第 1 章: 紹介

このセクションには、以下のトピックが含まれています。

概要

CA Embedded Entitlements Manager（CA EEM）を使用すると、共通するアクセス ポリシーの管理、認証、および許可サービスを複数のアプリケーション間で共有できます。

特長

CA EEM は、複数のセキュリティ サービスを提供します。 以下のセキュリティ サービスを使用できます。

- 設定サービス:
  - アプリケーション インスタンスの登録と登録解除
  - アプリケーション管理者の管理上のスコープ設定
  - 管理権限の委任
  - ユーザおよびグループの管理
- 管理セキュリティ サービス:
  - アクセス、イベント、責任ポリシーの管理
  - カレンダの管理
- ランタイム セキュリティ サービス:
  - ユーザの認証
  - アクセスの許可
  - セキュリティ イベントのログ記録

機能

CA EEM は以下の機能で構成されています。

全般

- ポリシーが分離されているため、登録された各アプリケーション インスタンスは専用の領域を使用してそのアプリケーション固有のデータを格納できます。
- Java、C++、および C# 向けのランタイム SDK
- Java、C++、および C# 向けの管理 SDK
- 管理機能(オブジェクトの挿入/変更/削除)のコマンド ライン インターフェース サポート:
  - XML のインポートおよびエクスポート
  - ランタイム チェック
  - 移行ツール
- スタンドアロンおよびコンテキスト起動によるアクセス
- Secure HTTP 通信
- CA Security Command Center および CA Audit との統合によるセキュリティ イベントの管理
- CA SiteMinder との統合による、CA SiteMinder データ ストアからのユーザおよびグループ情報の取得

ID 管理

- 全アプリケーション間でのグローバル ユーザおよび属性の共有
- グローバル ユーザの複数モード サポート
  - パスワード ポリシー管理機能を備えた内部グローバル ユーザ
  - LDAP ディレクトリ サーバからの外部グローバル ユーザ
  - CA Identity Manager からの外部グローバル ユーザ
- CA Identity Manager との統合によるロール ベースのユーザ プロビジョニングと管理
- シングル サインオンのためのポータブルなセッションのエクスポートおよびインポートのサポート

アクセス管理

- アクセス制御リスト(ACL)とビジネス ポリシーの両方に対応するアクセス管理
- ポリシー言語により、ポリシーの決定時にユーザ、セッション、環境およびリソースの属性を使用可能
- 全オブジェクトを対象とする組み込みの管理スコープ機能

- 組み込みの管理委任サポート機能
- アプリケーション固有のアクションを必要なカスタム責任チェックの組み込みサポート
  - 権限チェックのローカル インプロセス評価
  - アクセス ポリシー、ACL、管理スコープ ポリシー、委任権限を定義するための SDK および Web インターフェース

クライアント アクセス

CA EEM サーバには、サードパーティ統合を実現する標準的な Web および Web サービス インターフェースを介してアクセスでき、クライアント モジュールは必要ありません。 インターフェースは以下のとおりです。

- 設定と管理用の HTML および iTechnology
- CA Audit イベント配信用の iTechnology

iTechnology は、HTTP、HTTPS、HTML、XML、および SSL などの Web 標準に基づいた CA の技術です。 iTechnology は、フレームワークを提供することによって、インターネット全体に Web サービスを作成して展開できるようにします。

データ ストア サポート

CA EEM では、Microsoft Active Directory など、単一の外部ユーザ ソースを指定できます。 ユーザ オブジェクトの格納場所にかかわらず、CA EEM はその設定情報とポリシーを CA Directory に格納します。

第 2 章: Windows へのインストール

このセクションには、以下のトピックが含まれています。

インストールの概要

CA EEM を Windows 動作環境にインストールする場合は、以下のアプリケーションをインストールする必要があります。

CA EEM サーバ

CA EEM サーバでは、Web インターフェースを使用してアプリケーション リソースに関する許可ポリシーを定義できます。 Web ベースの管理インターフェースを使用して、ID およびアクセス ポリシーの管理が可能です。 既存のセキュリティ インフラストラクチャを利用し、一元化されたユーザ ストアおよびその他のエンタープライズ システムで定義したリソースやユーザ属性を使用して、ビジネス ロジックに基づくルールを実装します。

CA EEM Software Development Kit(SDK)

CA EEM SDK を使用すると、ID ベースのセキュリティ制御をアプリケーションに組み込むことができます。 SDK はライブラリ、Java クラス、ヘッダ ファイル、およびチュートリアルで構成されます。 SDK を使用して任意のアプリケーションに CA EEM を実装できます。 SDK を使用した CA EEM の実装方法の詳細については、「**プログラミング ガイド**」を参照してください。

各アプリケーションは、個別にインストールする必要があり、それぞれ独立して動作します。

サーバのインストール

CA EEM サーバは、インストール ウィザードか、コマンドラインからインストールすることができます。 CA EEM をサイレント モードでインストールする場合はコマンドラインを、対話形式でインストールする場合はインストール ウィザードをそれぞれ使用します。

JRE は、CA EEM のインストールや使用に必要な最小要件ではなくなりました。 JRE の有無に関係なく、CA EEM をインストールしたり使用したりすることができます。 最小要件として JRE を含まずに CA EEM をインストールする場合は、インストール ウィザードの JRE 選択パス画面をスキップしてください。 JRE を含まずに、サイレントモードで CA EEM サーバをインストールする場合は、javahome パラメータを「None」に設定してください。

以下のセクションでは、CA EEM サーバのインストール方法について説明します。

詳細情報

インストール ウィザードの JRE パス選択画面をスキップする方法 (P. 7)
ウィザード設定チェックリスト (P. 6)
インストール ウィザードを使用したサーバのインストール (P. 8)
サイレント モードによる CA EEM サーバのインストール (P. 15)

ウィザード設定チェックリスト

Windows に CA EEM サーバをインストールする場合は、以下の情報が必要です。

フィールド	値
CA EEM インストール パス	CA EEM をインストールする予定のコンピュータ上の場所
JRE インストール パス	コンピュータ内の JRE のインストール場所 **注:** JRE がインストールされていない状態で CA EEM をインストールおよび使用する場合は、コマンド ラインから Javahome 変数を「None」に設定してから CA EEM インストール ウィザードを実行してください。
EiamAdmin のパスワード	CA EEM の管理者 EiamAdmin と関連付けられているパスワード
バックアップ ディレクトリ	CA EEM の旧バージョンのファイルをバックアップする予定のコンピュータの場

フィールド	値
	所。 **注:** この情報は、CA EEM の以前のリリースを現在のリリースにアップグレードする場合にのみ必要です。

インストール ウィザードの JRE パス選択画面をスキップする方法

JRE は、CA EEM のインストールや使用に必要な最小要件ではなくなりました。 JRE がインストールされていない状態で CA EEM をインストールするには、以下のことを行う必要があります。

1. javahome パラメータを「None」に設定します。

   注: javahome パラメータを「None」に設定すると、インストール ウィザードで[Java のパス]選択画面は表示されなくなります。

2. インストール ウィザードを使用して CA EEM をインストールします。

Javahome パラメータの設定

インストール ウィザードで CA EEM をインストールする前に、javahome パラメータを「none」に設定する必要があります。 javahome パラメータを設定するには、コマンド ラインから以下のコマンドを入力します。

```
EEMServer_[version number]_win32.exe -s -a /z"javahome=None; "
```

インストール ウィザードを使用したサーバのインストール

CA EEM サーバのインストール ウィザードを使用すると、順を追ってインストールを進めることができ、インストール パラメータを定義するためのオプションも提示されます。

CA EEM サーバをインストールするには、以下の手順に従います。

1. 以下のいずれかの操作を実行します。
   - インストール先のコンピュータ上で Windows エクスプローラを起動して、インストール パッケージの EEMServer_[releasenumber].[build_number]_win32.exe をダブルクリックします。
   - コマンド プロンプトで、インストール パラメータを使用して以下のコマンドを入力します。

     ```
     EEMServer_[releasenumber].[build_number]_win32.exe -s -a /z "eiampath=<Custom installation path for CA EEM>;etdirpath=<Custom installation path for CA Directory>;igpath=<Custom installation path for iGateway>;"
     ```

     インストール パラメータを使用してカスタム インストール パスを指定できます。 インストールパラメータの詳細については、「サーバのインストール パラメータ」を参照してください。

     インストール ウィザードが表示されます。
2. インストール ウィザードの指示に従ってインストールを完了します。

詳細情報:

インストール ウィザードの JRE パス選択画面をスキップする方法 (P. 7)

サーバのアップグレード

既存の CA EEM サーバ システムを最新バージョンにアップグレードできます。

既存の CA EEM サーバ システムをアップグレードするには、以下の手順に従います。

1. アップグレード対象のコンピュータ上で EEMServer_<version number>_win32.exe を実行します。
2. インストールされている CA EEM サーバのバージョンに応じて、以下のいずれかの処理が行われます。
   - 既存の CA EEM サーバ バージョンがインストール中のサーバ バージョンより古い場合は、インストール ウィザードが自動的に既存のバージョンをバックアップし、新しいバージョンにアップグレードします。
   - 既存の CA EEM サーバ バージョンがインストールするサーバと同じバージョンの場合は、インストール ウィザードによって、CA EEM サーバをアンインストールするかどうか確認を求められます。 CA EEM サーバをアンインストールして再インストールできます。
   - インストールするサーバ バージョンが既存のサーバ バージョンより古い場合は、インストール ウィザードにエラーが表示され、インストール処理は終了します。

CA EEM サーバをアップグレードすると、以下の項目が更新されます。

- \\CA\SharedComponents\iTechnology フォルダの CA EEM サーバ
- iGateway
- CA Directory

また、P12 証明書はすべて PEM 証明書にマイグレートされます。

詳細情報:

ウィザード設定チェックリスト (P. 6)
インストール ウィザードを使用したサーバのインストール (P. 8)

サーバの起動

登録されたアプリケーションの ID とアクセス ポリシーを管理するには、CA EEM サーバを起動する必要があります。

CA EEM サーバを使用して起動するには、以下の手順に従います。

1. 以下のいずれかの操作を実行します。
   - ブラウザに「https://hostname」または「ipaddress:5250/spin/eiam」と入力します。CA EEM サーバ コンピュータで作業している場合は、「http://localhost:/5250/spin/eiam」と指定します。
   - Windows 動作環境では、[スタート]-[プログラム]-[CA]-[Embedded Entitlements Manager]-[EEM UI]を選択します。

     ログイン ページが表示されます。
2. このログイン ダイアログ ボックスに以下の情報を入力します。
   a. アプリケーション ドロップダウン リストから、登録したアプリケーション インスタンスを選択します。 デフォルトは、<グローバル> です。 デフォルトの管理者ユーザ名は EiamAdmin です。

      注: ログイン用に他のグローバル ユーザを追加し、基本設定に従いそのユーザ名を設定することができます。

   b. パスワードを入力します。 ここには、CA EEM サーバのインストール時に EiamAdmin 用に指定したパスワードと同じものを入力してください。

   c. [設定を保存する]ボックスをオンにすると、次回のログイン時に同じ設定で CA EEM サーバにログインできます。
3. [ログイン]をクリックします。

   CA EEM インターフェースのホームページが開きます。 CA EEM サーバの使用方法の詳細については、オンライン ヘルプを参照してください。

CA EEM サーバでのアクセシビリティの有効化

CA EEM サーバのアクセシビリティ機能を使用すると、さまざまな利用環境で、お客様が製品とサポート ドキュメントを正しく使用して重要なビジネス業務を遂行することができます。 アクセシビリティを有効にすると、キーボードのみまたは画面リーダを使用し、重要なビジネス業務を実行することができます。

アクセシビリティを有効にする方法

1. 以下のいずれかの操作を行います。
   - ブラウザに「https://hostname」または「ipaddress:5250/spin/eiam」と入力します。 CA EEM サーバ コンピュータで作業している場合は、「http://localhost:/5250/spin/eiam」と指定します。
   - Windows 動作環境では、[スタート]-[プログラム]-[CA]-[Embedded Entitlements Manager]-[EEM UI]を選択します。

     ログイン ページが表示されます。
2. このログイン ダイアログ ボックスに以下の情報を入力します。
   a. アプリケーション ドロップダウン リストから、登録したアプリケーション インスタンスを選択します。 デフォルトは、<グローバル>です。 デフォルトの管理者ユーザ名は EiamAdmin です。

      注: ログイン用に他のグローバル ユーザを追加し、基本設定に従いそのユーザ名を設定することができます。

   b. パスワードを入力します。 ここには、CA EEM サーバのインストール時に EiamAdmin 用に指定したパスワードと同じものを入力してください。

   c. [設定を保存する]ボックスをオンにすると、次回のログイン時に同じ設定で CA EEM サーバにログインできます。
3. [アクセシビリティの有効化]をクリックします。

   CA EEM GUI でアクセシビリティが有効になります。
4. [ログイン]をクリックします。

   CA EEM インターフェースのホームページが開きます。

サーバの削除

CA EEM サーバは、[コントロール パネル]の[プログラムの追加と削除]を使用してアンインストールできます。

注: CA EEM に登録されたアプリケーションがある場合は、CA EEM サーバは削除できません。 CA EEM サーバを削除する前に、アプリケーションの登録を解除する必要があります。 アプリケーションの登録解除については、**オンライン ヘルプ**を参照してください。

SDK のインストール

CA EEM SDK 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。

CA EEM SDK をインストールするには、以下の手順に従います。

1. Windows エクスプローラを起動して、インストール パッケージ EEMSDK_<version number>_win32.exe をダブルクリックするか、またはコマンド プロンプトからインストール ファイルを実行します。

   インストール ウィザードが表示されます。

2. 条件に同意する場合は、[同意する]をクリックします。

   注: [同意する]ボタンは、条件のテキストをスクロールして読まないと使用できません。

   [インストール先の選択]ダイアログ ボックスが表示されます。 デフォルトでは、CA EEM SDK はインストール ウィザードによって C:¥Program Files¥CA¥Embedded IAM SDK にインストールされます。

3. [次へ]をクリックします。

   または

   [参照]をクリックして CA EEM SDK をインストールするコンピュータ上のディレクトリを選択し、[次へ]をクリックします。

   これにより、CA EEM SDK のインストールが開始されます。

4. [完了]をクリックします。

   これで、CA EEM SDK がインストールされます。

   注: 環境変数 %EIAM_SDK% はインストール中に作成され、インストール パスを指します。 エクスプローラのパスにこの環境変数を使用して、インストール フォルダを開いてください。

SDK の起動

CA EEM SDK を起動するには、[スタート]-[プログラム]-[CA]-[Embedded Entitlements Manager]-[EEM SDK]をクリックします。

CA EEM SDK ドキュメント ウィンドウが開きます。

SDK の削除

CA EEM SDK は、[コントロール パネル]の[プログラムの追加と削除]を使用してアンインストールできます。

サーバのインストール パラメータ

Windows に CA EEM をインストールする場合は、以下のコマンド ライン パラメータに関する情報を収集する必要があります。

-eiampath

CA EEM サーバのインストール先のパスを指定します。 デフォルトは、C:¥Program Files¥CA¥SharedComponents¥Embedded IAM です。

-etdirpath [path]

CA Directory のインストール先のパスを指定します。 デフォルトは、C:¥Program Files¥CA¥Directory です。

-igpath [path]

iGateway のインストール先のパスを指定します。 デフォルトは、C:¥Program Files¥CA¥SharedComponents¥iTechnology です。

backupdir

既存のシステムのデータをバックアップする場所を指定します。

-capkiinstalldir

CAPKI モジュールのインストール フォルダのパスを指定します。 デフォルトは、C:¥Program Files¥CA¥SC¥CAPKI です。

-javahome [ディレクトリ]

iGateway インストーラを呼び出す際、[directory] に JAVA_HOME 変数を設定します。 CA EEM インストーラでは、すでに設定されている場合でも、変数を設定するように求めるメッセージが表示されます。 このパラメータにはデフォルト値はありません。

注: java を含めずに CA EEM をインストールする場合は、javahome を none に設定する必要があります。

CA Directory のインストール時は、CA EEM は以下のパラメータを使用します。 必要に応じて、パラメータを設定することができます。

重要: デフォルトのポート番号をカスタマイズする前に、他のサービスが同じポートを使用するように設定されていないことを確認してください。

-dxadminport

DXadmind が DXmanager のリクエストをリスンするポートを指定します。 このポートは DXadmind と DXmanager の LDAP 通信に使用されます。 DXadmind は DSA を含む各ホスト上で実行されているバックグラウンド プロセスです。 DXmanager は、DSA と通信するために DXadmind を使用します。

