CA Risk Authentication

インストール ガイド
（UNIX プラットフォーム用）

8.0

ca technologies

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

第 1 章: はじめに

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、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。

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。

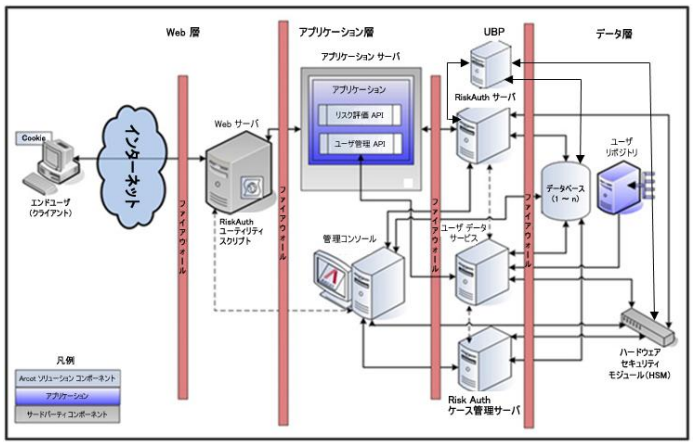
このガイドでは、さまざまなソリューション要件に基づく Risk Authentication の展開の計画について説明します。 各シナリオは複数のコンポーネントで構成され、これらのコンポーネントが相互に、および企業内のほかのシステムや複数のネットワークで形成されるシステムと通信します。

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

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. 13）
- アプリケーション層（P. 14）
- データ層（P. 15）



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、「*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（Risk Authentication サーバ、UDS、管理コンソール、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。

- **Risk Authentication サーバ：** このサーバ コンポーネントは、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、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 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、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、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、Risk Authentication Java API の使用方法およびアプリケーションと Risk Authentication の統合方法の例を示します。また、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。

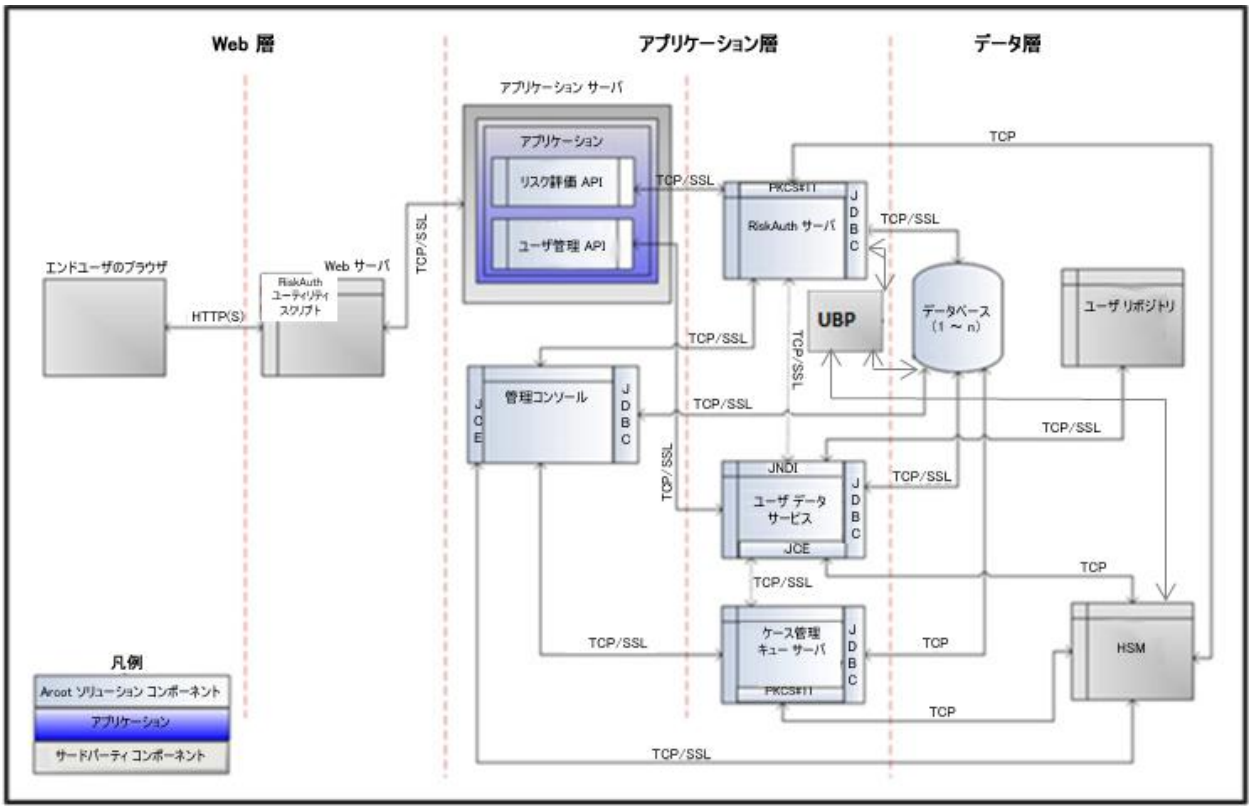
ストレージ用のデータ層

この層は、各トランザクションを分析するために 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。

Risk Authentication コンポーネント間の通信

以下の図は、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。 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（双方向および一方向通信）をサポートしています。

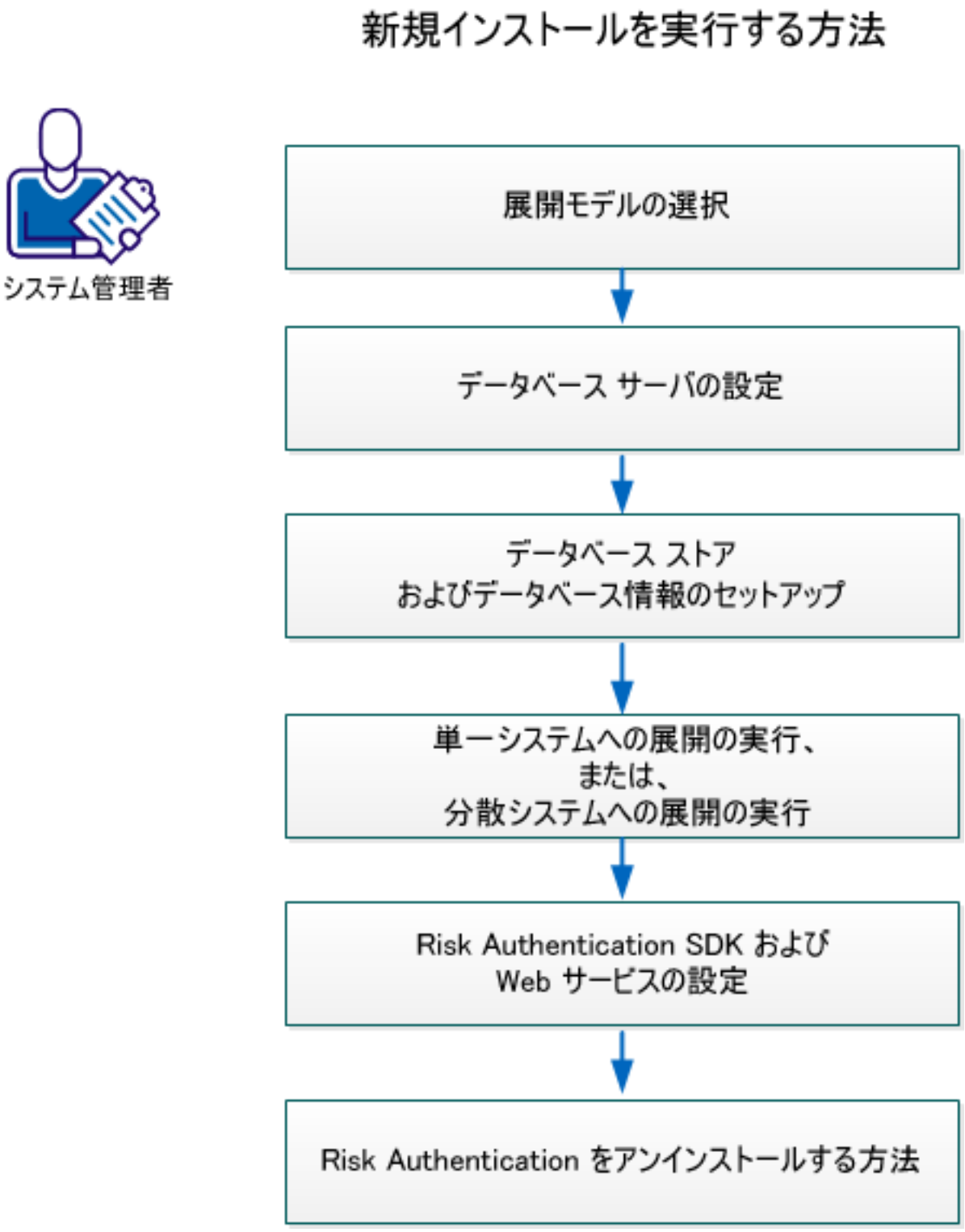
- ケース管理キュー サーバ
- Risk Authentication データベース
- ユーザ データ サービス
- Risk Authentication SDK （リスク評価）
- サンプル アプリケーション
- 評価コールアウト
- スコアリング コールアウト

