CA Advanced Authentication

インストール ガイド
（UNIX プラットフォーム用）

8.0

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。

Chapter 1: 基本の理解

過去数年でインターネット詐欺の件数は急増し、ユーザ名とパスワードに頼る認証方式では万全ではなくなってきました。エンド ユーザを守るため、また政府当局によるセキュリティ規制、社内ポリシー、またはベスト プラクティスを遵守するため、より強固な認証方式が求められています。

しかし、強力な認証方式の導入に際しては、往々にしてコンプライアンス要件を重視するあまりユーザの利便性が損なわれがちです。組織は、複雑な工程は排除しながらも組織内の認証処理のセキュリティを強化することを望んでいます。

また、顧客やパートナーがアプリケーションやデータにアクセスする機会を増やしながらも財務損失やブランドへの悪影響といったリスクを回避する必要があります。

CA Strong Authentication は、ユーザの身元を確認し、保護するための強力な認証サービスで、以下のような特長があります。

- ネットワーク上でパスワード(クリア テキスト形式または暗号化形式のいずれも)を転送しない。
- 各種ユーザのセキュリティや利便性に対する最適な認証方式を選択できる。
- CA AuthID® および CA AuthID OTP を使用する(両者とも特許取得のキー隠蔽技術 Cryptographic Camouflage に基づく)。

  Cryptographic Camouflage では、キーを総当たり攻撃対策となる長さのパスワード 1 つのみで暗号化する方法は取りません。実際にキーを正しく復号化できるパスワードは 1 つのみですが、複数のパスワードでキーを復号化して有効なキーを入手できるように見せかけて攻撃者を欺きます。この方法では、スマート カードと同様に辞書攻撃や MITM (Man-in-the-Middle、中間者)攻撃から秘密キーを保護できますが、これを完全にソフトウェア形式で実現します。

  詳細については、「Cryptographic Camouflage の仕組み (14P.)」を参照してください。

このガイドでは、さまざまなソリューション要件に基づく CA Strong Authentication の展開の計画について説明します。各ソリューション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、および企業内の他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。

注: CA Strong Authentication では、現在でもコード オブジェクトやその他の製品の一部に Arcot や WebFort という用語が使用されています。そのため、ドキュメントに Arcot と WebFort という記述が見られます。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。これらの不整合については、今後のリリースで修正される予定です。

汎用認証サーバとしての CA Strong Authentication

CA Strong Authentication は、独自認証メカニズムとオープン認証メカニズムの実装を幅広くサポートしている点で、*VAS*（*Versatile Authentication Server*、汎用認証サーバ）といえます。また、PKI（Public Key Infrastructure）や OTP/Activation Code（ワンタイム パスワード）を利用した認証をサポートするだけでなく、既存の認証方式を組み込むことができるよう設計されています。これによって、組織では、クリティカル システムおよびパートナー アプリケーションへの変更にシームレスに対処できます。

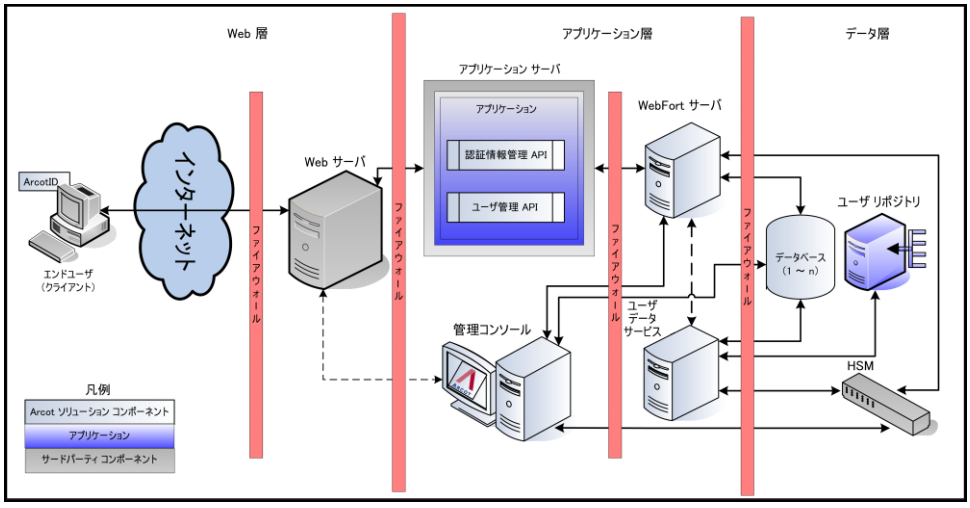
CA Strong Authentication の VAS 機能によって、組織では、エンド ユーザのニーズに最適な認証方式を柔軟に選択できます。以下を選択できます。

- 各種の標準的な認証インターフェースと統合する。
- 標準ベースのハードウェアまたはソフトウェア認証メカニズムを実装する。
- OTP/Activation Code トークンなどの従来の技術を引き続きサポートする一方で、CA AuthID などの新しい認証方式を追加する。
- プラグインによって CA Strong Authentication VAS を拡張し、独自の認証方式を実行する。

CA Strong Authentication のアーキテクチャ

CA Strong 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、トランザクションのセキュリティを最大限に高めるために、以下の図に示されているアーキテクチャを実装することをお勧めします。

- Web 層（8P.）
- アプリケーション層（9P.）
- データ層（10P.）



Web 層

この層は静的な(HTML)コンテンツで形成され、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。

この層がエンド ユーザのブラウザに CA AuthID クライアント(Java、Flash、またはネイティブ)を提供します。CA AuthID クライアントは、CA Strong Authentication サーバと対話してユーザ認証を行います。また、CA AuthID パスワードを収集し、チャレンジに署名し、署名済みのチャレンジを CA Strong Authentication サーバに送信して検証します。

注: CA AuthID クライアントについては、「*CA CA AuthID クライアント リファレンス ガイド*」を参照してください。

アプリケーション層

この層は、CA Strong Authentication サーバ、SDK を使用するアプリケーション、管理コンソールとユーザ データ サービス(UDS)が存在するアプリケーション サーバで構成されます。

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

- **CA Strong Authentication サーバ**

  アプリケーションからの発行リクエストと認証リクエストを CA Strong Authentication SDK を利用して処理するサーバ コンポーネント。

- **管理コンソール**

  サーバ インスタンス、CA Strong Authentication コンポーネント間の通信モード、認証ポリシー、認証情報プロファイル、および認証情報の管理を設定し、組織、管理者、およびユーザの管理を行うための Web ベースのコンソール。

- **ユーザ データ サービス**

  RDBMS (リレーショナル データベース管理システム)やディレクトリ サーバ(LDAP)などの各種ユーザ リポジトリのユーザ関連データおよび組織関連データへのアクセスを提供する抽象層。

- **認証 API**

  認証リクエストを CA Strong Authentication サーバに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 Java API。

- **認証情報管理 API**

  CA Strong Authentication でユーザ認証情報を作成および管理するために CA Strong Authentication サーバに発行リクエストを転送する Java API。アプリケーションから呼び出すことができます。

- **ユーザ管理 API**

  CA Strong Authentication でユーザを作成および管理するためにユーザ データ サービスに発行リクエストを転送する Web サービス クライアント。アプリケーションから呼び出すことができます。

- **サンプル アプリケーション**

  サンプル アプリケーションは、CA Strong Authentication Java API の使用方法およびアプリケーションと CA Strong Authentication の統合方法の例を示します。

データ層

この層には、ほかのユーザ リポジトリが設定されていない場合に、設定、認証情報、およびユーザ データを格納するために CA Strong Authentication が使用する RDBMS があります。

HSM（ハードウェア セキュリティ モジュール）は、ユーザの機密データを暗号化するために使用される場合、この層の一部になります。

CA AuthID の主要概念

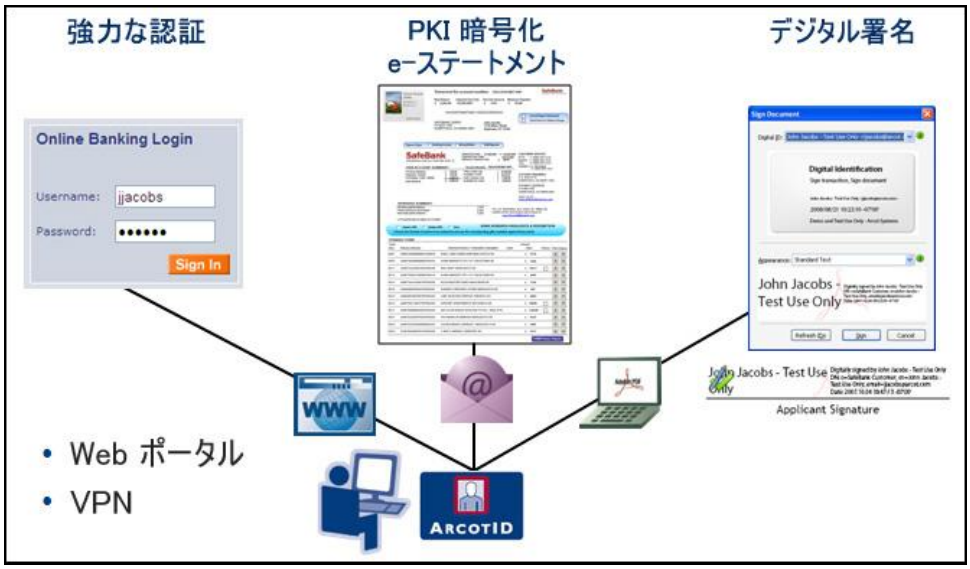
このセクションでは、CA Strong Authentication がサポートする第 1 の認証情報である CA AuthID の主要概念について説明します。

- CA AuthID の概要（11P.）
- CA AuthID のファイル構造（13P.）
- Cryptographic Camouflage の仕組み（14P.）
- ローミング ダウンロードのサポート（14P.）
- セキュア コンテナ（Key Authority）としての CA AuthID（15P.）
- CA AuthID クライアント（16P.）

CA AuthID の概要

CA AuthID は、エンド ユーザのハードウェアを使用せずに、PKI 対応アプリケーションの認証、デジタル署名、暗号化、および復号化に、スマート カードと同じ機能を提供します。Web アプリケーションが PKI ベースの認証をサポートしていない場合でも、CA AuthID は Web アプリケーションに対して認証を行うことができます。

以下の図は、CA AuthID のユース ケースを示しています。



CA AuthID はデータ ファイルであり、セキュリティ保護されたオンデマンド認証のために、エンド ユーザのコンピュータや USB ドライブに保存されるか、またはリモートでダウンロードされます。単純なパスワードとは異なり、CA AuthID はパスワードの総当たり攻撃に対して脆弱 *ではありません*。また、CA AuthID は中間者攻撃に対して脆弱ではなく、フィッシング攻撃からユーザを保護します。

CA AuthID は、Web や VPN（仮想プライベート ネットワーク）など、さまざまなアプリケーションの強力な認証に使用できます。

CA AuthID は、シンプルであるが安全ではないユーザ名/パスワード認証と、高価で展開が難しいが非常に安全なスマート カードおよび USB トークン ソリューションの間のギャップを埋める、設定可能なソリューションです。

CA AuthID は、業界標準および CA の特許取得済みの Cryptographic Camouflage 技術に基づいており、総当たり攻撃に対して保護されている強力な認証をソフトウェアのみで提供します。

CA AuthID はパスワードで保護されており、強力な認証を提供するために以下の機能をサポートしています。

- CA AuthID にアクセスできるのは正しい CA AuthID パスワードのみです。
- 入力されるすべての CA AuthID パスワードに対して、それが正しくない場合でも、偽装したレスポンスが生成されます。その結果、CA AuthID パスワードのオフラインでの識別を困難にしています。
- CA AuthID 認証はチャレンジ/レスポンス認証プロトコルであり、ユーザのパスワードはローカルでのみ使用され、転送されたり、サーバ側で検証されたりすることはありません。
- 正しくない CA AuthID パスワードが繰り返し入力されると、設定されている最大の認証試行数に応じて CA AuthID がロックアウトされます。
- CA AuthID を発行したドメインでのみ有効です。
- CA AuthID はオンラインでのみ使用できます。つまり、ユーザが自分の CA AuthID パスワードを検証するためには、CA Strong Authentication サーバに接続する*必要があります*。

CA AuthID のファイル構造

CA AuthID には、以下の主要なコンポーネントが含まれています。

1. 標準の X.509v3 デジタル証明書（CA 固有の拡張あり）。
2. CA Strong Authentication サーバに対する認証のために生成される、公開キーと秘密キーの 2 番目のペア。これは、一般的な署名や暗号化には使用されません。

   公開キーは暗号化された形式で格納されます。公開キーは、CA AuthID の作成および認証に使用される*ドメイン キー*を使用して暗号化されます。ドメイン キーは、グローバル レベルまたは組織レベルで設定できます。組織に固有のドメイン キーを使用して発行された CA AuthID は、組織を越えて使用することは *できません*。

   秘密キーは、CA AuthID パスワードを使用して暗号的に隠蔽されます。
3. 署名、暗号化、および復号化に使用できる、ユーザのオープン PKI キーおよび証明書を格納するセクション。詳細については、「セキュア コンテナ（Key Authority）としての CA AuthID（15P. ）」を参照してください。

Cryptographic Camouflage の仕組み

Web ブラウザでの公開キー暗号化のサポートにより、公開キー暗号化法署名および認証プロトコルの使用は以前より一般的になっています。

ただし、秘密キーのセキュリティは課題として残っています。最も基本的な脅威は、ディスクに保存されている秘密キーの盗難です。通常、秘密キーはソフトウェア キー コンテナ、ファイルに格納されていて、キーはパスワードを使用して暗号化されています。

コンテナを盗む攻撃者は、辞書攻撃を使用して、パスワードを推測しようとします。

このような問題を打開するために、CA Strong Authentication は、Cryptographic Camouflage を使用して、ソフトウェア内に秘密キーの安全なストレージを確保する方式を提供しています。この方式では、キー コンテナへの攻撃は本質的に監視されます。

キー コンテナでは、偽装の秘密キーの中にユーザの秘密キーを埋め込みます。キー コンテナのクラッキングを試みる攻撃者は、多くの偽装の秘密キーを復元化しますが、そのキーを使用してチャレンジに署名して、CA Strong Authentication サーバに送信するまで、正しい秘密キーと偽装キーを区別することはできません。このような場合、CA Strong Authentication サーバは複数の認証の失敗を検知して、ユーザのアクセスを一時的に停止します。

ローミング ダウンロードのサポート

CA AuthID は、移動中に任意のデバイスを使用してダウンロードできます。この機能は*ローミング*と呼ばれています。CA Strong Authentication サーバは、必要な場合に、ユーザが CA AuthID を安全にダウンロードして任意のシステムから認証することができるローミング機能を提供します。この方法は、データを不正なアクセスから保護しながら、必要な場合はいつでも、クリティカルなデータおよびサービスへの即時アクセスを提供します。

ローミング ユーザは、Q&A、OTP/Activation Code、またはカスタマイズしたサードパーティのソリューションを使用して認証できます。

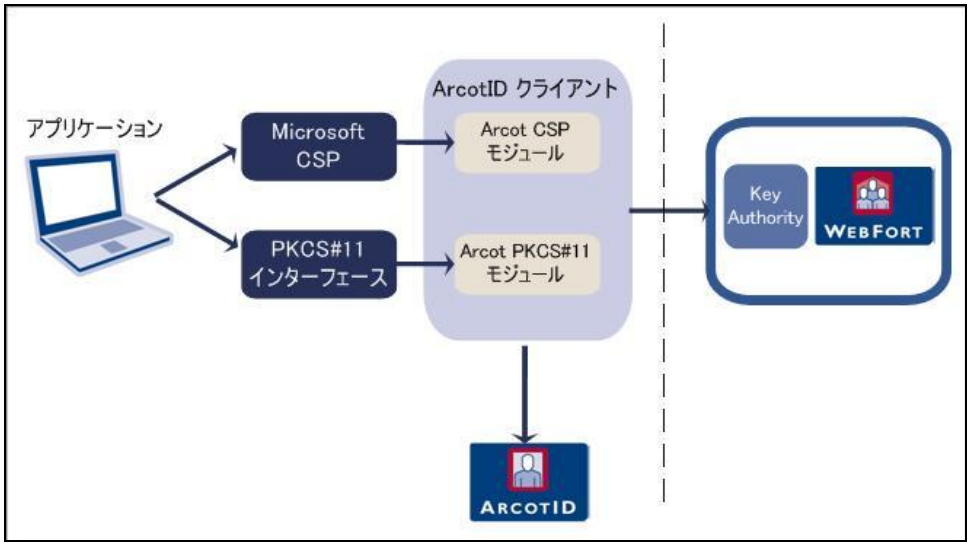
セキュア コンテナ（Key Authority）としての CA AuthID

強力な認証の提供に加えて、CA AuthID は、電子メール署名（S/MIME）、ドキュメント署名、証明書ベースの認証（オープン PKI）など、さまざまなアプリケーションや操作に使用できるデジタル証明書および秘密キーを格納するためのセキュア コンテナとしても使用できます。CA AuthID 内の秘密キー ストレージを管理するプロセスは、KA（*Key Authority*）によって実行されます。

これらの認証情報を格納するために CA AuthID に無署名属性が作成されます。この属性は、キー *バッグ*または*キー ボールト*と呼ばれます。デジタル証明書は暗号化されていない形式でキー バッグに格納されますが、秘密キーは、CA Strong Authentication データベースに格納されている *Key Authority* キーと呼ばれるキーを使用して暗号化されます。