デフォルト: 2123

-dsaport

dsa がリクエストのリスニングに使用するポートを指定します。

デフォルト: 509

-ssldport

CA Directory が SSLD サーバのリスニングに使用するポートを指定します。 SSLD サーバは、SSL と TLS の認証、および CA Directory の暗号化と復号化を処理するバックグラウンド プロセスです。

デフォルト: 21847

-routerport

dsa がルータ dsa との接続に使用するポートを指定します。 ルータ DSA には、ローカル データもデータストアもなく、 他の DSA へトラフィックをルーティングすることのみが可能です。

デフォルト: 1684

-dxdbsize

CA EEM のデータストアの最大サイズを指定します。

デフォルト: 500 MB

-dxuser

CA Directory をインストール、管理、およびアンインストールできる非 DSA ユーザを指定します。 dxuser には、ローカル システム ユーザもネットワーク ユーザも指定することができます。

注: dxuser としてローカル システム ユーザを使用して CA Directory をインストールした場合は、アンインストール時にそのユーザは削除されます。 そのため、CA Directory のインストールに dxuser としてローカル システム ユーザを使用する場合は、そのユーザが他のプログラムを実行するように設定されていないことを確認してください。

注: Microsoft Windows Server 2003 がインストールされているコンピュータでは、コマンド プロンプトで使用できる文字列の最大長は 8,191 文字です。 Microsoft Windows 2000 では、コマンド プロンプトで使用できる文字列の最大長は 2,047 文字です。 InstallShield のコマンドの長さの詳細については、「**リリース ノート**」を参照してください。

サイレント モードによる CA EEM サーバのインストール

サイレント モードで CA EEM サーバをインストールするには、以下の 2 つのタスクが必要になります。

1. 応答ファイルを作成する。
2. 応答ファイルを指定するコマンドを実行する。

   サイレント インストールの場合、インストール エラーを記録するためにログファイル eiaminstall.log が作成されます。

注: CA EEM サーバをサイレント インストールした場合は、サイレント モードで削除することもできます。

応答ファイルの作成

インストール時の入力内容を応答ファイルに記録しておき、そのファイルを使用して CA EEM サーバをサイレント インストールすることができます。 インストールする各ビルドごとに新しい応答ファイルを作成する必要があります。

応答ファイルを作成するには、以下の手順に従います。

1. インストール先のコンピュータで、CA EEM サーバのインストール パッケージを実行します。
2. コマンド プロンプトで以下のコマンドを入力すると、指定したディレクトリに応答ファイルが作成されます。

   ```
   EEMServer_[releasenumber].[build_number]_win32.exe -s -a /r /f1"pathname of response file"
   ```

   例:

   ```
   EEMServer_8.4.0.55_win32.exe -s -a /r /f1"c:\resp.iss"
   ```

3. インストール パラメータの値を入力します。この値は応答ファイルに保存されます。

応答ファイルを指定するコマンドの実行

以下の例は、サイレント インストールを実行するためのオプションを示しています。

- CA EEM サーバをサイレント モードでインストールするには、コマンド プロンプトで以下のコマンドを入力します。

  ```
  EEMServer_[releasenumber].[build_number]_win32.exe -s -a /s /f1"pathname of response file"
  ```

  例:

  ```
  EEMServer_8.4.0.55_win32.exe -s -a /s /f1"c:\resp.iss"
  ```

- CA EEM サーバのサイレント インストール時にインストール ログ ファイルを作成するには、コマンド プロンプトで以下のコマンドを入力します。

  ```
  EEMServer_[releasenumber].[build_number]_win32.exe -s -a /s /v"/qn /L*v <path to create log file>" /f1"pathname of response file"
  ```

  例:

  ```
  EEMServer_8.4.0.55_win32.exe -s -a /s /v"/qn /L*v c:\install.txt" /f1"c:\resp.iss"
  ```

これで、CA EEM サーバは、指定した応答ファイルを使用してサイレント モードでインストールされます。

注: インストール スクリプトを記述する際に、インストール パラメータを同時に指定することができます。 パラメータの詳細については、「サーバのインストール パラメータ」を参照してください。

サイレント モードによる CA EEM サーバの削除

製品を正しく削除するには、CA EEM サーバと同じビルドから作成された応答ファイルを使用する必要があります。 CA EEM サーバをサイレント モードでアンインストールするには、コマンド プロンプトから以下のコマンドを入力します。

```
EEMServer_[releasenumber].[build_number]_win32.exe -s -a /s /f1”pathname of response file” /z"uninstall"
```

例:

```
EEMServer_8.4.0.55_win32.exe -s -a /s /f1”c:\resp.iss” /z"uninstall"
```

これで、CA EEM サーバがサイレント モードで削除されます。

注: CA EEM に登録されたアプリケーションがある場合は、CA EEM サーバは削除できません。 CA EEM サーバを削除する前に、アプリケーションの登録を解除する必要があります。 アプリケーションの登録解除については、**オンライン ヘルプ**を参照してください。

第 3 章: Linux および UNIX へのインストール

このセクションには、以下のトピックが含まれています。

インストールの概要

Linux および UNIX 動作環境に CA EEM をインストールするには、以下のアプリケーションをインストールする必要があります。

CA EEM サーバ

CA EEM サーバでは、Web インターフェースを使用してアプリケーション リソースに関する許可ポリシーを定義できます。 Web ベースの管理インターフェースを使用して、ID およびアクセス ポリシーの管理が可能です。 既存のセキュリティ インフラストラクチャを利用し、一元化されたユーザ ストアおよびその他のエンタープライズ システムで定義したリソースやユーザ属性を使用して、ビジネス ロジックに基づくルールを実装します。

CA EEM Software Development Kit(SDK)

CA EEM SDK を使用すると、ID ベースのセキュリティ制御をアプリケーションに組み込むことができます。 SDK はライブラリ、Java クラス、ヘッダ ファイル、およびチュートリアルで構成されます。 SDK を使用して任意のアプリケーションに CA EEM を実装できます。 SDK を使用した CA EEM の実装方法の詳細については、「**プログラミング ガイド**」を参照してください。

各アプリケーションは、個別にインストールする必要があり、それぞれ独立して動作します。

サーバのインストール

Linux および UNIX 用の CA EEM サーバは、自己解凍式のシェル スクリプトを使用します。このスクリプトに従ってインストール処理を進めることができます。 インストール中に、スクリプトによってライセンス情報が表示され、インストール パラメータを指定するよう求められます。 インストール パラメータを入力すると、インストールが開始されます。

Linux および UNIX 用 CA EEM サーバをインストールするには、以下の手順に従います。

1. インストール先のコンピュータで、インストール スクリプト EEMServer_[releasenumber].[build_number]_[name of operating system].sh を実行します。

   例:

   ```
   EEMServer_8.4.0.55_sunos.sh
   ```

   ファイルが解凍され、インストールが開始されます。

2. 使用許諾契約書の条件に同意する場合は「Y」を入力します(同意せずにインストールを中止する場合は「N」を入力します)。

   インストール パラメータを入力するように要求されます。

3. インストール パラメータを入力します。

   注: 使用可能なインストール パラメータについては、「サーバのインストール スクリプト パラメータ」を参照してください。

   例:

   a. CA EEM サーバのインストール パスを入力します(または、デフォルト値をそのまま使用します)。

   入力したインストール パラメータ値を示す確認画面が表示されます。

4. 確認画面の情報が正しければ、「Y」を入力してインストールを続行します(「N」と入力すると、インストールは終了します)。

5. EiamAdmin のパスワードを入力します。

   注: デフォルトの管理者ユーザ名は EiamAdmin です。

   インストール手順は、コマンド ライン パラメータと、インストールする CA EEM サーバ パッケージのタイプによって異なります。

   インストーラ スクリプトにより、コンピュータへの CA EEM サーバのインストールが最後まで実行されます。

サーバのアップグレード

既存の CA EEM サーバ システムを最新バージョンにアップグレードできます。

既存の CA EEM サーバ システムをアップグレードするには、以下の手順に従います。

1. インストール先のコンピュータで、インストールスクリプト EEMServer_[releasenumber].[build_number]_[name of operating system] を実行します。
2. インストールされている CA EEM サーバのバージョンに応じて、以下のいずれかの処理が行われます。
   - 既存の CA EEM サーバ バージョンがインストール中のサーバ バージョンより古い場合は、インストール ウィザードが自動的に新しいバージョンにアップグレードします。
   - 既存の CA EEM サーバ バージョンがインストールするサーバと同じバージョンの場合は、インストール ウィザードによって、CA EEM サーバをアンインストールするかどうか確認を求められます。 CA EEM サーバをアンインストールして再インストールできます。
   - インストールするサーバ バージョンが既存のサーバ バージョンより古い場合は、インストール ウィザードにエラーが表示され、インストール処理は終了します。

   CA EEM サーバのインストール方法の詳細については、「サーバのインストール」(P. 20)を参照してください。

CA EEM サーバをアップグレードすると、以下の項目が更新されます。

- \\CA\SharedComponents\iTechnology フォルダの CA EEM サーバ
- iGateway
- CA Directory

サーバの削除

CA EEM サーバを削除するには、インストール ディレクトリから eiamuninstall.sh スクリプトを実行します。

注: CA EEM に登録されたアプリケーションがある場合は、CA EEM サーバは削除できません。 CA EEM サーバを削除する前に、アプリケーションの登録を解除する必要があります。 アプリケーションの登録解除については、**オンライン ヘルプ**を参照してください。

SDK のインストール

Linux および UNIX 用の CA EEM SDK は、自己解凍式のシェル スクリプトを使用します。このスクリプトに従ってインストール処理を進めることができます。 インストール中に、スクリプトによってライセンス情報が表示され、インストール パラメータの指定を求められます。 インストール パラメータを入力すると、インストールが開始されます。

Linux および UNIX 用 CA EEM SDK をインストールするには、以下の手順に従います。

1. インストール先のコンピュータで、インストールスクリプト EEMSDK_[releasenumber].[build_number]_[name of operating system].sh を実行します。

   例:

   ```
   EEM_8.4.0.55_sunos.sh
   ```

   ファイルが解凍され、インストールが開始されます。
2. 使用許諾契約書の条件に同意する場合は「Y」を入力します(同意せずにインストールを中止する場合は「N」を入力します)。
3. CA EEM SDK のインストール パスを入力します(または、デフォルト値をそのまま使用します)。
4. インストールする製品を選択します。

   CA EEM SDK がコンピュータにインストールされます。

CA EEM SDK の起動

CA EEM SDK を起動するには、Web ブラウザで「/opt/CA/eIAMSDK/Doc/index.html」(または、CA EEM SDK のインストール先)にアクセスします。

SDK の削除

Linux および UNIX オペレーティング システムから CA EEM SDK を削除できます。

CA EEM SDK を削除するには、以下の手順に従います。

1. インストール先のコンピュータで、インストールスクリプト EEMSDK_[releasenumber].[build_number]_[name of operating system].sh を実行します。

   例:

   ```
   EEM_8.4.0.55_sunos_linux.sh
   ```

   ファイルが解凍されます。

2. [製品のアンインストール/削除]を選択します。

   インストール スクリプトにより、CA EEM SDK が削除されます。

サーバのインストール スクリプト パラメータ

CA EEM をインストールする際、インストール中にスクリプトによってプロンプトが表示され、以下のコマンド ライン パラメータに関する情報を収集する必要があります。

スクリプトは、以下のコマンド ライン パラメータを受け付けます。

backupdir

既存のシステムのデータをバックアップする場所を指定します。

-capkiinstalldir

CAPKI モジュールのインストール フォルダのパスを指定します。

デフォルト: /opt/CA/SharedComponents/capki

CA Directory のインストール時は、CA EEM は以下のパラメータを使用します。 必要に応じて、パラメータを設定することができます。

重要: デフォルトのポート番号をカスタマイズする前に、他のサービスが同じポートを使用するように設定されていないことを確認してください。

-dxadminport

DXadmind が DXmanager のリクエストをリスンするポートを指定します。 このポートは DXadmind と DXmanager の LDAP 通信に使用されます。 DXadmind は DSA を含む各ホスト上で実行されているバックグラウンド プロセスです。 DXmanager は、DSA と通信するために DXadmind を使用します。

デフォルト: 2123

-dsaport

dsa がリクエストのリスニングに使用するポートを指定します。

デフォルト: 509

-ssldport

CA Directory が SSLD サーバのリスニングに使用するポートを指定します。 SSLD サーバは、SSL と TLS の認証、および CA Directory の暗号化と復号化を処理するバックグラウンド プロセスです。

デフォルト: 21847

-routerport

dsa がルータ dsa との接続に使用するポートを指定します。 ルータ DSA には、ローカル データもデータストアもなく、 他の DSA へトラフィックをルーティングすることのみが可能です。

デフォルト: 1684

-dxdbsize

CA EEM のデータストアの最大サイズを指定します。

デフォルト: 500 MB

-dxuser

CA Directory をインストール、管理、およびアンインストールできる非 DSA ユーザを指定します。 dxuser には、ローカル システム ユーザもネットワーク ユーザも指定することができます。

注: dxuser としてローカル システム ユーザを使用して CA Directory をインストールした場合は、アンインストール時にそのユーザは削除されます。 そのため、CA Directory のインストールに dxuser としてローカル システム ユーザを使用する場合は、そのユーザが他のプログラムを実行するように設定されていないことを確認してください。

-eiamadminpw [password]

[password] の部分に、EiamAdmin パスワードを設定します。

-eiampath

CA EEM サーバのインストール パスを指定します。 デフォルトは、「/opt/CA/SharedComponents/EmbeddedIAM」です。

-etdirpath [path]

CA Directory のインストール パスを指定します。

-igpath [directory]

iGateway パスを設定します。 -iisystem のように完全修飾パスを指定する必要があります。 デフォルトは /opt/CA/SharedComponents/iTechnology です。

-javahome [directory]

JAVA_HOME を設定します。 このパラメータは、デフォルトでは JAVA_HOME 環境変数のコンテンツに設定されています。$JAVA_HOME が設定されていない場合にのみ入力を要求されます。

注: java を含めずに CA EEM をインストールする場合は、javahome=none に設定する必要があります。 これは HP-UX には適用されません。

-logfile [filename]

インストーラがログ情報を [ファイル名] に書き込むように指定します。デフォルトは /tmp/eiam-install.log です。

-silent

サイレント モードでインストールを実行します。 必要なパラメータがコマンド ラインで指定されていない場合、インストールは終了して適切なメッセージを出力します。必要なパラメータがすべて指定されない限り、システムに変更は適用されません。

-tempdir [directory]

一時ファイルを格納するディレクトリを指定します。 デフォルトは /tmp/eiam_temp です。 これは完全修飾パスで、固有のサブディレクトリに格納されている必要があります。 このスクリプトでは、ここで指定されるディレクトリをスクリプトの完了時に削除するために、rm -rf を使用します。

サイレント モードでのサーバのインストール

Linux または UNIX に CA EEM サーバをサイレント モードでインストールするには、コマンド プロンプトで以下のコマンドを入力します。

```
EEMServer_[releasenumber].[build_number]_[name of operating system].sh -silent
-eiamadminpw password -javahome directory
```

例: Sun の動作環境用の以下のコマンドに含まれているのは、最低限必要なパラメータです。

```
EEMServer_8.4.0.55_sunos.sh -silent -eiamadminpw password -javahome directory
```

追加のインストール パラメータを指定することもできます。 ほとんどのインストール パラメータにはデフォルト値があります。 スクリプト パラメータの詳細については、「サーバのインストール スクリプト パラメータ」を参照してください。

ファイルが解凍され、インストールが開始されます。

サイレント モードによる CA EEM サーバの削除

CA EEM サーバを削除するには、インストール ディレクトリから eiamuninstall.sh -silent を実行します。

注: 登録されたアプリケーションがある場合は、CA EEM サーバは削除できません。アンインストールを正しく完了するには、すべてのアプリケーションの登録を解除する必要があります。 アプリケーションの登録解除の詳細については、「オンライン ヘルプ」を参照してください。

第 4 章: CA EEM SDK の設定

CA EEM SDK を使ったアプリケーションの構築に必要な新しいバイナリ

アプリケーションに CA EEM SDK r8.4 SR02 を組み込むには、CA EEM SDK DLL に加えて、以前のリリースから新しくなった以下のバイナリを使用する必要があります。