注: 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。 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、「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 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、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
展開モデルの選択
データベース サーバの設定
データベース ストア
およびデータベース情報のセットアップ
単一システムへの展開の実行、
または、
分散システムへの展開の実行
Risk Authentication SDK および
Web サービスの設定
Risk Authentication をアンインストールする方法

以下の手順に従います。

1. 展開モデルを選択します。 (P. 23)
2. データベース サーバを設定します。 (P. 34)
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. 41)
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. Risk Authentication SDK および Web サービスを設定 (P. 199) します。

展開モデルの選択

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。 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 Risk Authentication サーバに統合できます。

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、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、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。

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、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、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。

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、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、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、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. 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。

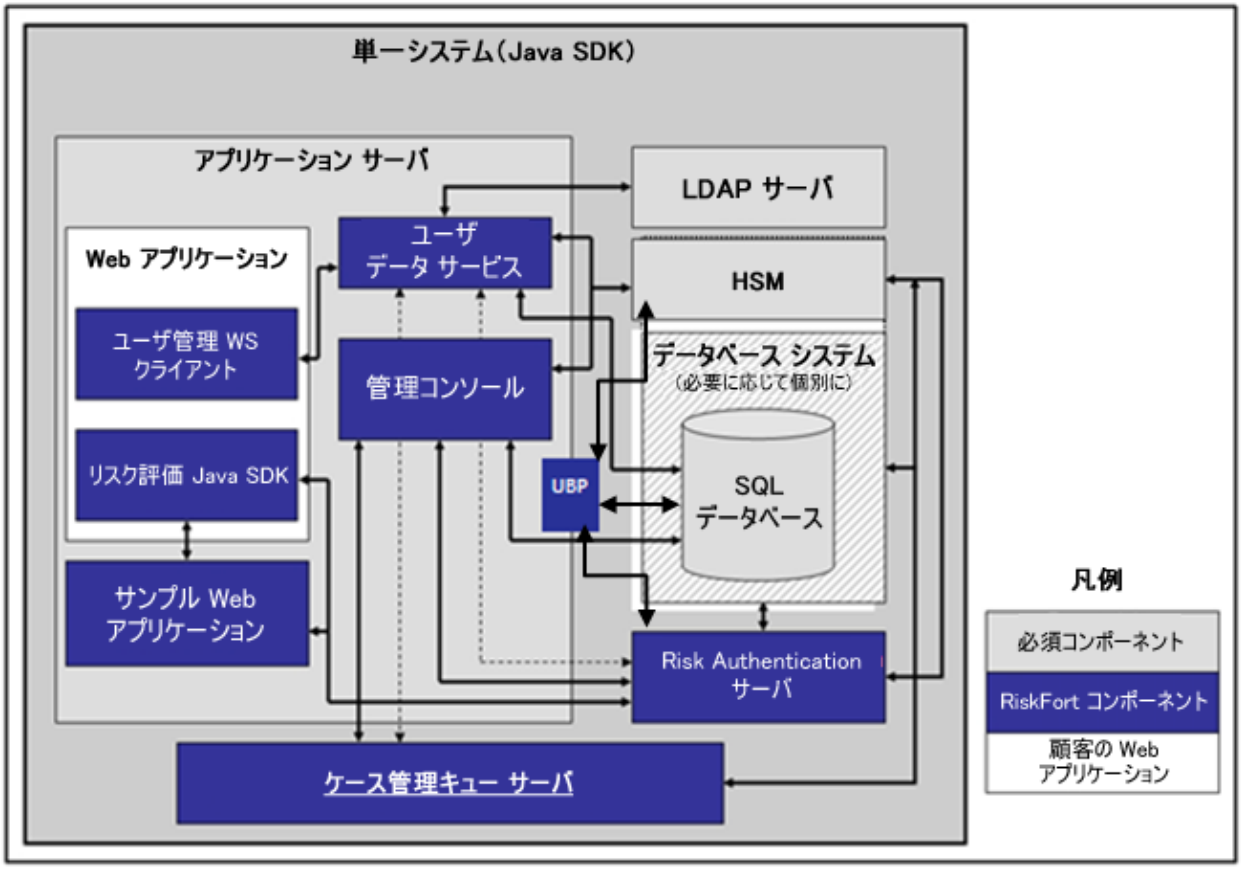
b. サンプル アプリケーションを使用するか、独自の Web アプリケーションを作成します。

   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 Risk Authentication サーバおよび Java SDK を示しています。

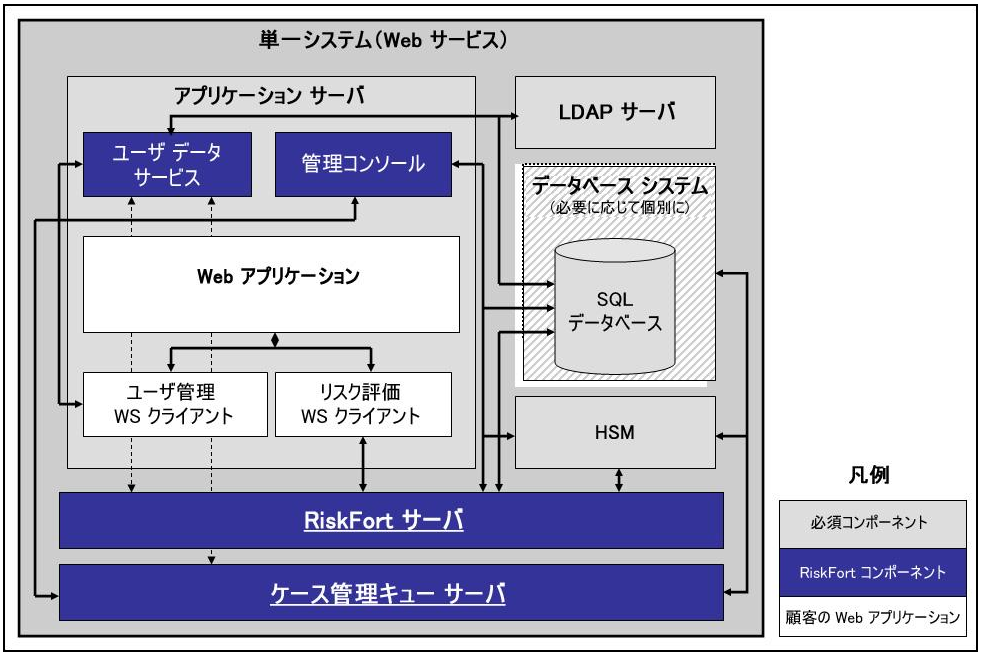

単一システム(Java SDK)
アプリケーション サーバ
Web アプリケーション
ユーザ管理 WS クライアント
リスク評価 Java SDK
サンプル Web アプリケーション
ユーザ データ サービス
管理コンソール
UBP
LDAP サーバ
HSM
データベース システム
(必要に応じて個別に)
SQL データベース
Risk Authentication サーバ
ケース管理キュー サーバ
凡例
必須コンポーネント
RiskFort コンポーネント
顧客の Web アプリケーション

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、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 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 Risk Authentication サーバ モジュール自体に組み込まれているので、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 Risk Authentication を展開する理由は、以下のとおりです。

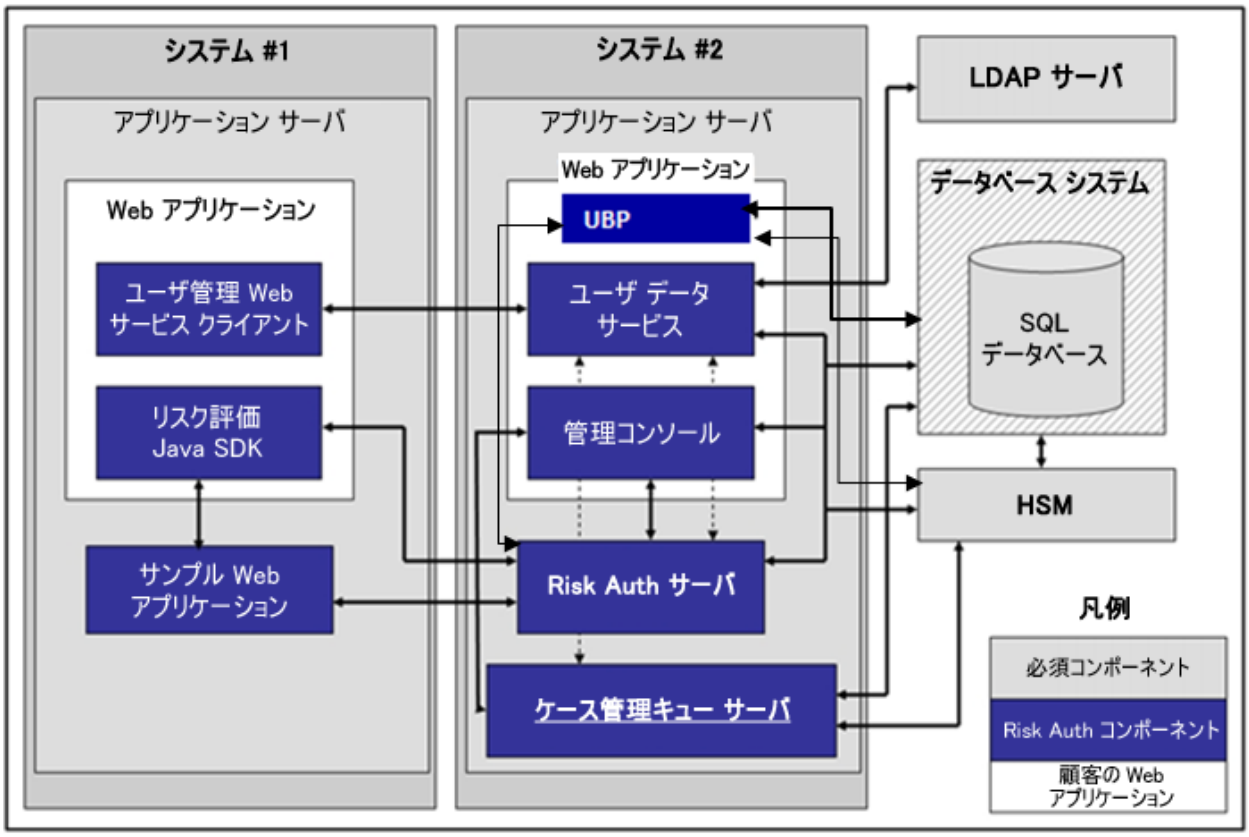
- 高可用性（フェールオーバとロード バランシング）
- 高パフォーマンス
- スループットの増加