キー バッグに格納されている秘密キーを使用するために、CA AuthID クライアント（「CA AuthID クライアント (16P.)」を参照）は、ユーザの秘密キーを使用してリクエストに署名して、CA Strong Authentication サーバに KA キーをリクエストします。CA Strong Authentication サーバは受信リクエストを認証して、クライアントに KA キーを送信します。クライアントはこのキーを使用して、キー バッグを開き、秘密キーにアクセスします。

以下の図は、オープン PKI コンテナとして CA AuthID を使用する方法を示しています。



CA AuthID クライアント

CA AuthID クライアント ソフトウェアは CA Strong Authentication サーバと一緒に使用されます。CA AuthID クライアントでは、エンド ユーザは Web ブラウザで CA AuthID を使用して、Web サイト、VPN、またはその他のオンライン リソースへの認証を行うことができます。

さまざまなアプリケーション環境（オペレーティング システム、ブラウザ、JVM）をサポートするために、CA AuthID クライアントは以下のようなさまざまなもので利用できます。

- ネイティブ クライアント
- Flash クライアント
- Java 署名済みアプレット
- Java 未署名アプレット
- JavaScript クライアント

注: これらのクライアント タイプの詳細については、「*CA AuthID クライアント リファレンス ガイド*」を参照してください。

CA Strong Authentication プラグイン

CA Strong Authentication には、以下の既定の認証方式が用意されています。

- CA AuthID

  CA AuthID は、2 要素認証を提供する CA 独自のセキュア ソフトウェア認証情報です。CA AuthID は小さなデータ ファイルであり、それ自体を Web や VPN（仮想プライベート ネットワーク）などのさまざまなクライアントに対する強力な認証に使用できます。

  CA AuthID の詳細については、「CA AuthID の主要概念 (10P.)」を参照してください。

- Password

  ユーザがシステムにログインするためにユーザ名およびパスワードを発行する、標準の認証情報。

- **ワンタイム パスワード**

  ワンタイム パスワードは、CA Strong Authentication サーバによって生成されるもう 1 つの認証情報です。OTP/Activation Code は、数字の文字列または英数字の文字列です。また、使用できる回数を設定することもできます。

- OATH **準拠のワンタイム パスワード**

  OATH（オープン認証）標準に準拠したワンタイム パスワード。CA Strong Authentication では、カウンタベースの OATH OTP/Activation Code（HOTP）と時間ベースの OATH OTP Token（TOTP）の両方がサポートされています。

- **質問と回答**

  質問と回答（*Q&A*）は、チャレンジ/レスポンス認証メカニズムです。ユーザは、尋ねられた質問に対して正しい回答を行うことで CA Strong Authentication サーバに認証されます。これらの質問と回答は、登録時にユーザ自身が設定します。

- CA MobileOTP

  CA AuthID は、OATH、EMV（Europay、MasterCard、VISA）標準に準拠しています。お使いのアプリケーションが CA AuthID OTP に統合されると、ユーザのパスワードを入力として受け入れ、ユーザのデバイス上でパスワード（*パスコード*とも呼ばれる）を生成します。その後、ユーザはこの生成されたパスコードをサブミットして、Web アプリケーションを認証します。ユーザは、認証結果に基づき、保護されているアプリケーションへのアクセス権を付与されるか、またはアクセスを拒否されます。

パスコードの生成はオフライン プロセスです。つまり、パスコードを生成するためにアプリケーションを CA Strong Authentication に接続する必要はありません。

- LDAP Username-Password

  CA Strong Authentication は LDAP 認証をサポートしています。この認証では、ディレクトリ サービス内のユーザ認証情報を使用してユーザが認証されます。

1 つ以上の認証情報をユーザに対して発行できます。同じタイプの認証情報を複数発行することもできます。たとえば、単一ユーザに 2 つのパスワード認証情報（CA AuthID 認証情報と Q&A 認証情報）を発行できます。

デフォルトの認証メカニズムを拡張する場合、CA Strong Authentication はプラグインを記述することで、ユーザにその柔軟性を提供します。

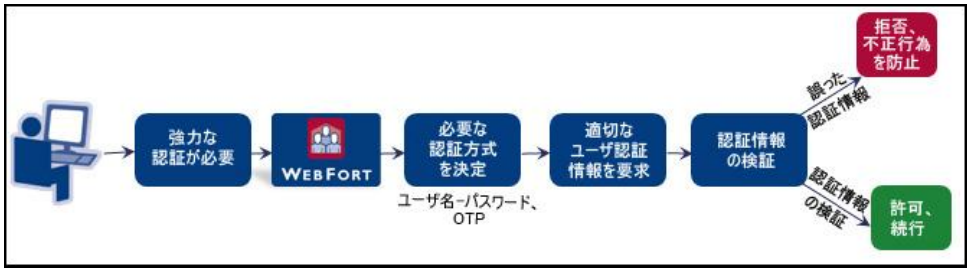
ユーザ認証

CA Strong Authentication で保護されている Web アプリケーションにアクセスを試みるユーザは、既定の認証情報のいずれかを使用して認証できます。

すべての認証メカニズムで、認証が成功するたびにクライアントに認証トークンが提供されます。認証トークンはさらに、クライアントがすでにサーバに認証されていることを証明するために使用されます。認証トークンは一定の期間のみ有効です。その期間後は、クライアントはサーバに再度認証される必要があります。

すべてのパスワード タイプの認証情報（パスワード、OTP/Activation Code、CA AuthID OTP、および OATH OTP Token）は、シングル ステップ認証モデルに基づいています。つまり、認証情報はクライアントによってユーザに渡され、サーバがユーザ認証情報を確認します。

以下の図は、一般的な認証フローを示しています。



ただし、CA AuthID および Q&A はチャレンジ/レスポンス認証モデルに基づいています。これらの認証メカニズムには、ユーザを認証する複数の手順が含まれています。

CA AuthID でのユーザ認証の仕組み

CA AuthID を使用する認証は、PKI ベースのチャレンジ/レスポンス メカニズムです。クライアントはユーザの秘密キーを提供することで、認証トークンを取得します。認証中のクライアントとサーバ間の対話は以下のとおりです。

1. **ユーザ認証情報の取得**

CA Strong Authentication で保護されているアプリケーションまたはリソースがユーザ認証情報を取得します。たとえば、ユーザの CA AuthID がシステムまたは USB に存在しない場合などです。

1. **適切なチャレンジの取得**

   アプリケーションがユーザの認証に使用されたチャレンジをリクエストします。

   CA Strong Authentication サーバが一意のチャレンジを作成して、ユーザのアプリケーションに送信します。

2. **署名の生成**

   ユーザは、CA AuthID を発見するために正しい CA AuthID パスワードを入力します。クライアントが、発見の結果利用可能になったユーザの秘密キーを使用してこのチャレンジに署名します。チャレンジは、クライアント マシンに事前にロードするか、またはサーバからダウンロードすることができます。

3. **署名済みチャレンジの検証**

   署名済みチャレンジが、検証のために CA Strong Authentication サーバに送信されます。署名が正常に検証されると、ユーザはログインしたり、保護されているリソースにアクセスしたりできます。また、CA Strong Authentication は、トランザクションが成功するたびにユーザの認証トークンを返します。

以下の図は、CA AuthID の認証フローを示しています。

Chapter 2: 展開を計画する方法

この章の情報を使用して、展開モデルを選択し、各システムにインストールする CA Strong Authentication コンポーネントおよび事前インストール ソフトウェアを判断します。展開計画がしやすくなるように、各展開モデルのアーキテクチャ図も示します。

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。

この章は以下のトピックで構成されます。

- 新規インストールの展開の概要（22P.）
- アップグレードの展開の概要
- 展開モデルの選択（25P.）

CA Strong Authentication を展開する方法

このプロセスでは、CA Strong Authentication を展開する手順の概要について説明します。

1. ビジネス ニーズに適した展開モデルを選択します。

   詳細については、「展開モデルの選択 (25P.)」を参照してください。

2. CA Strong Authentication とそのコンポーネントのインストール先となるシステムが、すべてのハードウェア要件を満たしていることを確認します。

   詳細については、「ハードウェア要件」を参照してください。

3. 事前にインストールが必要なソフトウェアをインストールします。

   詳細については、「ソフトウェア要件」を参照してください。

4. データベース サーバを設定します。

   - Microsoft SQL Server データベースの設定の詳細については、「Microsoft SQL Server の設定 (38P.)」を参照してください。
   - Oracle データベースの設定の詳細については、「Oracle データベースの設定 (40P.)」を参照してください。
   - IBM DB2 データベースの設定の詳細については、「IBM DB2 Universal Database の設定 (43P.)」を参照してください。
   - MySQL データベースの設定の詳細については、「MySQL の設定 (45P.)」を参照してください。

5. CA Strong Authentication をインストールします。

   - 単一システムへの展開については、「単一システムに CA Strong Authentication を展開する方法 (56P.)」を参照してください。
   - 分散システムへの展開では、「分散システムに CA Strong Authentication を展開する方法 (89P.)」を参照してください。

6. データベースで SQL スクリプトを実行し、スキーマを作成して、初期設定値を設定します。

   - 単一システムへの展開の場合は、「データベース スクリプトの実行 (68P.)」を参照してください。
   - 分散システムへの展開の場合は、「データベース スクリプトの実行 (107P.)」を参照してください。

7. アプリケーション サーバ上の必要なファイルと JAR をコピーします。管理コンソールおよびユーザ データ サービスは正常に機能するためにこれらのファイルを使用します。
   - 単一システム展開でのファイルのコピーの詳細については、「アプリケーション サーバの準備 (70P.)」を参照してください。
   - 分散システム展開でのファイルのコピーの詳細については、「アプリケーション サーバの準備 (109P.)」を参照してください。
8. 管理コンソールを展開します。
   - 単一システム展開での管理コンソールの展開の詳細については、「管理コンソールの展開」を参照してください。
   - 分散システム展開での管理コンソールの展開の詳細については、「管理コンソールの展開」を参照してください。
9. マスタ管理者として管理コンソールにログインします。
   - 単一システム展開での管理コンソールの初期化の詳細については、「管理コンソールへのログイン方法 (81P.)」および「システムのブートストラップ (83P.)」を参照してください。
   - 分散システム展開での管理コンソールの初期化の詳細については、「管理コンソールへのログイン方法」および「システムのブートストラップ (83P.)」を参照してください。
10. CA Strong Authentication サーバを起動し、サービスが正常に開始されていることを確認します。
    - 単一システム展開での CA Strong Authentication サーバの起動の詳細については、「CA Strong Authentication サーバの起動 (86P.)」および「インストールの確認 (86P.)」を参照してください。
    - 分散システム展開での CA Strong Authentication サーバの起動の詳細については、「CA Strong Authentication サーバの起動 (86P.)」および「インストールの確認 (86P.)」を参照してください。
11. サンプル アプリケーションを展開および実行して、インストールをテストします。
    - 単一システム展開でのこの実行方法の詳細については、「サンプル アプリケーションの展開 (89P.)」および「サンプル アプリケーションの使用」を参照してください。
    - 分散展開でのこの実行方法の詳細については、「サンプル アプリケーションの展開 (120P.)」、「サンプル アプリケーションの通信サーバの設定 (121P.)」、および「サンプル アプリケーションの使用」を参照してください。

12. (**オプション**)ユーザ データ サービス(UDS)を展開します。ユーザ リポジトリとしてディレクトリ サービスを使用する*場合のみ*、展開する必要があります。
    - 単一システム展開での UDS の展開および起動の詳細については、「ユーザ データ サービスの展開」を参照してください。
    - 分散システム展開での UDS の展開および起動の詳細については、「ユーザ データ サービスの展開」を参照してください。

展開モデルの選択

CA Strong Authentication の展開の一部として、CA Strong 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。CA Strong Authentication サーバに付属している Java SDK または Web サービスを使用して、アプリケーションと CA Strong Authentication サーバを統合します。

CA Strong 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、開発および単純なテストが目的の場合は、CA Strong Authentication のすべてのコンポーネントを単一のシステムにインストールします。ただし、運用展開およびステージング環境の場合は、CA Strong Authentication サーバを専用のシステムにインストールする必要があります。付属の SDK または Web サービスは、ユーザがログインするアプリケーションが配置された別のシステムにインストールします。

CA Strong Authentication にはサンプル アプリケーションも付属しています。このサンプル アプリケーションを使用して、CA Strong Authentication が正しくインストールされているかどうか、およびユーザ認証を実行できるかどうかを確認できます。また、CA Strong Authentication を既存のアプリケーションと統合するためのサンプル コードとしても役立ちます。

CA Strong Authentication でサポートされている高レベルの展開タイプは以下のとおりです。

- **単一システム展開** - 開発またはテスト用

  詳細については、「単一システムへの展開 (26P.)」を参照してください。

- **分散システム展開** - 運用環境またはステージング環境用

  詳細については、「分散システムへの展開 (29P.)」を参照してください。

- **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

  詳細については、「高可用性環境での展開 (32P.)」を参照してください。

単一システムへの展開

単一システムへの展開では、CA Strong Authentication および Web アプリケーションのすべてのコンポーネントが、単一システム上にインストールされます。この展開モデルは通常、開発、概念実証、または初期テストで使用されます。

単一システム展開では Java SDK および Web サービスの両方を使用できます。これらのコンポーネントの事前インストール ソフトウェアについては、「ソフトウェア要件」を参照してください。

単一システムに CA Strong Authentication を展開するには、インストール時に［*Complete*］オプションを選択する必要があります。インストールおよびインストール後の手順の詳細については、「単一システムに CA Strong Authentication を展開する方法 (56P.)」を参照してください。

コンポーネント図

コンポーネント図には、事前インストール ソフトウェアおよび CA Strong Authentication コンポーネントの可能な展開オプションがいくつか示されています。Complete インストールを実行すると、Java SDK と Web サービスの両方がシステムにインストールされます。CA Strong Authentication と Web アプリケーションの統合には、これらのいずれの方法も使用できます。

- Java SDK の展開
- Web サービスの展開

単一システムの展開を実行する場合、以下の選択を行います。

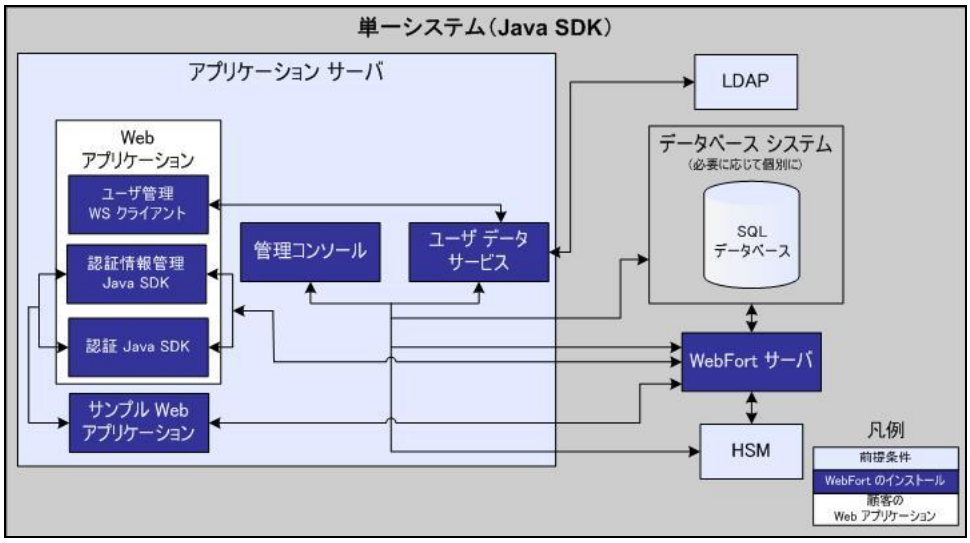
- CA Strong Authentication サーバが配置されているシステムにデータベースサーバをインストールするか、別のシステム上にある既存のデータベースを使用する。
- サンプル アプリケーションを使用するか、独自の Web アプリケーションを作成する。

重要: サンプル アプリケーションを運用展開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。

- 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。

以降の各セクションでは、展開の決定に役立つ情報を提供します。

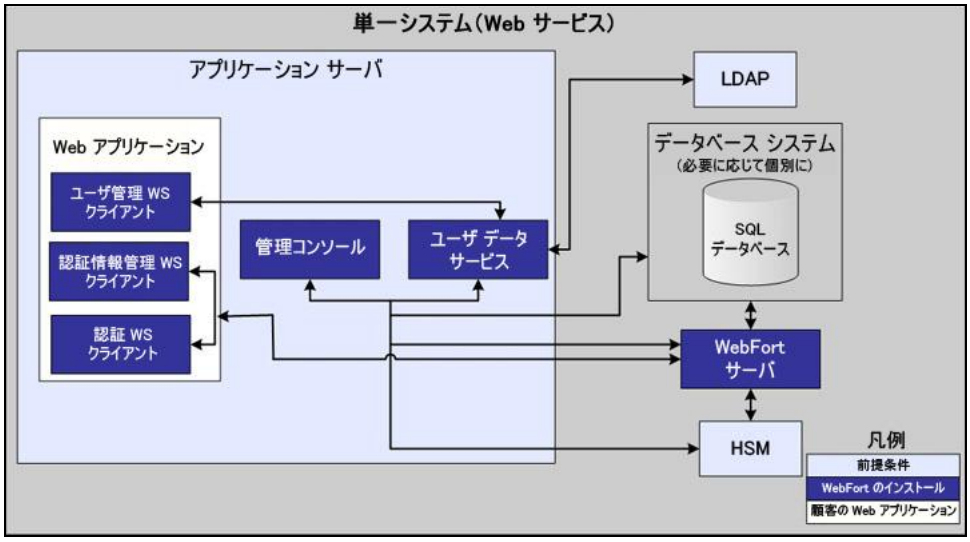
以下の図は、単一システムへの CA Strong Authentication サーバおよび Java SDK の展開を示しています。