Java

- xml-apis.jar

C++

EIAMSDK/lib/$OS フォルダから、各オペレーティング システムに合わせて以下のファイルをコピーします。

Windows

- log4cxx.dll
- log4cxx.lib
- libexpat-2.0.1.dll
- libexpat-2.0.1.lib

HP-UX

- *log4cxx*（liblog4cxx.sl、liblog4cxx.sl.10、liblog4cxx.sl.10.0 など）
- libapr*（libapr-1.sl.3、libaprutil-1.sl.3、libapr-1.sl.3.3、libaprutil-1.sl.3.4 など）
- libexpat-2.0.1.sl

UNIX（HP-UX を除く）

- *log4cxx*（liblog4cxx.so、liblog4cxx.so.10、liblog4cxx.so.10.0 など）
- libexpat*

新しい Java バイナリを使用して、アプリケーションを構築する方法

1. xml-apis.jar への参照を付けて ClassPath を更新します。
2. 新しいバイナリおよびロガー設定ファイルをパッケージするためにインストーラを更新します。
3. CA EEM SDK バイナリと一緒に新しいバイナリおよびロガー設定ファイルを展開します。

CA EEM C++ SDK を使用してアプリケーションを実行するために必要なファイル

CA EEM C++ SDK を使用し、アプリケーションを組み込んで実行するには、以下のバイナリが必要です。

Windows

- ipthread.dll
- libcurl_7_18_2.dll
- libexpat-2.0.1.dll
- log4cxx.dll
- msvcm80.dll
- msvcm90.dll
- msvcp71.dll
- msvcp80.dll
- msvcp90.dll
- msvcr70.dll
- msvcr71.dll
- msvcr80.dll
- msvcr90.dll
- pcre.dll
- pthread.dll
- pthreadVCE.dll
- xerces-c_2_8.dll
- zlib.dll
- Microsoft.VC80.CRT.manifest
- Microsoft.VC90.CRT.manifest

HP-UX

- liblog4cxx.sl、liblog4cxx.sl.10、liblog4cxx.sl.10.0
- libapr-1.sl.3、libapr-1.sl.3.3、libaprutil-1.sl.3.4
- libexpat-2.0.1.sl、libxerces-c.sl.28、libcurl.sl.4、libpcre.sl.0、libexpat.sl.2 libz.sl、liblog4cxx.sl.10.0

Linux

- libxerces-c.so.28
- libcurl.so.4
- linux_k24 用の libexpat.so.2 および linux_26 用の libexpat-2.0.1.so
- libpcre.so.0
- libz.so.1
- liblog4cxx.so.10.0.0

AIX

- libxerces-c28.a libcurl.4.so libexpat-2.0.1.a libpcre.a libz.so
- liblog4cxx.a

Sun Solaris

- libxerces-c.so.28
- libcurl.so.4 libpcre.so
- libexpat.so.2 libz.so
- liblog4cxx.so.10.0.0

新しい C++ バイナリを使用して、アプリケーションを構築する方法

1. CA EEM SDK に付属している新しいライブラリをインクルードするには、作成する makefile に以下の行を追加します。

   ```
   -llog4cxx -llibexpat
   ```

2. 新しいバイナリ、eiam.config ファイルおよび eiam.log4cxx.config ファイルをパッケージするためにインストーラを更新します。
3. CA EEM SDK バイナリと一緒に新しいバイナリ、eiam.config ファイルおよび eiam.log4cxx.config ファイルを展開します。

注: ソース コードにロガー ヘッダ ファイルを入れる必要はありません。

CA EEM C# SDK を使用してアプリケーションを構築および実行するために必要なファイル

CA EEM C# SDK を使用して、アプリケーションを組み込んで実行するには、以下のバイナリが必要です。

- log4net.dll
- CPoz.dll
- iclient.dll
- CsharpSDK.dll

注: CAPICOM.dll と InterOP.CAPICOM.dll は、C# SDK を使用してアプリケーションを構築する際に必要ではありません。 パッケージからこれらの DLL を削除します。

ユーザのアプリケーションを CA EEMC# SDK をパッケージに含める方法

1. ユーザのアプリケーションを構築する場合、参照先のアセンブリとして以下のフォルダから DLL を追加します。

   ```
   %EIAM_SDK%\lib\csharp
   ```

2. インストーラを更新して、参照先の DLL、eiam.config ファイルおよび eiam.log4net.config ファイルをパッケージに含めます。

   注: eiam.config ファイルおよび eiam.lognet.config ファイルは %EIAM_SDK%¥bin フォルダにあります。

3. 参照先の DLL、eiam.config ファイル、および iam.log4net.config ファイルをクライアント コンピュータ上に展開します。

CA EEM SDK の設定

以下のトピックは、Safe::Configurator クラスを使用して CA EEM SDK を設定する方法について説明します。

詳細情報:

eiam.config ファイルについて (P. 31)
CA EEM SDK ログ記録 (P. 79)
CA EEM C++ SDK の設定 (P. 38)
SafeConfigurator を使った CA EEM Java SDK の設定 (P. 37)

eiam.config ファイルについて

以下に示す CA EEM SDK 設定データを制御するには、eiam.config ファイルを使用する必要があります。

- 循環バッファ
- ロガー設定ファイル
- 監査ファイルを格納する SAF フォルダ
- FIPS 互換モード

eiam.config ファイルは以下の設定可能なパラメータで構成されています。

CyclicBuffer size

循環バッファに含むログ メッセージの数を指定します。 循環バッファは最新ログ メッセージを指定された数だけメモリに格納します。 バッファが指定されたサイズに達すると、新しいログ メッセージがバッファで最も古いログ メッセージと置き換えられます。 アプリケーションがクラッシュしても、コアから最新のログ メッセージを回復できます。

デフォルト: 500

最小: 0

最大: 1000

enable

循環バッファが有効かどうか指定します。 有効が false に設定されている場合、循環バッファは無効です。 このため、CyclicBuffer size、dump、file の各パラメータを指定する必要はありません。

値: [true|false]

デフォルト: true

重要: ログ記録を有効にしているかどうかに関係なく、循環バッファはデフォルトで有効です。 循環バッファを有効にすると、CA EEM のパフォーマンスが影響を受けます。

dump

eiam.config ファイルが変更または更新される場合に、循環バッファの内容をファイルに書き込むかどうか指定します。

値: [true|false]

デフォルト: false

file

ダンプ ファイルのファイル名を指定します。 dump が false に設定されている場合、ログ メッセージはダンプ ファイルに書き込まれません。 ファイルのファイル拡張子は .log です。

LoggerConfiguration ファイル

CA EEM Java および C++ SDK のロガー設定ファイルの絶対パスを指定します。CA EEM ログ記録情報はロガー設定ファイルに格納されます。eiam.log4cxx.config と eiam.log4j.config は、CA EEM C++ SDK および CA EEM Java SDK のロガー設定ファイルです。

Saf directory

監査ファイルが処理のために格納される SAF フォルダ。

Network sockettimeout

ソケットのタイムアウトをミリ秒単位で指定します。

デフォルト: 120000（2 秒）

詳細情報:

CA EEM SDK ログ記録 (P. 79)

eiam.config ファイルの例

以下は eiam.config ファイルの例です。

```
<EiamConfiguration>
  <!-- EIAM Internal: Configure cyclic buffer -->
  <CyclicBuffer size="500" dump="false" file="dump.log" enable="true" />
  <!-- Absolute file path for logger configuration, For Java use:-
file="eiam.log4j.config" -->
  <LoggerConfiguration file="eiam.log4cxx.config"/>
  <!-- Absolute folder path for SAF folder where audit files will be stored for
processing-->
  <Saf directory="audit"/>
   <!-- Socket timeout in milli seconds. Default value is 2 mins -->
  <Network sockettimeout="120000"/>
  <SDK type="Java">
        <iTechSDK>
                <FIPSMode>true</FIPSMode>
                <JCEProvider>JsafeJCE</JCEProvider>
                <Security>
                        <digestAlgorithm>SHA1</digestAlgorithm>
                </Security>
                <Debug>
                        <logLevel>trace</logLevel>
                </Debug>
        </iTechSDK>
  </SDK>
  <SDK type="C++">
        <iTechSDK>
                <FIPSMode></FIPSMode>
                <Commons>
                        <etpkiCryptoLib></etpkiCryptoLib>
                </Commons>
                <TransportConfig>
                        <!--possible values are SSLV23 / SSLV3 / TLSV1-->
                        <secureProtocol></secureProtocol>
                </TransportConfig>
                <Security>
                        <!--possible values are MD5/SHA1/SHA256/SHA384/SHA512-->
                        <digestAlgorithm></digestAlgorithm>
                </Security>
                <Debug>
                        <!--possible values are ERROR/WARNING/TRACE/NOLEVEL-->
                        <logLevel></logLevel>

                        <!--possible values are true/false -->
                        <logToFile></logToFile>

                        <!--log file name-->
                        <logFile></logFile>
```

```
                    <!--log file size in MB(positive integer)-->
                    <maxLogSize></maxLogSize>
                </Debug>
            </iTechSDK>
        </SDK>
    </EiamConfiguration>
```

iTechnology SDK のログの有効化

iTechnology SDK のログを有効にできるのは、CA EEM C++ SDK および CA EEM Java SDK のみです。 CA EEM C# SDK では、ロガー設定ファイルを使用します。

iTechnology SDK のログ記録を有効にするには、eiam.config ファイルを開き、以下のタグを編集します。

- logLevel
- logToFile
- logFile
- maxLogSize

CA EEM Java SDK では、前の <SDK type ="Java"> セクションで説明したタグを編集します。 CA EEM C++ SDK では、<SDK type ="C++"> で説明したタグを編集します。

CA EEM Java SDK を FIPS のみのモードに設定する前に

CA EEM Java SDK を FIPS のみのモードに設定するには、以下のタスクを実行します。

1. サードパーティの Java 暗号化拡張機能(JCE)ライブラリを使用するには、JRE を設定します。
2. Java.security ファイルに JCE プロバイダとして Crypto-J ライブラリを追加します。

   注: JCE を使用して JRE を設定する方法の詳細については、対応する JCE のドキュメントを参照してください。
3. eiam.config ファイルで FIPS のみのモードを有効にします。

CA EEM Java SDK の FIPS のみのモードの設定

CA EEM SDK を FIPS のみのモードに設定すると、CA EEM は FIPS 140-2 に準拠した暗号化ライブラリを使用し、機密データの暗号化や複合化を行います。

CA EEM Java SDK を FIPS のみのモードに設定する方法

1. eiam.config ファイルを開き、<SDK type="Java"> セクションにある以下のタグを編集します。
   - FIPSMode
   - JCEProvider
   - digestAlgorithm
2. eiam.config ファイルを保存して閉じます。
3. アプリケーションを再起動します。

   CA EEM Java SDK が FIPS のみのモードに設定されます。

CA EEM C++ SDK の FIPS のみのモードの設定

CA EEM SDK を FIPS のみのモードに設定すると、CA EEM は FIPS 140-2 に準拠した暗号化ライブラリを使用し、機密データの暗号化や複合化を行います。

CA EEM C++ SDK を FIPS のみのモードに設定する方法

1. eiam.config ファイルを開き、<SDK type="C++"> セクションにある以下のタグを編集します。
   - FIPSMode
   - etpkiCryptoLib
   - secureProtocol
   - digestAlgorithm
2. eiam.config ファイルを保存して閉じます。
3. アプリケーションを再起動します。

   CA EEM C++ SDK が FIPS のみのモードに設定されます。

CA EEM C# SDK の FIPS のみのモードの設定

CA EEM C# SDK を FIPS のみのモードに設定する場合、CA EEM は FIPS 140-2 に準拠した暗号化ライブラリを使用し、機密データの暗号化や複合化を行います。

注: CA EEM C# SDK では、P11 証明書はサポートされません。

CA EEM C# SDK を FIPS のみのモードに設定する方法

1. eiam.config ファイルを開き、<SDK type="C#"> セクションにある以下のタグを編集します。
   - FIPSMode
   - digestAlgorithm
2. eiam.config ファイルを保存して閉じます。
3. アプリケーションを再起動します。

   CA EEM C# SDK が FIPS のみのモードに設定されます。

SafeContext 情報の設定

eiam.config ファイル内の <SafeContext> タグには SafeContextFactory クラスを使用して SafeContext を生成するのに必要な情報が含まれています。 eiam.config ファイル内の各 SafeContext タグは、一意の refID タグを使用して識別されます。 SafeContext を生成するには、この refID を SafeContextFactory に渡す必要があります。 eiam.config ファイルで SafeContext 関連の情報を指定する手順は以下になります。

SafeContext 関連情報を設定する方法

1. eiam.config ファイルを開き、<SafeContext> セクションを編集して以下のタグを設定します。
   - refID
   - Backend
   - Application
   - Locale
   - Authentication Type
2. eiam.config ファイルを保存して閉じます。

SafeConfigurator を使った CA EEM Java SDK の設定

Safe::Configurator クラスを使用して、CA EEM SDK を設定する必要があります。 CA EEM SDK を設定するには、以下のプロセスを実行します。

注: CA EEM SDK を設定する前に、eiam.config を設定する必要があります。

1. アプリケーションの起動中に CA EEM SDK を初期化するために以下の API をコードに含めます。

   ```
   SafeConfigurator.getInstance().init(filename);
   ```

   この中で、

   filename

   アプリケーションに定義した eiam.config ファイルの絶対パスを指定します。

   注: この行の後の CA EEM SDK オペレーションはすべて、ロガー設定のログ記録のトレース レベルに基づいて記録されます。

2. アプリケーションのシャットダウン時のコードには以下の API を含めます。

   ```
   m_config.term();
   ```

   注意: m_config.init(filename) を使ったすべての初期化コールは、対応する m_config.term() を使って終了する必要があります。 init メソッドと term メソッドはスレッド セーフであり、参照がカウントされます。 Safe ライブラリは最初の init () コール中に初期化され、参照数がゼロになると終了します。

詳細情報:

eiam.config ファイルについて (P. 31)
ロガー設定ファイルについて (P. 80)

CA EEM C++ SDK の設定

Safe::Configurator クラスを使用して、CA EEM SDK を設定する必要があります。 CA EEM SDK を設定するには、以下のプロセスを実行します。

注: CA EEM SDK を設定する前に eiam.config を設定する必要があります。

1. アプリケーションの起動中に CA EEM SDK を初期化するために以下の API をコードに含めます。

   ```
   Safe::Configurator::getInstance()->init(filename);
   ```

   この中で、

   filename

   アプリケーションに定義した eiam.config ファイルの絶対パスを指定します。

2. アプリケーションのシャットダウン時のコードには以下の API を含めます。

   ```
   Safe::Configurator::getInstance()->term();
   ```

   注: Safe::Configurator::getInstance()->init(filename) を使ったすべての初期化コールは、対応する Safe::Configurator::getInstance()->term() を使って終了する必要があります。 init メソッドと term メソッドはスレッド セーフであり、参照がカウントされます。 Safe ライブラリは最初の init () コール中に初期化され、参照数がゼロになると終了します。

詳細情報:

eiam.config ファイルについて (P. 31)
ロガー設定ファイルについて (P. 80)

CA EEM C# SDK の初期化

SafeConfigurator クラスを使用して、CA EEM SDK を設定します。 CA EEM SDK を設定するには、以下のプロセスを実行します。

注: CA EEM SDK を設定する前に、eiam.config ファイルを設定します。 eiam.config ファイルを設定しない場合、CA EEM SDK は以下のデフォルト設定を使用して初期化されます。

- FIPS 非準拠モード
- ログ記録はエラーに設定され、コンソールのログ記録のみが有効
- SAF の場所は無効

CA EEM SDK を初期化するには、以下のプロセスを実行します。

1. アプリケーションの起動中に CA EEM SDK を初期化するには、コードに以下の API を含めます。

   ```
   SafeConfigurator.getInstance().Init(filename);
   ```

   場所

   filename

   アプリケーションに定義した eiam.config ファイルの絶対パスを指定します。

   注: filename を指定しない場合、CA EEM SDK はデフォルト値を使用して初期化されます。

2. アプリケーションのシャットダウン中に以下の API をコードに含めます。

   ```
   SafeConfigurator.getInstance().term();
   ```

   注: SafeConfigurator クラスの詳細については、「プログラミング ガイド」を参照してください。

第 5 章: FIPS 140-2 のサポート

このセクションには、以下のトピックが含まれています。

FIPS 140-2 の概要

FIPS（Federal Information Processing Standards）140-2 は、機密性は高いが機密扱いではないデータを保護するセキュリティ システム内で暗号化アルゴリズムを使用するための要件を規定します。 CA EEM サーバには RSA の CryptoC ME v2.0 暗号化ライブラリが組み込まれています。このライブラリは、FIPS 140-2（**暗号化モジュールに関するセキュリティ要件**）に適合していることが確認されています。 このモジュールの認証証明書番号は 608 です。