分散システム展開では、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 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、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

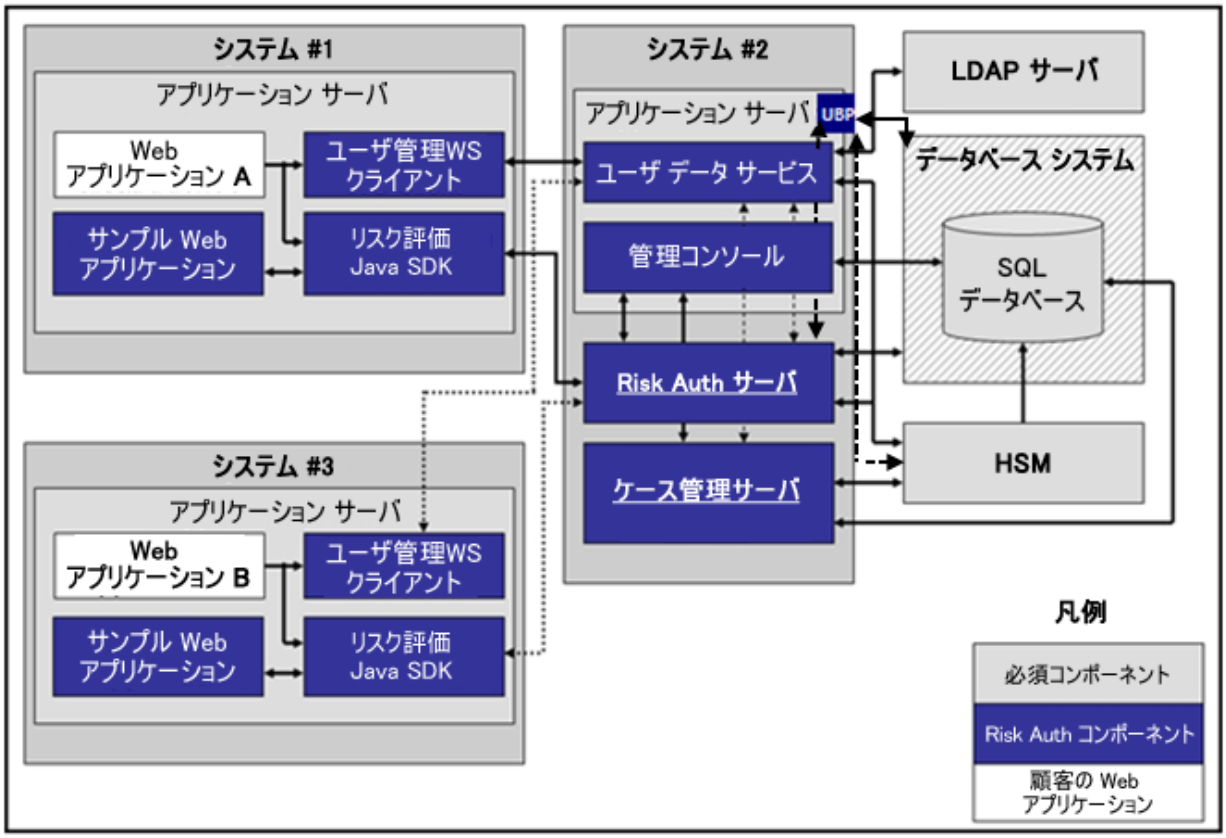
以下の図は、Java SDK を使用した単一アプリケーションへの Risk Authentication の展開を示しています。


システム #1
アプリケーション サーバ
Web アプリケーション
ユーザ管理 Web サービス クライアント
リスク評価 Java SDK
サンプル Web アプリケーション
システム #2
アプリケーション サーバ
Web アプリケーション
UBP
ユーザ データ サービス
管理コンソール
Risk Auth サーバ
ケース管理キュー サーバ
LDAP サーバ
データベース システム
SQL データベース
HSM
凡例
必須コンポーネント
Risk Auth コンポーネント
顧客の Web アプリケーション

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。

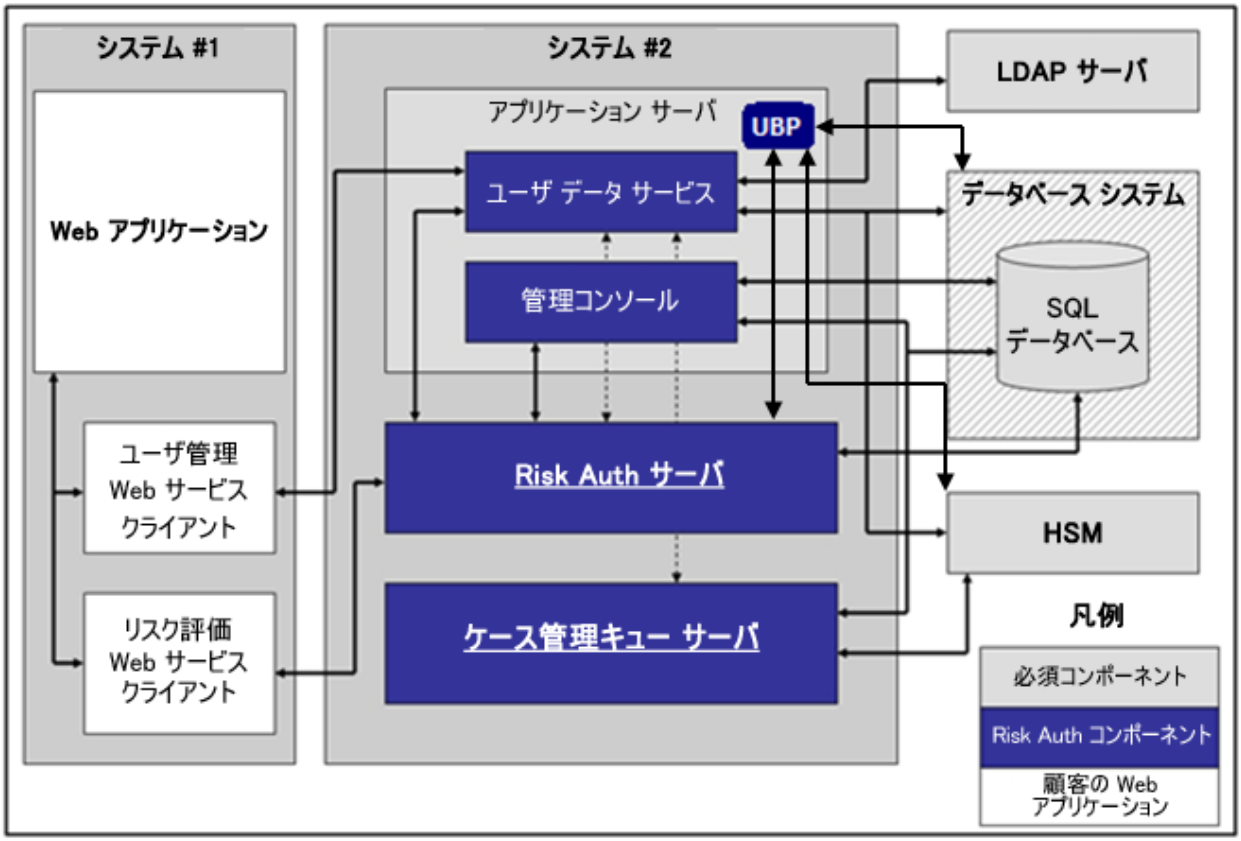
Java SDK を使用した複数アプリケーションの展開

以下の図は、Java SDK を使用した複数アプリケーションへの 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 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、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 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。