注: アプリケーション サーバの HTML ページを配信するための Web サーバの使用はオプションであり、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 サービスを展開する場合について、以下の図は、単一システムの CA Strong Authentication サーバと Web サービスを示しています。

注: 現在すべての Web サービスは CA Strong Authentication サーバ自体に組み込まれているため、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。



分散システムへの展開

分散システムへの展開では、CA Strong Authentication コンポーネントを異なるシステムにインストールします。このタイプの展開は、セキュリティおよびパフォーマンスを向上させます。このモデルは通常、運用環境の展開またはステージングの環境で使用されます。

最も一般的な展開では、1 つのシステムに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。分散システムに CA Strong Authentication を展開するには、インストール時に[*Custom*]オプションを選択します。インストールおよびインストール後の手順の詳細については、「分散システムに CA Strong Authentication を展開する方法 (89P.)」の章を参照してください。

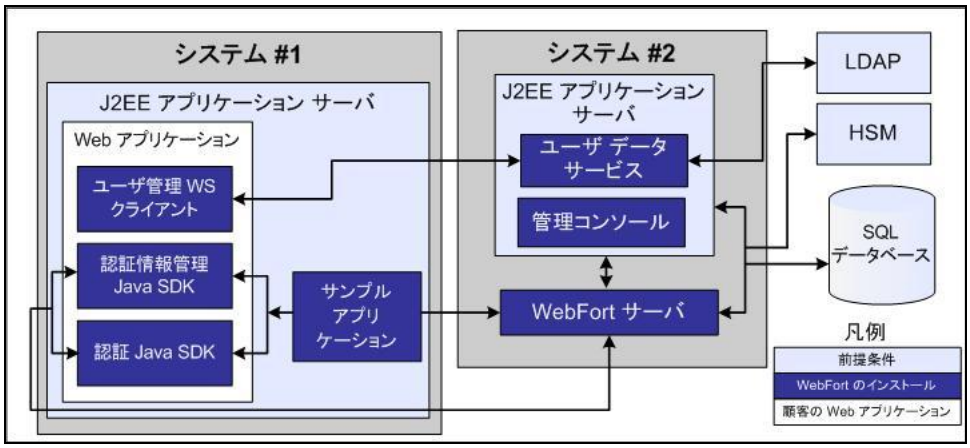
コンポーネント図

このセクションの図は、事前インストール ソフトウェアと CA Strong 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。

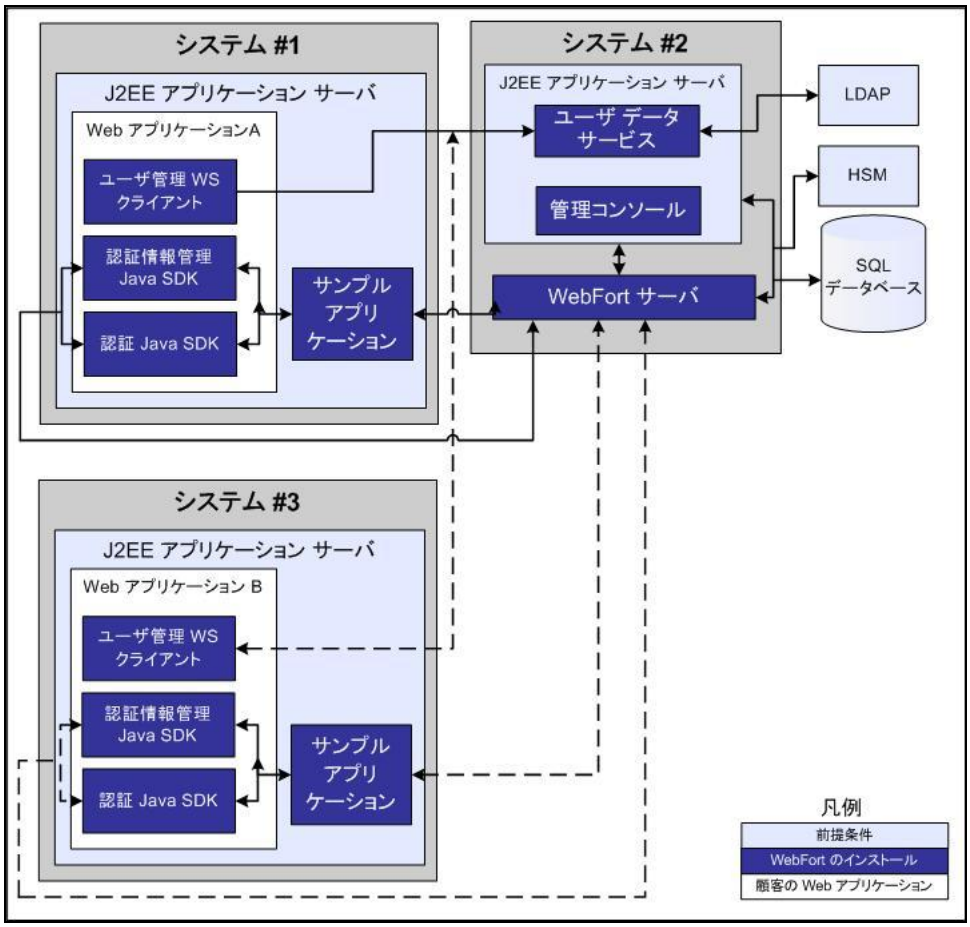
- Java SDK を使用した単一アプリケーションの展開
- Java SDK を使用した複数アプリケーションの展開
- Web サービスを使用した単一アプリケーションの展開
- どの CA Strong Authentication コンポーネントを各システムにインストールするのか。

以降の各セクションでは、展開の決定に役立つ情報を提供します。

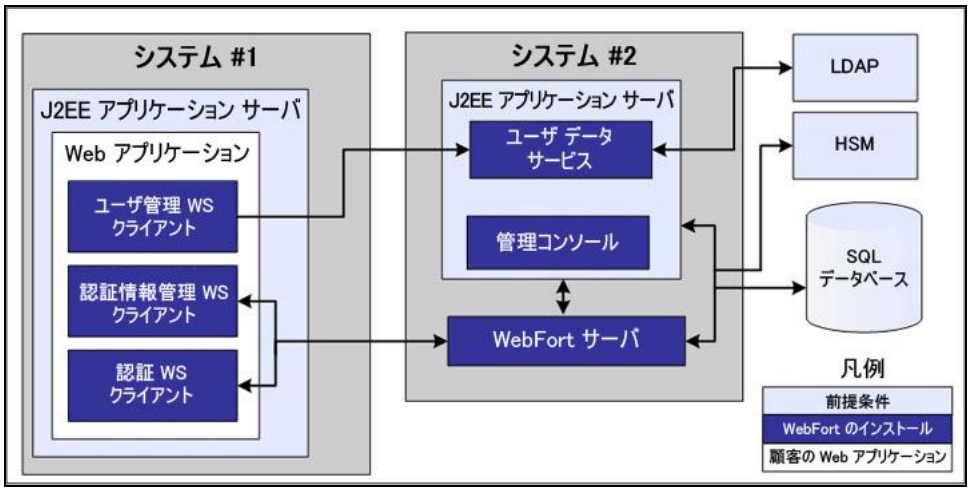
以下の図は、Java SDK を使用した単一アプリケーションへの CA Strong Authentication の展開を示しています。



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



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



高可用性展開

高可用性展開では、高可用性と拡張性を実現するために、CA Strong 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。

このセクションの図は、事前インストール ソフトウェアと CA Strong Authentication コンポーネントを高可用性展開用の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。

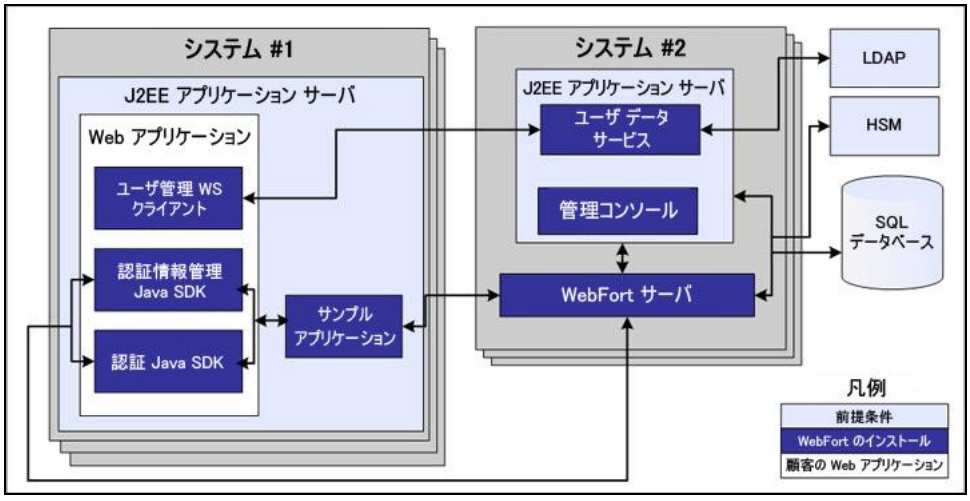
- 新しいサーバ インスタンスをいつ追加するか。

  通常、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、新しいサーバ インスタンスを追加する必要があります。

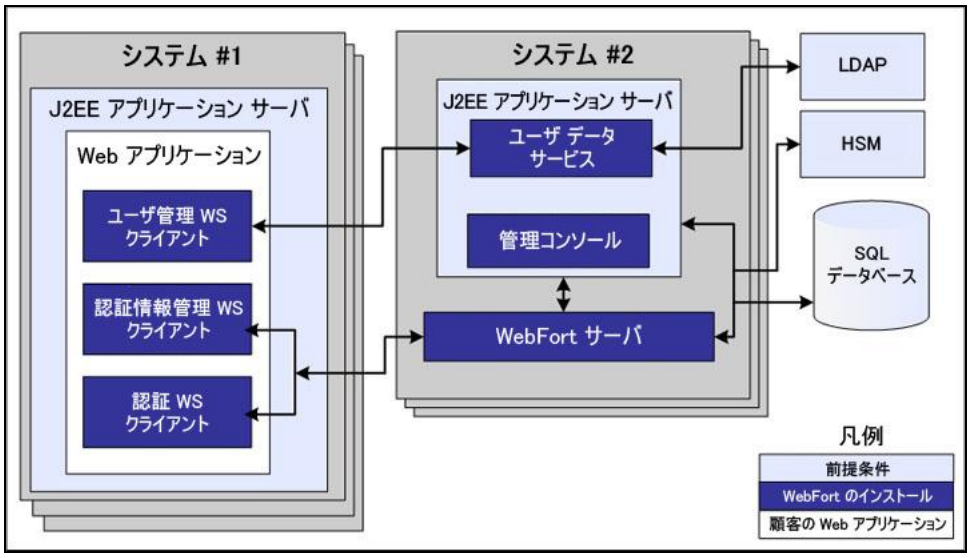
- サーバ、管理コンソール、UDS、および SDK インスタンスはそれぞれいくつ必要か。

  - CA Strong 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。
  - **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。複数の UDS インスタンスが必要な場合は、ロード バランサの背後に配置する必要があります。ただし、UDS フェイルオーバはサポートされていません。
  - 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、Java SDK を使用した CA Strong Authentication の複数インスタンス展開を示しています。



以下の図は、Web サービスを使用した CA Strong Authentication の複数インスタンス展開を示しています。



コンポーネント図

このセクションの図は、事前インストール ソフトウェアと CA Strong Authentication コンポーネントを高可用性展開用の複数のシステムにインストールする場合のオプションを示しています。

決定のポイント

- 新しいサーバ インスタンスをいつ追加するか。

  通常、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、新しいサーバ インスタンスを追加する必要があります。

- CA Strong Authentication サーバ、CA Advanced Authentication、UDS、および SDK インスタンスはそれぞれいくつ必要か。

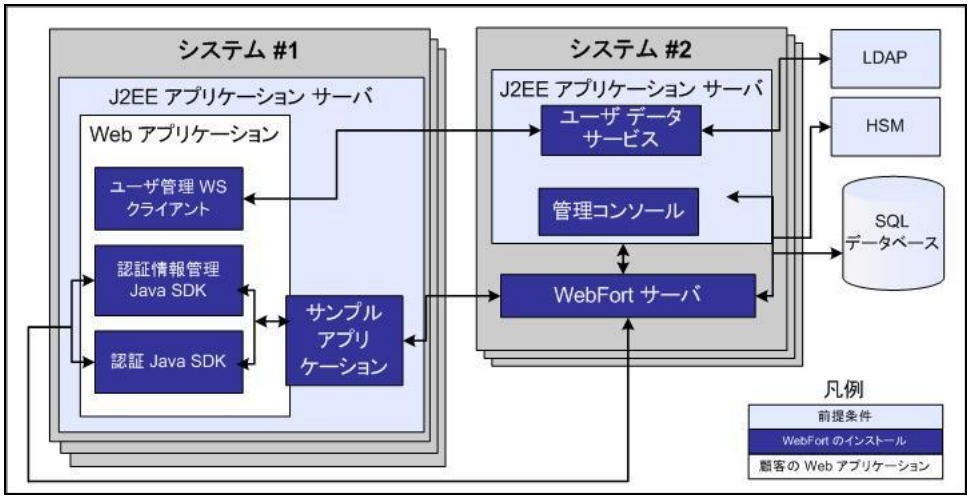
  - CA Strong Authentication **サーバ**: 複数インスタンスがサポートされています。数は、目標のトランザクション レートによって異なります。
  - CA Advanced Authentication: 複数インスタンスがサポートされています。数は、管理コンソールに同時にログインするシステム内の管理者の数によって異なります。
  - UDS **サーバ**: 現在、1 つのみサポートされています。複数の UDS インスタンスが必要な場合は、ロード バランサの背後に配置する必要があります。ただし、UDS フェイルオーバはサポートされていません。
  - SDK: 複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。

以降の各セクションでは、展開の決定に役立つ情報を提供します。

- Java SDK を使用した高可用性展開
- Web サービスを使用した高可用性展開

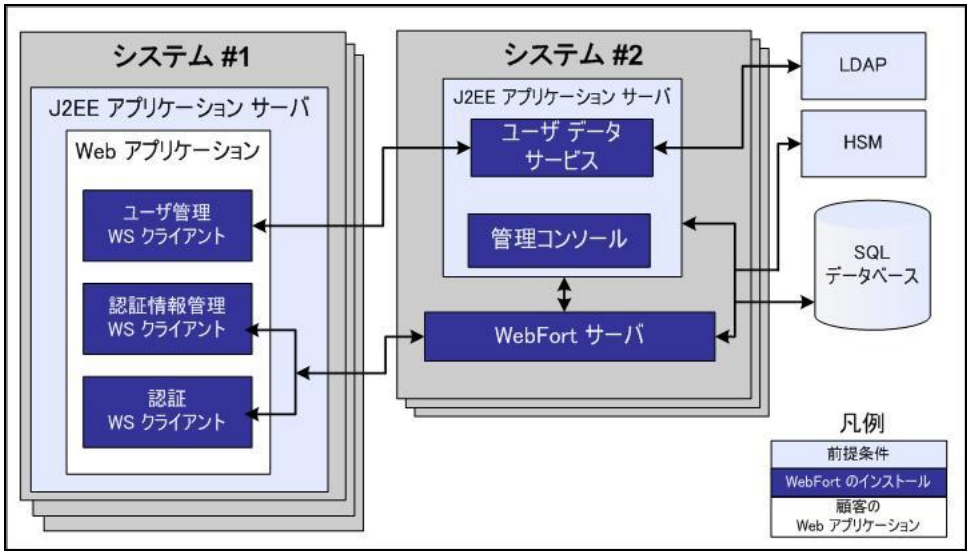
Java SDK を使用した高可用性展開

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



Chapter 3: データベース サーバを設定する方法

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

CA Strong 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。データベース接続は、以下のいずれかの方法で設定できます。

- 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。
- 以下の方法で、手動で設定します。
  - DSN（データ ソース名）を作成します。
  - arcotcommon.ini ファイルを編集します。
  - dbutil ツールを使用して、securestore.enc を更新します。

サポートされているデータベースごとに固有の設定要件があります。データベース サーバを自分で設定する場合は、以下の情報を使用してください。あるいは、データベース アカウントを要求するとき、データベース管理者（DBA）に以下の情報を提供してください。

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: データベースを保護するために、データベース サーバを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で保護し、関連するすべての製品と同じタイム ゾーンに設定することをお勧めします。

- Microsoft SQL Server の設定（38P.）
- Oracle データベースの設定（40P.）
- IBM DB2 Universal Database の設定（43P.）
- MySQL の設定（45P.）

SQL Server の設定

このセクションでは、Microsoft SQL Server 用の以下の設定情報を示します。

重要: Microsoft SQL Server が「SQL Server **認証**」認証方式を使用するように設定されていることを確認します。

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、Microsoft SQL Server のドキュメントを参照してください。

以下の条件に従ってデータベースを設定します。

1. 推奨される名前は arcotdb です。
2. 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. データベース ユーザの作成

データベース ユーザの作成

データベース ユーザを作成するには、以下の手順に従います。

次の手順に従ってください:

1. 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。

2. [**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。
3. ログイン名を入力します。推奨される名前は arcotuser です。
4. 以下のパラメータを設定します。
   a. SQL Server **認証**に対する認証。
   b. 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。

      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。
   c. 作成したデータベース(arcotdb)に対する[**既定のデータベース**]。
   d. ログイン([**このログインにマップされたユーザー**]セクション内)用の[**ユーザー マッピング**]。
   e. db_owner ([<db_name> **のデータベース ロール メンバシップ**]セクション)に対するデフォルト データベース用の[**ユーザー マッピング**](SQL 2005)。