CA EEM Java SDK は、FIPS に準拠した RSA の BSAFE Crypto-J 4.0 のバージョンを使用します。 CA EEM C++ SDK は ETPKI 4.1.x を組み込みます。これは、RSA の暗号化ライブラリを使用します。

CA EEM は、FIPS 非準拠モードまたは FIPS のみのモードで動作できます。 暗号化の相違点として、CA EEM が暗号化を適用する方法は両方のモードで同じですが、アルゴリズムが異なります。

FIPS 認定モードで FIPS 140-2 認定暗号化モジュールを使用するコンピュータ製品は、AES（Advanced Encryption Algorithm）、 SHA-1（Secure Hash Algorithm）などの FIPS 認定セキュリティ関数や、FIPS 140-2 標準および実装ガイドで明示的に認証されている TLS v1.0 などの高度なレベルのプロトコルのみを使用します。

FIPS のみのモードでは、CA EEM は以下のアルゴリズムを使用します。

- パスワードの暗号化およびサーバ リクエストの署名を行うためのデフォルト ダイジェスト アルゴリズムとして SHA1 を使用します。 FIPS のみのモードでは、以下のアルゴリズムのいずれかを使用できます。
  - SHA1
  - SHA256
  - SHA384
  - SHA512
- LDAP 接続が TLS を使用している場合、外部 LDAP ディレクトリとの通信に TLS v1.0 を使用します。

CA EEM でサポートされるセキュリティ モード

CA EEM では、FIPS 非準拠と FIPS のみの 2 つの操作モードをサポートします。 CA EEM の機能は、両方のモードで同じです。 これらの 2 つのモードでは、パスワードの格納や検証、CA EEM と CA SiteMinder などのその他の製品との機密データの通信に使用するアルゴリズムが異なります。

FIPS 非準拠

暗号化に FIPS に準拠していない技術を使用するモードを指します。 このモードでは、デフォルトのアルゴリズムとして MD5 を使用し、機密データの暗号化や複合化を実行します。 新規インストールやアップグレードは、常に FIPS 非準拠のモードで実行されます。 FIPS 非準拠モードでは、CA EEM サーバは CA EEM クライアントと下位互換性があります。 たとえば、CA EEM r8.4 SDK を使用して CA EEM r8.4 SP3 サーバに接続できます。

FIPS のみ

暗号化に FIPS に準拠した技術のみを使用しているモードを指します。 このモードは、FIPS 非準拠で実行しているクライアントとは互換性がありません。 FIPS のみに設定されている CA EEM r8.4 SP3 SDK クライアントは、FIPS モードで実行している CA EEM r8.4 SP3 サーバとのみ使用することができます。

CA EEM サーバを FIPS のみのモードに設定する方法

FIPS のみのモードでは、CA EEM は FIPS に準拠したアルゴリズムを使用するように設定する必要があります。 CA EEM サーバと CA EEM SDK クライアントは、両方が FIPS のみのモードに設定されている場合にのみ通信できます。 同様に、FIPS モードの CA EEM サーバは、FIPS に準拠したアルゴリズムを使用するように設定されている LDAP ディレクトリとのみ通信できます。 CA EEM の環境を FIPS のみのモードに設定するには、以下を実行します。

- CA EEM サーバを FIPS のみのモードに設定するための前提条件を確認します。
- CA EEM サーバの FIPS のみのモードの設定

CA EEM サーバを FIPS のみのモードに 設定するための前提条件

CA EEM サーバを FIPS のみのモードに設定するための前提条件は以下になります。

- CA ITM、CA ELM などの iGateway を使用する他の CA 製品が FIPS のみのモードに設定されていることを確認します。 iGateway は、FIPS のみのモードおよび FIPS 非準拠モードが混在していると初期化できません。 iGateway を FIPS のみのモードで初期化する場合、iGateway を使用するすべての製品が FIPS のみのモードに設定されている必要があります。- iGateway.conf ファイルを開き、以下のタグの値を確認します。

  `FIPSMode`

  このタグの値が False に設定されている場合、iGateway を使用する製品が FIPS 非準拠モードであることを意味します。 FIPS のみのモードで CA EEM を有効にする場合は、iGateway の既存の設定を元に適切に判断します。

- 他の CA 製品によって使用されるスピンドルのバージョンを確認するには、spin.conf ファイルを開き、<Spindle Name> および <version> のタグの値をメモします。 各製品のドキュメントを使用し、これらのバージョンが FIPS に準拠しているかどうかを確認します。

  注: iGateway.conf ファイルおよび spin.conf ファイルは以下の場所に格納されています。

  - **Windows:** %IGW_LOC%
  - **Linux および UNIX:** /opt/CA/SharedComponents/iTechnology

CA EEM を FIPS のみのモードに設定する前に

FIPS のみのモードを使用するように環境を移行する前に、ユーザの環境が最小要件を満たしていることを確認します。 以下を印刷して、チェックリストとして使用します。

- ☐ CA EEM サーバを CA EEM r8.4 SP3 にアップグレードする。
- ☐ CA EEM に統合または接続される製品が FIPS のみのモードを使用するように設定されていることを確認する。

CA EEM サーバの FIPS のみのモードの設定

CA EEM サーバを FIPS のみのモードに設定する場合、CA EEM は FIPS 140-2 に準拠した暗号化ライブラリを使用し、機密データの暗号化や複合化を行います。

注:

- FIPS のみのモードでは、IE7（またはそれ以上）、Firefox 3.0（またはそれ以上）を使用して CA EEM 管理者 GUI を表示します。 FIPS 140-2 モードで Firefox を設定する方法の詳細については、Firefox のサポート サイトを参照してください。
- また、CA EEM サーバのセキュリティ モードを FIPS のみから FIPS 非準拠に、または FIPS 非準拠から FIPS のみに変更する場合、以下の手順が有効です。

CA EEM を FIPS のみのモードに設定する方法

1. iGateway サービスを停止します。
2. 以下のコマンドを使用し、CA Directory サービスを停止します。

   Windows

   ```
   dxserver stop all
   ssld stop
   ```

   Linux および UNIX の場合

   ```
   su - dsa -c "dxserver stop all"
   su - dsa -c "ssld stop"
   ```

3. iGateway.conf ファイルを開き、以下のタグを ON に設定します。

   ```
   <FIPSMode>ON<FIPSMode>
   ```

   注: FIPS のみから FIPS 非準拠モードに変更するには、FIPSMode タグを OFF に設定します。

4. コマンド プロンプトで以下のスクリプトを実行します。

   Windows

   ```
   ssld remove iTechPoz-Server
   ssld install iTechPoz-Server -certfiles "%DXHOME%/config/ssld/personalities" -ca
   "%DXHOME%/config/ssld/iTechPoz-trusted.pem" -port 21847 -fips
   ```

   Linux および UNIX の場合

   ```
   su - dsa
   ssld remove iTechPoz-Server
   ssld install iTechPoz-Server -certfiles $DXHOME/config/ssld/personalities -ca
   $DXHOME/config/ssld/iTechPoz-trusted.pem -port 21847 -fips
   ```

   注: オプション -port は ssld ポートを指定します。 別の ssld ポートを設定している場合は、前のコマンドで 21847 を正しいポート番号に置き換えます。 また、FIPS のみから FIPS 非準拠にセキュリティ モードを変更中の場合、この手順のコマンドを -fips オプションを指定せずに使用します。

5. 以下のコマンドを使用して、CA Directory サービスを開始します。

   Windows

   ```
   ssld start
   dxserver start all
   ```

   Linux および UNIX の場合

   ```
   su - dsa -c "ssld start"
   su - dsa -c "dxserver start all"
   ```

6. iGateway サービスを開始する。

   CA EEM が FIPS のみのモードに設定されます。

CA EEM サーバが FIPS のみのモードに設定されていることを確認します。

CA EEM サーバが FIPS のみのモードに設定されていることを確認するには、以下を実行します。

1. ブラウザに「https://hostname」または「IP address:5250/spin/eiam/about.csp」と入力します。

   [バージョン情報]ページが開きます。

2. FIPS: ラベルが「有効」に設定されていることを確認します。

   ラベルが「有効」に設定されている場合、CA EEM サーバが FIPS のみのモードに設定されていることを示しています。

CA EEM サーバと外部 LDAP ディレクトリ間の通信

CA EEM サーバと外部ディレクトリ間の通信は、暗号化使用または暗号化未使用いう 2 つの LDAP 接続のタイプによって決定されます。 暗号化を使用した CA EEM サーバと外部ディレクトリでサポートされる操作モードは以下になります。

CA EEM サーバで LDAP の通信に暗号化が有効になっている場合

CA EEM サーバが外部 LDAP ディレクトリとの通信に暗号化されたチャネルを使用するように設定されている場合、CA EEM サーバが FIPS モードに設定されていると、LDAP ディレクトリも FIPS 互換モードを使用するように設定する必要があります。

PKCS#11 デバイスのサーバ証明書を使用するように CA EEM を設定する

CA EEM サーバまたは CA EEM SDK と共に nCipher PKCS#11 デバイスを使用するには、nCipher デバイスを設定し、以下のプロパティを指定したように設定する必要があります。

```
CKNFAST_OVERRIDE_SECURITY_ASSURANCES=all
```

注: ハード トークンを使用して nCipher デバイスを設定する方法の詳細については、nCipher のドキュメントを参照してください。

PKCS#11 デバイスに格納された証明書を使用するように CA EEM サーバを設定するには、以下を実行します。

1. iGateway サービスを停止します。
2. iGateway.conf を開き、<Connector name="defaultport"> CA Portal5250</port> タグを編集して以下の値を設定します。

   certType

   使用する証明書のタイプを指定します。 サポートされる証明書のタイプは、p12、pem および p11 です。

   デフォルト: pem

   タイプ: Childnode

P11 証明書の使用

<pkcs11Lib/> -- トークンによって提供される PKCS11 ライブラリへのパス

<token/> -- トークン ID

<userpin/> -- 暗号化されたユーザ ピン

<id/> -- 証明書および秘密鍵 ID

<sensitive/> -- 秘密鍵は機密情報です。 ソフトウェア鍵と暗号化操作は cryptopki ハードウェアを使用して実行されるため、sensitive に設定されているキーは変換されません。(sensitive ではない鍵を sensitive として扱うことはできます。ただし、sensitive に設定されている鍵を sensitive ではない鍵に変換したり、sensitive ではない鍵として扱うことはできません)。

デフォルト: **False**

3. iGateway.conf ファイルを保存して閉じます。
4. iGateway サービスを開始します。

PKCS#11 デバイスにサーバ証明書を格納するように CA EEM を設定する

PKCS#11 デバイスに CA EEM 証明書を格納するには、以下を実行します。

1. iGateway サービスを停止します。
2. iGateway.conf ファイルを開き、<CertificateManager> タグを編集して以下の値を設定します。

   certType

   使用する証明書のタイプを指定します。 サポートされる証明書のタイプは、p12、pem および p11 です。

   デフォルト: pem

   タイプ: Childnode

P11 証明書の使用

<pkcs11Lib> <pkcs11Lib/> -- トークンによって提供される PKCS11 ライブラリへのパス

<token><token/> -- トークン ID

<userpin><userpin/> -- 暗号化されたユーザ ピン

<id><id/> -- 証明書および秘密鍵 ID

<sensitive><sensitive/> -- 秘密鍵は機密情報です。 ソフトウェア鍵と暗号化操作は cryptopki ハードウェアを使用して実行されるため、sensitive に設定されているキーは変換されません。(sensitive ではない鍵を sensitive として扱うことはできます。ただし、sensitive に設定されている鍵を sensitive ではない鍵に変換したり、sensitive ではない鍵として扱うことはできません) - オプションはデフォルトでは false に設定されています。

3. iGateway.conf ファイルを保存して閉じます。
4. iGateway サービスを開始します。

ユーザのアプリケーションの FIPS のみのモードの設定

ユーザのアプリケーションを FIPS のみのモードに設定するには、CA EEM SDK が FIPS のみのモードに設定されており、CA EEM SDK が FIPS に準拠した技術のみを暗号化に使用することを確認します。 CA EEM SDK 設定ファイルである eiam.config は、CA EEM SDK のセキュリティで保護された操作モードを制御します。 CA EEM SDK FIPS のみのモードに設定する前に、以下を確認します。

- ユーザの CA EEM SDK がバージョン r8.4 SP3 であることを確認します。
- CA EEM によって使用される既存の P12 証明書を PEM 証明書に移行します。
- CA EEM SDK を FIPS のみのモードで初期化します。

ユーザのアプリケーションによって使用される P12 証明書を PEM 証明書に移行します。

CA EEM は、P12、PEM、PKCS#11 証明書をサポートしますが、以下の考慮事項があります。

- FIPS のみのモードでは、P12 のサポートは無効です(利用できません)。 代わりに、FIPS のみのモードでは、PEM および PKCS#11 の証明書のサポートが追加されました。

注: CA EEM C# SDK は、FIPS のみのモードでは PEM 証明書のみをサポートします。FIPS 非準拠モードでは、P12、PEM 証明書のみをサポートします。

そのため、P12 証明書を使用している場合は、この証明書を FIPS のみのモードでサポートされている証明書の形式のいずれかに移行します。 P12 証明書を pem 証明書に変換するには、igwCertUtil ユーティリティを使用します。 igwCertUtil は、証明書の変換、作成、削除を実行するユーティリティです。 igwCertUtil は以下のフォルダにあります。

Windows

%IGW_LOC%

UNIX および Linux

$IGW_LOC

igwcertutil Utility -- 証明書の作成、コピー、変換および削除

Windows、UNIX および Linux で有効

作成コマンドの形式は以下のとおりです。

```
igwCertUtil -version version -create -cert inputcert-params -issuer issuercert -params
[-debug] [-silent]
```

変換コマンドの形式は以下のとおりです。

```
igwCertUtil -version version -conv -cert inputcert-params -target newcert-params
[-debug] [-silent]
```

コピー コマンドの形式は以下のとおりです。

```
igwCertUtil -version version -copy -cert inputcert-params -target newcert-params
[-debug] [-silent]
```

削除コマンドの形式は以下のとおりです。

```
igwCertUtil -version version -delete -cert cert-params [-debug] [-silent]
```

-version version

証明書の作成、変換、コピー、または削除の実行時に使用する igwCertUtil のバージョンを指定します。 バージョンは下位互換性用に使用されます。 igwCertUtil を変更すると、バージョン タグは指定したバージョンの動作を取得します。

-cert inputcert-parms

証明書の作成、変換、コピーの実行時に XML 文字列として証明書を指定します。

-issuer issuercert-parms

証明書の作成時に、新しく生成された証明書に署名するために使用する証明書を指定します。 証明書を指定しない場合、自己署名証明書が作成されます。

-target newcert-parms

既存の証明書の変換(またはコピー)実行時に、新規証明書用の設定を指定します。

-cert cert-parms

-debug

(オプション)igwCertUtil のデバッグをオンにします。

-silent

(オプション)igwCertUtil のサイレント モードをオンにします。

igwCertUtil から以下のようなエラー コードが返されます。

- CERTUTIL_ERROR_UNKNOWN (-1): 不明または未定義のエラーが発生しました
- CERTUTIL_SUCCESS (0): 正常な操作です
- CERTUTIL_ERROR_USAGE (1): 不正なコマンド ライン引数が渡されました
- CERTUTIL_ERROR_READCERT (2): 証明書を読み取れません
- CERTUTIL_ERROR_WRITECERT (3): 証明書に書き込めません
- CERTUTIL_ERROR_WRITECERT (3): 証明書を削除できません

例: P12 証明書を PEM 証明書に変換する

以下の例では、P12 証明書を PEM 証明書に変換する使用用法について説明します。

```
igwCertUtil -version 4.6.0.0 -conv -cert
"<Certificate><certType>p12</certType><certURI>testCert.p12</certURI><certPW>pass
word</certPW></Certificate>" -target "<Certificate><certType>pem</certType>
<certURI>testCert.cer</certURI><keyURI>testCert.key</keyURI></Certificate>"
```

例: P12 証明書を PKCS#11 証明書に変換する

```
igwCertUtil -version 4.6.0.0 -conv -cert
"<Certificate><certType>p12</certType><certURI>testCert.p12</certURI><certPW>pass
word</certPW></Certificate>" -target "<Certificate><certType>p11</certTyp
><pkcs11Lib>pathto-pkcs11Lib</pkcs11Lib><token>pkcs11token</token><userpin>user
in</userpin><id>certid</id></Certificate>"
```

FIPS のみのモードでの CA EEM SDK の初期化

eiam.config ファイルを設定することによって、CA EEM SDK を FIPS のみのモードで初期化できます。 eiam.config ファイルを設定するには、「CA EEM SDK の設定 (P. 27)」を参照します。