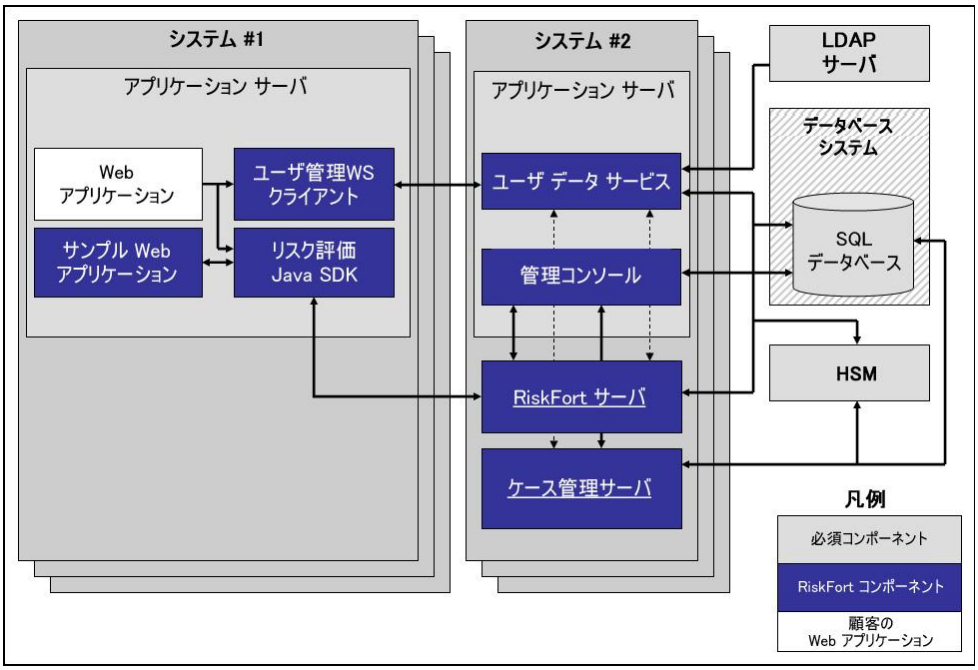
トランザクション レートが許容されるしきい値（組織のポリシーによって決定）を超えた場合、サーバ インスタンスを追加する必要があります。以下の Risk Authentication コンポーネントにより、ほとんどの場合、複数のインスタンスが機能できるようになります。

- **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。

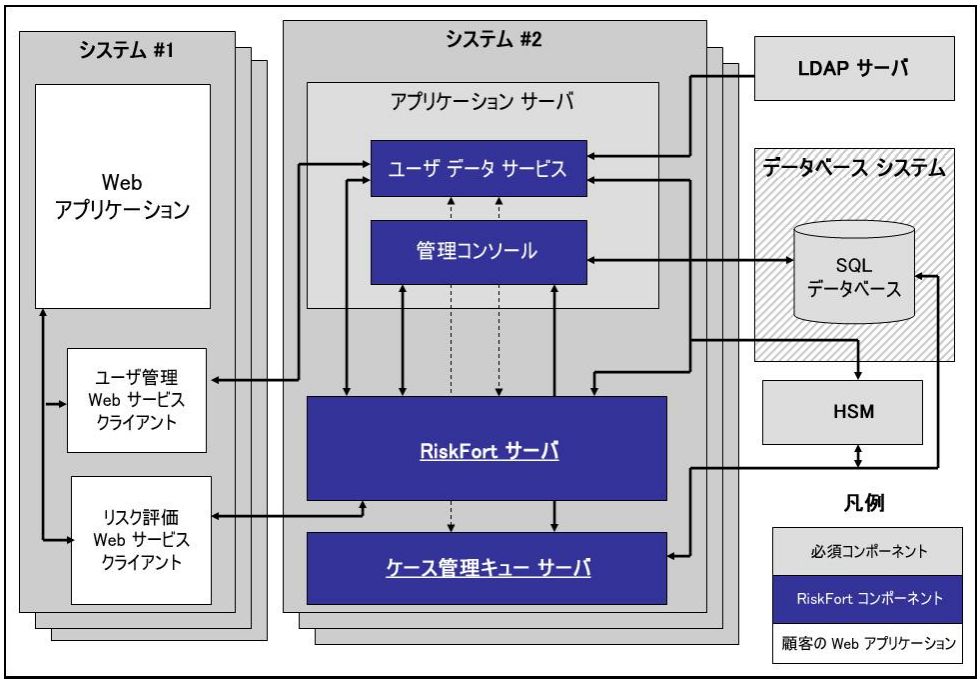
Java SDK を使用した高可用性展開

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

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. 34)
- 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. 41)

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。

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、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、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 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、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. 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。

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、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: 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。

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

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）。

   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

Risk Authentication のインストールに進む前に、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 Risk Authentication コンポーネントをインストールするシステム（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（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、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。

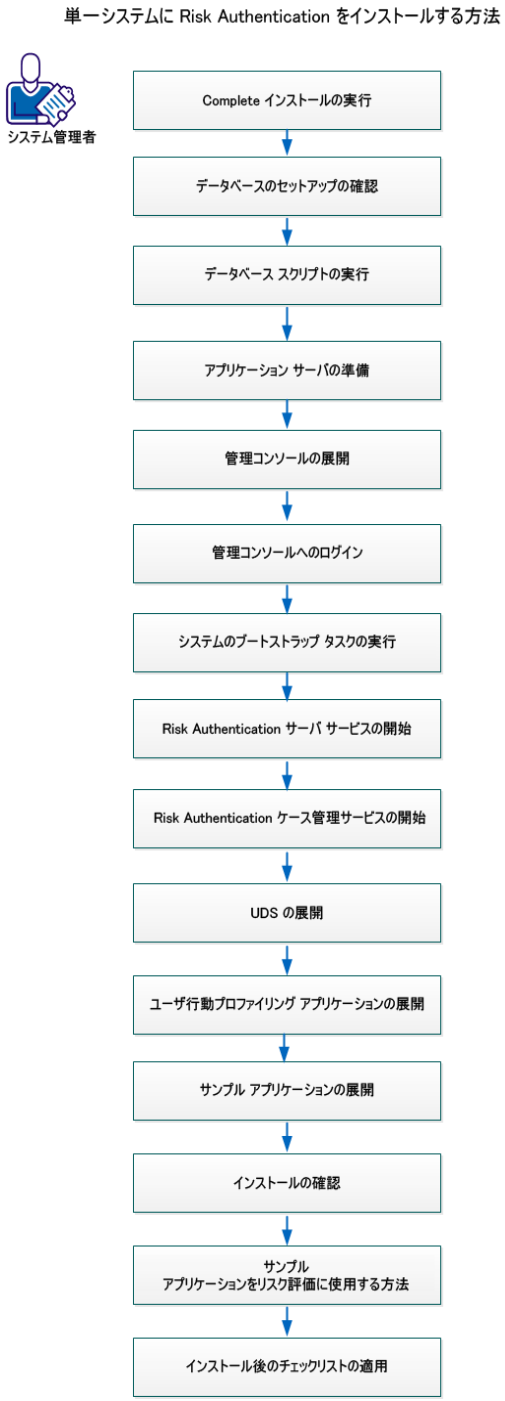
- Application Server

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

Risk Authentication コンポーネントをインストールするには、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 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、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。



以下のタスクを実行します。

1. Complete インストールの実行
2. データベースのセットアップの確認 (P. 59)
3. データベース スクリプトの実行 (P. 58)
4. アプリケーション サーバの準備 (P. 60)
5. 管理コンソールの展開
6. 管理コンソールへのログイン (P. 71)
7. システムのブートストラップ タスクの実行 (P. 72)
8. Risk Authentication サーバ サービスの開始
9. Risk Authentication ケース管理サービスの開始
10. UDS の展開
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. 78)
12. サンプル アプリケーションの展開 (P. 80)
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. 78)
14. インストールの確認
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. 82)
16. インストール後のチェックリストの適用 (P. 86)

重要:

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 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。
  - Strong Authentication は、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。 Strong Authentication をすでにインストールしており、Risk Authentication インストール手順を開始しようとしている場合、Risk Authentication インストーラは 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、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

Risk Authentication をインストールするには、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。

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、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. ログインし、インストーラを解凍したディレクトリに移動します。
2. インストーラを実行する権限があることを確認します。 ない場合は、以下のコマンドを実行します。
   - *（Solaris の場合）* chmod a=rx Risk Authentication-8.0-Solaris-Installer.bin
   - *（Linux の場合）* chmod a=rx Risk Authentication-8.0-Linux-Installer.bin
3. 以下のコマンドを入力した後に Enter キーを押して、インストーラを実行します。
   - *Solaris の場合：* prompt> sh Risk Authentication-8.0-Solaris-Installer.bin
   - *Linux の場合：* prompt> sh Risk Authentication-8.0-Linux-Installer.bin

     注: root ログインでインストーラを実行すると、警告メッセージが表示されます。 続行する場合は「**Y**」を入力し、インストールを終了する場合は「**N**」を入力します。インストーラ画面を終了していた場合は、インストーラを再度実行します。
4. ［次へ］をクリックします。
5. 使用許諾契約書の内容をよく読み、Enter キーを押して使用許諾契約書のテキストの次の画面を表示します。 Enter キーを複数回押す必要がある場合があります。

   使用許諾契約書に同意する場合は、「**Y**」を入力してインストールを続行します。

   注: 「**N**」を入力すると、警告メッセージが表示され、インストールが停止されます。

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。

6. ［次へ］をクリックします。
7. 以下のいずれかの手順に従って、インストール場所を選択します。
   - Risk Authentication をインストールするディレクトリの絶対パスを入力し、**Enter** キーを押して続行します。