Oracle データベースの設定

このセクションでは、Oracle データベースおよび CA Strong Authentication サーバ用の設定情報を示します。

注: 以下のセクションに示すタスクの実行の詳細については、Oracle データベースのドキュメントを参照してください。

Oracle を使用して CA Strong Authentication を実行するには 2 つのテーブルスペースが必要です。

- 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、デフォルト ユーザ テーブルスペースにすることができます。

  データベースの作成については、「新規データベースの作成 (41P.)」を参照してください。

- 2 つ目のテーブルスペースはレポートの実行に使用されます。パフォーマンスを高めるため、別のテーブルスペースを使用することをお勧めします。

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、データベース設定スクリプト (arcot-db-config-for-common-8.0.sql) によってレポート テーブルスペースが自動的に作成されます。ユーザに必要な権限がない場合、DBA は手動でレポート テーブルスペースを作成し、テーブルスペースを作成するこのスクリプト内のセクションを削除できます。

同じ名前のテーブルスペースがすでに存在する場合は、削除され再作成されます。

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、DBA の希望に応じて変更できます。ただし、テーブルスペース名が ARReports であることを確認します。

データベースの作成

UTF-8 文字セットで情報を格納するデータベースを作成します。この文字セットにより、CA Strong 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。

次の手順に従ってください:

1. 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。
2. 以下のコマンドを実行します。
   ```
   Update sys.props$ set value$='UTF8' where
   name='NLS_NCHAR_CHARACTERSET' Or name = 'NLS_CHARACTERSET';
   ```
3. 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。
4. 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)。
5. 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(主に監査ログに使用されます)。

   注: 本稼働、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、付録「データベースリファレンス」を参照してください。
6. ユーザに以下の権限を付与します。

   ```
   CREATE TABLE
   CREATE INDEX
   CREATE SEQUENCE
   CREATE PROCEDURE
   CREATE SESSION
   DML PRIVILEGES
    RESOURCE PRIVILEGES
   CONNECT PRIVILEGES
   ALTER TABLE
   ```

   - アップグレード専用の追加権限

   ```
    ALTER EXTENT PARAMETERS
    CREATE TABLESPACE
   ```

   - レポートを使用するための追加権限

   ```
   UNLIMITED TABLESPACE
   ```

（オプション） DROP TABLESPACE

IBM DB2 Universal Database の設定

このセクションでは、IBM DB2 Universal Database（UDB）用の以下の設定情報を示します。

UTF-8 文字セットで情報を格納するデータベースを作成します（推奨される名前は arcotdb）。この文字セットにより、CA Strong 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。

1. IBM DB2 UDB データベース サーバにログインします。
2. 以下のコマンドを実行して、UTF-8 サポートを有効にします。
   `create db <DB-NAME> using codeset utf-8 territory us;`
3. テーブルスペースのページ サイズを 16K に設定します。デフォルトでは、テーブルスペースのページ サイズは 4K です。

   テーブルスペースのページ サイズの変更の詳細については、ベンダーのドキュメントを参照してください。
4. データ量が多い場合、トランザクション ログ ファイルのデフォルト サイズでは不十分な場合があります。そのため、ログ ファイル サイズを増やすことをお勧めします。

   ログ ファイル サイズの変更の詳細については、ベンダーのドキュメントを参照してください。
5. 設定変更が適用されたことを確認します。

   代替スキーマを使用する場合の詳細については、付録「IBM DB2 Universal Database の代替スキーマの設定」を参照してください。
6. 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）。
7. ユーザに以下の権限を付与します。

   ```
   CREATE TABLE
   CREATE INDEX
   CREATE SEQUENCE
   CREATE PROCEDURE
   CREATE SESSION
   DML PRIVILEGES
   CONNECT PRIVILEGES
   ALTER TABLE
   ```

   - アップグレード専用の追加権限

CREATE TABLESPACE（AUTORESIZE = yes を指定）

- レポートを使用するための追加権限

DROP TABLESPACE

MySQL データベースの設定

このセクションでは、MySQL データベース用の以下の設定情報を示します。

CA Strong Authentication は、MySQL の InnoDB ストレージ エンジンを使用します。このストレージ 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。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。

注: InnoDB のサポートを有効にする手順については、MySQL のドキュメントを参照してください。

1. MySQL コマンド ウィンドウを開きます。
2. 以下のコマンドを実行して、データベース スキーマを作成します。
   `CREATE SCHEMA '<schema-name>' DEFAULT CHARACTER SET utf8;`
3. 以下のコマンドを実行して、データベース ユーザを作成します。
   `CREATE USER '<user-name>' identified by '<user-password>';`
4. 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）。
5. 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`

Chapter 4: クライアント システムの UTF-8 サポートの設定

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

UNIX プラットフォームで UTF-8 サポートを有効にするには、以下の環境変数を設定します。

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

Chapter 5: Java 依存コンポーネントの要件

管理コンソール、CA Strong 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 環境変数を設定します。path 変数は $JAVA_HOME/bin/ を参照している必要があります。また、アプリケーション サーバが同じ JAVA_HOME を使用することを確認します。そうなっていない場合、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
- UDS

Chapter 6: HSM の要件

HSM を使用して暗号化キーを格納する場合は、以下を設定してから先に進みます。

1. HSM Server
2. HSM クライアント
3. HSM の少なくとも 1 つの 3DES キー

   重要: これらの 3DES キーのラベルをしっかりと書き留めたことを確認します。これは、データベース内の情報を暗号化する際に使用します。

HSM サーバおよびクライアント コンポーネントのインストールと設定、および必要なキーの生成方法の詳細については、プラットフォーム ベンダーのドキュメントを参照してください。

Chapter 7: インストール前のチェックリスト

CA Strong Authentication のインストールと設定に進む前に、以下のチェックリストを確認することをお勧めします。

注: 以下のチェックリストの項目および値はサンプルです。インストール手順を開始する前に、お使いのオペレーティング環境の要件を満たすように、このチェックリストを変更してください。

情報	入力例	記入欄
データベース		
Type	Oracle	
データベース名 (*MS SQL および DB2 のみ*)	arcotdb	
DSN 名	arcotdsn	
ホスト名 (またはサーバ IP アドレス)	51.100.25.24	
ポート	1521	
サービス ID (Oracle データベースのみ)	oradb1	
データベース ユーザ	arcotuser	
データベース ログイン パスワード	password1234!	
設定済みの権限: **注**: CREATE 権限の場合はすべて、対応する DROP 権限があります。		
Oracle **データベース**		
CREATE TABLE		
CREATE INDEX		
CREATE SEQUENCE		
CREATE PROCEDURE		
CREATE SESSION		
DML PRIVILEGES		

情報	入力例	記入欄
RESOURCE PRIVILEGES		
CONNECT PRIVILEGES		
ALTER TABLE *(アップグレードの場合のみ)*		
ALTER EXTENT PARAMETERS		
CREATE TABLESPACE (*レポートの場合*)		
UNLIMITED TABLESPACE (*レポートの場合、オプション*)		
DROP TABLESPACE		
Microsoft SQL Server **注**: これらのアクションを実行するユーザは、ddladmin ロールに属している必要があります。		
CREATE TABLE		
CREATE INDEX		
CREATE PROCEDURE		
REFERENCES		
DML PRIVILEGES		
CONNECT PRIVILEGES		
ALTER *(アップグレードの場合のみ)*		
IBM DB2 UDB		
CREATE TABLE		
CREATE INDEX		
CREATE SEQUENCE		
CREATE PROCEDURE		
CREATE SESSION		
DML PRIVILEGES		

情報	入力例	記入欄
CONNECT PRIVILEGES		
ALTER TABLE *(アップグレードの場合のみ)*		
CREATE TABLESPACE (AUTORESIZE = yes を指定) *(レポートの場合)*		
DROP TABLESPACE		
MySQL		
SELECT		
INSERT		
UPDATE		
DELETE		
EXECUTE		
CREATE		
ALTER		
CREATE ROUTINE		
ALTER ROUTINE		
DROP		
GRANT OPTION		
アプリケーション サーバ		
Type	Apache Tomcat 5.5	
ホスト名	localhost	
ポート	8080	
JDK	1.5.0_10	
ディレクトリ サービス(オプション)		
ホスト名	ds.myldap.com	
ポート	389	
スキーマ名	inetorgperson または user	

情報	入力例	記入欄
ベース識別名	dc=myldap,dc=com	
User Name	cn=admin,cn=Administrators,cn=dscc	
Password	mypassword1234!	
Web サーバ（オプション）		
Type	IIS 6	
ホスト名	mywebserver.com	
ポート	443	

Chapter 8: 単一システムに CA Strong Authentication を展開する方法

インストール ウィザードを使用して、順を追ってインストールを進めることができます。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。単一のコンピュータ上に CA Strong 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。

大まかな作業の流れは以下のとおりです。

1. インストーラを実行してファイル システムに CA Strong Authentication コンポーネントを追加し、SQL データベースにアクセスするように設定します。

   インストール手順については、「Complete インストールの実行 (60P.)」を参照してください。

2. データベース スクリプトを実行し、スキーマおよびデータベース テーブルを作成します。また、データベースが正常に設定されていることを確認します。

   詳細については、「データベース スクリプトの実行 (68P.)」および「データベース セットアップの確認 (69P.)」を参照してください。

3. アプリケーション サーバを準備して、Web コンポーネントが使用するファイルをコピーします。

   詳細については、「アプリケーション サーバの準備 (70P.)」を参照してください。

4. アプリケーション サーバに管理コンソールを展開して、展開を確認します。

   詳細については、「管理コンソールの展開」および「管理コンソールの確認」を参照してください。

5. マスタ管理者として管理コンソールにログインし、CA Strong Authentication を初期化します。

   詳細については、「管理コンソールへのログイン方法 (81P.)」および「システムのブートストラップ (83P.)」を参照してください。

6. CA Strong Authentication サーバを起動し、サービスが正常に開始されていることを確認します。

   詳細については、「CA Strong Authentication サーバの起動 (86P.)」および「インストールの確認 (86P.)」を参照してください。

7. アプリケーション サーバにユーザ データ サービスを展開して、展開を確認します。

   詳細については、「ユーザ データ サービスの展開」を参照してください。

8. サンプル アプリケーションを展開し、これを使用して CA Strong Authentication 設定をテストします。

   注: サンプル アプリケーションは、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。

   詳細については、「サンプル アプリケーションの展開 (89P.)」および「サンプル アプリケーションの使用」を参照してください。

9. (オプション) 製品コンポーネント間の安全な通信を確保するために、SSL をサポートするよう設定できます。

   詳細については、「CA CA Strong Authentication 管理ガイド」の「SSL の設定」を参照してください。

10. インストール チェックリストを完了します。

    詳細については、「インストール後のチェックリスト」を参照してください。

以下の点に注意してください。

- *<install_location>* に特殊文字(~ ! @ # $ % ^ & * ( ) _ + = { }[ ] ’’ " など)が含まれていないことを確認してください。
- 現時点では、インストーラを使用して CA Strong 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/ およびそのサブディレクトリは手動でクリーンアップします。
- 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。
  - 暗号化のセットアップが要求されることもありません。

Complete インストールの実行

すべてのコンポーネントを単一システムにインストールするには、[*Complete*]オプションを使用します。*Custom* インストールでは、選択したコンポーネントのみをパッケージからインストールできます。このオプションは上級ユーザが実行することをお勧めします。

以下の手順に従います。

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。
   `chmod a=rx CA-StrongAuthentication-8.0-Linux-Installer.bin`
3. 以下のようにインストーラを実行します。
   `sh CA-StrongAuthentication-8.0-Linux-Installer.bin`
4. Enter キーを押して、インストールを続行します。

   [使用許諾契約書]が表示されます。
5. 使用許諾契約書に同意します。
   a. 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。

   [Choose Installation Location]オプションが表示されます。
6. 以下の手順のいずれかを実行します。
   - インストーラによって既存のホーム ディレクトリが検出された場合、そのディレクトリのパスが表示されます。このディレクトリ パスを使用するには Enter キーを押します。
   - インストーラによって既存のホーム ディレクトリが検出されない場合、デフォルトのディレクトリ パスが表示されます。その場合は、そのデフォルトのパスを使用するか、または新しいパスを指定します。デフォルトのポートをそのまま使用する場合は、Enter キーを押します。または、製品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。

   CA Strong Authentication でサポートされているインストールのタイプ（Complete または Custom）が表示されます。

7. デフォルトの([Complete])オプションを選択して CA Strong Authentication のすべてのコンポーネントをインストールする場合は「1」を入力し、Enter キーを押して続行します。

   [Database Type]オプションが表示されます。

8. データベースに対応する数字を入力し、Enter キーを押して続行します。

   - 1 - Microsoft SQL Server
   - 2 - IBM DB2 (UDB)
   - 3 - Oracle データベース
   - 4 - MySQL

   [Primary Database Access Configuration]オプションが表示されます。

   注: CA Strong Authentication では、Oracle Real Application Clusters (Oracle RAC)がサポートされています。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 9)を実行してから、「Oracle RAC 用の CA CA Strong Authentication の設定 (128P.)」の手順を実行します。

9. 使用するデータベースに応じて、以下の設定を行います。

   - Microsoft SQL Server を指定した場合は、以下の表に示されている情報を定義します。

パラメータ	Description
Primary ODBC DSN	インストーラによって、CA Strong Authentication がデータベースへの接続に使用する ODBC 接続が作成されます。 推奨される入力値は arcotdsn です。
User Name	CA Strong 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)。 **注**: プライマリおよびバックアップの DSN に対して別のユーザ名を使用することをお勧めします。

パラメータ	Description
Password	上記のフィールドで指定したユーザ名に関連付けられているパスワード。CA Strong 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。
Server Name	CA Strong 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
Port Number	データベース サーバが受信リクエストを待ち受けるポート。 **注**: デフォルトのポートをそのまま使用する場合は、Enter キーを押します。
データベース	MS SQL データベース インスタンスの名前。

- IBM DB2 (UDB)を指定した場合は、以下の表に示されている情報を定義します。

パラメータ	Description
Primary ODBC DSN	インストーラによって、CA Strong Authentication がデータベースへの接続に使用する ODBC 接続が作成されます。 推奨される入力値は arcotdsn です。
User Name	CA Strong 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。 **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*。

パラメータ	Description
Password	上記のフィールドで指定したユーザ名に関連付けられているパスワード。CA Strong 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。
ホスト名	CA Strong 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
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。 **注**: デフォルトのポートをそのまま使用する場合は、Enter キーを押します。
データベース	IBM DB2 データベース インスタンスの名前。

- Oracle データベースを指定した場合は、以下の表に示されている情報を定義します。

パラメータ	Description
Primary ODBC DSN	インストーラによって、CA Strong Authentication がデータベースへの接続に使用する ODBC 接続が作成されます。 推奨される入力値は arcotdsn です。
User Name	CA Strong 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。 **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*。

パラメータ	Description
Password	上記のフィールドで指定したユーザ名に関連付けられているパスワード。CA Strong 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。 **注**：デフォルトのポートをそのまま使用する場合は、Enter キーを押します。
ホスト名	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 を指定した場合は、以下の表に示されている情報を定義します。

パラメータ	Description
Primary ODBC DSN	インストーラによって、CA Strong Authentication がデータベースへの接続に使用する ODBC 接続が作成されます。 推奨される入力値は arcotdsn です。
User Name	CA Strong 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）。 **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。CA Strong 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。

パラメータ	Description
Server Name	CA Strong 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
Port Number	データベース サーバが受信リクエストを待ち受けるポート。 **注**: デフォルトのポートをそのまま使用する場合は、Enter キーを押します。
データベース	MS SQL データベース インスタンスの名前。

[Backup Database Access Configuration]オプションが表示されます。

1. 以下の手順のいずれかを実行します。
   - 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 キーを押して続行します。
   - 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 キーを押して続行します。

     パラメータの詳細については、使用しているデータベースに対する前述の手順の表を参照してください。

   [Encryption Configuration]オプションが表示されます。このオプションを使用して、暗号化モードを選択し、暗号化に使用される情報を設定します。
2. 以下の情報を指定します。
   - Master Key: データベースに格納されるデータを暗号化するために使用されるマスタ キーを入力します。デフォルトでは、マスタ キーの値は MasterKey に設定されます。このキーは securestore.enc ファイルに格納されます。このファイルは *<install_location>*/arcot/conf にあります。

     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 キーで securestore.enc ファイルを再生成します。詳細については、「CA CA Strong Authentication 管理ガイド」を参照してください。

a. 機密データを暗号化するためにハードウェア セキュリティ モジュール（HSM）を使用する場合は「y」を入力します。ソフトウェア暗号化を使用する場合は、「n」を入力します、この場合、以下の HSM 情報を入力する必要はありません。

注：以下のオプションは、HSM を使用することを選択した場合のみ表示されます。

b. Luna HSM を使用する場合は「1」、nCipher netHSM を使用する場合は「2」を入力します。

- HSM PIN：HSM に接続するために使用されるパスワードを入力します。
- Shared Library：HSM に対応する PKCS#11 共有ライブラリへの絶対パス。

注：libcknfast.so ファイルの絶対パスを入力します。

- Storage Slot Number：データの暗号化に使用される 3DES キーが存在する HSM スロット。Luna のデフォルト値は 0 です。また、nCipher netHSM のデフォルト値は 1 です。

注：HSM パラメータ値は、*<install_location>*/arcot/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、付録「設定ファイルおよびオプション」の説明に従い、arcotcommon.ini ファイルを編集します。

