Oracle® Universal Content Management

Security Providers コンポーネント管理ガイド

10*g* リリース 3

部品番号 : B51299-01

2008 年 10 月

ORACLE®

Oracle Universal Content Management Security Providers コンポーネント管理ガイド , 10*g* リリース 3

部品番号 : B51299-01

原本名 : Oracle Universal Content Management Security Providers Component Administration Guide, Version 10*g*R3

目次

索引

1

はじめに

概要

この章では、次の内容について説明します。

製品の概要

Security Providers コンポーネントでは、2 種類の新しいプロバイダで基本的なソケット・プロバイダの機能を拡張することにより、Content Server のセキュリティを向上させます。

- Secure Socket Layer（SSL）プロバイダ
- keepalive プロバイダ

Security Providers コンポーネントを使用する利点は、次のとおりです。

- SSL によって通信の暗号化と認証が行われることで、Web 通信のセキュリティが高まります。

- セキュリティ・プロバイダによって、ソケットまたはサーバー認証に対し証明書を使用できるようになります。
- keepalive および接続プーリング・ロジックにより、SSL ソケットの作成とティアダウンが削減され、SSL コスト・オーバーヘッドを防ぐことができます。

要件

Security Providers コンポーネントの使用を決める際、次の事項を考慮することが重要です。

- このコンポーネントは、サポートされている Windows および UNIX オペレーティング・システムで稼働する Content Server で実行されます。
- このドキュメントでは、コンポーネントは、Content Server バージョン 10*g*R3 で実行されるものとします。

コンポーネントの内容

Security Providers コンポーネント・ファイル SecurityProvider.zip のダウンロードが可能です。これには、サンプルなども含まれています。次のファイルが含まれています。

説明	ファイル名
コンポーネント・ファイル	*.hda *.class *.htm *.zip

対象読者

このマニュアルは、ネットワーク通信、および Content Server と Web ブラウザとの間のサーバー / クライアント通信のプロバイダを設計、インストールおよび管理する人を対象としています。これらの設計者および管理者は、ネットワークとセキュリティの概要、および Content Server サービスとプロセスに精通している必要があります。

このマニュアルについて

このマニュアルでは、Content Server で、Security Providers コンポーネントをインストールおよび構成する手順を示します。このドキュメントに含まれる情報は、製品テクノロジの進歩や、ハードウェア、オペレーティング・システム、サード・パーティ・ソフトウェアの作成や変更に伴って変わる可能性があります。

表記規則

このマニュアルでは次の表記規則を使用します。

- *<install_dir>/<instance>* という表記は、特定の Content Server のインスタンスがインストールされているシステム上の場所の参照に使用されます。
- スラッシュ（/）は、インターネット・アドレスの区切りに使用されます。たとえば、http://www.microsoft.com/windows2000/ などです。インターネット・アドレスの最後に、スラッシュが付く場合と付かない場合があります。
- バックスラッシュ（\）は、パスのレベル（サーバー、ディレクトリまたはファイル）の区切りに使用されます。たとえば、C:\stellent\ です。Windows ファイル・システムのファイルを参照する場合も、UNIX システムのファイルを参照する場合もこれを使用します。バックスラッシュは、サーバー、ディレクトリまたはファイル・パスの最後に必ず付加されます。
- オペレーティング・システムのダイアログまたはウィンドウにアクセスするパスには、次の形式を使用します。

  「スタート」→「設定」→「コントロール パネル」
- 必要なユーザー入力は、次のフォント書式を使用して示されます。

  `xyz_name`

2

インストール

概要

この章では、次の内容について説明します。

SECURITY PROVIDERS コンポーネントのインストール

Stellent Content Server に Security Providers をインストールして有効にするには、Component Wizard または Component Manager を使用します。