     注: 指定するインストール ディレクトリ名にはスペースを含めないでください。 スペースを含めると、一部の Risk Authentication スクリプトとツールが想定どおりに機能しない場合があります。
   - **Enter** キーを押して、インストーラによって表示されたデフォルトのディレクトリを受け入れます。
8. *（既存の Advanced Authentication 製品がすでにインストールされているシステムにインストールする場合にのみ該当）*以下のいずれかのオプションを選択し、Enter キーを押します。
   - **1**： 新しいパスを入力する。
   - **2**： 既存の Advanced Authentication 製品がインストールされている場所を使用する。
9. 「**1**」と入力して、すべてのコンポーネントをインストールするデフォルト（Complete）インストールを選択し、Enter キーを押します。
10. 選択するデータベースに対応する番号（**1.** MS SQL Server **2.** Oracle データベース **3.** MySQL）を入力して、Enter キーを押します。
    - 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: 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。 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 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。

11. 以下の情報を入力して、Enter キーを押します。

- 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。 Risk Authentication サーバは、この DSN を使用して 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

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 とで異なっている必要があります。

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。 Risk Authentication サーバは、この DSN を使用して 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

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 とで異なっている必要があります。

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

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。 Risk Authentication サーバは、この DSN を使用して 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

  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

  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 とで異なっている必要があります。

  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

12. バックアップ データベース アクセスの設定で、以下のいずれかの手順を実行します。
    - 入力を求められたら、「**n**」を入力してセカンダリ DSN の設定をスキップし、Enter キーを押します。
    - 入力を求められたら、「**y**」を入力してセカンダリ DSN を設定し、Enter キーを押します。
13. 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。

    - **1.** Luna HSM
    - **2.** 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。

14. ［Pre-Installation Summary］画面の情報を確認し、Enter キーを押します。
15. Enter キーを押してインストールを開始します。 前の画面での設定を変更したい場合は、その画面に戻るまで［Back］をクリックします。必要な変更を行った後、Enter キーを押して続行します。
16. **Enter** キーを押します。インストーラは以下のタスクを実行するため、数分かかることがあります。
    - すべてのコンポーネントおよび関連するバイナリがインストールディレクトリにコピーされます。
    - データベース設定が arcotcommon.ini ファイルに格納され、パスワードが securestore.enc ファイルに格納されます。
    - 必要な INI ファイルへの書き込みが行われます。
    - 管理コンソール用の JNI_LIBRARY_PATH や、ODBC_HOME、ODBCINI、ORACLE_HOME、ORACLE_LIB_PATH などの環境変数を arrfenv ファイル内に設定します。
    - 前の画面で指定したとおり、odbc.ini ファイルで選択された ODBC ドライバを使用して、プライマリ DSN およびバックアップ DSN（選択および設定されている場合）が作成または上書きされます。

    上記のタスクが正常に完了すると、インストールは完了します。
17. **Enter** キーを押してインストーラを終了します。

    プロンプトが再度表示されるまで、（インストーラが一時ファイルをクリーンアップするため）数分間待機する必要がある場合があります。
18. UTF-8 サポートが有効になっていることを確認します。そのためには、以下の手順に従います。
    a. *<install_location>*/arcot/odbc32v70wf/odbc.ini ファイルに移動します。

b. [ODBC] セクションを見つけます。

c. IANAAppCodePage=106 エントリがこのセクションにあることを確認します。

d. このエントリがない場合は、追加します。

e. ファイルを保存して閉じます。

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（Arcot_RiskFort_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 /opt 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 /opt 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: Strong Authentication をインストール済みの場合は、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、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 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 ARRFSERVERS;
   ```

   上記のクエリの結果、以下の出力が表示されます。

   ```
   SERVERNAME                    VERSION
   ------------------------  ---------------
   Risk Authentication                    8.0
   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、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、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.so を以下のいずれかの場所にコピーします。
   - Solaris の場合： *Tomcat_JAVA_HOME*/jre/bin/
   - RHEL の場合： *Tomcat_JAVA_HOME*/jre/bin/
3. libArcotAccessKeyProvider.so ファイルがコピーされるディレクトリに LD_LIBRARY_PATH を設定しエクスポートします。
4. 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：*install_location*/arcot/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 ローダを設定します。
   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］を選択します。［Class Loader］を選択します。
   d. ［New］を選択します。
   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 ローダ ID を選択します。
   g. ［Shared Library References］を選択します。
   h. ［Add］を選択し、［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.so を以下のいずれかの場所にコピーします。
   - Solaris の場合： *WebSphere_JAVA_HOME*/jre/bin/
   - RHEL の場合： *WebSphere_JAVA_HOME*/jre/bin/

   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.so を以下のいずれかの場所にコピーします。
   - Solaris の場合： *WebLogic_JAVA_HOME*/jre/bin/
   - RHEL の場合： *WebLogic_JAVA_HOME*/jre/bin

   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］を選択します。 arcot-crypto-util.jar ファイルがあるディレクトリに移動します。
7. Enter キーを押します。
8. Enter キーを押して、［Summary］ページを表示します。
9. ［完了］を選択します。
10. 変更を有効にします。
11. libArcotAccessKeyProvider.so ファイルがコピーされるディレクトリに LD_LIBRARY_PATH を設定しエクスポートします。
12. 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. 以下に対して libArcotAccessKeyProvider.so をコピーします。
   - RHEL の場合： *JBoss_JAVA_HOME*/jre/bin/