第 6 章: CA EEM サーバのバックアップとリストア

このセクションには、以下のトピックが含まれています。

ファイル システムのバックアップ

定期的に、または CA EEM サーバ環境を変更したら常に、CA EEM サーバをバックアップすることをお勧めします。 データが破損した場合でも、CA EEM サーバのバックアップを使用して CA EEM サーバをリストアできます。

バックアップする必要があるのは、以下の CA EEM のファイルとフォルダです。

データの説明	Windows のファイル名	Linux のファイル名
設定ファイル	■ iPoz.conf ■ Eiam.conf ■ iPoz.map ■ Spin.conf ■ iPozDsa.pem ■ iPozRouterDsa.pem ■ logDepot.conf ■ calmReporter.conf	■ iPoz.conf ■ Eiam.conf ■ iPoz.map ■ Spin.conf ■ iPozDsa.pem ■ iPozRouterDsa.pem ■ eiam-type ■ Sponsorfiles ■ logDepot.conf ■ calmReporter.conf

データの説明	Windows のファイル名	Linux のファイル名
イベント情報	■ logdepotdb ■ calm_catalog フォルダ ■ calm_archive フォルダ	■ logdepotdb ■ calm_catalog フォルダ ■ calm_archive フォルダ
フォルダ	■ システム レジストリ ■ iTechnology フォルダ	■ iTechnology フォルダ ■ 環境設定

CA EEM サーバのファイルとフォルダのバックアップ

定期的に、または CA EEM サーバ環境を変更したら常に、CA EEM サーバをバックアップすることをお勧めします。 データが破損した場合でも、CA EEM サーバのバックアップを使用して CA EEM サーバをリストアできます。

CA EEM サーバのファイルとフォルダをバックアップするには、以下の手順に従います。

1. iGateway を停止します。
2. CA EEM の設定ファイル、イベント情報、およびフォルダをバックアップします。
3. CA Directory に保存されている CA EEM データをバックアップします。

   CA EEM サーバの設定ファイル、イベント、およびフォルダがバックアップされます。

詳細情報:

ファイル システムのバックアップ (P. 53)
CA Directory に保存されている CA EEM データのバックアップ (P. 57)

リストアの手順

CA EEM データをリストアする必要があるのは、以下の状況に対応できるようにする場合です。

- 破損した CA EEM システムを修復する
- 想定どおりに動作しない CA EEM サーバ環境を修復する

CA EEM の設定ファイルとデータをリストアするには、以下の手順に従います。

1. iGateway を停止します。
2. バックアップした CA EEM の .conf ファイルすべての名前を .conf.merge に変更し、名前を変更した設定ファイルを iTechnology フォルダにコピーします。 .conf.merge ファイルは、バックアップした設定ファイルを新しい設定ファイルにマージするために必要です。
3. CA EEM データをリストアします。
4. iGateway を開始します。

詳細情報:

CA Directory に保存されている CA EEM データのバックアップ (P. 57)

iGateway サービスの開始

iGateway サービスを開始するには、以下のコマンドを入力します。

Windows

```
net start igateway
```

Linux および UNIX の場合

```
$IGW_LOC/S99igateway start
```

iGateway サービスの停止

iGateway サービスを停止するには、以下のコマンドを入力します。

Windows

```
net stop igateway
```

Linux および UNIX の場合

```
$IGW_LOC/S99igateway stop
```

第 7 章: CA Directory に保存されている CA EEM データのバックアップ

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 Directory 用語の紹介

このセクションでは、本書で使用されている CA Directory の用語について説明します。

DSA

DSA は、ディレクトリのネームスペースの一部またはすべてを管理するプロセスです。

This also needs to be expanded to make it accessible to readers new to CA Directory. CA EEM サーバのインストール時に、以下の CA Directory 関連パラメータを設定することができます。

DXmanager

DXmanager は、ディレクトリのバックボーンを作成、設定、監視、および管理する Web アプリケーションです。

DSA コンソール

DSA **コンソール**を使用すると、DSA へ接続して DXserver コマンドを指定したり、トレース情報を受信したり、ユーザ エージェントとして機能させたりすることができます。

DXtool

DXtool は、CA Directory に付属するコマンド ライン ユーティリティのセットです。これらのツールを使用して、ディレクトリ管理、LDIF データの操作、ディレクトリとのデータのロードとアンロード、および CA Directory で使用するスキーマの抽出と変換を行うことができます。

LDIF（LDAP Data Interchange Format）

LDIF **ファイル**は、LDIF 形式でディレクトリ情報を保存したテキスト ファイルです。LDIF ファイルを使用すると、LDAP ディレクトリ サーバ間でディレクトリ情報を転送したり、ディレクトリに適用する変更をまとめて記述したりすることができます。

DXtool の使用方法

DXtool を実行するには、以下の方法があります。

- DSA コンソールを使用して、ホスト上で DXtool コマンドを実行する。
- TCP/IP ネットワーク経由で DSA コンソールを使用して、リモート ホスト上で DXtool コマンドを実行する。
- スクリプトの中に DXtool コマンドを埋め込む。

すべてのツールは、成功時には 0 を、エラー時には 0 以外の値をそれぞれ返します。

DXHOME 環境変数

一部のツールでは、DXHOME 環境変数を DXserver のホーム パスに設定しておく必要があります。 これは、CA Directory のインストール時に自動的に設定されます。

ツールによっては、DSA 設定ファイルは DXHOME のパスの下層にある config フォルダに存在している必要があります。

DXtool の終了ステータス コード

DXtool の終了コードは一般的には共通ですが、すべての終了コードがすべてのツールに適用されるわけではありません。 終了コードは以下のとおりです。

0

成功

1

対応する DSA が実行中です。

2

1 つまたは複数のデータストア ファイルがすでに存在します。

3

指定したディレクトリの場所が存在しないか、またはディレクトリではありません。

4

指定したファイルのタイプが間違っています（ディレクトリなど）。

5

このファイルの権限に問題があります。

6

データストア ファイルのフル パス名が長すぎます。 考えられる理由は、データストア ディレクトリに対して指定した場所のパス名が長すぎることです。

7

古いデータストア ファイルを削除しようとした際にエラーが発生しました。

8

古いデータストア ファイルの名前を変更しようとした際にエラーが発生しました。

9

ファイルの 1 つを作成またはパディングしようとした際にエラーが発生しました。

10

データストアのサイズが 0 以下です。

11

ファイルを作成しようとした際に、デバイスに十分な領域がなかったか、メモリが不足していました。

12

アクセス権が不十分なため（権限が不十分な可能性があります）、ファイルを作成するか、またはファイルにアクセス権限を設定できません。

13

DXHOME 環境変数が設定されていません。

14

DXHOME 環境変数が無効です。

15

対応する DSA がすでに存在します。

16

作成した DSA の起動に失敗しました。 詳細については、ログ ファイルを参照してください。

17

不正または不明なコマンド ライン パラメータが指定されました。

18

対応する DSA が存在しません。

CA Directory データのバックアップ方法

CA Directory のデータをバックアップするには、以下の手順に従います。

1. ローカルの DSA に接続します。
2. 実行中のデフォルトの DSA のデータストアのスナップショット コピーを取ります。このプロセスをオンライン ダンプと呼びます。 スナップショットを取るには、以下のコマンドを実行します。

   ```
   dump dxgrid-db
   ```

   注: CA EEM をバックアップする場合は、dxgrid-db を DSA 名 iTechPoz-Servern に置き換えてください。
3. DXdumpdb ツールを使用して、オンライン ダンプ(.ZDB ファイル)、つまりデータストアのスナップショット コピーを LDIF ファイルにバックアップします。

詳細情報:

ローカルの DSA コンソールへの接続 (P. 60)
オンライン データストア ダンプ (P. 61)
dump dxgrid-db コマンド - データストアのスナップショット コピーの定期的な取得 (P. 61)

ローカルの DSA コンソールへの接続

UNIX または Windows では、DSA にコンソール ポートが設定されていれば、その DSA にローカルに接続することができます。

DSA コンソールにローカルに接続するには、以下の手順に従います。

1. DSA が実行されているホストで、コマンド プロンプトを開きます。
2. 以下のコマンドを入力します。

   ```
   telnet localhost local-port-number
   ```

   local-port-number

   接続先の DSA コンソールのポート番号を指定します。

オンライン データストア ダンプ

DSA の実行中も常にデータストアのスナップショット コピーを取ることができます（オンライン ダンプ）。 更新があれば、DSA はオンライン ダンプの実行前に更新をすべて完了し、コピーが完了するまで更新は一切開始されません。

データストア ファイルは、データベース ファイルが dxgrid-db.zdb であるため、.z で始まる拡張子を持つファイルにコピーされます。

注： ダンプを実行するたびに、前回のバックアップ ファイルは上書きされます。 バックアップ ファイルを保存しておきたい場合は、次回のダンプを実行する前に目的のファイルを別の場所にコピーしてください。

dump dxgrid-db コマンド - データストアのスナップショット コピーの定期的な取得

dump dxgrid-db コマンドは、実行中の DSA のデータストアのスナップショット コピーを定期的に取ります。 更新があれば、DSA はこのコマンドの実行前に更新をすべて完了し、コピーが完了するまで更新は一切開始されません。

データストア ファイルは、データベース ファイルが dxgrid-db.zdb であるため、.z で始まる拡張子を持つファイルにコピーされます。

注： ダンプを実行するたびに、前回のバックアップ ファイルは上書きされます。 バックアップ ファイルを保存しておきたい場合は、次回のダンプを実行する前に目的のファイルを別の場所にコピーしてください。

DXdumpdb ツールを使用すると、ダンプ コマンドで作成したデータストアからデータをエクスポートすることができます。

コマンドの形式は以下のとおりです。

```
dump dxgrid-db [period start period];
```

period start period

（オプション）オンライン ダンプが定期的に実行されます。

start

日曜日の AM00:00:00 （GMT）からの秒数を指定します。

注： 開始時間の定義には、GMT を使用し、ローカル時間は使用しないでください。

period

オンライン ダンプの間隔を秒数で指定します。

例: オンライン ダンプを 1 時間ごとに実行する場合

以下のコマンドは、1 時間ごとにデータストアのスナップショット コピーを取ります。

```
dump dxgrid-db 0 3600
```

注: UNIX の場合は cron ジョブ、Windows の場合はスケジュール タスクを作成して、バックアップしたファイルを安全な場所にコピーしてください。 ダンプを実行するたびに、前回のバックアップ ファイルは上書きされます。

LDIF ファイルを使用したデータのバックアップとロード

LDIF **ファイル**は、LDIF 形式でディレクトリ情報を保存したテキスト ファイルです。 LDIF ファイルを使用すると、LDAP ディレクトリ サーバ間でディレクトリ情報を転送したり、ディレクトリに適用する変更をまとめて記述したりすることができます。

CA Directory には DXdumpdb ツールが付属しており、このツールを使用してデータストアから LDIF ファイルにデータをアンロードすることができます。 後日、LDIF ファイルからデータストアにデータをロードして、ディレクトリの内容を復旧することも可能です。

LDIF ファイルへのディレクトリのバックアップ

LDIF ファイルにディレクトリをバックアップするには、以下の手順に従います。

1. ユーザ dsa (UNIX の場合)または DXserver 管理者(Windows の場合)としてログインします。
2. LDIF ファイルにデータストアをバックアップするには、以下のコマンドを実行します。

   ```
   dxdumpdb -f filename -z dsaname
   ```

   -f filename

   データのダンプ先ファイルのパスとファイル名を指定します。

   -z

   コンソール コマンド dump dxgrid-db によって生成されたデータストアのコピーから DXdumpdb がダンプを実行するよう指定します。

   dsaname

   DSA の名前を指定します。

DXdumpdb ツール - LDIF ファイルへのデータストアのデータのエクスポート

DXdumpdb ツールを使用して、データストアのデータを LDIF ファイルにエクスポートできます。

注: このコマンドを含む DXtool のすべてのコマンドが返すステータス コードの一覧については、「DXtool の終了ステータス コード」(P. 58)を参照してください。

このコマンドの形式は以下のようになります。

```
dxdumpdb options DSA
```

オプション

以下のオプションを 1 つまたは複数指定します。

-f filename

エクスポートしたデータを受け取るファイルを指定します。 このオプションを指定しなかった場合は、標準出力または画面に出力されます。

-v

詳細モードで実行します。 このオプションを指定すると、エラー トレースとステータス トレースがオンに切り替わります。 -v オプションを有効にするには、-f オプションも指定する必要があります。

-z

コンソール コマンド dump dxgrid-db によって生成されたデータストアのコピーから DXdumpdb がダンプを実行するよう指定します。

DSA

DSA を定義します。 DXdumpdb は、この DSA の設定ファイルを参照して、LDIF ファイルにエクスポートするデータストアを検索します。

例: 画面への Democorp のデータの抽出

以下の例では、democorp DSA のデータストアから画面に LDIF 形式のデータを出力します。

```
dxdumpdb democorp
```

例: オンライン データストア ダンプのバックアップ

以下の例では、オンライン データストア ダンプを LDIF ファイルにエクスポートします。

```
dxdumpdb -f eembackup -z iTechPoz-Servern
```

CA Directory データのリストア方法

CA Directory をリストアするには、以下の手順に従います。

1. DSA を停止します。
2. DXloaddb を使用して LDIF からデータストアにデータストアをロードします。

DXloaddb ツール - LDIF ファイルからのデータストアのロード

DXloaddb を使用すると、LDIF ファイルからデータストアをロードすることができます。ただし、データストアがすでに存在している必要があります。 データストア内の既存の情報はすべて削除されます。

使用上の注意

- LDIF ファイルは並べ替える必要はありません。
- DXloaddb は、LDIF ファイル内にあるクリア テキスト形式のパスワード エントリをハッシュします。

  DSA の設定でハッシュ アルゴリズムが指定されている場合、DXloadbdb はそのアルゴリズムを使用し、 指定されていない場合は、SHA-1 を使用します。
- DXloaddb は、動作属性の処理に DSA の設定をデフォルトで使用します。
  - 設定が op-attrs = true の場合、LDIF ファイルの動作属性はデータストアにロードされます。

    createTimestamp 属性が設定されていない LDIF ファイル内のエントリがあれば、データストアに creatTimestamp 属性が追加されます。
  - 設定が op-attrs = false の場合、LDIF ファイルの動作属性は無視され、DXloaddb は動作属性を作成しません。

このコマンドの形式は以下のようになります。

```
dxloaddb [options] dsa ldif-file
```

オプション

以下のオプションを 1 つまたは複数指定します。

-n

DXloaddb が一切アクションを実行しないよう指定します。

-O

パスワード ポリシー（ログイン試行回数など）やタイムスタンプ属性といった標準的な動作属性を DXloaddb に含めるよう指定します。 このオプションを指定した場合、DXloaddb は LDIF ファイルで定義されていない動作属性があれば作成します。

-s

DXloaddb が以下のデータストア関連統計を生成するよう指定します。

- 合計データサイズ（MB）
- エントリ総数
- 無視されたエントリ数
- データストアファイルへのパディング容量（KB）
- 1MB あたりの平均エントリ数

-v

詳細出力を指定します。

ldif-file

データストアにロードする LDIF ファイルの名前。

DSA

ロードするデータストアが含まれる DSA を定義します。

例：データストアの作成とロード

データストアを作成してロードする正しい順序は、以下のとおりです。

```
dxnewdb
dxloaddb
```

例：データストアへの LDIF データのロード

以下の例では、democorp.ldif ファイルからデータストア democorp へデータをロードします。

```
dxloaddb democorp democorp.ldif
```

以下は、democorp.ldif に含まれる可能性がある内容の一部です。

```
dn: o=Democorp, c=US
oc: organization
dn: ou=Administration, o=Democorp, c=US
oc: organizationalUnit
dn: cn=Fred Jones, ou=Administration, o=Democorp, c=US
oc: organizationalPerson
postalAddress: 11 Main Street $ Newtown
surname: Jones
title: Manager
telephonenumber: +1 (123) 456 7890
telephonenumber: +1 (987) 654 3210
dn: ou=Sales, o=democorp, c=US
oc: organizationalUnit
```

Telephonenumber が 2 回記述されているのは、これが複数値属性であるためです。

第 8 章: フェールオーバの設定

このセクションには、以下のトピックが含まれています。

フェールオーバ

フェールオーバは、データが使用できなくなったときでもデータ フローや運用性を中断しないための機能です。

CA EEM のフェールオーバを機能させるには、サーバにインストールした CA EEM にアプリケーションを接続して、他のサーバに関する情報を取得する必要があります。他のサーバの設定に関する情報は、フェールオーバに使用される iPoz.conf ファイルから取得できます。