3. Enter キーを押して続行します。

   ［Pre-Installation Summary］が表示されます。

4. 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。

   インストール中であることを示すメッセージが表示されます。数分かかる場合があります。

   上記のタスクが正常に完了すると、［Installation Complete］メッセージが表示されます。

5. 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。

6. UTF-8 サポートが有効になっていることを確認します。

   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。

インストール後の作業

このセクションでは、CA Strong Authentication のインストール後に実行するインストール後のタスクについて説明します。

1. データベース スクリプトの実行 (68P.)
2. データベース セットアップの確認 (69P.)
3. アプリケーション サーバの準備 (70P.)
4. 管理コンソールの展開
5. 管理コンソールの確認
6. 管理コンソールへのログイン方法 (81P.)
7. システムのブートストラップ (83P.)
8. CA Strong Authentication サーバの起動 (86P.)
9. インストールの確認 (86P.)
10. ユーザ データ サービスの展開
11. サンプル アプリケーションの展開 (89P.)
12. サンプル アプリケーションの使用

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

データベース スクリプトの実行

CA Strong Authentication には、CA Strong Authentication データベースでスキーマを作成して初期設定値を設定するデータベース スクリプトが付属しています。

次の手順に従ってください:

1. データベース タイプに対応するスクリプトが格納されたフォルダを見つけます。デフォルトの場所は以下のとおりです。

   （Microsoft SQL Server の場合）<*install_location*>/arcot/dbscripts/mssql

   （Oracle データベースの場合）<*install_location*>/arcot/dbscripts/oracle

   （IBM DB2 UDB の場合）<*install_location*>/arcot/dbscripts/db2

   （MySQL の場合）<*install_location*>/arcot/dbscripts/mysql

2. データベース ベンダー ツールを使用して、*以下の順でスクリプト*を実行します。

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

      重要: Risk Authentication 8.0 をインストール済みの場合は、Risk Authentication 8.0 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-webfort-8.0.sql

注: スクリプトの実行中にエラーが発生した場合は、必要な権限が付与されているかどうかをデータベース管理者に確認します。

データベースのセットアップの確認

データベース スクリプトを実行した後、arwfutil ツールを使用して、スキーマが正しくシードされていることを確認します。

次の手順に従ってください:

1. コマンド プロンプト ウィンドウを開きます。
2. 以下の場所に移動します。
   <install_location>/arcot/sbin
3. コマンド プロンプトで、以下のコマンドを入力します。
   ./arwfutil vdb

   このコマンドにより、*<install_location>*/arcot/logs ディレクトリに arcotwebfort-vdb-*<dd>*-*<mmm>*-*<yy>*.txt ファイルが作成されます。
4. テキスト エディタで arcotwebfort-vdb-*<dd>*-*<mmm>*-*<yy>*.txt ファイルを開き、以下のタイプのエントリを確認します。
   ARWF* FOUND

   これらの行は、データベース が正常にセットアップされたことを示します。
5. arcotwebfort-vdb-*<dd>*-*<mmm>*-*<yy>*.txt ファイルを閉じます。

アプリケーション サーバを準備する方法

CA Strong 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、以下のアプリケーション サーバをサポートしています。

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

UDS および管理コンソール WAR ファイルをアプリケーション サーバに展開する前に、CA Strong Authentication ファイルと JDBC JAR ファイルを、お使いのアプリケーション サーバ上の適切な場所にコピーします。

- 手順 1: Java ホームの設定 （70P. ）
- 手順 2: アプリケーション サーバへの Arcot ファイルのコピー
- 手順 3: アプリケーション サーバへの JDBC JAR のコピー
- 手順 4:（Oracle WebLogic 10.1 に必須）Enterprise Archive ファイルの作成 （79P. ）

手順 1: Java ホームの設定

アプリケーション サーバに UDS および管理コンソールを展開する前に、JAVA_HOME 環境変数が設定されていることを確認します。Apache Tomcat の場合、JAVA_HOME を、使用している JDK に対応する Java ホーム ディレクトリに設定します。

また、PATH 環境変数に $JAVA_HOME/bin を含めます。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。

手順 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

UDS および CA Advanced Authentication では、データベースに安全にアクセスするために以下のファイルを使用します。

- libArcotAccessKeyProvider.so。以下の場所にあります。
  <install_location>/arcot/native/*<platform name>*/*<32bit-or-64bit>*/
- arcot-crypto-util.jar。以下の場所にあります。
  <install_location>/arcot/java/lib/
- これらのファイルを、CA Strong Authentication を展開したアプリケーション サーバにコピーする必要があります。

Apache Tomcat へのデータベース アクセス ファイルのコピー

次の手順に従ってください:

1. 以下のディレクトリに libArcotAccessKeyProvider.so ファイルをコピーします。
   - RHEL **の場合**: *<Apache Tomcat で使用する JAVA_HOME>*/jre/bin
2. arcot-crypto-util.jar ファイルを以下のディレクトリにコピーします。
   <Apache Tomcat で使用する JAVA_HOME>/jre/lib/ext
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。

   注: 残りのインストール タスクの一環としてアプリケーション サーバの再起動が必要になります。アプリケーション サーバを再起動する回数を最小限に抑えるには、再起動を必要とする最後のタスクを実行した後に 1 回再起動します。

IBM WebSphere へのデータベース アクセス ファイルのコピー

次の手順に従ってください:

1. IBM WebSphere Administration Console にログインします。
2. [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。スコープには、アプリケーションを展開するターゲット サーバ/ノードが含まれる必要があります。
   b. [New]をクリックします。
   c. **名前**を入力します(たとえば、ArcotJNI)。
   d. **クラスパス**を指定します。このパスは、arcot-crypto-util.jar ファイルが存在し、ファイル名も含まれる場所を指している必要があります。 たとえば、*<install_location>*/arcot/java/lib/arcot-crypto-util.jar などです。
   e. JNI のライブラリ パスを入力します。このパスは、libArcotAccessKeyProvider.so ファイルがある場所を指している必要があります。たとえば、*<install_location>*/arcot/java/native/linux/*<32bit-or-64bit>* などです。
   f. [Apply]をクリックします。
3. サーバ レベルのクラス ローダを設定します。
   a. [Servers]-[Server Types]-[WebSphere Application Servers]に移動します。
   b. [Application Servers]で、設定が行われたサーバの設定ページにアクセスします。
   c. [Java and Process Management]をクリックしてから、[Class Loader]をクリックします。
   d. [New]をクリックします。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。
   e. 自動生成された**クラス ローダ** ID をクリックします。
   f. クラス ローダの[Configuration]ページで、[Shared Library References]をクリックします。
   g. [Add]をクリックし、この手順の前半で作成した共有ライブラリ(たとえば、ArcotJNI)を選択して、[Apply]をクリックします。
   h. 変更を保存します。

4. 以下のディレクトリに libArcotAccessKeyProvider.so ファイルをコピーします。
   - RHEL の場合: <*IBM WebSphere で使用する JAVA_HOME*>/jre/bin
5. アプリケーション サーバを再起動します。

   注: 残りのインストール タスクの一環としてアプリケーション サーバの再起動が必要になります。アプリケーション サーバを再起動する回数を最小限に抑えるには、再起動を必要とする最後のタスクを実行した後に 1 回再起動します。

Oracle WebLogic へのデータベース アクセス ファイルのコピー

次の手順に従ってください:

1. 以下のディレクトリに libArcotAccessKeyProvider.so をコピーします。
   - RHEL の場合: <*Oracle WebLogic インスタンスで使用する JAVA_HOME*>/jre/bin
2. 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。
3. arcot-crypto-util.jar を <*Oracle WebLogic インスタンスで使用する JAVA_HOME*>/jre/lib/ext ディレクトリにコピーします。
4. WebLogic Administration Console にログインします。
5. [Deployments]に移動します。
6. [Lock and Edit]オプションを有効にします。
7. [Install]をクリックして、arcot-crypto-util.jar ファイルが含まれるディレクトリに移動します。
8. [**次へ**]をクリックします。

   [Application Installation Assistant]画面が表示されます。
9. [**次へ**]をクリックします。

   [Summary]ページが表示されます。
10. [**完了**]をクリックします。
11. 変更を有効にします。
12. アプリケーション サーバを再起動します。

    注: 残りのインストール タスクの一環としてアプリケーション サーバの再起動が必要になります。アプリケーション サーバを再起動する回数を最小限に抑えるには、再起動を必要とする最後のタスクを実行した後に 1 回再起動します。

JBoss へのデータベース アクセス ファイルのコピー

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>
```

4. アプリケーション サーバを再起動します。

手順 3: アプリケーション サーバへの JDBC JAR のコピー

CA Advanced Authentication、UDS、およびサンプル アプリケーションは、Java 依存コンポーネントであり、データベースに接続するために JDBC JAR ファイルを使用します。これらのファイルはアプリケーション サーバにコピーする必要があります。

注: 以下のセクションで説明されている手順に進む前に、JDBC JAR ファイルをダウンロード済みであることを確認します。サポートされる JDBC JAR ファイルの詳細については、「インストールの準備」を参照してください。

Apache Tomcat への JDBC JAR のコピー

次の手順に従ってください:

1. JDBC JAR ファイルをダウンロードした場所に移動します。
2. JDBC JAR ファイルをコピーして、以下のディレクトリに貼り付けます。
   - Apace Tomcat 5.5.x **の場合**: *<TOMCAT-HOME>*\common\lib
   - Apace Tomcat 6.x **および** 7.x **の場合**: *<TOMCAT-HOME>*\lib

   または、JDBC JAR ファイルが含まれるパスを Classpath 環境変数に追加します。
3. Apache Tomcat を再起動します。

IBM WebSphere への JDBC JAR のコピー

次の手順に従ってください:

1. IBM WebSphere Administration Console にログインします。
2. [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。スコープには、アプリケーションを展開するターゲット サーバ/ノードが含まれる必要があります。
   b. [New]をクリックします。
   c. **名前**を入力します(たとえば、JDBCJAR)。
   d. **クラスパス**を指定します。このパスは、JDBC JAR ファイルが存在する場所で、ファイル名も含まれている必要があります。
   e. [Apply]をクリックします。
3. サーバ レベルのクラス ローダを設定します。

注: クラス ローダを作成するか、または「手順 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」の実行時に作成したクラス ローダを使用できます。

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]をクリックしてから、[Class Loader]をクリックします。

d. [New]をクリックします。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。

e. 自動生成された**クラス ローダ** ID をクリックします。

f. クラス ローダの[Configuration]ページで、[Shared Library References]をクリックします。

g. [Add]をクリックし、[JDBCJAR]を選択して、[Apply]をクリックします。

h. 変更を保存します。

4. IBM WebSphere を再起動します。

Oracle WebLogic への JDBC JAR のコピー

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、Oracle WebLogic Server はデフォルトで Oracle データベースをサポートしているため、このセクションで説明されている手順を実行する必要はありません。

1. JDBC JAR ファイルを以下のディレクトリにコピーします。

   <*Oracle WebLogic インスタンスで使用する JAVA_HOME*>/jre/lib/ext

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、JDBC 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。

   [Application Installation Assistant]画面が表示されます。

7. [**次へ**]をクリックします。

   [Summary]ページが表示されます。

8. [**完了**]をクリックします。
9. 変更を有効にします。
10. Oracle WebLogic Server を再起動します。

JBoss への JDBC JAR のコピー

次の手順に従ってください:

1. 任意のソースから必要な JAR をダウンロードし、ダウンロードした場所に移動します。
2. 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。
3. 同じフォルダ（*<JBOSS_HOME>\modules\advauth-jdbc-driver\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-jdbc-driver">
<resources>
<resource-root path="<JDBC Jar Name>"/>
</resources>
    <dependencies>
        <module name="javax.api"/>
<module name="javax.transaction.api"/>
    </dependencies>
</module>
```

   注: 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

4. アプリケーション サーバを再起動します。

手順 4: Enterprise Archive ファイルの作成

Weblogic 10.1 で有効

CA Strong Authentication には、管理コンソールおよびユーザ データ サービスを展開するための WAR ファイルが付属しています。これらのファイルの形式を EAR に変更して、その EAR ファイルを展開することができます。

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/tools/common/bundlemanager ディレクトリに移動します。
3. 以下のコマンドを使用して bundlemanager ツールを実行し、EAR ファイルを作成します。
   ```
   java -jar bundle-manager.jar  -ear <filename.ear> -warList
   <war_file_name>
   ```
   注: 上記のコマンドの <*war_file_name*> は、管理コンソールの EAR ファイルを生成する場合は arcotadmin.war、UDS の EAR ファイルを生成する場合は、arcotuds.war に置き換えます。

   このコマンドは <*install_location*>/arcot/Java/webapps に EAR ファイルを生成します。

管理コンソールの展開

注: WebSphere 7.0、8.0、および 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。

CA Strong Authentication 管理コンソールを展開するには、arcotadmin.war ファイルが必要です。このファイルは以下から入手できます。
`<install_location>/arcot/java/webapps/`

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。
   `<install_location>/arcot/sbin`
2. 「source arwfenv」と入力し、Enter キーを押して $ARCOT_HOME 環境変数を設定します。
3. 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。
4. 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。
   たとえば、Apache Tomcat の場合は、<*APP_SERVER_HOME*>\webapps\ に WAR ファイルを展開します。
5. アプリケーション サーバを再起動します。

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、インストール時にデータベースで自動的に作成されたマスタ管理者の認証情報を使用します。

次の手順に従ってください:

1. 以下の URL を使用して、Web ブラウザ ウィンドウで管理コンソールを開始します。

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

   注: 上記の URL で指定するホストおよびポートの情報は、管理コンソールが展開されるアプリケーション サーバのものである必要があります。

2. デフォルトのマスタ管理者のアカウント認証情報でマスタ管理者として管理コンソールにログインします。認証情報は以下のとおりです。

   - **ユーザ名** *:* masteradmin
   - **パスワード**: master1234!

webfortserver ユーティリティの使用

webfortserver ツールを使用して、インストールした CA Strong Authentication のリリースを確認できます。このツールの詳細については、「CA Strong Authentication 管理ガイド」を参照してください。

以下の手順に従います。

1. 以下の場所に移動します。
   `<install_location>/arcot/bin`
2. 対話モードでツールを開始するには、以下のコマンドを実行します。
   `./webfortserver -i`
3. プロンプトでバージョン番号を入力します。

   webfort-ver-<dd>-<mmm>-<yy>.txt ファイルが <*install_location*>/arcot/logs フォルダに作成されます。
4. このファイルを開き、以下の項目をチェックして、最新のバージョンを使用していることを確認します。
   - bin セクションの CA Strong Authentication ライブラリ ファイルのバージョンが 8.0 である。
   - bin セクションの UDS ライブラリ ファイル（arwfuds.dll）のバージョンが 2.0.3 である。
5. ファイルを閉じます。

システムのブートストラップ

管理コンソールを使用して CA Strong Authentication を管理できるようにするには、以下の手順を実行してシステムを初期化します。

- 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 ラベルを指定する
- デフォルトの組織の認証メカニズムを指定する

*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。これらのタスクを実行したら、ほかの管理リンクが有効になります。

「ブートストラップ タスクの実行 (84P.)」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

管理コンソールを展開すると、デフォルトで組織が作成されます。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。

ブートストラップ タスクの実行

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. [**古いパスワード**]、[**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。

   [グローバル キー ラベルの設定]画面が表示されます。

3. **グローバル キー ラベル**を指定して、[**次へ**]をクリックします。

   CA Strong Authentication では、機密データに対してハードウェア ベースまたはソフトウェア ベースの暗号化を使用できます。ハードウェア暗号化かソフトウェア暗号化かに関係なく行えます。

   ハードウェア暗号化を使用する場合、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、キー ラベルは HSM キー ラベルと一致している*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には、このラベルがデータベース内の実際のソフトウェア キーへの参照として機能します。

   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**]フィールドには、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が示されます。

   [デフォルト組織の設定]画面が表示されます。

4. [**デフォルト組織設定**]セクションで、デフォルトの組織の以下のパラメータを指定します。

   - **表示名**: 組織の名称。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。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。
     - **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。
     - 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 サービスの展開」の説明に従い、UDS を展開します。
     - CA Strong Authentication **ユーザ パスワード**

       ユーザがここで[CA Strong Authentication **ユーザ パスワード**]オプションを選択すると、サーバによって認証情報が発行されて認証されます。

     注: この実行方法の詳細については、「CA CA Strong Authentication 管理ガイド」を参照してください。

5. [デフォルト組織の設定]画面の[**キー ラベル設定**]セクションで、以下を指定します。
   - **グローバル キーの使用**: デフォルトでは、このオプションが選択されています。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**]オプションを選択解除した場合は、デフォルトの組織に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**: このフィールドには、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が示されます。
6. [**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。

   [終了]画面に示されるとおりに、管理コンソールの初期化が完了します。
7. [**続行**]をクリックして、管理コンソールを使用するほかの設定に進みます。

CA Strong Authentication の起動

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

1. 以下のディレクトリに移動します。
   `<install_location>/arcot/bin/`
2. 以下のコマンドを実行します。
   `./webfortserver start`

   注: サーバを停止する場合は、./webfortserver stop コマンドを実行します。

インストールの確認

以下により、CA Strong Authentication サーバおよび Web アプリケーションが正常に起動したかどうかを確認できます。

- ログ ファイルの使用 (87P.)
- CA Strong Authentication サーバのユーティリティの使用 (82P.)
- ポートの確認 (87P.)