   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>
```

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

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ので、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 <JBOSS_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. <JBOSS_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 Risk Authentication を管理するためには、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. 作業ディレクトリを、次のディレクトリに変更します。
2. <install_location>/arcot/sbin
3. 「source arrfenv」と入力し、Enter キーを押して Arcot 環境変数を設定します。
4. 変更を有効にするために、アプリケーション サーバを再起動します。
5. 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。
6. *(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。

7. アプリケーション サーバを再起動します。
8. 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 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。

     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。

Strong Authentication パスワード： ユーザがここで［*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: Strong Authentication のインストールと設定の詳細については、「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。

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. 以下のコマンドを実行します。
   `source <install_location>/arcot/sbin/arrfenv`

   このコマンドで、*<install_location>* を Risk Authentication がインストールされているディレクトリのパスに置き換えます。
2. 以下のディレクトリに移動します。

   `install_location/arcot/bin/`
3. 以下のコマンドを実行します。

   `./riskfortserver start`

   注: Risk Authentication サーバを停止する場合は、bin ディレクトリに移動し、「./riskfortserver stop」 コマンドを入力します。

Risk Authentication ケース管理サービスの開始

次の手順に従ってください：

1. 以下のコマンドを実行します。
   `source <install_location>/arcot/sbin/arrfenv`

   このコマンドで、*<install_location>* を Risk Authentication がインストールされているディレクトリのパスに置き換えます。

2. 以下のディレクトリに移動します。
   install_location/arcot/**bin**/

3. 以下のコマンドを実行します。

   ```
   ./casemanagementserver start
   ```

   注: ケース管理サーバを停止する場合は、bin ディレクトリに移動し、「./casemanagementserver stop」コマンドを入力します。

ユーザ データ サービス(UDS)の展開

Risk Authentication 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、Risk Authentication に対して、組織で展開されているサードパーティ 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. 作業ディレクトリを以下の場所に変更します。
   install_location/arcot/sbin/
2. 「source arrfenv」と入力し、Enter キーを押して必要な環境変数を設定します。
3. 以下の場所にある *arcotuds.war* をアプリケーション サーバ上に展開します。
   install_location/arcot/java/webapps/

   例： Apache Tomcat では、この WAR ファイルを APP_SERVER_HOME/webapps/ に展開します。

   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。
4. *(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. ［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 for application］を選択します。
   d. ［適用］をクリックします。
5. アプリケーション サーバを再起動します。
6. 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/**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。

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。 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。

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、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 Risk Authentication のワークフロー
- Risk Authentication API の基本操作（呼び出しと後処理）
- 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、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/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。
   - *Solaris の場合：* STARTING Risk Authentication 8.0
   - *RHEL の場合：* STARTING Risk Authentication 8.0

   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。
   - Solaris の場合： STARTING Risk Authentication Case Management 8.0
   - RHEL の場合： STARTING Risk Authentication Case Management 8.0

   Risk Authentication Case Management Service READY

注: 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。

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、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、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>/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 Risk Authentication サーバに提供します。 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、「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、「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

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	
システム情報		
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	Risk Authentication サーバ 管理コンソール ユーザ データ サービス	
管理コンソール情報		
ホスト名	localhost	
ポート	8080	
マスタ管理者パスワード	mypassword1234!	
ユーザ データ サービス情報		
ホスト名	localhost	
ポート	8080	
アプリケーション コンテキスト ルート	arcotuds	

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

Risk Authentication コンポーネントのインストールは、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 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、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
システム管理者
1 つ目のシステムへのインストール
データベース スクリプトの実行
アプリケーション サーバを準備する方法
管理コンソールの展開
管理コンソールへのログイン
ブートストラップ タスクの実行
Risk Authentication サーバ サービスの開始
Risk Authentication ケース管理サービスの開始
Risk Authentication サーバのインストールの確認
ユーザ データ サービスの展開
ユーザ行動プロファイリング アプリケーションの展開
2 つ目のシステムへの
Risk Authentication のインストール
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
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
サンプル アプリケーションをリスク評価操作に使用
インストール後のチェックリストの適用

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
2. データベース スクリプトの実行 (P. 103)
3. アプリケーション サーバを準備する方法 (P. 104)
4. 管理コンソールの展開
5. 管理コンソールへのログイン
6. ブートストラップ タスクの実行 (P. 117)
7. Risk Authentication サーバ サービスの開始
8. Risk Authentication ケース管理サービスの開始
9. Risk Authentication サーバのインストールの確認
10. ユーザ データ サービスの展開
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. 78)
12. 2 つ目のシステムへの Risk Authentication のインストール
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. 139)
14. 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. 140)
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. 141)
16. インストール後のチェックリストの適用 (P. 86)

重要:

単一のシステムまたは分散環境に 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 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. 92)
データベース スクリプトの実行 (P. 103)
アプリケーション サーバを準備する方法 (P. 104)

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 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. ログインし、インストーラを解凍したディレクトリに移動します。
2. インストーラの実行に必要な権限があることを確認します。 ない場合は、以下のコマンドを実行します。
   - Linux の場合： chmod a=rx Risk Authentication-8.0-Linux-Installer.bin
3. 以下のコマンドを入力し、Enter キーを押すことによりインストーラを実行します。
   - Linux の場合：

     ```
     prompt> sh Arcot-RiskFort-8.0-Linux-Installer.bin
     ```

     注: root ログインでインストーラを実行している場合は、警告メッセージが表示されます。 続行する場合は「Y」を入力し、インストールを終了する場合は「N」を入力します。 インストーラ画面を終了した場合は、再度インストーラを実行します。
4. Enter キーを押して、インストールを続行します。
5. 使用許諾契約書の内容をよく読みます。 「y」と入力して使用許諾契約書に同意し、Enter キーを押して次の手順へ進みます。

   注:「N」を入力すると、警告メッセージが表示され、インストールが中止されます。
6. ［Choose Installation Location］画面で以下の手順を実行します。
   - Risk Authentication をインストールするディレクトリの絶対パスを入力し、Enter キーを押して続行します。

注: 指定するインストール ディレクトリ名にはスペースを含めないでください。 スペースを含めると、Risk Authentication のスクリプトとツールの一部が想定どおりに機能しない場合があります。

- Enter キーを押して、インストーラによって表示されたデフォルトのディレクトリを受け入れます。

Risk Authentication でサポートされているインストール オプションが表示されます。

7. （既存の Advanced Authentication 製品がすでにインストールされているシステムにインストールする場合にのみ適用可能）インストーラに以下のオプションが表示されます。
   - **1** - 新しいパスを入力する。
   - **2** - 既存の CA Advanced Authentication 製品がインストールされている場所を使用する。
8. （既存の CA Advanced Authentication 製品がすでにインストールされているシステムにインストールする場合にのみ該当）必要なオプションを選択し、Enter キーを押してインストールを続行します。

   注: オプション 1 または 2 を選択した場合、指定した場所に arcot という新しいディレクトリが作成されます。
9. Customize インストール オプションを受け入れてインストールを続行する場合は、「2」を入力して Enter キーを押します。
10. インストールする Risk Authentication コンポーネントを表す番号をカンマ区切りリスト（カンマと番号の間にスペースを入れない）で指定し、Enter キーを押して続行します。

    コンポーネントに関する情報を以下の表に示します。

コンポーネント	説明

コンポーネント	説明
リスク評価サーバ	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（Risk Authentication サーバ）がインストールされます。 ■ リスク評価 ■ 設定 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。 ■ **リスク評価 Web サービス：** 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。 ■ **ユーザ管理 Web サービス**：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。 ■ **管理 Web サービス：** 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。
Risk Authentication ケース管理サーバ	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（Risk Authentication ケース管理サーバ）をインストールします。 **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、Risk Authentication ケース管理サーバの単一のインスタンスにのみ接続できます。
Risk Authentication SDK およびサンプル アプリケーション	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。 ■ **リスク評価 SDK：** 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。 ■ **サンプル アプリケーション：** Risk Authentication Java API の使用方法の例を示します。 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。 詳細については、「Risk Authentication SDK および Web サービスの設定」を参照してください。
管理コンソール	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。

コンポーネント	説明
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、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。
ユーザ行動プロファイリング	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。

例： 現在のシステムに Risk Authentication サーバ、Risk Authentication ケース管理キュー サーバ、および管理コンソールを（SDK およびサンプル アプリケーションなしで）インストールする場合は、以下を指定します。

1,2,4,5

注: この画面でサーバ コンポーネントがインストール対象として選択されていない場合、手順 11 から手順 16 の画面は表示されません。

Advanced Authentication 製品がすでにインストールされている場所にインストールする場合、インストーラは、インストールされている製品と同じデータベース設定を使用します。 そのため、手順 11 から手順 15 の画面は表示されません。

1. 選択するデータベースに対応する番号を指定し、Enter キーを押して続行します。
   - 1 - Microsoft SQL Server
   - 2 - Oracle データベース
   - 3 - MySQL

   注: 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。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 12）を実行してから、「Oracle RAC 用の Risk Authentication の設定」の手順を実行します。
2. 前の手順で 1 （SQL Server）を指定した場合は、以下の表に示されている情報を入力します。

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。 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。 **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。

パラメータ	Description
サーバ	Risk Authentication データストアのホスト名または IP アドレス。 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	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）。 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*。 **注**: ユーザ名はプライマリ DSN とバックアップ用 DSN とで異なっている*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。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。
データベース	MS SQL データベース インスタンスの名前。
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。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。

- 前の手順で 2（Oracle）を指定した場合は、以下の表に示されている情報を入力します。

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。 Risk Authentication サーバは、この DSN を使用して 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。 **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。
User Name	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）。 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*。 **注**: ユーザ名はプライマリ DSN とバックアップ用 DSN とで異なっている*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。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	Risk Authentication データストアのホスト名または IP アドレス。 ■ **構文**： *<server_name>* ■ **例**： demodatabase

- 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。 Risk Authentication サーバは、この DSN を使用して 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。 **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	Risk Authentication データストアのホスト名または IP アドレス。 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	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。 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*。 **注**: ユーザ名はプライマリ DSN とバックアップ用 DSN とで異なっている*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。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。
データベース	MySQL データベース インスタンスの名前。
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。 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。

1. バックアップ データベース アクセスの設定画面で、以下のいずれかの手順を実行します。
   - 入力を求められたら、「N」を入力してセカンダリ DSN の設定をスキップし、Enter キーを押して次の画面に進みます。
   - 入力を求められたら、「Y」を入力してセカンダリ DSN を設定し、Enter キーを押して続行します。

   実行されるタスクに関するデータベース固有の情報については、前の手順の表を参照してください。
2. Enter キーを押して続行します。
3. 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。

4. Enter キーを押します。
5. 表示された製品の詳細をよく確認し、Enter キーを押してインストールを続行します。

   インストーラがバックエンドで以下のタスクを実行するため、インストールに数分かかることがあります。

   - すべてのコンポーネントおよび関連するバイナリがインストールディレクトリにコピーされます。
   - データベース設定が *arcotcommon.ini* ファイルに格納され、パスワードが securestore.enc ファイルに格納されます。
   - 必要な INI ファイルに書き込みが行われます。
   - 管理コンソールの JNI_LIBRARY_PATH や、ODBC_HOME、ODBCINI、ORACLE_HOME、ORACLE_LIB_PATH などの環境変数を arrfenv ファイル内に設定します。
   - 前の画面で指定したとおり、odbc.ini ファイル内の選択済み ODBC ドライバを使用して、プライマリ DSN およびバックアップ用 DSN （選択され設定されている場合）を作成または上書きします。

   上記のタスクが正常に完了すると、［Installation Complete］画面が表示されます。
6. Enter キーを押してインストーラを終了します。
7. インストール ログ ファイル（Arcot_RiskFort_Install_<timestamp>.log）を確認します。これは、<install_location>/arcot/ ディレクトリにあります。
8. UTF-8 サポートが有効になっていることを確認するには、以下の手順に従います。

   a. <install_location>/arcot/odbc32v70wf/odbc.ini ファイルに移動します。

   b. [ODBC] セクションを見つけます。

   c. IANAAppCodePage=106 エントリがこのセクションにあることを確認します。

d. このエントリがない場合は、追加します。

e. ファイルを保存して閉じます。

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: 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、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 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 ARRFSERVERS;
```

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

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. 105）
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
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. 110）
4. Enterprise Archive ファイルの作成（P. 113）

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、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、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.so を以下のいずれかの場所にコピーします。
   - Solaris の場合： *Tomcat_JAVA_HOME*/jre/bin/
   - RHEL の場合： *Tomcat_JAVA_HOME*/jre/bin/
3. libArcotAccessKeyProvider.so ファイルがコピーされるディレクトリに LD_LIBRARY_PATH を設定しエクスポートします。
4. 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：*install_location*/arcot/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 ローダを設定します。
   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］を選択します。［Class Loader］を選択します。
   d. ［New］を選択します。
   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 ローダ ID を選択します。
   g. ［Shared Library References］を選択します。
   h. ［Add］を選択し、［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.so を以下のいずれかの場所にコピーします。
   - Solaris の場合： *WebSphere_JAVA_HOME*/jre/bin/
   - RHEL の場合： *WebSphere_JAVA_HOME*/jre/bin/

   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.so を以下のいずれかの場所にコピーします。
   - Solaris の場合： *WebLogic_JAVA_HOME*/jre/bin/
   - RHEL の場合： *WebLogic_JAVA_HOME*/jre/bin