Component Wizard でのインストール

Component Wizard を使用してコンポーネントをインストールするには、この手順を使用します。

1. リリース 10*g* のコンポーネント更新バンドルから Security Providers のコンポーネント・ファイル SecurityProviders.zip を入手します。ファイルを一時的な場所に置きます。
2. 「スタート」→「プログラム」→「**Stellent Content Server**」→「**< インスタンス >**」→「**Utilities**」→「**Component Wizard**」と選択します。

   Component Wizard のメイン画面と「Component List」画面が表示されます。
3. 「Component List」画面で、「**Install**」をクリックします。

   「Install」画面が表示されます。

4. 「**Select**」をクリックします。SecurityProviders.zip ファイルをダウンロードした場所へ移動し、これを選択します。
5. 「**Open**」をクリックします。

   zip ファイルの内容が、「Install」画面のリストに追加されます。
6. 「**OK**」をクリックします。

   Component Wizard により、Security Providers コンポーネントを有効にするかどうかが確認されます。
7. 「**Yes**」をクリックします。

   「Component List」画面に、Security Providers コンポーネントが有効として表示されます。
8. コンテンツ・サーバーを再起動します。

Component Manager でのインストール

Component Manager を使用してコンポーネントをインストールするには、この手順を使用します。

1. リリース 10*g* のコンポーネント更新バンドルから Security Providers のコンポーネント・ファイル SecurityProviders.zip を入手します。ファイルを一時的な場所に置きます。
2. 新たにブラウザ・ウィンドウを開き、システム管理者として Stellent Content Server にログインします。
3. 「Administration」トレーを開きます。
4. 「**Admin Applets**」をクリックして「Administration」ページを開きます。
5. 「**Admin Server**」をクリックします。
6. 該当するコンテンツ・サーバー・インスタンスをクリックします。
7. サイドバーで「**Component Manager**」をクリックします。

   Component Manager の画面が表示されます。
8. 「Install New Component」ボックスの隣にある「Browse」ボタンを選択します。SecurityProviders.zip ファイルをクリックして選択します。
9. 「**Install**」をクリックします。

   インストールされるコンポーネント・アイテムがリストされたページが表示されます。
10. 「**Continue**」をクリックしてインストールを続けます。

    インストールが成功したことを示す Content Server のメッセージが表示されます。

11. メッセージ・ページをクリックして Component Manager に戻ります。
12. 「Disabled Components」ボックスでコンポーネント名をクリックします。
13. **「Enable」**をクリックしてコンポーネントを有効にします。

    「Enabled Components」ボックスにコンポーネントがリストされます。
14. サイドバーで**「Start/Stop Content Server」**をクリックします。
15. Content Server を再起動します。

3

SECURITY PROVIDERS の使用

概要

この章では、次の内容について説明します。

SECURITY PROVIDERS について

Security Providers コンポーネントでは、標準の Stellent Content Server 受信 / 送信ソケット・プロバイダに Secure Socket Layer（SSL）の暗号化および認証を使用します。このコンポーネントでは、ソケット・プロバイダに対する keepalive 機能も有効になり、SSL ソケットの作成およびティアダウンが最小限になります。セキュリティ・プロバイダと鍵、証明書を適切に使用することで、Content Server とのネットワークおよびインターネット通信のセキュリティが向上します。

Security Providers コンポーネントを使用するには、ソケット・プロバイダ、セキュリティと認証、SSL、keepalive、およびその他のネットワーク通信のセキュリティに精通している必要があります。Security Providers コンポーネントを使用する場合、次の情報源を利用できます。

❖ 『システム設定およびプロセスの管理』

❖ 『セキュリティおよびユーザー・アクセスの管理』

- 「Sun Java Secure Socket Extension (JSSE) Reference Guide for the Java 2 SDK, Standard Edition, v. 1.4.2」

  このオンライン・ドキュメントは、www.sun.com から参照できます。広範な関連ドキュメントのセクションがあり、書籍、セキュリティ規格、政府のセキュリティ政策と規定を参照できる Web リンクや、暗号化と SSL に関する書籍のリストが記載されています。