CA EEM は、以下の 2 種類のフェールオーバ シナリオをサポートするように設定できます。

- データ ストア フェールオーバ
- サーバ フェールオーバ (P. 72)

注: このシナリオでは、ホスト名が Server1、Server2、～ ServerN と続くことを想定しています。

アプリケーション データ ストアのフェールオーバ

CA EEM サーバはアプリケーション データ ストアとして CA Directory を使用します。このアプリケーション データ ストアは、フェールオーバと回復に組み込み型のサポートを提供します。 フェールオーバの設定ですべてのサーバ上の以下の項目を同期します。

1. システム時間
2. セキュリティ モード（FIPS 非準拠または FIPS のみ）
3. アプリケーション データ ストア
4. DNS 参照が適切であることを確認します

重要： 同期を実行する前にアプリケーション データ ストアをバックアップします。 データ ストアのバックアップ方法の詳細については、「CA Directory に保存されている (P. 57)CA EEM データのバックアップ」を参照してください。

アプリケーション データ ストアのフェールオーバの設定

注： プライマリ サーバ上で以下の手順を実行します。 セカンダリ サーバ上で実行する手順は、明示的に指定されています。

この手順では、以下のデフォルト値を使用して CA EEM サーバをインストールしたとみなします。

- dsa ユーザ：dsa
- データ dsa ポート：509
- グループ メンバシップ：etrdir

これらのパラメータのいずれかをカスタム値にカスタマイズした場合、このデフォルト値をカスタマイズした値に置き換えてください。

アプリケーション データ ストアのフェールオーバを設定する方法

1. フェールオーバ セットアップで以下のコマンドをすべてのサーバ上で使用し、CA EEM サービスを停止します。

   Windows

   ```
   net stop igateway
   dxserver stop all
   ssld stop
   ```

   Linux および UNIX の場合

   ```
   $IGW_LOC/S99igateway stop
   su - dsa -c "dxserver stop all"
   su - dsa -c "ssld stop"
   ```

2. 以下のファイルを各セカンダリ CA EEM サーバからプライマリ サーバ(例: Server 1)の対応するフォルダにコピーします。

   Windows

   ```
   %DXHOME%\config\knowledge\iTechPoz-HostnameOfServerN.dxc
   %DXHOME%\config\knowledge\iTechPoz-HostnameOfServerN-Router.dxc
   %DXHOME%\config\ssld\personalities\iTechPoz-HostnameOfServerN.pem
   %DXHOME%\config\ssld\personalities\iTechPoz-HostnameOfServerN-Router.pem
   ```

   Linux および UNIX の場合

   ```
   $DXHOME/config/knowledge/iTechPoz-HostnameOfServerN.dxc
   $DXHOME/config/knowledge/iTechPoz-HostnameOfServerN-Router.dxc
   $DXHOME/config/ssld/personalities/iTechPoz-HostnameOfServerN.pem
   $DXHOME/config/ssld/personalities/iTechPoz-HostnameOfServerN-Router.pem
   ```

3. セカンダリ サーバからプライマリ サーバである Server 1 の一時フォルダに以下のファイルをコピーします。

   UNIX および Linux

   ```
   $DXHOME/config/ssld/iTechPoz-trusted.pem
   ```

   Windows

   ```
   %DXHOME%\config\ssld\iTechPoz-trusted.pem
   ```

4. Server 1 上のすべてのサーバの環境設定ファイル (iTechPoz-HostnameOfServerN.dxc)を以下のように編集します。

   以下の行を変更します。

   ```
   address    = tcp localhost port 509

   #address = tcp HostnameOfServerN port 509, tcp localhost port 509

   #dsa-flags  = multi-write
   ```

   以下のように変更します。

   ```
   #address    = tcp localhost port 509

   address = tcp HostnameOfServerN port 509, tcp localhost port 509

   dsa-flags   = multi-write
   ```

   注:

   - CA EEM は、をデフォルトの データ dsa ポートとしてポート番号 509 使用します。 カスタムのデータ dsa ポートを使用するように CA EEM サーバを設定している場合は、509 をカスタム ポート番号で置き換えます。
   - hostname の代わりに IP アドレスを使用するには、ダブルクォート(" ")で IP アドレスを囲みます。

5. Server 1 の iTechPoz.dxg を編集し、セカンダリ サーバの参照を追加します。

   例:

```
# iTechPoz - iTechnology Repozitory
# Source the knowledge files of the iTechPozRouter and iTechPoz DSAs.
source "iTechPoz-HostnameofServer1-Router.dxc";
source "iTechPoz-HostnameofServer1.dxc";
source "iTechPoz-HostnameOfServer2-Router.dxc";
source "iTechPoz-HostnameOfServer2.dxc";
source "iTechPoz-ServerN-Router.dxc";
source "iTechPoz-ServerN.dxc";
```

6. 各セカンダリ サーバの iTechPoz-trusted.pem のコンテンツを Server 1 のコンテンツと結合して、新しい iTechPoz-trusted.pem ファイルを作成します。

   Windows

   ```
   type <Server 2 の iTechPoz-trusted.pem への絶対パス>>><Server 1 の
   iTechPoz-trusted.pem への絶対パス>
   ```

   UNIX/Linux

   ```
   cat <Server 2 の iTechPoz-trusted.pem への絶対パス>>><Server 1 の
   iTechPoz-trusted.pem への絶対パス>
   ```

   例: type "C:¥Program Files¥CA¥Directory¥dxserver¥config¥ssld¥iTechPoz-trusted_2.pem" >> "C:¥Program Files¥CA¥Directory¥dxserver¥config¥ssld¥iTechPoz-trusted.pem

7. 各セカンダリ サーバの iTechPoz-trusted.pem のコンテンツを最終的に生成される Server 1 の iTechPoz-trusted.pem のコンテンツに結合します。

8. 以下のファイルをプライマリ サーバからすべてのセカンダリ サーバの対応するフォルダにコピーします。

   注: コピーを実行する前に、セカンダリ サーバの iTechPoz-trusted.pem、データ dsa およびルータ ファイル(iTechPoz*)をバックアップします。

   UNIX および Linux

   ```
   $DXHOME/config/ssld/iTechPoz-trusted.pem
   $DXHOME/config/ssld/personalities/iTechPoz-*.pem
   $DXHOME/config/knowledge/iTechPoz*
   ```

   Windows

   ```
   %DXHOME%\config\ssld\iTechPoz-trusted.pem
   %DXHOME%\config\ssld\personalities\iTechPoz-*.pem
   %DXHOME%\config\knowledge\iTechPoz*
   ```

9. 各セカンダリ サーバ上の iTechPoz.dxg ファイルを編集します。 iTechPoz.dxg ファイルを以下のように変更します。

   ```
   # iTechPoz - iTechnology Repozitory
   # Source the knowledge files of the iTechPozRouter and iTechPoz DSAs.
   source "iTechPoz-HostnameOfServerN-Router.dxc";
   source "iTechPoz-HostnameOfServerN.dxc";
   source "iTechPoz-HostnameOfServer1-Router.dxc";
   source "iTechPoz-HostnameOfServer1.dxc";
   ```

```
source "iTechPoz-HostnameOfServer2-Router.dxc";
source "iTechPoz-HostnameOfServer2.dxc";
source "iTechPoz-ServerKRouter.dxc";
source "iTechPoz-ServerK.dxc";
```

注: ローカル ホストのエントリは他のサーバのエントリの前に表示される必要があります。

10. UNIX または Linux 上で実行しているすべての CA EEM サーバに対し、以下の各ファイルの所有権およびグループ メンバシップを dsa と etrdir に変更します。以下のコマンドを実行します。

```
chown dsa:etrdir /opt/CA/Directory/dxserver/config/ssld/iTechPoz-trusted.pem
chown dsa:etrdir /opt/CA/Directory/dxserver/config/knowledge/iTechPoz*
```

11. すべてのサーバ上で以下のコマンドを使用し、CA EEM サービスを開始します。

Windows

```
ssld start
dxserver start all
net start igateway
```

Linux および UNIX の場合

```
su - dsa -c "ssld start"
su - dsa -c "dxserver start all"
$IGW_LOC/S99igateway start
```

アプリケーション データ ストアのフェールオーバ設定が保存されます。

CA EEM サーバのフェールオーバ

注: フェールオーバ セットアップのすべてのサーバ(Server1、Server2、～ ServerN)には必ず同じバージョンの CA EEM サーバをインストールし、それらのシステム時刻を同期してください。

フェールオーバ セットアップの他のすべてのサーバからのセッションおよび証明書を信頼するように Server1 を設定できます。 フェールオーバ セットアップのすべてのサーバに、次の手順を繰り返します。

Server1 をフェールオーバ用に設定するには、以下の手順に従います。

1. URL 「https://server1:5250/spin」を入力します。
2. iTech 管理者を選択して、[実行]をクリックします。

   [ログイン]画面が表示されます。
3. ログイン画面で選択したオプション タイプに従い、ログイン認証情報を入力します。

   ホスト

   root または administrator としてログインします。
4. [設定]タブをクリックし、[信頼済み iAuthority ホスト]ペインで[ホスト名]として ServerN を追加し、[信頼]をクリックします。

   iControl.conf ファイルにエントリが追加され、Server1 は ServerN からのセッションを信頼するようになります。

   注: フェールオーバ セットアップの他のすべてのサーバを、[信頼済み iAuthority ホスト]ペインに追加します。
5. [iAuthority]タブをクリックし、[ラベル]に「ServerN」と入力し、[信頼できるルートの追加]ペインで PEM 証明書ファイルの場所を参照して、[信頼できるルートの追加]をクリックします。

   注: PEM 証明書ファイル(rootcert.pem)は、ServerN の iTechnology ディレクトリにあります。

   iAuthority.conf ファイルにエントリが追加され、Server1 は ServerN からの証明書を信頼するようになります。

   注: フェールオーバ セットアップの他のすべてのサーバに証明書エントリを追加します。

CA EEM ファイルの設定

フォール バックに使用できるサーバのリストを受け取るように CA EEM Server1 を設定する必要があります。これはレプリケーション バージョンです。

CA EEM Server1 を設定するには、以下の手順に従います。

1. Server1 の iTechnology ディレクトリを開きます。
   - **Windows**：%IGW_LOC%
   - **Linux および UNIX**：/opt/CA/SharedComponents/iTechnology（デフォルト）
2. iPoz.conf ファイルを開き、以下のタグを追加します。

   ```
   <BackboneMember>Server2</BackboneMember>
   ```
3. iGateway を停止してから開始します。

   Windows

   ```
   net stop igateway
   net start igateway
   ```

   Linux および UNIX の場合

   ```
   /opt/CA/SharedComponents/iTechnology/S99igateway stop
   /opt/CA/SharedComponents/iTechnology/S99igateway start
   ```

CA EEM Server2 も、フォール バックに使用できるサーバのリストを受け取るように設定する必要があります。これは、レプリケーション バージョンです。

CA EEM Server2 を設定するには、以下の手順に従います。

1. Server2 の iTechnology ディレクトリを開きます。
   - **Windows**：%IGW_LOC%
   - **Linux および UNIX**：/opt/CA/SharedComponents/iTechnology（デフォルト）
2. iPoz.conf ファイルを開き、以下のタグを追加します。

   ```
   <BackboneMember>Server1</BackboneMember>
   ```
3. iGateway を停止してから開始します。

   Windows

   ```
   net stop igateway
   net start igateway
   ```

   Linux および UNIX の場合

   ```
   /opt/CA/SharedComponents/iTechnology/S99igateway stop
   /opt/CA/SharedComponents/iTechnology/S99igateway start
   ```

注：フェールオーバ セットアップで設定したすべての CA EEM サーバで、前の手順を繰り返します。

第 9 章: アーティファクト連携

アーティファクト連携の有効化

アーティファクト連携を使用する場合は、すべての CA EEM サーバのフェールオーバのセットアップで以下の手順を実行します。

アーティファクト連携の有効化

1. iGateway サービスを停止します。
2. iPoz.conf ファイルを探して開きます。
3. 以下のタグを編集します。

```
<ArtifactManager SessionTimeout="10"
RequestTimeout="30"ArtifactStore="local/federated"></ArtifactManager>
```

この中で、

SessionTimeOut

エクスポート セッションの有効時間を分単位で指定します。

デフォルト: 10 分

範囲:

RequestTimeOut

起動リクエストの有効時間を分単位で指定します。

デフォルト: 30 分

範囲:

Store

アーティファクトの格納先を指定します。 値を「local」に指定すると、フェールオーバのセットアップでは、ある CA EEM サーバのアーティファクトを他の CA EEM サーバで利用できません。 すべての CA EEM サーバでアーティファクトを利用できるようにするには、このパラメータの値を「federated」にします。

値: [local|federated]

デフォルト: local

注: SessionTimeOut と RequestTimeOut のパラメータは eiam.conf ファイルにもあります。 eiam.conf ファイルでこれらのパラメータを指定すると、eiam.conf ファイルの値が優先されます。

4. ファイルを保存して閉じます。
5. iGateway サービスを再起動します。

   アーティファクト連携は有効になります。

第 10 章: CA SiteMinder との統合

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 SiteMinder と CA EEM の統合方法

CA SiteMinder を CA EEM と統合するには、CA SiteMinder Administrator で以下を実行します。

- CA EEM と CA SiteMinder のポリシー サーバ間の通信のために、CA SiteMinder にエージェントを作成します。 エージェントが 4.x エージェントをサポートすることを確認します。
- 管理者を作成するか、またはシステム レベル スコープを持つ既存のデフォルト管理者「SiteMinder」を使用します。
- CA EEM が LDAP 属性を取得するために使用する、許可用の CA SiteMinder ユーザ ディレクトリを作成します。
- sAMAccountName や UID など、ディレクトリのユーザを一意に識別できるように[UniversallD]フィールドを設定します。 UniversalID は、SiteMinder UI、[ユーザ ディレクトリ]、[プロパティ]、[ユーザ属性]タブから設定できます。
- [ユーザ属性]タブのパスワード属性(RW)を userPassword に設定します。
- CA EEM がユーザの認証に使用する、認証用の CA SiteMinder データストアを作成します。

  注: 認証用と許可用のユーザ ストアが同じ場合は、許可用に作成されている既存のユーザ ストアを使用します。
- リソース フィルタを「/iamt.html」に設定して領域を作成します。
- CA SiteMinder ドメインを作成し、ユーザ ディレクトリ、管理者、および領域をドメインに追加します。

CA SiteMinder の詳細については、CA SiteMinder のマニュアルを参照してください。

CA SiteMinder モジュールに対する CA EEM のサーバ側のログ記録の設定

CA SiteMinder 統合にログ レベルを設定する方法

1. 以下の内容でファイルを作成し、sm_log.properties という名前でファイルを保存します。

```
#filename: sm_log.properties
#set the default logging level for the root logger
.level = INFO
#set the default logging level for the logger name com.ca.eiam
com.ca.eiam.level = ALL
```

2. sm.properties ファイルで com.ca.eiam ロガー用のログ レベルを以下のいずれかの値に変更します。

   SEVERE

   重大な失敗を示すメッセージのレベルを指定します。

   WARNING

   警告を示すレベルを指定します。

   INFO

   通知メッセージのレベルを指定します。

   CONFIG

   静的な設定メッセージのレベルを指定します。

   FINE

   トレース情報のレベルを指定します。

   ALL

   すべてのレベルのメッセージをログに記録するように指定します。

3. ファイルを以下の場所に保存します。

   Windows

   %IGW_LOC%

   Linux および UNIX の場合

   /opt/CA/SharedComponents/iTechnology

4. iGateway サービスを停止します。

5. 手順 3 で指定した場所から iGateway.conf ファイルを開き、<JVMSettings></JVMSettings> タグの間に以下のタグを追加します。

```
<Properties name="eiam.sm">
<system-properties>java.util.logging.config.file=sm_log.properties</system-pr
operties>
</Properties>
```

6. ファイルを保存して閉じます。
7. iGateway サービスを開始します。

第 11 章: CA EEM SDK ログ記録

Java と C++ の SDK の場合、CA EEM の新しいログ記録プロセスは、ロガー フレームワークとして log4j と log4cxx をそれぞれ利用します。 以前のログ記録プロセスは、safe::util ロガー ユーティリティを使っていました。 この新機能には、以下の利点があります。

- ログ レベルを更新、または変更しても、アプリケーションを再起動する必要がありません。
- ロガー設定ファイルでパラメータを編集する方法で、ファイル名、ファイル サイズ、バックアップ ログ ファイルの数などのログ記録プロパティを管理できます。
- ネットワーク コールやパフォーマンス統計に CA EEM SDK ログ メッセージを分類できます。