   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］を選択します。 arcot-crypto-util.jar ファイルがあるディレクトリに移動します。
7. Enter キーを押します。
8. Enter キーを押して、［Summary］ページを表示します。
9. ［完了］を選択します。
10. 変更を有効にします。
11. libArcotAccessKeyProvider.so ファイルがコピーされるディレクトリに LD_LIBRARY_PATH を設定しエクスポートします。
12. 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. 以下に対して libArcotAccessKeyProvider.so をコピーします。
   - RHEL の場合： *JBoss_JAVA_HOME*/jre/bin/

   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>
```

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

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ので、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 <JBOSS_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. <JBOSS_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 Risk Authentication を管理するためには、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. 作業ディレクトリを、次のディレクトリに変更します。
2. <install_location>/arcot/sbin
3. 「source arrfenv」と入力し、Enter キーを押して Arcot 環境変数を設定します。
4. 変更を有効にするために、アプリケーション サーバを再起動します。
5. 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。
6. *(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。

7. アプリケーション サーバを再起動します。

8. 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!

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 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。 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*。 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 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、Strong Authentication をインストールします。

Strong Authentication のインストールおよび設定の詳細については、「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。

第 8 章: 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. 以下のコマンドを実行します。
   `source <install_location>/arcot/sbin/arrfenv`

   このコマンドで、*<install_location>* を Risk Authentication がインストールされているディレクトリのパスに置き換えます。

2. 以下のディレクトリに移動します。

   `install_location/arcot/bin/`

3. 以下のコマンドを実行します。

   `./riskfortserver start`

   注: Risk Authentication サーバを停止する場合は、bin ディレクトリに移動し、「./riskfortserver stop」コマンドを入力します。

第 9 章: Risk Authentication ケース管理サービスの開始

次の手順に従ってください：

1. 以下のコマンドを実行します。
   `source <install_location>/arcot/sbin/arrfenv`

   このコマンドで、*<install_location>* を Risk Authentication がインストールされているディレクトリのパスに置き換えます。

2. 以下のディレクトリに移動します。
   install_location/arcot/**bin**/

3. 以下のコマンドを実行します。

   ```
   ./casemanagementserver start
   ```

   注: ケース管理サーバを停止する場合は、bin ディレクトリに移動し、「./casemanagementserver stop」コマンドを入力します。

第 10 章: ユーザ データ サービス(UDS)の展開

Risk Authentication 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、Risk Authentication に対して、組織で展開されているサードパーティ 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. 作業ディレクトリを以下の場所に変更します。
   install_location/arcot/sbin/
2. 「source arrfenv」と入力し、Enter キーを押して必要な環境変数を設定します。
3. 以下の場所にある *arcotuds.war* をアプリケーション サーバ上に展開します。
   install_location/arcot/java/webapps/

   例： Apache Tomcat では、この WAR ファイルを APP_SERVER_HOME/webapps/ に展開します。

   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。
4. *(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. ［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 for application］を選択します。
   d. ［適用］をクリックします。
5. アプリケーション サーバを再起動します。
6. 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/**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 11 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。 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 12 章: 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/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。
   - *Solaris の場合：* STARTING Risk Authentication 8.0
   - *RHEL の場合：* STARTING Risk Authentication 8.0

   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。
   - Solaris の場合： STARTING Risk Authentication Case Management 8.0
   - RHEL の場合： STARTING Risk Authentication Case Management 8.0

   Risk Authentication Case Management Service READY

注: 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。

第 13 章: 2 つ目のシステムへの Risk Authentication のインストール

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 ファイルをターゲット（2 つ目の）システムにコピーします。

   Solaris の場合：

   Risk Authentication-8.0-Solaris-Installer.bin

   Linux の場合：

   Risk Authentication8.0--Linux-Installer.bin

2. インストーラの実行に必要な権限があることを確認します。 ない場合は、以下のコマンドを実行します。

   Solaris の場合：

   chmod a=rx Arcot-RiskFort-8.0-Solaris-Installer.bin

   Linux の場合：

   chmod a=rx Arcot-RiskFort-8.0-Linux-Installer.bin

3. 以下のようにインストーラを実行します。

   Solaris の場合：

   prompt> sh Arcot-RiskFort-8.0-Solaris-Installer.bin

   Linux の場合：

   prompt> sh Arcot-RiskFort-8.0-Linux-Installer.bin

4. ［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。

5. 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。

6. 「1 つ目のシステムへのインストール (P. 92)」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、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 Risk Authentication のワークフロー
- Risk Authentication API の基本操作（呼び出しと後処理）
- 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 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

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

Risk Authentication.risk-evaluation.properties ファイルには、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、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、Risk Authentication サンプル アプリケーション WAR ファイル（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 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¥Risk Authentication-8.0-sample-application>¥WEB-INF¥classes¥properties¥`

   `<App_Home¥Risk Authentication-8.0-sample-application¥>`

   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。

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、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、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. 142）
- ユーザの作成（P. 144）
- 既知のユーザのリスク評価および後評価の実行（P. 145）
- 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. 146）

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 Risk Authentication サーバに提供します。 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. 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。
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。
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: 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、「Risk Authentication 8.0 管理ガイド」の「SSL の設定」を参照してください。

第 14 章: インストール後のチェックリストの適用

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	
システム情報		
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	Risk Authentication サーバ 管理コンソール ユーザ データ サービス	
管理コンソール情報		
ホスト名	localhost	
ポート	8080	
マスタ管理者パスワード	mypassword1234!	
ユーザ データ サービス情報		
ホスト名	localhost	
ポート	8080	
アプリケーション コンテキスト ルート	arcotuds	

第 15 章: サイレント モード インストール

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。

Risk Authentication プロパティ ファイル

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。

名前

installer.properties

場所

```
risk_auth_home/
```

risk_auth_home

Risk Authentication のインストール パスを指定します。

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、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、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、空白にままにします。

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、空白にままにします。

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、空白にままにします。

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、空白にままにします。

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 インストールの実行

Risk Authentication をユーザによる操作なしでインストールするには、サイレント インストールを実行します。

次の手順に従ってください：

1. サイレント インストールのガイドラインを確認します。
2. Risk Authentication ホスト システムから Risk Authentication プロパティ ファイルをコピーします。
3. 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. 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. 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

   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 /opt/ca/arcot/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 Risk Authentication をインストールします。
6. Risk Authentication のインストールを確認します。

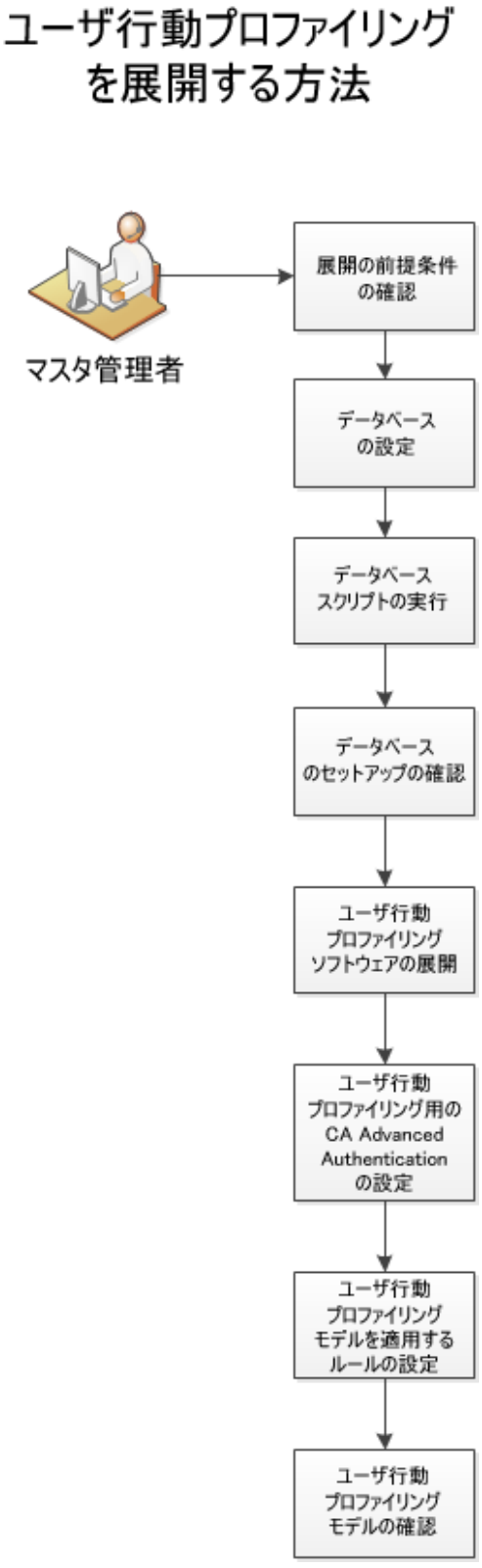
第 16 章: ユーザ行動プロファイリング モデルを展開する方法

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。

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
マスタ管理者
展開の前提条件
の確認
データベース
の設定
データベース
スクリプトの実行
データベース
のセットアップの確認
ユーザ行動
プロファイリング
ソフトウェアの展開
ユーザ行動
プロファイリング用の
CA Advanced
Authentication
の設定
ユーザ行動
プロファイリング
モデルを適用する
ルールの設定
ユーザ行動
プロファイリング
モデルの確認

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. 165)
2. データベースの設定 (P. 165)
3. データベース スクリプトの実行 (P. 170)
4. データベースのセットアップの確認 (P. 171)
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. 174)
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. 175)
8. ユーザ行動プロファイリング モデルの確認 (P. 176)

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 Strong Authentication/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。

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、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、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 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、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. 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、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: 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。

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

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）。

   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、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、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 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. 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. ［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、ニーモニック（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. 新しいルールの横にある［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. ［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. 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. Risk Authentication サーバ
   b. Risk Authentication ケース管理サービス
   c. 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 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. Risk Authentication を選択して［変更と削除］をクリックします。
9. ［Uninstall Risk Authentication］ウィンドウが表示されます。
10. Uninstall 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 17 章: Oracle RAC 用の Risk Authentication の設定

このセクションの手順は、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. 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

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. 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 18 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 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。

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.prefsize=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.prefsize	32
com.sun.jndi.ldap.connect.pool.initsize	8
com.sun.jndi.ldap.connect.pool.timeout	240000
com.sun.jndi.ldap.connect.pool.protocol	plain ssl
com.sun.jndi.ldap.connect.pool.authentication	simple

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.prefsize=32
-Dcom.sun.jndi.ldap.connect.pool.initsize=8
-Dcom.sun.jndi.ldap.connect.pool.timeout=240000
-Dcom.sun.jndi.ldap.connect.pool.protocol="plain ssl"
-Dcom.sun.jndi.ldap.connect.pool.authentication=simple
-Dweblogic.Name=%SERVER_NAME%
-Djava.security.policy=%WL_HOME%¥server¥lib¥weblogic.policy %PROXY_SETTINGS%
%SERVER_CLASS%
) else (
echo Redirecting output from WLS window to %WLS_REDIRECT_LOG%
%JAVA_HOME%¥bin¥java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%
-Dweblogic.Name=%SERVER_NAME%
-Djava.security.policy=%WL_HOME%¥server¥lib¥weblogic.policy %PROXY_SETTINGS%
%SERVER_CLASS% >"%WLS_REDIRECT_LOG%" 2>&1
)
```

