:

JAVA ライブラリに、以下のいずれかのファイルへのパスが含まれていません。

- ArcotAccessKeyProvider.dll
- arcot-crypto-util.jar

#### 解決方法:

以下の手順を実行します。

1. PATH 変数に以下のファイルへの絶対パスが含まれていることを確認します。
  - ArcotAccessKeyProvider.dll
  - arcot-crypto-util.jar
2. アプリケーションサーバを再起動します。

#### 問題:

ARCOT\_HOME のログ ディレクトリにログ ファイル（arcotadmin.log、arcotuds.log、casemanagementserver.log、または riskfortserver.log）がありません。

#### 原因:

この問題の考えられる原因は以下のとおりです。

- ARCOT\_HOME がインストール時に正しく設定されていない。
- アプリケーションサーバの JAVA ホームが、JDK ホームではなく JRE を指している。

### 解決方法:

これらの問題を解決するには、以下の手順に従います。

- ARCOT\_HOME をリセットして、正しい場所を指すように設定されていることを確認します。通常は、<installation\_location>%Arcot Systems% を指します。

この結果として、コマンドプロンプトウィンドウで  
cd %ARCOT\_HOME% コマンドを使用する際には、現在のディレクトリを <installation\_location>%Arcot Systems% に変更する必要があります。

- アプリケーションサーバの JAVA HOME の場所に ArcotAccessKeyProvider.dll ファイルおよび arcot-crypto-util.jar ファイルをコピーしたことを確認します。

### 問題:

UDS WAR (arcotuds.war) を展開しましたが、UDS が起動しません。

### 原因:

考えられる原因の 1 つは、アプリケーションサーバの JAVA ホームが JDK ホームではなく JRE を指しているというものです。

### 解決方法:

この問題を解決するには、以下の手順に従います。

- アプリケーションサーバの JAVA ホームの場所に ArcotAccessKeyProvider.dll ファイルおよび arcot-crypto-util.jar ファイルをコピーしたことを確認します。

## データベース関連のエラー

### 問題:

RiskMinder データベース スクリプトを実行しましたが、エラーは表示されませんでした。しかし、RiskMinder テーブルにアクセスしようとする、テーブルが存在しないというエラー メッセージが表示されます。

または

Oracle データベースを使用しています。テーブルにアクセスしようとする、以下のエラーが表示されます。

```
ERROR : common.database.DBF0ManagerImpl(65) : Failed to retrieve Database error codes for Datasource[1]. Error: ORA-00942: table or view does not exist.
```

### 原因:

- スクリプトの実行に使用したデータベース ユーザに、必要なファイル権限がありません。
- 指定された順序でデータベース スクリプトを実行していません。

### 解決方法:

以下の手順を実行します。

1. ユーザに正しいファイル権限があることを確認します。
2. データベースをクリーンアップします。

詳細な手順については、「RiskMinder スキーマの削除」を参照してください。

3. 正しい順序で、データベース スクリプトを再度実行します。

単一システム インストールでの詳細については、「データベース スクリプトの実行」を参照してください。

分散システム インストールでの詳細については、「データベース スクリプトの実行」を参照してください。

4. データベースが正しくシードされたかどうかを確認するには、以下のクエリを実行します。

```
SELECT SERVERNAME, VERSION FROM ARRFSEVERES;
```

### 問題:

Oracle データベースへの接続に失敗し、RiskMinder サーバ ログ ファイルに以下のエントリが記録されます。

```
ReportError: SQL Error State:08001, Native Error Code: 30FD, ODBC  
Error: [DataDirect][ODBC Oracle driver][Oracle]ORA-12541: TNS:no  
listener
```

### 解決方法:

以下の点をチェックする。

- データベース サーバのリスナ サービス。
- RiskMinder サーバがインストールされているシステムの TNSnames.ora ファイルの設定。

### 問題:

Oracle データベースへの接続に失敗し、RiskMinder サーバ ログ ファイルに以下のエントリが記録されます。

```
TNS:listener could not resolve SERVICE_NAME given in connect  
descriptor
```

### 解決方法:

以下を確認します。

- データベースが起動している。起動していない場合、上記のメッセージが表示されます。
- データベースが起動している場合は、データベースがまだリスナに登録されていない可能性があります。これは、データベースまたはリスナの起動直後に発生します。通常、この問題は 1 分程度待てば解決します。
- 静的な登録を使用している場合は、接続文字列 (TNSNAMES.ORA、NAMES、OID など) で使用される SERVICE\_NAME エントリが、リスナが認識している有効なサービスと一致することを確認します。
- C:>tnsping SERVICE\_NAME を使用して、ステータスを確認したり、C:>lsnrctl services を使用して、リスナが認識しているすべてのサービスを確認したりできます。