ログ ファイルの使用

CA Strong Authentication サーバが正しく起動したかどうかを確認するには、以下の手順に従います。

1. 以下の場所に移動します。
   <install_location>/arcot/logs
2. 任意のエディタで arcotwebfortstartup.log ファイルを開き、以下の行を見つけます。
   ```
   INSTANCE_VER.................................: [8.0]
   CA Strong Authentication Service READY
   ```
   これらの行は、CA Strong Authentication サーバが正常にインストールされていることを示しています。

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

ポートの確認

CA Strong Authentication サーバがデフォルト ポートで各プロトコルをリスンしているかどうかを確認するには、以下の手順に従います。

1. 以下の場所に移動します。
   <install_location>/arcot/logs
2. 任意のエディタで arcotwebfortstartup.log ファイルを開いてプロトコル名を検索し、以下のスニペットに示されているように、それらが正しいポートをリスンしているかどうかを確認します。
   ```
   PROTOCOLNAME : [Administration-WS]
   PORTNO : 9745
   PROTOCOLID : [Transaction-Native]
   PORTNO : 9742
   PROTOCOLID : [ServerManagement-WS]
   PORTNO : 9743
   PROTOCOLID : [Transaction-WS]
   PORTNO : 9744
   ```

注: デフォルトのポートおよびプロトコルの詳細については、付録「デフォルトのポート番号および URL」を参照してください。

ユーザ データ サービスの展開

CA Strong Authentication は、リレーショナル データベース(RDBMS)から、または LDAP サーバから直接ユーザ データにアクセスできます。

ユーザ データ サービス(UDS)を展開するには、ファイル arcotuds.war が必要です。このファイルは以下から入手できます。
`<install_location>/arcot/java/webapps/`

次の手順に従ってください:

1. アプリケーション サーバの適切なディレクトリに arcotuds.war をインストールします。

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

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

   a. [Applications]-[Application Types]-[WebSphere Enterprise Applications]に移動し、[UDS settings]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. [Apply]をクリックします。

3. アプリケーション サーバを再起動します。

4. UDS が正しく開始したかどうかを確認するには、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/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。

注: FATAL および ERROR メッセージがある場合は確認して解決します。予期しない状態に関する WARNING メッセージはすべて確認します。

サンプル アプリケーションの展開

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

- 一般的な CA Strong Authentication の ワークフロー
- CA Strong Authentication Java API を使用して実行できるタスク
- CA Strong 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展開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。

次の手順に従ってください:

1. 以下の場所からサンプル アプリケーションの war ファイルを展開します。
   `<install_location>/arcot/samples/java`
2. サンプル アプリケーションを開始します。
3. 以下の URL を使用してサンプル アプリケーションにアクセスします。

   http://<host>:<app_server_port>/sample-application/

Chapter 9: 分散システムに CA Strong Authentication を展開する方法

InstallAnywhere ウィザードを使用して、CA Strong Authentication コンポーネントをインストールします。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。ただし、分散環境に CA Strong Authentication をインストールして設定する場合は、インストーラを実行する際に[*Custom*]オプションを使用してください。

以下の手順は、プロセスの概要です。

1. CA Strong Authentication インストーラを実行し、CA Strong Authentication サーバと管理コンソールをインストールして、SQL データベースにアクセスするよう設定します。また、同じシステム上に Web サービスをインストールすることも選択できます。

   インストール手順については、「1 つ目のシステムへのインストール (93P.)」を参照してください。

2. データベース スクリプトを実行し、CA Strong Authentication スキーマおよびデータベース テーブルを作成します。また、データベースが正常に設定されていることを確認します。

   詳細については、「データベース スクリプトの実行 (107P.)」および「データベース セットアップの確認 (69P.)」を参照してください。

3. アプリケーション サーバを準備して、Web コンポーネントが使用するファイルをコピーします。

   詳細については、「アプリケーション サーバの準備 (109P.)」を参照してください。

4. アプリケーション サーバに管理コンソールを展開して、展開を確認します。

   詳細については、「管理コンソールの展開」および「管理コンソールの確認」を参照してください。

5. マスタ管理者の認証情報を使用して管理コンソールにログインし、CA Strong Authentication を初期化します。

   詳細については、「管理コンソールへのログイン方法 (81P.)」および「システムのブートストラップ (83P.)」を参照してください。

6. CA Strong Authentication サーバを起動し、サービスが正常に開始されていることを確認します。

   詳細については、「CA Strong Authentication サーバの起動 (86P.)」および「インストールの確認 (86P.)」を参照してください。

7. アプリケーション サーバにユーザ データ サービスを展開して、展開を確認します。

   詳細については、「ユーザ データ サービスの展開」を参照してください。

8. 1 つ以上のシステムに Java SDK および Web サービスをインストールします。

   詳細については、「2 つ目のシステムへのインストール (68P.)」を参照してください。

サンプル アプリケーションを展開して設定し、これを使用して設定をテストします。

注: サンプル アプリケーションのみをインストールするには、[SDKs and Sample Application]オプションのみ選択していることを確認してから、インストールを続行します。

詳細については、「サンプル アプリケーションの展開 (120P.)」、「サンプル アプリケーションの通信サーバの設定 (121P.)」、および「サンプル アプリケーションの使用」を参照してください。

1. （オプション）CA Strong Authentication コンポーネント間の安全な通信を確保するために、SSL をサポートするよう設定できます。

   詳細については、「CA Strong Authentication 管理ガイド」の「SSL の設定」を参照してください。

2. インストール チェックリストを完了します。

   詳細については、「インストール後のチェックリスト」を参照してください。

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)。
- 現時点では、インストーラを使用して CA Strong 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/ およびそのサブディレクトリは手動でクリーンアップします。
- 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*。

1 つ目のシステムへのインストール

以下の手順に従って、CA Strong Authentication および関連するコンポーネントをインストールします。

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。必要な権限がない場合は、以下のコマンドを実行します。
   ```
   chmod a=rx CA-StrongAuthentication-8.0-Linux-Installer.bin
   ```

3. 以下のようにインストーラを実行します。
   sh CA Strong Authentication-<version_number>-<*platform_name*>-Installer.bin

   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」、インストールを終了する場合は「N」を入力します。インストーラを終了した場合は、再度インストーラを実行します。

   [Welcome]メッセージが表示されます。

4. Enter キーを押して、インストールを続行します。

   [使用許諾契約書]が表示されます。

5. 使用許諾契約書に同意します。

   a. 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。

   [Choose Installation Location]オプションが表示されます。

6. 以下の手順のいずれかを実行します。

   - インストーラによって既存のホーム ディレクトリが検出された場合、そのディレクトリのパスが表示されます。このディレクトリ パスを使用するには Enter キーを押します。
   - インストーラによって既存のホーム ディレクトリが検出されない場合、デフォルトのディレクトリ パスが表示されます。その場合は、そのデフォルトのパスを使用するか、または新しいパスを指定します。デフォルトのポートをそのまま使用する場合は、Enter キーを押します。または、CA Strong 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。

   インストール タイプ(Complete または Custom)を選択する画面が表示されます。

7. 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。
8. Custom 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。

   [Choose Product Features]オプションが表示されます。

9. インストールする CA Strong 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*)で指定します。

   1 つ目のシステムで、以下のコンポーネントをインストールします。

   a. CA Strong Authentication **認証サーバ**

   b. **管理コンソール**

   c. **ユーザ データ サービス**

以下の表に、インストーラでインストールされるすべてのコンポーネント、およびコンポーネントをインストールするために入力する必要がある番号を示します。

オプション	[Component]	Description

オプション	[Component]	Description
1	CA Strong Authentication 認証サーバ	このオプションは、SDK、管理コンソール、および Web サービスからの以下のリクエストを処理するコア処理エンジン（CA Strong Authentication **サーバ**）をインストールします。 ■ 認証情報発行の設定 ■ 認証情報認証の設定 ■ サーバの設定 また、このコンポーネントでは、以下の Web サービスにアクセスできます。 ■ **認証と許可 Web サービス** - ユーザを認証および許可するためのプログラミング インターフェースを提供します。 ■ **発行 SDK および Web サービス** - CA Strong Authentication データベース内のユーザ認証情報を作成、読み取り、更新するためのプログラミング インターフェースを提供します。 ■ **認証 Web サービス** - ユーザを認証するためのプログラミング インターフェースを提供します。 ■ **認証情報管理 Web サービス** - ユーザ認証情報の作成および管理用のプログラミング インターフェースを提供します。 ■ **管理 Web サービス** - CA Strong Authentication 管理コンソールで使用されるプログラミング インターフェースを提供します。 ■ **バルク操作 Web サービス**: OATH トークンをアップロードおよび取得するためのプログラミング インターフェースを提供します。

オプション	[Component]	Description
2	Java SDK および WS	このオプションは、CA Strong Authentication サーバに認証およびユーザ認証情報発行リクエストを転送するためにアプリケーションが呼び出せるプログラミング インターフェース（API および Web サービスの形式）を提供します。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。 ■ **認証 Java SDK および Web サービス** - CA Strong Authentication サーバを使用して認証するためのプログラミング インターフェースを提供します。 ■ **認証情報管理 Java SDK および Web サービス** - ユーザ認証情報の作成および管理用のプログラミング インターフェースを提供します。 ■ **管理 Web サービス** - 設定を作成するためのプログラミング インターフェースを提供します。 ■ **バルク操作 Web サービス**: OATH トークンをアップロードおよび取得するためのプログラミング インターフェースを提供します。 これらのコンポーネントの設定の詳細については、「CA Strong Authentication Java SDK および Web サービスの設定」の章を参照してください。
3	CA Strong Authentication のサンプル アプリケーション	このオプションは、Java API の使用方法を示すための Web ベースのインターフェースを提供します。また、CA Strong Authentication が正常にインストールされているかどうかの確認、および認証情報管理リクエストと認証リクエストを実行できるかどうかの確認を行うためにも使用できます。

オプション	[Component]	Description
4	管理コンソール	このオプションは、CA Strong Authentication サーバおよび認証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。
5	ユーザ データ サービス	このオプションは、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。

1. Enter キーを押して続行します。

   データベース タイプを選択する画面が表示されます。

2. 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 - IBM DB2 (UDB)
   - 3 - Oracle データベース
   - 4 - MySQL

   [Primary Database Access Configuration]オプションが表示されます。

   注: CA Strong Authentication では、Oracle Real Application Clusters (Oracle RAC)がサポートされています。CA Strong 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 CA Strong Authentication の設定 (128P.)」の手順を実行します。

3. 使用しているデータベースに応じた設定:

   - Microsoft SQL Server を指定した場合は、以下の表に示されている情報を指定します。

パラメータ	Description
Primary ODBC DSN	インストーラによって、CA Strong Authentication がデータベースへの接続に使用する ODBC 接続が作成されます。 推奨される入力値は arcotdsn です。

パラメータ	Description
User Name	CA Strong 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)。 **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。CA Strong 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。
Server Name	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
Port Number	データベース サーバが受信リクエストを待ち受けるポート。 **注**: デフォルトのポートをそのまま使用する場合は、Enter キーを押します。
データベース	MS SQL データベース インスタンスの名前。

- IBM DB2 (UDB)を指定した場合は、以下の表に示されている情報を指定します。

パラメータ	Description
Primary ODBC DSN	インストーラによって、CA Strong Authentication がデータベースへの接続に使用する ODBC 接続が作成されます。 推奨される入力値は arcotdsn です。

パラメータ	Description
User Name	CA Strong 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。 **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。CA Strong 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。
ホスト名	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
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。 **注**: デフォルトのポートをそのまま使用する場合は、Enter キーを押します。
データベース	CA Strong Authentication がアクセスするデータベースの名前。

- Oracle データベース サーバを指定した場合は、以下の表に示されている情報を指定します。

パラメータ	Description
Primary ODBC DSN	インストーラによって、CA Strong Authentication がデータベースへの接続に使用する ODBC 接続が作成されます。 推奨される入力値は arcotdsn です。

パラメータ	Description
User Name	CA Strong 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。 **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。CA Strong 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。 **注**: デフォルトのポートをそのまま使用する場合は、Enter キーを押します。
ホスト名	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 を指定した場合は、以下の表に示されている情報を指定します。

パラメータ	Description
Primary ODBC DSN	インストーラによって、CA Strong Authentication がデータベースへの接続に使用する ODBC 接続が作成されます。 推奨される入力値は arcotdsn です。

パラメータ	**Description**
User Name	CA Strong 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。(Microsoft SQL Server では一般的に、このユーザ名を「ログイン」と呼びます)。 **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。CA Strong 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。
Server Name	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
Port Number	データベース サーバが受信リクエストを待ち受けるポート。 **注**: デフォルトのポートをそのまま使用する場合は、Enter キーを押します。
データベース	Microsoft SQL Server データベース インスタンスの名前。

[Backup Database Access Configuration]オプションが表示されます。

1. 以下の手順のいずれかを実行します。
   - 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 キーを押して続行します。
   - 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 キーを押して続行します。

     実行されるタスクの詳細については、上記の手順の表を参照してください。

   [Encryption Configuration]オプションが表示されます。
2. 以下の情報を指定します。
   - Master Key: データベースに格納されるデータを暗号化するために使用されるマスタ キーを入力します。デフォルトでは、マスタ キーの値は MasterKey に設定されます。このキーは securestore.enc ファイルに格納されます。このファイルは *<install_location>*/arcot/conf にあります。

     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 キーの値で securestore.enc ファイルを再生成する必要があります。詳細については、「CA Strong Authentication 管理ガイド」を参照してください。
   - 機密データを暗号化するためにハードウェア セキュリティ モジュール（HSM）を使用する場合は「y」を入力します。ソフトウェア暗号化を使用する場合は、「n」を入力します、この場合、以下の HSM 情報を入力する必要はありません。

   a. Luna HSM を使用する場合は「1」、nCipher netHSM を使用する場合は「2」を入力します。

   b. HSM PIN: HSM に接続するために使用されるパスワードを入力します。

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

     Luna（libCryptoki2.so）および nCipher netHSM（libcknfast.so）の場合は、ファイルの絶対パスとフル ネームを入力します。
   - Storage Slot Number: データの暗号化に使用される 3DES キーが存在する HSM スロット。Luna のデフォルト値は 0 です。また、nCipher netHSM のデフォルト値は 1 です。

     注: HSM パラメータ値は、*<install_location>*/arcot/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、付録「設定ファイルおよびオプション」の説明に従い、arcotcommon.ini ファイルを編集します。

3. Enter キーを押して続行します。

   [Pre-Installation Summary]が表示されます。

4. 表示された製品の詳細を確認し、Enter キーを押して続行します。

   上記のタスクが正常に完了すると、[Installation Complete]メッセージが表示されます。

5. 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。

6. UTF-8 サポートが有効になっていることを確認します。

   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。

1 つ目のシステムでのインストール後のタスク

このセクションでは、以下のインストール後の手順について説明します。

1. データベース スクリプトの実行（107P.）
2. データベース セットアップの確認（69P.）
3. アプリケーション サーバの準備（109P.）
4. 管理コンソールの展開
5. 管理コンソールの確認
6. 管理コンソールへのログイン方法（81P.）
7. システムのブートストラップ（83P.）
8. CA Strong Authentication サーバの起動
9. インストールの確認（86P.）
10. ユーザ データ サービスの展開

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

データベース スキーマの作成

CA Strong Authentication には、CA Strong Authentication データベースでスキーマを作成して初期設定値を設定する SQL スクリプトが付属しています。

次の手順に従ってください:

1. データベース タイプに対応するスクリプトが格納されたフォルダを見つけます。デフォルトの場所は以下のとおりです。

   （Microsoft SQL Server の場合）<*install_location*>/arcot/dbscripts/mssql

   （Oracle データベースの場合）<*install_location*>/arcot/dbscripts/oracle

   （IBM DB2 UDB の場合）<*install_location*>/arcot/dbscripts/db2

   （MySQL の場合）<*install_location*>/arcot/dbscripts/mysql

2. データベース ベンダー ツールを使用して、*以下の順でスクリプト*を実行します。

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

      重要: Risk Authentication 8.0 をインストール済みの場合は、Risk Authentication 8.0 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-webfort-8.0.sql

注: スクリプトの実行中にエラーが発生した場合は、必要な権限を持っているかどうかを確認します。

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、arwfutil ツールを使用して、スキーマが正しくシードされていることを確認します

次の手順に従ってください:

1. コマンド プロンプト ウィンドウを開きます。
2. 以下の場所に移動します。
   <install_location>/arcot/sbin
3. コマンド プロンプトで、以下のコマンドを入力します。
   `./arwfutil vdb`

   このコマンドにより、*<install_location>*/arcot/logs ディレクトリに arcotwebfort-vdb-*<dd>*-*<mmm>*-*<yy>*.txt ファイルが作成されます。
4. テキスト エディタで arcotwebfort-vdb-*<dd>*-*<mmm>*-*<yy>*.txt ファイルを開き、以下のタイプのエントリを確認します。
   `ARWF*  FOUND`

   これらの行は、データベース が正常にセットアップされたことを示します。
5. arcotwebfort-vdb-*<dd>*-*<mmm>*-*<yy>*.txt ファイルを閉じます。

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、Web ベースのアプリケーションであり、以下のアプリケーション サーバをサポートしています。

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

UDS および管理コンソール WAR ファイルをアプリケーション サーバに展開する前に、CA Strong Authentication ファイルと JDBC JAR ファイルを、お使いのアプリケーション サーバ上の適切な場所にコピーします。

- 手順 1: Java ホームの設定 (109P.)
- 手順 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
- 手順 3: アプリケーション サーバへの JDBC JAR のコピー
- 手順 4: (Oracle WebLogic 10.1 に必須) Enterprise Archive ファイルの作成

Java ホームの設定

選択したアプリケーション サーバに UDS および管理コンソールの WAR ファイルを展開する前に、JAVA_HOME 環境変数を設定していることを確認します。Apache Tomcat の場合、JAVA_HOME を、使用している JDK に対応する Java ホーム ディレクトリに設定します。

また、PATH 環境変数に $JAVA_HOME/bin を含めます。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。

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 CA Advanced Authentication では、データベースにアクセスするために以下のファイルを使用します。

- libArcotAccessKeyProvider.so。以下の場所にあります。
  `<install_location>/arcot/native/<platform name>/<32bit-or-64bit>/`
- arcot-crypto-util.jar。以下の場所にあります。
  `<install_location>/arcot/java/lib/`