5. 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.prefsize=32
   - -Dcom.sun.jndi.ldap.connect.pool.initsize=8
   - -Dcom.sun.jndi.ldap.connect.pool.timeout=240000
   - -Dcom.sun.jndi.ldap.connect.pool.protocol=plain ssl
   - -Dcom.sun.jndi.ldap.connect.pool.authentication=simple
7. ［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。
   <JBOSS_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、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 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.sepa
rator}-", "read";
permission java.util.PropertyPermission "adb.converterutil", "read";
permission java.lang.RuntimePermission "accessDeclaredMembers";
permission java.security.SecurityPermission "putProviderProperty.BC";
permission java.security.SecurityPermission "insertProvider.BC";
permission java.security.SecurityPermission "putProviderProperty.SHAProvider";
permission java.io.FilePermission "${arcot.home}${file.separator}-",
"read,write";
permission java.net.SocketPermission "*:1024-65535", "connect,accept,resolve";
permission java.net.SocketPermission "*:1-1023", "connect,resolve";
};
```

10. 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。

第 19 章: Risk Authentication SDK および Web サービスの設定

このセクションでは、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。

Risk Authentication API の設定

Risk Authentication には、Java API のセットが付属しています。 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、「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-evaluation

および
<*install_location*>¥Arcot Systems¥sdk¥java¥properties¥riskfort.risk-evaluation.properties

- 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 Risk Authentication Javadoc を参照してください。

Risk Authentication Web サービスの設定

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。

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

Risk Authentication パッケージのインストール後、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、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、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
admin/**ArcotRiskFortAdminWebService.wsdl**	ルールの作成および管理に対して使用されます 管理コンソールの使用により通常実行される設定。
riskfort/**ArcotRiskFortEvaluateRiskService.wsdl**	リスク評価の実行に使用します。
uds/**ArcotUserRegistryMgmtSvc.wsdl**	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。
uds/**ArcotConfigRegistrySvc.wsdl**	ユーザ アカウント タイプの作成と管理に使用します。
uds/**ArcotUserRegistrySvc.wsdl**	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	*http://<rf_hostname>:<rf_port>/services/ArcotRiskFortAdminSvc* ここで指定するデフォルト ポートは **7777** です。
ArcotRiskFortEvaluateRiskService	*http://<rf_hostname>:<rf_port>/services/RiskFortEvaluateRiskSvc* ここで指定するデフォルト ポートは **7778** です。
ArcotUserRegistryMgmtSvc	*http://<app_server_hostname>:<appserver_port>/arcotuds/services/ArcotUserRegistrySvc*
ArcotConfigRegistrySvc	*http://<app_server_hostname>:<appserver_port>/arcotuds/services/ArcotConfigRegistrySvc*
ArcotUserRegistrySvc	*http://<app_server_hostname>:<appserver_port>/arcotuds/services/ArcotUserRegistryMgmtSvc*

注: Java クライアントの生成の詳細については、「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

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、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: 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 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。

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

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。

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 Risk Authentication を初期化します。

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、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、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、Risk Authentication データベースに接続するために 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 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、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 20 章: カスタム アクションの追加

Risk Authentication の各チャネルには、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。 Risk Authentication のルールは、チャネルまたはチャ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ネルを指定します。 そのチャ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ネル ID に置き換えます。 同様に、*<action-name>* をチャ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、「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: Risk Authentication のエラーのトラブルシューティング

この付録では、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、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
- 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、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、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。

Risk Authentication コンポーネント	ファイル名	Description
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。 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、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。

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、「Risk Authentication 管理ガイド」の付録「Risk Authentication のログ」を参照してください。

第 21 章: Risk Authentication のアンインストール

Windows のコントロール パネルから 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 ファイルを実行して 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: 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。

Strong Authentication と Risk Authentication の両方をインストールした後に 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、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。

Risk Authentication サーバのアンインストール

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。 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「Risk Authentication スキーマの削除」を参照してください。

重要: Risk Authentication の後に Strong Authentication をインストールした場合は、Strong Authentication サーバをアンインストールしてから 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. 以下のコンポーネントをシャットダウンします。
   - Risk Authentication サーバ
   - ケース管理キュー サーバ
   - 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. 管理コンソールが開いている場合は閉じます。
3. INI ファイル、および Risk Authentication の設定関連のその他のファイルがすべて閉じられていることを確認します。
4. arcot/ ディレクトリに移動します。
5. 以下のコマンドを実行して Risk Authentication のアンインストールを開始します。
   ```
   sh <install_directory>/arcot/"Uninstall_Uninstall_CA Risk Authentication"/Uninstall CA Risk Authentication
   ```
6. アンインストール ウィザードで以下の手順を実行します。
   - ［**1**］を指定すると、［**1-Completely remove all features and components.**］ オプションが選択されます。このオプションでは、インストールされているすべてのコンポーネントをアンインストールできます。
   - ［**2**］を指定すると、［**2-Choose specific features that were installed by InstallAnywhere.**］ オプションが選択されます。このオプションでは、選択したコンポーネントだけを現在のシステムからアンインストールできます。

   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。

例： 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 Risk Authentication サーバをアンインストールします。

7. **Enter** キーを押して確認し、アンインストールを続けます。
   - ［**1**］を指定した場合は、手順 9 に進みます。

   注: アンインストールが完了するまで数分かかる場合があります。
   - ［**2**］を指定した場合は、手順 8 に進みます。

     ［**2**］を指定すると、［Choose Product Features］画面が表示されます。この画面には、現在のシステムにインストールされている Risk Authentication コンポーネントが表示されます。
8. （**特定のコンポーネントをアンインストールする場合のみ**）コンポーネント番号を入力し（カンマで区切って）、**Enter** キーを押します。

   注: アンインストールが完了するまで数分かかる場合があります。

   アンインストールが完了すると、［Uninstall Complete］画面が表示され、コマンドプロンプトに戻ります。
9. **Enter** キーを押してウィザードを終了し、アンインストールを完了します。

アンインストール後のタスクの実行

Risk Authentication コンポーネントがすべて削除されていることを確認するには、以下の手順に従います。

次の手順に従ってください:

1. アンインストール後に必要なくなった場合は、<*install_location*>**/arcot/** ディレクトリを削除します。

   注: 複数の Advanced Authentication 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が Risk Authentication である場合に限り、このディレクトリを削除します。

2. アプリケーション サーバを停止します。
3. <*APP-SERVER-HOME*> 内の適切なサブディレクトリから以下の WAR ファイルを削除します。

   注: ここで、*APP-SERVER-HOME* は、アプリケーション サーバ（例：Apache Tomcat）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 ファイルの削除の詳細については、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）
   - Risk Authentication-8.0-sample-application.war : サンプル アプリケーション
   - Risk Authentication-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、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. Risk Authentication のインストール時に作成した DSN エントリが自動的に削除されていない場合は、そのファイルを削除します。

このエントリを削除するには、odbc.ini ファイルの保存先に移動してテキスト エディタでこのファイルを開き、対応するデータベース エントリを削除します。 ODBC の設定に基づき、このファイルは以下のいずれかの場所にある可能性があります。

- /etc/odbc.ini
- /usr/local/etc/odbc.ini