注: CA EEM C# SDK のログ記録はアップグレードされていません。 CA EEM C# SDK では、safe::util を継続して使用する必要があります。

ログ記録により、CA EEM SDK で生成されたメッセージ、エラー、情報を記録できます。 CA EEM SDK では、以下のファイルによってログ記録を制御します。

- eiam.log4cxx.config
- eiam.log4j.config

これらの 2 ファイルは CA EEM SDK パッケージの一部であり、デフォルトでは以下のように Bin フォルダに格納されています。

UNIX

/opt/CA/eIAMSDK/bin

Windows

C:¥Program Files¥CA¥Embedded IAM SDK¥bin

ロガー設定ファイルについて

ロガー設定ファイルの eiam.log4cxx.config および eiam.log4j.config は CA EEM SDK ログ記録を設定するために使われます。 これらのファイルには、以下の主要コンポーネントが含まれます。

- アペンダ
- ロガー
- ルート ロガー

これらのコンポーネントには設定可能なパラメータが含まれているため、それぞれのビジネス要件に合わせてログ記録プロセスをカスタマイズできます。

アペンダ

アペンダには、各ロガーのログ記録を制御するパラメータが含まれています。 ロガー設定ファイルには、デフォルトで以下のアペンダが含まれています。

SDK

SDK メッセージをログ ファイルにログ記録します。 ログ ファイルのファイル名を含むパスを指定します。

デフォルト: eiam.cppsdk.log

注: Windows 上の Tomcat サーバ下でアプリケーションを展開している場合、パスには左上がりスラッシュ('¥')の代わりに必ず右上がりスラッシュ('/')を使ってください。 左上がりスラッシュを使うと、ログ ファイルは指定したパスではなく、Apache Tomcat フォルダに作成されます。

ネットワーク

ネットワーク コール関連のメッセージをログ ファイルにログ記録します。

デフォルト: eiam.network.cpp.log

パフォーマンス

パフォーマンス コール関連のメッセージをログ ファイルにログ記録します。

デフォルト: eiam.performance.cpp.log

コンソール

コンソールにログ メッセージを表示します。

SDK アペンダはデフォルトで有効です。 ほかのアペンダを有効にするには、それぞれのコードからコメント文字列(<!-- and -->)を削除します。

アペンダは以下の設定可能なパラメータで構成されています。

file

アペンダのログ ファイル名を指定します。

append

ログ メッセージのセットをログ ファイルに追加するかどうか指定します。 値が true の場合、ログ メッセージのセットがログ ファイルの最後のログ メッセージに追加されます。

BufferedIO

最新のログ メッセージをバッファするかどうか指定します。 値が true の場合、最新の数件のログ メッセージがログ ファイルに書き込まれる前にメモリに格納されます。 これは IO 操作を最小化し、ログ レベルが高い場合に効果があります。

値: [true|false]

デフォルト: false

注: BufferedIO のデフォルト サイズは 8 KB です。

maxFileSize

ログ ファイルの最大サイズを指定します。 ログ ファイルが最大サイズを超えると、ファイル名が log.1 の新しいログ ファイルが作成され、ログ ファイルの内容は log.1 ファイルに転送されます。 ログ ファイルには、最新のログ メッセージが含まれます。 このファイルが最大サイズを再度超えると、ファイル名が log.2 の新しいログ ファイルが作成され、log.1 の内容は log.2 ファイルに転送され、ログ ファイルの内容は log.1 ファイルに転送されます。

デフォルト: 10 MB

最小: 10 KB

最大: 2 GB

注: maxFileSize の最小サイズは BufferedIO のサイズ以上でなければなりません。

maxBackupIndex

古いログの保存に使用するバックアップ ログ ファイルの最大数を指定します。 ログ ファイルの数が最大バックアップ インデックス値を超えると、最も古いログ メッセージのファイルが削除されます。

デフォルト: 1

最小: 1

最大: 12

ConversionPattern

ログ メッセージのフォーマットを指定します。 変換パターンを定義するフォーマット修飾子および変換文字を設定します。

注: 変換パターンの詳細については、www.apache.org のトピック log4j を参照してください。

例: SDK アペンダ

```
<appender name="SDK" class="org.apache.log4j.RollingFileAppender">
    <!-- The active sdk log file -->
    <param name="file" value="eiam.cppsdk.log" />
    <param name="append" value="true" />
    <param name="BufferedIO" value="false"/>
    <param name="maxFileSize" value="10000KB" />
    <param name="maxBackupIndex" value="1" />
    <layout class="org.apache.log4j.PatternLayout">
    <!-- The log message pattern -->
    <param name="ConversionPattern" value="%5p %d{ISO8601} [%t] [%c] %m%n"/>
    </layout>
</appender>
```

eiam.log4net.config のアペンダ

アペンダには、各ロガーのログ記録を制御するパラメータが含まれています。 デフォルトでは、ロガー設定ファイルには以下のアペンダが含まれています。

SDK

SDK メッセージをログ ファイルに記録します。 ログ ファイルのパス(ファイル名を含む)を指定します。

デフォルト: EIAM.C#SDK.log

注: Windows 上の Tomcat サーバにアプリケーションを展開している場合は、円記号(¥)の代わりにスラッシュ(/)を使用していることを確認してください。 円記号を使用すると、ログ ファイルは指定したパスに作成されません。代わりに、ログ ファイルが Apache Tomcat フォルダに作成されます。

Network

ネットワーク コールに関するメッセージをログ ファイルに記録します。

デフォルト: EIAM.NETWORK.C#SDK.log

Performance

パフォーマンス コールに関するメッセージをログ ファイルに記録します。

デフォルト: EIAM.PERFORMANCE.C#SDK.log

Console

コンソール上にログ メッセージを表示します。

デフォルトでは、SDK アペンダが有効に設定されています。 他のアペンダを有効にするには、対応するコードからコメント文字列(<!-- and -->)を削除します。

アペンダは、設定可能な以下のパラメータで構成されています。

file

アペンダのログ ファイル名を指定します。

appendToFile

ログ メッセージのセットをログ ファイルに追加するかどうかを指定します。 値が true の場合、ログ メッセージのセットがログ ファイルの最後のログ メッセージに追加されます。

maxSizeRollBackups

古いログの保持に使用するバックアップ ログ ファイルの最大数を指定します。 ログ ファイルの数がバックアップ インデックスの最大値を超えた場合、最も古いログ メッセージのファイルが削除されます。

デフォルト: 1

最小: 1

最大: 12

rollingStyle

最新のログ メッセージをバッファするかどうかを指定します。 値が true の場合、ログ ファイルに書き込む前に最新のいくつかのログ メッセージがメモリに保存されます。 この設定は、ログ レベルが高い場合に便利です。また、IO 操作を最小限に抑えます。

値: [true|false]

デフォルト: false

注: BufferedIO のデフォルト サイズは 8 KB です。

maximumFileSize

ログ ファイルの最大サイズを指定します。 ログ ファイルが最大サイズを超えると、新しいログ ファイルのファイル名として log.1 が作成されます。また、ログ ファイルのコンテンツは log.1 ファイルに転送されます。 このログ ファイルには、最新のログ メッセージが含まれています。 このファイルが再度最大サイズ超えると、新しいログ ファイルのファイル名として log.2 が作成されます。log.1 のコンテンツは log.2 ファイルに転送されます。また、ログ ファイルのコンテンツは log.1 ファイルに転送されます。

デフォルト: 10 MB

最小: 10 KB

最大: 2 GB

注: maxFileSize の最小サイズは、rollingStyle のサイズ以上に設定する必要があります。

ConversionPattern

ログ メッセージのフォーマットを指定します。 変換パターンを定義するには、変換文字および形式修飾子を設定します。

注: 変換パターンの詳細については、www.apache.org の log4net の記述を参照してください。

ロガー

ロガーでは、ネットワークとパフォーマンスのログ メッセージをレベルに従って分類する方法、および実行時に表示する方法を制御できます。 デフォルトでは、ネットワーク ロガーとパフォーマンス ロガーは無効です。 ロガーを有効にするには、それぞれのコードからコメント文字列を削除します。

ロガーには以下のパラメータが含まれています。

logger name

ロガーの名前を指定します。

additivity

ネットワーク、またはパフォーマンス ログ メッセージを SDK ログ ファイルに複製するかどうか指定します。

値: [true|false]

デフォルト: false

level value

ロガーのログ レベルを指定します。

値: [Trace|Debug|Info|Warn|Error|Fatal|Off]

以下にログ レベルを優先する順番で示します。

注: ログ レベルが高くなるほど、CA EEM のパフォーマンスが低下します。

トレース

低いレベルのデバッグを示します。 制御フローが含まれており、引数を渡します。

デバッグ

問題の診断で使用するメッセージを示します。 コンテキスト情報が含まれます。

情報

本稼働環境の実行をきめの粗いレベルでトレースするコンテキスト情報を示します。

警告

システムの潜在的な問題を示します。 たとえば、セキュリティに対応するメッセージ カテゴリでは、辞書攻撃が検出されると、警告メッセージが表示される必要があります。

クリティカル

システムの重大問題を示します。 この問題は回復できないため、手動介入を必要とします。

致命的

致命的なアプリケーション例外を示します。

オフ

ログ記録がないことを示します。

注: デフォルトの SDK アペンダのログ レベルはクリティカルである必要があります。

例: パフォーマンス ロガー

```
<logger name="Perform" additivity="false">
    <level value="trace"/>
    <appender-ref ref="Performance" />
</logger>
```

ルート ロガー

ルート ロガーは、すべてのアペンダのログ レベルを制御します。 ただし、ルート ロガー内の参照アペンダのログ レベルが親アペンダで指定されたログ レベルと異なる場合、優先度の高いログ レベルが優先度の低いログ レベルに優先されます。

たとえば、ルート ロガーのログ レベルがクリティカル、ネットワーク アペンダのログ レベルがトレースの場合、トレースはクリティカルより優先されるため、実行時、システムはログ レベルをトレースとしてログ メッセージを処理します。

例:ルート ロガー

```
<root>
    <priority value="error" />
    <appender-ref ref="SDK" />
    <appender-ref ref="Console" />
</root>
```

ロガー ファイルの設定

CA EEM では、ネットワーク、パフォーマンス、コンソールおよび SDK クラスに関連するログ メッセージを設定できます。

ロガー ファイルを設定する方法

1. テキスト エディタで、ロガー設定ファイルの eiam.log4cxx.config または eiam.log4j.config を開きます。
2. ロガーとアペンダを有効にします。
3. アペンダ パラメータを更新します。
4. ロガー設定ファイルを保存します。

eiam.log4cxx.config ファイルの例

以下は、eiam.log4cxx.config ファイルの例です。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<!-- Note that this file is read by the sdk every 60 seconds -->

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

    <appender name="SDK" class="org.apache.log4j.RollingFileAppender">
        <!-- The active sdk log file -->
        <param name="file" value="eiam.cppsdk.log" />
        <param name="append" value="true" />
        <param name="BufferedIO" value="false"/>
        <param name="maxFileSize" value="10000KB" />
        <param name="maxBackupIndex" value="1" />
        <layout class="org.apache.log4j.PatternLayout">
        <!-- The log message pattern -->
        <param name="ConversionPattern" value="%5p %d{ISO8601} [%t] [%c] %m%n"/>
        </layout>
    </appender>

    <appender name="Network" class="org.apache.log4j.RollingFileAppender">
        <!-- The file to log Network calls -->
        <param name="file" value="eiam.network.cpp.log" />
        <param name="append" value="true" />
        <param name="BufferedIO" value="true"/>
        <param name="maxFileSize" value="10000KB" />
        <param name="maxBackupIndex" value="1" />
        <layout class="org.apache.log4j.PatternLayout">
        <!-- The log message pattern -->
        <param name="ConversionPattern" value="%5p %d{ISO8601} [%t] [%c] %m%n"/>
        </layout>
    </appender>

    <appender name="Performance" class="org.apache.log4j.RollingFileAppender">
        <!-- The file to log Performance calls -->
        <param name="file" value="eiam.performance.cpp.log" />
        <param name="append" value="true" />
        <param name="BufferedIO" value="true"/>
        <param name="maxFileSize" value="10000KB" />
        <param name="maxBackupIndex" value="1" />
        <layout class="org.apache.log4j.PatternLayout">
        <!-- The log message pattern -->
        <param name="ConversionPattern" value="%5p %d{ISO8601} [%t] [%c] %m%n"/>
        </layout>
    </appender>

    <appender name="Console" class="org.apache.log4j.ConsoleAppender">
        <!-- Logs to Console -->
        <layout class="org.apache.log4j.PatternLayout">
```

```
        <!-- The log message pattern -->
        <param name="ConversionPattern" value="%5p %d{ISO8601} [%t] [%c] %m%n"/>
        </layout>
    </appender>

    <!-- Remove comment to enable Performance Logging -->
    <!--
    <logger name="Perform" additivity="false">
        <level value="trace"/>
        <appender-ref ref="Performance" />
    </logger>
    -->

    <!-- Remove comment to enable Network Logging -->
    <!--
    <logger name="Network" additivity="false">
        <level value="trace"/>
        <appender-ref ref="Network" />
    </logger>
    -->

    <root>
        <priority value="error" />
        <appender-ref ref="SDK" />
        <!-- <appender-ref ref="Console" /> -->
    </root>
</log4j:configuration>
```

eiam.log4net.config ファイルの例

eiam.log4net.config ファイルの例を以下に示します。

```
<?xml version="1.0" encoding="utf-8" ?>

<log4net>
        <appender name="SDK" type="log4net.Appender.RollingFileAppender">
                <file value="EIAM.C#SDK.log" />
                <appendToFile value="true" />
                <maxSizeRollBackups value="1" />
                <maximumFileSize value="10000KB" />
                <rollingStyle value="Size" />
                <layout type="log4net.Layout.PatternLayout">
                        <conversionPattern value="%date [%thread] %-5level %logger
- %message%newline" />
                </layout>
        </appender>

        <appender name="Network" type="log4net.Appender.RollingFileAppender">
                <file value="EIAM.NETWORK.C#SDK.log" />
                <appendToFile value="true" />
                <maxSizeRollBackups value="1" />
                <maximumFileSize value="10000KB" />
                <rollingStyle value="Size" />
                <layout type="log4net.Layout.PatternLayout">
                        <conversionPattern value="%date [%thread] %-5level %logger
- %message%newline" />
                </layout>
        </appender>

        <appender name="Performance" type="log4net.Appender.RollingFileAppender">
                <file value="EIAM.PERFORMANCE.C#SDK.log" />
                <appendToFile value="true" />
                <maxSizeRollBackups value="1" />
                <maximumFileSize value="10000KB" />
                <rollingStyle value="Size" />
                <layout type="log4net.Layout.PatternLayout">
                        <conversionPattern value="%date [%thread] %-5level %logger
- %message%newline" />
                </layout>
        </appender>

        <appender name="ConsoleAppender" type="log4net.Appender.ConsoleAppender">
                <layout type="log4net.Layout.PatternLayout">
                        <conversionPattern value="%date [%thread] %-5level %logger
- %message%newline" />
                </layout>
        </appender>
```

```
        <!-- Uncomment to enable Performance Logging -->
        <!--
        <logger name="Perform" additivity="false">
                <level value="ERROR"/>
                <appender-ref ref="Performance" />
        </logger>-->

        <!-- Uncomment to enable Network Logging -->
        <!--<logger name="Network" additivity="false">
                <level value="ERROR"/>
                <appender-ref ref="Network" />
        </logger>-->