これらのファイルを、CA Strong Authentication を展開したアプリケーション サーバ上の適切な場所にコピーする必要があります。

Tomcat

次の手順に従ってください:

1. 以下のディレクトリに libArcotAccessKeyProvider.so ファイルをコピーします。
   - RHEL **の場合**: *<Apache Tomcat で使用する JAVA_HOME>*/jre/bin
2. arcot-crypto-util.jar ファイルを *<Apache Tomcat で使用する JAVA_HOME>*/jre/lib/ext ディレクトリにコピーします。
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。

   注: 残りのインストール タスクの一環としてアプリケーション サーバの再起動が必要になります。アプリケーション サーバを再起動する回数を最小限に抑えるには、再起動を必要とする最後のタスクを実行した後に 1 回再起動します。

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]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。スコープには、アプリケーションを展開するターゲット サーバ/ノードが含まれる必要があります。
   b. [新規]をクリックします。
   c. 名前を入力します(たとえば、ArcotJNI)。
   d. クラスパスを指定します。このパスは、arcot-crypto-util.jar ファイルが存在し、ファイル名も含まれる場所を指している必要があります。 たとえば、<install_location>/arcot/java/lib/arcot-crypto-util.jar などです。
   e. JNI のライブラリ パスを入力します。このパスは、libArcotAccessKeyProvider.so ファイルがある場所を指している必要があります。たとえば、*<install_location>*/arcot/java/native/linux/*<32bit-or-64bit>* ディレクトリなどです。
   f. [適用]をクリックします。

3. サーバ レベルのクラス ローダを設定します。
   a. [Servers]-[Server Types]-[WebSphere Application Servers]をクリックします。
   b. [Application Servers]で、設定が行われたサーバの設定ページにアクセスします。
   c. [Java and Process Management]をクリックしてから、[Class Loader]をクリックします。
   d. [新規]をクリックします。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。
   e. 自動生成されたクラス ローダ ID をクリックします。
   f. クラス ローダの[Configuration]ページで、[Shared Library References]をクリックします。
   g. [Add]をクリックし、この手順の前半で作成した共有ライブラリ(たとえば、ArcotJNI)を選択して、[Apply]をクリックします。
   h. 変更を保存します。
4. 以下のディレクトリに libArcotAccessKeyProvider.so ファイルをコピーします。
   - RHEL **の場合**: <*IBM WebSphere で使用する JAVA_HOME*>/jre/bin
5. アプリケーション サーバを再起動します。

   注: 残りのインストール タスクの一環としてアプリケーション サーバの再起動が必要になります。アプリケーション サーバを再起動する回数を最小限に抑えるには、再起動を必要とする最後のタスクを実行した後に 1 回再起動します。

WebLogic

次の手順に従ってください:

1. 以下のディレクトリに libArcotAccessKeyProvider.so をコピーします。
   - RHEL **の場合**: <*Oracle WebLogic インスタンスで使用する JAVA_HOME*>/jre/bin
2. 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。
3. arcot-crypto-util.jar を <*Oracle WebLogic インスタンスで使用する JAVA_HOME*>/jre/lib/ext ディレクトリにコピーします。
4. WebLogic Administration Console にログインします。
5. [Deployments]に移動します。
6. [Lock and Edit]オプションを有効にします。
7. [Install]をクリックして、arcot-crypto-util.jar ファイルが含まれるディレクトリに移動します。
8. [**次へ**]をクリックします。

   [Application Installation Assistant]画面が表示されます。
9. [**次へ**]をクリックします。

   [Summary]ページが表示されます。
10. [**完了**]をクリックします。
11. 変更を有効にします。
12. アプリケーション サーバを再起動します。

    注: 残りのインストール タスクの一環としてアプリケーション サーバの再起動が必要になります。アプリケーション サーバを再起動する回数を最小限に抑えるには、再起動を必要とする最後のタスクを実行した後に 1 回再起動します。

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 のコピー

CA Advanced Authentication、UDS、サンプル アプリケーションは、CA Strong Authentication の Java 依存コンポーネントであり、データベースに接続するために JDBC JAR ファイルを必要とします。これらのファイルはアプリケーション サーバにコピーします。

注: 以下のサブセクションで示されている手順を実行する前に、JDBC JAR をダウンロードしていることを確認してください。サポートされている JDBC JAR の詳細については、「インストールの準備」を参照してください。

以下のセクションでは、データベースに必要な JDBC JAR を以下のいずれか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
- IBM WebSphere
- Oracle WebLogic
- JBoss アプリケーション サーバ

Apache Tomcat

Apache Tomcat インストール ディレクトリに JDBC JAR ファイルをコピーする方法

1. JDBC JAR ファイルをダウンロードした場所に移動します。
2. JDBC JAR ファイルをコピーして、以下のディレクトリに貼り付けます。
   - Apace Tomcat 5.5.x **の場合**: *<TOMCAT-HOME>*\common\lib
   - Apace Tomcat 6.x **および** 7.x **の場合**: *<TOMCAT-HOME>*\lib

   または、JDBC JAR ファイルが含まれるパスを Classpath 環境変数に追加します。
3. Apache Tomcat を再起動します。

IBM WebSphere

IBM WebSphere に JDBC JAR ファイルをコピーする方法

1. WebSphere Administration Console にログインします。
2. [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。スコープには、アプリケーションを展開するターゲット サーバ/ノードが含まれる必要があります。

b. [New]をクリックします。

c. **名前**を入力します(たとえば、JDBCJAR)。

d. **クラスパス**を指定します。このパスは、JDBC JAR ファイルが存在する場所で、ファイル名も含まれている必要があります。

e. [Apply]をクリックして、変更を保存します。

3. サーバ レベルのクラス ローダを設定します。

注: クラス ローダを作成するか、または「手順 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 (110P.)」の実行時に作成したものを使用できます。

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]をクリックしてから、[Class Loader]をクリックします。

d. [New]をクリックします。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。

e. 自動生成された**クラス ローダ** ID をクリックします。

f. クラス ローダの[Configuration]ページで、[Shared Library References]をクリックします。

g. [Add]をクリックし、[JDBCJAR]を選択して、[Apply]をクリックします。

h. 変更を保存します。

4. IBM WebSphere を再起動します。

Oracle WebLogic

Oracle WebLogic Server に JDBC JAR ファイルをコピーする方法

注: Oracle データベースを使用している場合、Oracle WebLogic Server 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う必要はありません。

1. JDBC JAR ファイルを以下のディレクトリにコピーします。
<Oracle WebLogic インスタンスで使用する JAVA_HOME>/jre/lib/ext ディレクトリ。

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、JDBC 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。

   [Application Installation Assistant]画面が表示されます。
7. [**次へ**]をクリックします。

   [Summary]ページが表示されます。
8. [**完了**]をクリックします。
9. 変更を有効にします。
10. Oracle WebLogic Server 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. 任意のソースから必要な JAR をダウンロードし、ダウンロードした場所に移動します。
2. 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。
3. 同じフォルダ(*<JBOSS_HOME>\modules\advauth-jdbc-driver\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-jdbc-driver">
<resources>
<resource-root path="<JDBC Jar Name>"/>
</resources>
    <dependencies>
        <module name="javax.api"/>
<module name="javax.transaction.api"/>
    </dependencies>
</module>
```

   注: 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。

手順 4: Enterprise Archive ファイルの作成

Weblogic 10.1 で有効

CA Strong Authentication には、管理コンソールおよびユーザ データ サービスを展開するための WAR ファイルが付属しています。これらのファイルの形式を EAR に変更して、その EAR ファイルを展開することができます。

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/tools/common/bundlemanager ディレクトリに移動します。
3. 以下のコマンドを使用して bundlemanager ツールを実行し、EAR ファイルを作成します。
   ```
   java -jar bundle-manager.jar  -ear <filename.ear> -warList
   <war_file_name>
   ```
   注: 上記のコマンドの <*war_file_name*> は、管理コンソールの EAR ファイルを生成する場合は arcotadmin.war、UDS の EAR ファイルを生成する場合は、arcotuds.war に置き換えます。

   このコマンドは <*install_location*>/arcot/Java/webapps に EAR ファイルを生成します。

ユーザ データ サービスの展開

CA Strong Authentication は、リレーショナル データベースから、または LDAP サーバから直接ユーザ データにアクセスします。

ユーザ データ サービス(UDS)を使用するには、arcotuds.war ファイルを展開する必要があります。このファイルは以下から入手できます。
`<install_location>/arcot/java/webapps/`

次の手順に従ってください:

1. アプリケーション サーバの適切なディレクトリに arcotuds.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。
   たとえば、Apache Tomcat の場合は、<*APP_SERVER_HOME*>\webapps\ に WAR ファイルを展開します。

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

   a. [Applications]-[Application Types]-[WebSphere Enterprise Applications]に移動し、[UDS Settings]画面にアクセスします。

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

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

   d. [Apply]をクリックします。

3. アプリケーション サーバを再起動します。

4. UDS が正しく開始したかどうかを確認するには、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/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。

注: FATAL および ERROR メッセージがある場合は確認して解決します。予期しない状態に関する WARNING メッセージはすべて確認します。

追加のサーバに CA Strong Authentication を展開する方法

CA Strong 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、分散システム展開での追加のシステムにその他のコンポーネントをインストールします。

1. CA Strong Authentication インストーラである Installer.bin ファイルを見つけます。
2. 以下のコマンドを使用して、インストーラを実行します。
   `sh CA Strong Authentication-version_number-<platform name>-Installer.bin`

   インストーラによりインストールの準備が開始されます。
3. [Choose Product Features]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（93P.）」で説明されている手順を実行します。
4. インストールするコンポーネントを選択します。
5. 手順 12 から手順 19 に従って、インストールを完了します。

サンプル アプリケーションの展開

サンプル アプリケーションを使用して、CA Strong Authentication のインストールが成功し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の機能の例を示します。

- 一般的なワークフロー
- Java API を使用して実行できるタスク
- CA Strong Authentication とアプリケーションの統合

以下の手順に従います。

1. 以下の場所から CA Strong Authentication *version_number*-sample-application.war ファイルを展開します。
   `<install_location>/arcot/samples/java`
2. サンプル アプリケーションを開始します。
3. 以下の URL を使用してサンプル アプリケーションにアクセスします。

   http://<host>:<app_server_port>/CA Strong Authentication-version_number-sample-application/

サンプル アプリケーションの通信サーバの設定

サンプル アプリケーションと CA Strong Authentication サーバを異なるシステムにインストールした場合は、通信設定を行う必要があります。

以下の手順に従います。

Web ブラウザ ウィンドウのサンプル アプリケーションにアクセスします。

1. ナビゲーション ウィンドウで、[Setup]-[Server Connectivity]をクリックして、[CA Strong Authentication Server Connectivity]ページを開きます。
2. 以下の表に示す接続パラメータの値を指定します。

   注: これらのパラメータを使用して作成した設定は、現在のセッションに対して有効です。サンプル アプリケーションまたはアプリケーション サーバを再起動した場合は、再度これらの値を設定します。

フィールド	デフォルト値	Description
IP アドレス	localhost	CA Strong Authentication サーバが利用可能なシステムのホスト名または IP アドレス。
ポート	9742	認証および発行サービスが利用可能なポート。
Maximum Active Connections	64	サンプル アプリケーションによってメンテナンスされたデータベース接続の最大数。

3. [Set Up]をクリックして、接続を保存します。

Chapter 10: サイレント モード インストールを実行する方法

CA Strong Authentication をインストールした後に、サイレント インストール方式を使用して、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。

以下を入力します。

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. CA Strong Authentication ホスト システムから CA Strong Authentication プロパティ ファイルをコピーします。
3. CA Strong Authentication のインストール メディアをプロパティ ファイルと同じ場所にコピーします。
4. CA Strong Authentication インストーラのプロパティ ファイルを変更します。
5. CA Strong Authentication インストーラを実行します。

サイレント モード インストールのガイドライン

サイレント インストールを開始する前に、以下のガイドラインを確認します。

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

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

デフォルト プロパティ ファイル

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

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

Name

installer.properties

場所

`strong_auth_home/`

strong_auth_home

CA Strong Authentication の インストール パスを指定します。

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

このファイル内の以下のデフォルト パラメータでは、CA Strong 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。

WFSRV - CA Strong Authentication サーバ

CA Strong 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

WFSDK - CA Strong Authentication Java SDK および WS

CA Strong 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。

WFAPP - CA Strong Authentication のサンプル アプリケーション

Java SDK の使用方法の例を示す Web ベースのアプリケーション。

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。

USER_INSTALL_DIR_SILENT

CA Strong 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、db2、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 インストールを実行する方法

CA Strong 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. CA Strong Authentication インストーラを実行します。

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

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

   Installation_media

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

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

Chapter 11: Oracle RAC 用の CA Strong Authentication の 設定

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

データベース スクリプトの変更

データベース スクリプトは、CA Strong Authentication インストール手順の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。

次の手順に従ってください:

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-8.0.sql ファイルを開きます。このファイルは、install_location/arcot/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。

   - 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。
   - 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。

JDBC URL の設定

arcotcommon.ini ファイルで、Oracle RAC で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/conf/ ディレクトリにあります。
2. URL パラメータの値を、INI ファイルの [arcot/db/primarydb] セクションに指定し、必要に応じて [arcot/db/backupdb] セクションにも指定します。URL を以下の形式で入力します。

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

   例:

   ```
   URL.1=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=172.30.250.18)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=forwardinc)(SERVER=DEDICATED)))
   ```

   注: Oracle RAC が設定されたクライアントである場合は、すべてのノードをこの形式で含めます。
3. CA Strong Authentication インストーラの実行中に指定したデータベース ユーザが、Oracle RAC のデータベース ユーザと異なる場合は、以下の手順を実行します。
   a. 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。
   b. DBUtil 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/<*platform_name*> ディレクトリにあります。DBUtil の使用方法については、「securestore.enc ファイルの更新およびトラストストア パスワードの設定」を参照してください。
4. arcotcommon.ini ファイルを保存して閉じます。

odbc.ini ファイルの更新

odbc.ini ファイルには接続パラメータが含まれます。Oracle RAC の場合、Oracle RAC インストールに関連する値を odbc.ini ファイルに指定する必要があります。

以下の手順に従います。

1. CA Strong Authentication をインストールしたシステムで *.ora ファイルを作成します。たとえば、/var/opt/tns.ora です。
2. ファイルに以下の行を入力します。

```
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. *ARCOT_HOME*/odbc32v60wf/odbc.ini ファイルをテキスト エディタで開きます。
5. 必要な DSN セクションについて、以下のパラメータが含まれる行をコメントアウトします。
   - HostName
   - PortNumber
   - SID

   例:

```
#HostName=172.30.251.251
#PortNumber=1521
#SID=an
```

6. 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=/var/opt/tns.ora
   ServerName=fwdincrac
   ```

7. ファイルを保存して閉じます。

Appendix A: CA Adapter 2.2.7 用の追加設定

CA Strong Authentication を Adapter 2.2.7 インスタンスと統合するには、追加の設定を実行する必要があります。このセクションの手順は、CA Strong Authentication コンポーネントがすべてインストールされ、正しく実行されていることを確認した後にのみ実行します。

このセクションには、以下のトピックが含まれています。

CA Adapter 2.2.7 インスタンスの更新

CA Strong Authentication には、Arcot-Adpater-2.2.7-Compatibility-Package.zip ファイルが含まれています。このファイルの内容を既存の Adapter インストール環境にコピーします。

以下の手順に従います。

1. CA Strong Authentication-Adpater-2.2.7-Compatibility-Package.zip ファイルの内容を一時的な場所に抽出します。

   以下にそのファイル構造を示します。