### 問題:

Oracle データベースへの接続に失敗し、RiskMinder サーバ ログ ファイルに以下のエントリが記録されます。

```
ORA-03113: end-of-file on communication channel
```

**原因:**

これは接続が失われたことを示す一般的なエラーです。以下のようなさまざまな原因で発生します。

- ネットワークの問題
- サーバセッションの強制切断
- Oracle データベースのクラッシュ
- データベース サーバのクラッシュ
- 中断を引き起こす Oracle の内部エラー (ORA-00600 や ORA-07445 など)
- Oracle クライアントまたは TNS レイヤで接続を処理できない

**解決方法:**

上記のリストにある考えられる原因を確認します。

**問題:**

データベースへの接続に失敗し、RiskMinder サーバ ログ ファイルに以下のエントリが記録されます。  
Database password could not be obtained from securestore.enc

**原因:**

データベースの詳細が `securestore.enc` ファイルに含まれていない可能性があります。

**解決方法:**

DBUtil ツールを使用して、データベースの詳細を指定して `securestore.enc` ファイルを更新します。

注: DBUtil の使用方法の詳細については、「*CA RiskMinder 管理ガイド*」を参照してください。

**問題:**

MSSQL データベースへの接続が、以下のエラーで失敗します。  
java.sql.SQLException: No Datasource is set.

**原因:**

この問題の理由として、必要な JDBC JAR ファイルがコピーされないか、または使用しているアプリケーション サーバ上の正しい場所にコピーされていないことが考えられます。

その理由は、管理コンソール、ユーザ データ サービス (UDS) 、およびサンプルアプリケーションは、RiskMinder の Java 依存コンポーネントであり、データベースに接続するために JDBC (Java Database Connectivity) ファイルおよび JAR (Java Archive) ファイルが必要なためです。

### 解決方法:

以下の手順を実行します。

1. 必要に応じて、使用しているデータベース用の JDBC JAR ファイルをダウンロードします。
  - **Oracle データベースの場合** : ojdbc14.jar (バージョン 10.2.0.1.0)
  - **Microsoft SQL Server データベースの場合** : sqljdbc.jar (バージョン 1.2.2828)
  - **MySQL の場合** : mysql-connector-java-5.1.22-bin.jar (バージョン 5.1.22)
2. JDBC JAR をコピーまたは展開します。
  - **Apache Tomcat** を使用している場合は、「手順 3 : アプリケーション サーバへの JDBC JAR ファイルのコピー」の「Apache Tomcat」を参照してください。
  - **IBM WebSphere** を使用している場合は、「手順 3 : アプリケーション サーバへの JDBC JAR ファイルのコピー」の「IBM WebSphere」を参照してください。
  - **Oracle WebLogic** を使用している場合は、「手順 3 : アプリケーション サーバへの JDBC JAR ファイルのコピー」の「Oracle WebLogic」を参照してください。
  - **JBoss アプリケーション サーバ** を使用している場合は、「手順 3 : アプリケーション サーバへの JDBC JAR ファイルのコピー」の「JBoss アプリケーション サーバ」を参照してください。



## RiskMinder サーバのエラー

### 問題:

RiskMinder サーバを再起動しようとしても、起動しません。  
arcotriskfortstartup.log の最後の行に以下のエラーが表示されています。  
Cannot continue due to ARRF\_LIB\_init failure, SHUTTING DOWN

### 原因:

考えられる原因は、`$$rulelibname$$`.dll を必要とするルールを設定したものの、この DLL が `%ARCOT_HOME%\plugins\rules¥` ディレクトリに存在しないことです。

### 解決方法:

以下の手順を実行します。

1. 以下の文字列を検索します。  
`Couldn't find symbol [$$RULENAME$$] in library [$$rulelibname$$]`
2. この文字列が見つかった場合、（対応する）`$$rulelibname$$`.dll を `%ARCOT_HOME%\plugins\rules¥` ディレクトリにコピーします。
3. 以下の文字列を検索します。  
`"Couldn't get function pointer for symbol [ARRF_AddOnRule] in lib [$$rulelibname$$]`
4. この文字列が見つかった場合、（対応する）`$$rulelibname$$`.dll を `%ARCOT_HOME%\plugins\rules¥` ディレクトリにコピーします。
5. ログにこれらの文字列がない場合、ログファイル内で **ERROR** または **WARNING** メッセージを確認することを強く推奨します。この問題をデバッグするための十分な情報が見つかるはずですが。