        <root>
                <level value="ERROR" />
                <appender-ref ref="SDK" />
                <!--    <appender-ref ref="ConsoleAppender" />   -->
        </root>
</log4net>
```

eiam.lo4j.config ファイルの例

eiam.log4cxx.config ファイルの例を以下に示します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<!-- Note that this file is read by the sdk every 60 seconds -->

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

        <appender name="SDK" class="com.ca.eiam.log4j.RollingFileAppender">
                <!-- The active sdk log file -->
                <param name="file" value="eiam.javasdk.log" />
                <param name="append" value="true" />
                <param name="BufferedIO" value="false"/>
                <param name="maxFileSize" value="10000KB" />
                <param name="maxBackupIndex" value="1" />
                <layout class="com.ca.eiam.log4j.PatternLayout">
                        <!-- The log message pattern -->
                        <param name="ConversionPattern" value="%5p %d{ISO8601} [%t]
[%c] %m%n"/>
                </layout>
        </appender>

        <appender name="Network" class="com.ca.eiam.log4j.RollingFileAppender">
                <!-- The file to log Network calls -->
                <param name="file" value="eiam.network.java.log" />
                <param name="append" value="true" />
                <param name="BufferedIO" value="false"/>
                <param name="maxFileSize" value="10000KB" />
                <param name="maxBackupIndex" value="1" />
                <layout class="com.ca.eiam.log4j.PatternLayout">
                        <!-- The log message pattern -->
                        <param name="ConversionPattern" value="%5p %d{ISO8601} [%t]
[%c] %m%n"/>
                </layout>
        </appender>

        <appender name="Performance"
class="com.ca.eiam.log4j.RollingFileAppender">
                <!-- The file to log Performance calls -->
                <param name="file" value="eiam.performance.java.log" />
                <param name="append" value="true" />
                <param name="BufferedIO" value="false"/>
                <param name="maxFileSize" value="10000KB" />
                <param name="maxBackupIndex" value="1" />
                <layout class="com.ca.eiam.log4j.PatternLayout">
                        <!-- The log message pattern -->
```

```
                        <param name="ConversionPattern" value="%5p %d{ISO8601} [%t]
[%c] %m%n"/>
                </layout>
        </appender>

        <appender name="Console" class="com.ca.eiam.log4j.ConsoleAppender">
                <!-- Logs to Console -->
                <layout class="com.ca.eiam.log4j.PatternLayout">
                        <!-- The log message pattern -->
                        <param name="ConversionPattern" value="%5p %d{ISO8601} [%t]
[%c] %m%n"/>
                </layout>
        </appender>

        <!-- Uncomment to enable Performance Logging -->
        <!--
        <logger name="Perform" additivity="false">
                <level value="trace"/>
                <appender-ref ref="Performance" />
        </logger>
        -->

        <!-- Uncomment to enable Network Logging -->
        <!--
        <logger name="Network" additivity="false">
                <level value="trace"/>
                <appender-ref ref="Network" />
        </logger>
        -->

        <root>
                <priority value="error" />
                <appender-ref ref="SDK" />
                <!-- <appender-ref ref="Console" /> -->
        </root>

</log4j:configuration>
```

第 12 章: 外部ディレクトリ サーバ サポートの設定

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 EEM を使用した外部ディレクトリの設定

認証用と許可用に異なる外部ディレクト リストアを使用している場合、CA EEM を以下のように設定します。

- CA EEM に外部認証ディレクトリを設定するには、iPoz.conf ファイルを使用します。
- CA EEM サーバに外部許可ディレクトリを設定するには、CA EEM 管理者 GUI を使用します。

  注: 外部ディレクトリへの参照の設定方法については、「オンライン ヘルプ」を参照してください。

認証に外部ディレクトリを使用するように CA EEM サーバを設定するには、インストール後に /CA/SharedComponents/iTechnology フォルダにある iPoz.conf ファイルに以下のオプションを設定します。

注: iPoz.conf ファイルを変更する前に iGateway を停止し、後で再起動します。

UseExternalAuthDirectory

認証用に別の外部ディレクトリを使用するかどうかを指定します。 別の外部ディレクトリを使用する場合は、「True」を入力します。 デフォルトは「False」です。

ExternalAuthDirType:

外部ディレクトリのタイプを指定します。 現在サポートされているタイプは、CA Identity Manager、Custom Mapped Directory、Microsoft Active Directory、Novell eDirectory、Novell eDirectory-CN、および Sun One Directory です。

ExternalAuthDirUserDn

指定した外部ディレクトリのタイプに対する UserDn を指定します。

ExternalAuthDirPassword

暗号化された形式でユーザ パスワードを指定します。

注: 以下のコマンドを使用してパスワードを暗号化し、それを ipoz.conf ファイルに貼り付けてください。

```
/iTechnology/safex -munge <password in clear text>
```

ExternalAuthDirHost

外部ディレクトリが設定されているホスト名を指定します。

ExternalAuthDirPort

外部ディレクトリが listen するポートを指定します。

ExternalAuthDirUserSearchPreFilter

外部ディレクトリに従って事前検索フィルタを指定します。 ユーザなどの任意のオブジェクト クラスを検索できます。

ExternalAuthDirUserSearchPostFilter

外部ディレクトリに従って事後検索フィルタを指定します。 ユーザなどの任意のオブジェクト クラスを検索できます。

ExternalDirCacheFolder

CA EEM サーバが外部ディレクトリ フォルダをキャッシュする必要があるかどうかを指定します。 このタグを True に設定すると、CA EEM サーバは外部フォルダをキャッシュします。また、CA EEM 管理者 GUI を使用して、これらのフォルダにアクセスできます。 このタグを False に設定すると、CA EEM は CA EEM 管理者 GUI に外部ディレクトリ フォルダを表示しません。

値: [True|False]

デフォルト: True

外部ディレクトリによって返された DN 内のスラッシュをエスケープするように CA EEM サーバを設定する

外部ディレクトリを認証に使用するように CA EEM サーバを設定するには、インストール後に /CA/SharedComponents/iTechnology フォルダにある iPoz.conf ファイルの以下のオプションを設定します。

注: iPoz.conf ファイルを変更する前に iGateway を停止し、後で再起動します。

ExternalDirEscapeSlash

CA EEM が外部ディレクトリによって返された DN でスラッシュ(/)をエスケープする必要があるかどうかを指定します。 CA EEM でスラッシュをエスケープする必要がある場合は、このタグを True に設定します。

注: DN のスラッシュをエスケープするように CA EEM を設定する必要があります。設定しない場合、CA EEM はオブジェクトを正常に取得できない場合があります。

値: [True|False]

デフォルト: False

外部ディレクトリ フェールオーバ サポートの設定

CA EEM の機能を拡張し、サーバのレプリケーション バージョンである別の外部ディレクトリ サーバにフォールバックするようにできます。

このためには、iPoz.conf でマッピングを指定します。

注: iPoz.conf ファイルを変更する前に iGateway を停止し、変更した後で再起動する必要があります。

ExternalDirHostBackup

レプリケートされた外部ディレクトリ サーバのホスト名を指定します。

ExternalAuthDirHostBackup

ユーザ認証に使用する別の外部ディレクトリ サーバのホスト名を指定します。

TLS を使用した LDAP サーバへの接続

LDAP サーバへの TLS 接続を確立するには、匿名証明書を許可するように LDAP サーバを設定する必要があります。 TLS 使用して LDAP に接続するように EEM を設定するには、以下を実行します。

TLS 使用して LDAP に接続するように CA EEM を設定する方法

1. CA EEM GUI にログインします。
2. [設定]-[EEM サーバ]をクリックします。
3. [グローバル ユーザ/グローバル グループ]をクリックします。

   EEM サーバを設定するペインが表示されます。
4. [外部ディレクトリから参照]オプションを選択します。
5. 設定の詳細を入力します。

   注: 設定の詳細については、「オンライン ヘルプ」を参照してください。
6. (オプション)[TLS の使用]オプションを選択します。
7. [保存]をクリックします。

SSL での LDAP サーバへの接続

LDAP サーバと SSL 接続を確立するには、以下の証明書が必要です。

認証局の証明書

この証明書は、Verisign や Thwate などの認証局から入手できます。 この証明書は、認証局が発行した証明書が有効で信頼できることを示しています。

LDAP サーバの証明書

この証明書は、信頼できる認証局から入手する必要があります。 この証明書には LDAP サーバに関する情報が含まれており、クライアントが LDAP サーバを識別できるようになっています。

注: CA EEM では、SSL 接続に PEM 証明書のみをサポートしてます。

SSL で CA EEM と LDAP サーバを接続する方法

以下に、SSL で CA EEM サーバと LDAP サーバを接続するプロセスについて示します。

1. CA EEM サーバは、認証局の証明書を使用して LDAP サーバに接続します。
2. LDAP サーバは認証局の証明書を確認して、証明書が有効なら CA EEM サーバとのハンドシェイクを確立します。
3. ハンドシェイク中、LDAP サーバは CA EEM サーバに公開キーを送信します。この公開キーは、LDAP サーバに送信されるデータを暗号化するために使用されます。
4. CA EEM サーバは、公開キーを使用してデータを暗号化し、そのデータを LDAP サーバに送信します。
5. CA EEM サーバは、LDAP サーバを認証するユーザ名とパスワードを送信します。

SSL 接続の設定方法

LDAP サーバと CA EEM サーバ間の SSL 通信を設定するには、以下の手順に従います。

1. 証明書を使用する LDAP サーバを設定します。
2. SSL で通信するように CA EEM サーバを設定します。

SSL 証明書を使用するための LDAP サーバの設定

SSL を使用するように LDAP サーバを設定するには、以下の手順に従います。

1. 認証局から証明書を入手し、LDAP サーバのトラステッド証明書ストアに証明書をインストールします。
2. 認証局からサーバの証明書を取得し、LDAP サーバのサーバ証明書ストアに証明書をインストールします。
3. LDAP サーバが SSL 接続を受け付けることができるようにします。

CA EEM サーバでの SSL の有効化

サーバで SSL を有効にするには、以下の手順に従います。

1. LDAP サーバから認証局の証明書をコピーし、それを CA EEM サーバが実行されているコンピュータに保存します。
2. ipoz.conf ファイルを開き、以下のタグを編集します。

   <ExternalDirSSL>

   SSL 通信を有効にするか無効にするかを指定します。 SSL 通信を有効にするには、このタグを true に設定する必要があります。

   <ExternalDirCACertPath>

   認証局の証明書が保存されている、CA EEM サーバが実行中のコンピュータ上のパスを指定します。
3. igateway を再起動します。

第 13 章: 大量のポリシーをサポートする設定

このセクションには、以下のトピックが含まれています。

大量のポリシーのサポート

注: CA EEM が大量のポリシーのサポートを提供するのは、C++ SDK に対応したクライアント環境のみです。

大量のポリシーを使用するアプリケーションを登録する前に、CA EEM サーバとクライアントを設定しておく必要があります。

注: HP-UX プラットフォームでは、CA EEM は最大 20,000 ポリシーをサポートしています。

AIX における CA EEM サーバの追加設定

AIX で大量のポリシーを使用できるようにするには、以下の手順を実行して CA EEM サーバを追加設定する必要があります。

AIX で CA EEM サーバを設定するには、以下の手順に従います。

1. AIX コマンドプロンプトから以下のコマンドを実行して、ネットワーク設定を変更します。

   ```
   no -o tcp_nodelayack=1
   ```

2. AIX コマンドプロンプトから以下のコマンドを実行して、プロセス制限を増やします。

   ```
   ulimit -d unlimited
   ulimit -f unlimited
   ```

クライアントの設定

大量のポリシーを使用できるようにするには、クライアントを設定する必要があります。

すべてのオペレーティング システムに共通のクライアントの設定

大量のポリシーの展開をサポートするには、すべてのオペレーティング システムのクライアントを設定する必要があります。

- アプリケーションのキャッシュ更新時間を長くして、Safex を使用しているアプリケーションの登録中にキャッシュが更新されないようにします。

  キャッシュ更新の詳細については、「**プログラミング ガイド**」を参照してください。

  注: 登録中にキャッシュ更新が行われないようにするには、登録中におけるキャッシュ更新時間を 3600 秒に設定することをお勧めします。 登録後、キャッシュ更新時間をデフォルト設定の 30 秒に戻します。

- [信頼できるイベント配信]を有効にします。

  [信頼できるイベント配信]の詳細については、「**プログラミング ガイド**」を参照してください。

第 14 章: イベントのアーカイブ

このセクションには、以下のトピックが含まれています。

概要

CA EEM では、CA EEM サーバが生成したイベントのレポートを生成および管理することができます。 システムをアーカイブすると、アーカイブされたファイルは以下の 3 種類の状態になります。

ウォーム データベース ファイル

イベント数がイベント データベースの最大行数を超えたときに作成されるアーカイブ ファイルを指します。 ウォーム アーカイブ ファイルは、CA EEM サーバからクエリおよびレポートできます。 ウォーム データベース ファイルにデータを挿入することはできません。 ウォーム データベース ファイルは、[イベント ログ設定]の[最大アーカイブ日数]で指定した日数、CA EEM サーバで使用することができます。

コールド データベース ファイル

別の場所に手動でバックアップされているウォーム状態のアーカイブファイルを指します。 コールド データベース ファイルからクエリを実行したり、レポートを作成することはできません。 コールド データベース ファイルをクエリまたはレポートに使用するには、あらかじめ解凍しておく必要があります。

解凍データベース ファイル

リストアされたコールド状態のアーカイブファイルを指し、ユーザは CA EEM サーバからクエリを実行したり、レポートを生成したりすることができます。 解凍データベース ファイルは、[イベント ログ設定]の[イベント ポリシー]として指定した時間、アーカイブ ディレクトリで使用することができます。

[イベント ログ設定]を変更するには、以下の手順に従います。

1. CA EEM にログインします。

   CA EEM のホームページが表示されます。

2. [レポートの管理]-[設定]-[サービス]-[イベント ログ設定]をクリックします。

   [イベント ログ設定]画面が表示されます。

注: レポートを管理するためにサービスを設定する方法の詳細については、「**オンライン ヘルプ**」を参照してください。

コールド データベース ファイルの解凍ユーティリティ

CA EEM には、コールド データベース ファイルを解凍するためのユーティリティが用意されています。 ファイルに対してクエリを実行してライブ レポートを参照するには、あらかじめファイルをリストアして、コールド状態からウォーム状態に解凍しておく必要があります。 この機能は sem ユーティリティで提供されています。 sem ユーティリティは、テクニカル サポート サイト http://www.casupport.jp からダウンロードすることができます。

sem ユーティリティを設定するには、以下の手順に従います。

1. sem ユーティリティの圧縮ファイルを解凍します。
2. 使用しているオペレーティング システムに合わせて、環境変数を設定します。

   Linux または Solaris

   ```
   Export LD_LIBRARY_PATH = <sem_Extraction_Folder>:$LD_LIBRARY_PATH
   ```

   AIX

   ```
   Export LIBPATH = <sem_Extraction_Folder> :$LIBPATH
   ```

   HP-UX

   ```
   Export SHLIB_PATH= <sem_Extraction_Folder> :$SHLIB_PATH
   ```

注: Windows の場合、sem ユーティリティを設定するには、コマンドラインから、解凍したフォルダにアクセスして sem.exe を実行する必要があります。

SEM ユーティリティの構文

sem ユーティリティの構文は、以下のとおりです。

sem -h <ホスト名> -u <ユーザ> -p <パスワード> -listcolddb | -defrost <アーカイブ>

-h

コールド データベース ファイルが保存されているコンピュータのホスト名を指定します。

-u

CA EEM サーバへの認証に使用するユーザ名を指定します。

-p

CA EEM サーバへの認証に使用するユーザ名のパスワードを指定します。

-listcolddb

ホスト コンピュータに保存されているすべてのコールド データベース ファイルのリストを表示します。

-defrost <archive>

指定したアーカイブ ファイルを解凍します。

-fips

sem ユーティリティが FIPS に準拠したアルゴリズムを使用するように指定します。

注: CA EEM サーバが FIPS のみのモードに設定されている場合、sem ユーティリティを -fips オプションと共に使用する必要があります。

以下の表に、sem ユーティリティの戻り値を示します。

戻り値	説明
0	成功
1	無効な引数
2	無効なユーザ名
3	認証に失敗しました
4	コールド データベース ファイルの一覧表示に失敗しました
5	コールド データベース ファイルの解凍に失敗しました
6	初期化エラー

コールド データベース ファイルの解凍

ファイルに対してクエリを実行してライブ レポートを参照するには、あらかじめファイルをリストアして、コールド状態からウォーム状態に解凍しておく必要があります。

注: コールド データベース ファイルを解凍する前に、ファイルをアーカイブ ディレクトリ iTechnology¥calm_archive にコピーしておく必要があります。

コールド データベース ファイルのリストアおよび解凍方法

1. 現在の calm_archive フォルダにバックアップした calm_archive フォルダをコピーします。
2. コマンド ラインから sem ユーティリティを実行して、すべてのコールド データベース ファイルの一覧を取得します。

   sem -h <hostname> -u <username> -p <password> -listcolddb

   FIPS のみのモードの CA EEM サーバ

   sem -h <hostname> -u <username> -p <password> -fips -listcolddb
3. sem ユーティリティを実行して、コールド データベース ファイルを解凍します。

   sem -h <hostname> -u <username> -p <password> -defrost <archive>

   FIPS のみのモードの CA EEM サーバ

   sem -h <hostname> -u <username> -p <password> -fips -defrost <archive>

   コールド データベース ファイルがリストアおよび解凍されます。