```
arcotsm
    WEB-INF
        lib
            arcot-common.jar
            log4j-1.2.15.jar

arcotafm
    client
        arcotjsclient_jso.js
      vpn
             controller_vpn.jsp
WEB-INF
        classes
            jspStrings_en.properties
dbscripts
    mssql
        arcot-db-config-for-adapter-statemanager.sql
        drop-adapter-statemanager.sql
    oracle
        arcot-db-config-for-adapter-statemanager.sql
        drop-adapter-statemanager.sql
```

2. State Manager を以下のように更新します。
   a. State Manager 2.2.7 を展開したアプリケーション サーバ上の場所に移動し、arcot-common-1.0.9.jar ファイルを削除します。

      たとえば、Apache Tomcat の場合、この場所は *TOMCAT_HOME*\arcotsm\WEB-INF\lib です。

   b. 解凍されたファイル構造から、arcot-common.jar ファイルと log4j-1.2.15.jar ファイルをこの場所にコピーします。

   c. アプリケーション サーバを再起動します。
3. 以下のように Authentication Flow Manager を更新します。
   a. 解凍されたファイル構造から、arcotjsclient_jso.js ファイルを、Authentication Flow Manager 2.2.7 を展開したアプリケーション サーバ上の arcotafm¥client フォルダにコピーします。

      たとえば、Apache Tomcat の場合、この場所は *TOMCAT_HOME*/arcotafm/client です。

   b. 解凍されたファイル構造から、jspStrings_en.properties ファイルを、Authentication Flow Manager 2.2.7 を展開したアプリケーション サーバ上の arcotafm\WEB-INF\classes フォルダにコピーします。たとえば、Apache Tomcat の場合、この場所は *TOMCAT_HOME*/arcotafm/WEB-INF/classes です。
4. データベース スキーマで以下の手順に従います。
   a. State Manager と CA Strong Authentication のインスタンスが異なるデータベースを使用している場合は、State Manager によって使用されるデータベース内の ARCMNDBERRORCODES テーブルをドロップします。

   b. ARCMNDBERRORCODES テーブルを作成して行を挿入するスクリプトの一部を実行します。

LDAP プラグイン登録

このセクションの手順は、以下のシナリオの場合のみ実行します。

- CA Strong Authentication の新規インストールを実行し、Adapter 2.2.7 と共に使用する。
- バージョン 6.2.x から 7.x にアップグレードしたが、LDAP プラグインを登録せず、その後にアップグレードした。
- バージョン 6.2.x から 7.1.01 にアップグレードしたが、以前のリリースでは LDAP プラグインを登録していなかった。

CA Adapter 2.2.7 は、CA Strong Authentication を使用した LDAP 認証を有効にするためにプラグインを使用します。Adapter 2.2.7 で使用される LDAP プラグインが CA Strong Authentication で動作するようにするには、LDAP プラグインを登録する必要があります。

LDAP プラグインの登録

LDAP プラグインは、管理コンソールを使用して登録します。

以下の手順に従います。

1. マスタ管理者として管理コンソールにログインします。
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. CA Strong Authentication サブタブをクリックし、左ペインで[拡張設定]の下の[プラグイン登録]を選択します。

   右ペインに[プラグインの登録]画面が表示されます。
4. 以下のフィールドに適切な値を指定します。
   - 名前: プラグインの名前。
   - ハンドラ パス: arwfldapauthplugin.dll
   - 設定テンプレート: ファイル システム内のファイル ldapauthplugin-config.xml へのパスを選択します。通常、このファイルは *Install_Location*/arcot/samples/xml/webfort ディレクトリにあります。
   - UP_AUTH を[利用可能なイベント]リストから[サポート対象イベント]リストに移動させます。
5. [登録]ボタンをクリックします。

組織用のプラグインの設定

登録済みのプラグインは、別の組織用に設定することができます。

以下の手順に従います。

1. グローバル管理者として管理コンソールにログインします。
2. [組織]タブをクリックし、プラグインを使用する組織を検索します。

   注: ここで選択する組織は、AFM ウィザードで LDAP にマップする必要があります。

   組織の情報画面が表示されます。
3. CA Strong Authentication の設定サブタブをクリックし、左ペインで[拡張設定]の下の[プラグイン設定]を選択します。

   [プラグインの設定]画面が表示されます。
4. [名前]リストから、登録済みの LDAP 認証プラグインを選択します。
5. UP_AUTH を、[サポート対象イベント]リストから[選択したイベント]リストに移動させます。
6. [サブミット]をクリックします。

   プラグインが正常に設定されたことを示すメッセージが表示されます。

Appendix B: IBM WebSphere への管理コンソールの展開

IBM WebSphere 7.0、8.0、および 8.5 に CA Advanced Authentication を展開するには、以下の手順に従います。

1. 作業ディレクトリを、次のディレクトリに変更します。
   `<install_location>/arcot/sbin`
2. 「source arwfenv」と入力し、Enter キーを押して $ARCOT_HOME 環境変数を設定します。
3. 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。
4. CA Advanced Authentication の WAR ファイルがある以下の場所に移動します。
   `<install_location>/arcot/java/webapps`
5. arcotadmin.war ファイルを一時ディレクトリにコピーします。たとえば、opt/arcot_temp などです。
6. arcotadmin.war ファイルの内容を抽出します。

   /opt/arcot_temp/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

7. IBM WebSphere Administration Console にログインします。
8. [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。スコープには、アプリケーションを展開するターゲット サーバ/ノードが含まれる必要があります。
   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: /opt/arcot_temp/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。
9. CA Advanced Authentication の WAR ファイルがある以下の場所に移動します。
   `<install_location>/arcot/java/webapps`
10. アプリケーション サーバに arcotadmin.war を展開します。
11. 以下の手順に従って、共有ライブラリを設定します。
    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. [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。

12. 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。
13. アプリケーションが再起動されたことを確認します。

Appendix C: CA Strong Authentication のエラーのトラブルシューティング

この付録では、CA Strong Authentication の使用時に発生する可能性があるエラーを解決するのに役立つトラブルシューティング手順について説明します。トラブルシューティング トピックは、CA Strong 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（146P. ）
- Database-Related エラー（151P. ）
- CA Strong Authentication サーバ エラー（154P. ）

トラブルシューティング タスクを実行する前に、CA Strong 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/logs/ ディレクトリに保存されます。以下の表に、CA Strong Authentication コンポーネントのデフォルト ログ ファイル名を示します。

CA Strong Authentication コンポーネント	**ファイル名**	**Description**
CA Strong Authentication サーバ	arcotwebfortstartup.log	このファイルには、すべての起動(ブート)アクションが記録されます。CA Strong 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。
	arcotwebfort.log	このファイルには、サーバで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 サービスの操作が記録されます。

注: CA Strong Authentication のログ ファイルの詳細については、「CA Strong Authentication インストールおよび展開ガイド」の「CA Strong Authentication のログ」を参照してください。

インストール エラー

問題:

<install_location>/arcot/java/webapps ディレクトリに arcotadmin.war がありません。

原因

インストール時にファイルが作成されていない可能性があります。

解決方法:

arcotadmin.war 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 ウィンドウを開きます。
2. ARCOT_HOME 環境変数が設定されていることを確認します。
3. *<install_location>*/arcot/tools/common/bundlemanager ディレクトリに移動します。
4. 以下のコマンドで bundlemanager を実行します。
   ```
   java -jar bundle-manager.jar
   ```
   上記のコマンドにより、*<install_location>*/arcot/Java/webapps ディレクトリに arcotadmin.war ファイルが生成されます。

問題:

CA Strong Authentication サーバ（CA Strong Authentication サービス）を起動できません。arcotwebfortstartup.log に以下のエラーが表示されます。

```
Failed to initialize DB Pool Manager
```

または
```
Data source name not found and no default driver specified
```

原因

この問題の考えられる原因は以下のとおりです。

- データベースの DSN がシステム DSN として作成されていない。
- 64 ビットのプラットフォームを使用している。その結果、DSN が 64 ビットの ODBC Manager を使用して作成されている。

解決方法:

arcotcommon.ini ファイルで DSN 関連の問題を確認できます。問題が DSN 関連の場合、以下の手順に従います。

1. 最初の原因を解決するには、DSN がシステム DSN であることを確認する必要があります。以下の手順を実行します。
   a. [コントロール パネル]を開き、[管理ツール]-[データ ソース(ODBC)]に移動します。
   b. [システム DSN]タブをアクティブにし、DSN が存在することを確認します。存在しない場合は、以前と同じ名前で DSN を再作成します。
   c. サービスを再起動します。
2. 2 番目の原因(64 ビットのプラットフォームを使用している場合)を解決するには、ODBC Manager の 32 ビットのバージョンを使用します。

注: arcotcommon.ini およびその他の設定ファイルの詳細については、「CA Strong Authentication インストールおよび展開ガイド」の「設定ファイルおよびオプション」を参照してください。

問題:

CA Strong Authentication サーバ(CA Strong Authentication サービス)を起動できません。エラー メッセージは、サービスが起動し、自動的に停止していることを示しています。

原因

この問題の考えられる原因としては、インストール時にデータベースの詳細を指定したが、データ ソースが正常に作成されなかった可能性があります。

解決方法:

この問題を解決するには、以下の手順に従います。

1. DSN の対応するエントリが arcotcommon.ini にあるかどうかを確認します。
   - エントリがない場合は、手動で DSN を作成します。
2. エントリがある場合は、「データベース スクリプトの実行 (68P.)」の説明に従って、データベースをクリーンアップし(「CA Strong Authentication スキーマのアンインストール」を参照)、データベースを再シードします。
3. CA Strong Authentication サーバを再起動します。

問題

マスタ管理者として管理コンソールを初めて起動した際(「ブートストラップ タスクの実行」を参照)に、以下のメッセージが表示されます。

```
The server encountered an internal error that prevented it from
fulfilling this request.
```

arcotadmin.log ファイルに以下のエラーが記録されています。

```
adminLog: java.lang.UnsatisfiedLinkError: no ArcotAccessKeyProvider
in java.library.path
```

原因

JAVA ライブラリに、以下のいずれかのファイルへのパスが含まれていません。

- libArcotAccessKeyProvider.so
- arcot-crypto-util.jar

解決方法:

以下の手順に従います。

1. LD_LIBRARY_PATH 変数に以下のファイルへの絶対パスが含まれていることを確認します。
   - libArcotAccessKeyProvider.so
   - arcot-crypto-util.jar

2. アプリケーション サーバを再起動します。

問題:

ARCOT_HOME のログ ディレクトリにログ ファイル（arcotadmin.log、arcotuds.log、または webfortserver.log）がありません。

原因

この問題の考えられる原因は以下のとおりです。

- ARCOT_HOME がインストール時に正しく設定されていない。
- アプリケーション サーバの JAVA ホームが、JDK ホームではなく JRE を指していている。

解決方法:

これらの問題を解決するには、以下の手順に従います。

- ARCOT_HOME をリセットして、正しい場所を指すように設定されていることを確認します。通常は、*<installation_location>*/arcot/ を指します。

  この結果として、コマンド プロンプト ウィンドウで cd $ARCOT_HOME コマンドを使用する際には、現在のディレクトリを *<installation_location>*/arcot/ に変更する必要があります。
- アプリケーション サーバの JAVA ホームの場所に libArcotAccessKeyProvider.so ファイルおよび arcot-crypto-util.jar ファイルをコピーしたことを確認します。

問題:

UDS を展開しましたが、起動しません。

原因

考えられる原因の 1 つとして、アプリケーション サーバの JAVA ホームが JDK ホームではなく JRE を指していていることがあります。

解決方法:

アプリケーション サーバの JAVA ホームの場所に libArcotAccessKeyProvider.so ファイルおよび arcot-crypto-util.jar ファイルをコピーしたことを確認します。

問題:

UDS に接続できません。以下のエラー メッセージが表示されます。

```
Unable to contact User Data Service
```

原因

考えられる原因は以下のとおりです。

- UDS の **ホスト**、**ポート**、および**アプリケーション コンテキスト**情報を正しく指定していない可能性があります。
- UDS サービスが初期化されていない可能性があります。

解決方法:

この問題を解決するには、以下の手順に従います。

1. 管理コンソールの[ユーザ データ サービス設定]ページで指定した UDS 情報が正しいかどうかを確認します。[**ホスト**]、[**ポート**]、および[**アプリケーション コンテキスト ルート**]フィールドの詳細は正確である必要があります。
2. UDS ログ ファイルを確認して、サービスが正常に初期化されたことを確認します。

Database-Related エラー

問題:

データベースへの接続に失敗し、CA Strong Authentication サーバ ログ ファイルに以下のエントリが記録されます。

```
ReportError: SQL Error State:08001, Native Error Code: 30FD, ODBC Error: [DataDirect][ODBC Oracle driver][Oracle]ORA-12541: TNS:no listener
```

解決方法:

以下を確認します。

- データベース サーバのリスナ サービス。
- CA Strong Authentication サーバがインストールされているシステムの TNSnames.ora ファイルの設定。

問題:

データベースへの接続に失敗し、CA Strong Authentication サーバ ログ ファイルに以下のエントリが記録されます。

```
TNS:listener could not resolve SERVICE_NAME given in connect descriptor
```

解決方法:

以下を確認します。

- データベースが起動している。起動していない場合に、このメッセージが表示されます。
- データベースが起動している場合は、データベースがまだリスナに登録されていない可能性があります。これは、データベースまたはリスナの起動直後に発生します。通常、この問題は 1 分程度待てば解決します。
- 静的な登録を使用している場合は、接続文字列(TNSNAMES.ORA、NAMES、OID など)で使用される SERVICE_NAME エントリが、リスナが認識している有効なサービスと一致していることを確認します。
- 「C:>tnsping SERVICE_NAME」を使用してステータスを確認でき、「C:>lsnrctl services」を使用してリスナが認識しているすべてのサービスを確認できます。

問題:

データベースへの接続に失敗し、CA Strong Authentication サーバ ログ ファイルに以下のエントリが記録されます。
`Database password could not be obtained from securestore.enc`

原因

データベースの詳細が securestore.enc ファイルに含まれていない可能性があります。

解決方法:

DBUtil ツールを使用して、データベースの詳細を指定して securestore.enc ファイルを更新します。DBUtil の使用方法の詳細については、「CA Strong Authentication 管理ガイド」を参照してください。

問題:

データベースへの接続に失敗し、CA Strong Authentication サーバ ログ ファイルに以下のエントリが記録されます。
`ORA-03113: end-of-file on communication channel`

原因

これは接続が失われたことを示す一般的なエラーです。以下のような原因で発生します。

- ネットワークの問題
- サーバ セッションの強制切断
- Oracle データベースのクラッシュ
- データベース サーバのクラッシュ
- 中断を引き起こす Oracle の内部エラー(ORA-00600 や ORA-07445 など)
- Oracle クライアントまたは TNS レイヤで接続を処理できない

解決方法:

上記のリストにある考えられる原因をチェックします。

CA Strong Authentication サーバ エラー

問題:

CA Strong Authentication サーバを起動しようとしても、起動しません。
arcotwebfortstartup.log の最後の行に以下のエラーが記録されます。

```
WARN  STARTUP      -161388848 00WFMAIN - [11]: Protocol module
[SVRMGMT_WS] received portType error [bind: Address already in use]
```

原因

この問題の考えられる原因は、サーバ管理ポート(デフォルトのポート番号: 9743)が別のプロセスによってすでにホストで開かれているというものです。また、CA Strong Authentication サーバが起動するためには、少なくともサーバ管理ポートが必要です。

解決方法:

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. $ARCOT_HOME/bin に移動します。
3. 以下のコマンドを実行します。

   arwfserver -i
4. 「setsvrmgmtport <*new_port_number*>」を入力します。

   サーバ管理ポートを設定すると、マスタ管理者は管理コンソールにログインして、別のポートを設定できます。

問題:

「Access-Reject」メッセージで RADIUS リクエストが失敗します。

原因

以下を確認します。

- 共有秘密キーが正しく設定されている。
- CA Strong Authentication サーバ ログ内のエラー。

解決方法:

グローバル管理者または組織として管理コンソールにログインし、[RADIUS 設定]ページを使用して共有秘密キーを設定します。

問題:

CA AuthID 認証が失敗し、CA Strong Authentication サーバ ログ ファイルに以下のエントリが記録されます。

```
[Arcot Exception,No valid issuer certificate is available for this certificate: unknown or invalid certificate issuer in ArcotVerifier], Challenge verification failed -
```

原因

ドメイン キーが期限切れになっている可能性があります。

解決方法:

グローバル管理者または組織として管理コンソールにログインし、[認証キー管理設定]ページを使用してドメイン キーを設定します。

Chapter 12: CA Strong Authentication のアンインストール

この手順では、CA Strong Authentication および関連するコンポーネントをアンインストールする手順について説明します。

次の手順に従ってください:

1. 使用しているデータベースタイプに応じて、次のいずれかのフォルダに移動します。

   （Microsoft SQL Server の場合）<*install_location*>/arcot/dbscripts/mssql

   （Oracle データベースの場合）<*install_location*>/arcot/dbscripts/oracle

   （IBM DB2 UDB の場合）<*install_location*>/arcot/dbscripts/db2

   （MySQL の場合）<*install_location*>/arcot/dbscripts/mysql

2. スクリプトを*次に示す順序*で実行します。

   a. drop-webfort-8.0.sql

      注:（MySQL の場合）drop-webfort-8.0.sql の実行時には、「Safe Updates」が無効である必要があります。

   b. drop-arcot-common-8.0.sql

      これでデータベーステーブルがすべて削除されます。

3. CA Strong Authentication サーバを停止します。

4. 以下のディレクトリに移動します。
   ```
   sh <install_directory>/arcot/"Uninstall_CA Strong
   Authentication"/Uninstall CA Strong Authentication
   ```

5. 以下のコマンドを使用して、インストーラを実行します。
   ```
   sh Uninstall CA Strong Authentication
   ```

6. アンインストールのタイプを選択します。

   - 1-Completely remove all features and components: *すべてのコンポーネントを現在のシステムからアンインストールする場合は、このオプションを選択します。*
   - 2-Choose specific features that were installed by InstallAnywhere: *選択したコンポーネントのみを現在のシステムからアンインストールする場合は、このオプションを選択します。*

7. Enter キーを押して続行します。

   すべてのコンポーネントをアンインストールするオプションを選択した場合は、手順 8 に進みます。

   選択したコンポーネントをアンインストールするオプションを選択した場合は、[Choose Product Features]画面が表示されます。

8. 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コンポーネントが表示されます。コンポーネント番号を(カンマで区切って)入力し、Enter キーを押します。

   重要: 特定の機能をアンインストールするには、インストール時に実行したのとは逆の順序で行う必要があります。たとえば、CA Strong Authentication 認証サーバの後に管理コンソールをインストールした場合は、管理コンソールをアンインストールしてから CA Strong Authentication 認証サーバをアンインストールします。

   アンインストールが正常に終了すると、最後に[Uninstall Complete]画面が表示されます。

9. Enter キーを押してウィザードを終了します。

アンインストール後の作業手順

アンインストール後には、以下の作業を実行します。

1. <install_location>/arcot フォルダを削除します。
2. アプリケーション サーバから以下の Web アプリケーションをアンインストールします。
   - arcotadmin: 管理コンソール
   - arcotuds: ユーザ データ サービス
   - ca-strongauth-8.0-sample-application: サンプル アプリケーション
3. .com.zerog.registry.xml ファイルが削除されたことを確認します。この隠しファイルはインストール時にコピーされます。このファイルの場所は、CA Strong Authentication のインストールに使用したユーザ アカウントによって異なります。
   - root ユーザとして CA Strong Authentication をインストールした場合、このファイルは /var ディレクトリにあります。
   - その他のユーザとして CA Strong Authentication をインストールした場合、このファイルはそのユーザのホーム ディレクトリにあります。

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。