### 問題:

RiskMinder サーバを再起動しようとしても、起動しません。  
arcotriskfortstartup.log の最後の行に以下のエラーが表示されています。  
"Transport Exception on Admin channels: bind: Address already in use"  
"Cannot continue due to loadAdminProtocolsAndAddTranports failure, SHUTTING DOWN"

### 原因:

この問題の考えられる原因は、サーバ管理ポート（デフォルトのポート番号：7980）が別のプロセスによってすでにホストで開かれているというものです。一方、RiskMinder サーバは、起動するために最低限のサーバ管理ポートを必要とします。

### 解決方法:

以下の手順を実行します。

1. コマンドプロンプト ウィンドウを開きます。
2. %ARCOT\_HOME% に移動します。
3. 以下のように、デバッグ モードで RiskMinder サーバを起動します。

```
arrrserver.exe -debug -port <new_port>
```

サーバ管理ポートを開くと、マスタ管理者は管理コンソールにログインして、別のポートを設定できます。

## SDK エラー

### 問題:

新しい RiskMinder 設定を行い、この新しい設定を使用する Java API を呼び出そうとすると、以下のエラーが表示されます。  
設定が見つかりません

### 原因:

RiskMinder サーバを再起動しなかった可能性があります。

### 解決方法:

新しい設定を使用するには、RiskMinder サーバを再起動する必要があります。

# 第 20 章: CA Risk Authentication をアンインストールする方法

---

Windows のコントロールパネルから CA Risk Authentication をアンインストールするか、またはアンインストーラ ファイル (*Uninstall CA Risk Authentication.exe*) を実行して CA Risk Authentication をシステムから削除できます。

アンインストールプロセスを完了したら、システムに残っている WAR ファイルや入力内容をクリーンアップするため、アンインストール後のタスクを実行します。

**重要:** CA Risk Authentication をアンインストールするときは、最初にそのスキーマを削除してから、アンインストール処理を行います。

CA Strong Authentication と CA Risk Authentication の両方をインストールした後に CA Risk Authentication のみをアンインストールする場合は、以下のセクションに記載されているガイドラインに従います。これらのガイドラインに従うことによって、CA Strong Authentication で使用されている共通コンポーネントが削除または変更されません。

このセクションでは、CA Risk Authentication とそのコンポーネントをアンインストールするための手順について説明します。

1. [CA Risk Authentication スキーマの削除](#) (P. 204)
2. [CA Risk Authentication サーバのアンインストール](#) (P. 206)
3. [アンインストール後のタスクの実行](#) (P. 207)

## CA Risk Authentication スキーマの削除

CA Risk Authentication のみをアンインストールする場合は、以下のいずれかの手順を実行します。

- CA Risk Authentication をインストールしてから CA Strong Authentication をインストールした場合は、CA Strong Authentication スキーマを削除してから CA Risk Authentication スキーマを削除します。
- CA Strong Authentication をインストールしてから CA Risk Authentication をインストールした場合は、CA Risk Authentication スキーマのみを削除します。

注: 両方の製品をアンインストールする場合は、任意の順序で両方のスキーマを削除できます。

次の手順に従ってください：

1. 以下のディレクトリに移動します。  
`<install_location>%Arcot Systems%dbscripts%`
2. 使用しているデータベースに応じて、以下のいずれかのサブディレクトリに移動します。
  - Oracle の場合  
`<install_location>%Arcot Systems%dbscripts%oracle%`
  - Microsoft SQL Server の場合  
`<install_location>%Arcot Systems%dbscripts%mssql%`
  - MySQL の場合  
`<install_location>%Arcot Systems%dbscripts%mysql%`

3. CA Risk Authentication と関連コンポーネントのすべてのデータベーステーブルを削除するには、以下の順序でスクリプトを実行します。
  - a. `drop-riskauth-8.0.sql` を実行します。
  - b. 必要に応じて、`drop-riskauth-3dsecure-8.0.sql` を実行します。
  - c. インストールしていない場合は、`drop-arcot-common-8.0.sql` を実行します。

**注:** `drop-arcot-common-8.0.sql` スクリプトは、共通コンポーネント用のスキーマを削除するために使用されます。このスキーマは、CA Strong Authentication と CA Risk Authentication の両方によって使用されます。CA Strong Authentication がすでにインストールされている場合、このスキーマは CA Strong Authentication で引き続き使用するため、削除しないでください。

4. CA Risk Authentication スキーマに対して作成したデータベース ユーザアカウントを今後使用することがない場合は、そのユーザアカウントを削除します。

## CA Risk Authentication サーバのアンインストール

CA Risk Authentication をアンインストールすると、データベースをクリアするために必要なファイルおよびスクリプトがすべて削除されます。CA Risk Authentication データベースを削除する必要がある場合は、先に進む前に「CA Risk Authentication スキーマの削除 (P. 204)」を参照してください。

**重要:** CA Risk Authentication の後に CA Strong Authentication をインストールした場合は、CA Strong Authentication サーバをアンインストールしてから CA Risk Authentication サーバをアンインストールする必要があります。

次の手順に従ってください：

1. 以下のサーバをシャットダウンします。
  - a. CA Risk Authentication サーバ
  - b. ケース管理キュー サーバ
  - c. CA Risk Authentication のその他のコンポーネントが展開されているすべてのアプリケーションサーバ
2. 管理コンソールを閉じます。
3. INI ファイルおよびその他の CA Risk Authentication 設定ファイルがすべて閉じられていることを確認します。
4. [スタート] - [設定] - [コントロールパネル] - [プログラムの追加と削除] の順にクリックして、[プログラムの追加と削除] ウィンドウを開きます。
5. CA Risk Authentication を選択して [変更と削除] をクリックします。

**注:** <install\_location>%Arcot Systems%Uninstall CA Risk

Authentication%ディレクトリ内の Uninstall CA Risk Authentication.exe を実行して、CA Risk Authentication をアンインストールすることもできます。

6. 以下のいずれかのオプションを選択します。
  - CA Risk Authentication のすべてのコンポーネントをアンインストールするには、[Complete Uninstall] を選択し、手順 8 に進みます。

**注:** アンインストールプロセスが完了するまで数分かかる場合があります。
  - 選択したコンポーネントをアンインストールするには、[Uninstall Specific Features] を選択します。[次へ] をクリックします。

手順 7 に進みます。

- アンインストールするコンポーネントを選択し、[Uninstall] をクリックします。

**重要:** 特定の機能をアンインストールするには、コンポーネントをインストールしたのとは逆の順序で行います。たとえば、CA Risk Authentication サーバの後に管理コンソールをインストールした場合は、管理コンソールをアンインストールしてから CA Risk Authentication サーバをアンインストールします。

- [Done] をクリックして、処理を完了します。

## アンインストール後のタスクの実行

すべての CA Risk Authentication コンポーネントが削除されていることを確認するために実行する必要があるアンインストール後の手順は以下のとおりです。

次の手順に従ってください：

- <install\_location>¥Arcot Systems¥ ディレクトリを削除します。

注: 複数の CA 製品がインストールされているシステムでは、アンインストールする最後の製品である場合にこのディレクトリを削除します。

- アプリケーションサーバを停止します。

- <APP-SERVER-HOME> 内の適切なサブディレクトリから以下の WAR ファイルをアンインストールします。

注: ここで、APP-SERVER-HOME は、アプリケーションサーバ (Apache Tomcat など) がインストールされているディレクトリパスを表します。

WAR ファイルのアンインストールの詳細については、アプリケーションサーバベンダーのドキュメントを参照してください。

- arcotadmin.war : 管理コンソール
- arcotuds.war : ユーザデータサービス (展開されている場合)
- riskauth-8.0-sample-application.war : サンプルアプリケーション
- riskauth-8.0-sample-callouts.war : サンプルコールアウト

注: 分散システムに展開している場合は、該当するアプリケーションを展開したシステムでこれらのファイルを探してください。

4. データベースとして Oracle データベースを使用していた場合は、CA Risk Authentication データベースを実行しているシステムから `tabspace_arreports_<time_database_was_created>.dat` ファイルを削除します。
5. DSN エントリを削除します。

このエントリを削除するには、コントロールパネルを開きます。 [管理ツール] - [データ ソース (ODBC) ] - [システム DSN] に移動して、必要な DSN を選択します。

[削除] をクリックします。