- 「keytool Key and Certification Management Tool」

  このオンライン・ドキュメントは、www.sun.com から参照できます。

- RSA の「Public Key Cryptography Standards」

  このオンライン・ドキュメントは、RSA のサイト（www.rsasecurity.com）から参照できます。

- RSA の「Cryptography FAQ」

  このオンライン・ドキュメントは、RSA のサイト（www.rsasecurity.com）から参照できます。

- 「SSL Certificate FAQ」

  このオンライン・ドキュメントは、Linux Documentation Project（www.tldp.org）から参照できます。

次に、このマニュアルで使用する用語の定義をリストします。詳細は、前述の情報のリストや、セキュリティと認証の規格について参照してください。

- 証明書 : エンティティ（個人または企業）の ID および公開鍵を証明するデジタル署名。証明書は、認証局または個々のエンティティによって発行されます。
- 認証局（CA）: 他のエンティティに証明書を発行するエンティティ。認証局は、VeriSign や Thawte など、証明所の発行元として広く知られ、信頼できるとみなされています。
- キーストア : 鍵に関する情報のファイルまたはデータベース。認証プロセスに使用されます。
- 秘密鍵 : 発行元のエンティティのみが知る、鍵としてパッケージされる情報。秘密鍵は、署名の生成に使用されます。
- 公開鍵 : エンティティに関連付けられ、公開される鍵としてパッケージされる情報。公開鍵は、署名の証明に使用されます。
- SSL: Secure Socket Layer。公開鍵と秘密鍵のテクノロジを組み合せて使用するセキュアなネットワーク通信プロトコル。
- トラストストア : トラスト・マネージャが信頼できるとした鍵のファイルまたはデータベース。

計画

SSL ソケット・プロバイダまたは keepalive ソケット・プロバイダを実装する前に、セキュリティ・プロバイダの使用方法を決めることをお薦めします。keepalive および SSL の接続タイプを確認し、選択したセキュリティ・プロバイダを使用するために、キーストアまたはトラストストアの作成などの追加の構成が必要かどうかを確認します。追加の情報源については、3-1 ページの「Security Providers について」を参照してください。

次の項で、SSL と keepalive のプロバイダ・タイプについて説明します。また、プロバイダ・タイプの動作の制御に使用される Java クラスと、必要な追加構成についても説明します。

keepalive 接続

keepalive 機能では、サービス・リクエストに対して、永続的な接続とソケット接続のプーリングが可能になります。keepalive 接続の設定は、接続の設定とティアダウンに非常に時間がかかり、この時間を最小限にする必要がある場合に、最も役立ちます。Security Providers コンポーネントには、受信および送信の 2 つの keepalive ソケット・プロバイダが用意されています。

keepalive 受信ソケット・プロバイダの設定には、次の Java クラスが使用されます。

プロバイダ・クラス	idc.provider.ExtendedSocketIncomingProvider
接続クラス	idc.provider.KeepaliveSocketIncomingConnection
サーバー・スレッド・クラス	idc.server.KeepaliveIdcServerThread

keepalive 送信ソケット・プロバイダの設定には、次の Java クラスが使用されます。

プロバイダ・クラス	idc.provider.KeepaliveSocketOutgoingProvider
接続クラス	idc.provider.KeepaliveSocketOutgoingConnection
リクエスト・クラス	idc.provider.KeepaliveServerRequest

SSL 接続

SSL プロバイダの設定によって、keepalive 環境での SSL 接続の使用が可能になります。この設定では、SSL ソケットの設定とティアダウンのコストを最小限にできるため、単純な SSL プロバイダより推奨されします。Security Providers コンポーネントには、keepalive を使用する、受信および送信の 2 つの SSL ソケット・プロバイダが用意されています。

SSL keepalive 受信ソケット・プロバイダの設定には、次の Java クラスが使用されます。

プロバイダ・クラス	idc.provider.ssl.SSLSocketIncomingProvider
接続クラス	idc.provider.KeepaliveSocketIncomingConnection
サーバー・スレッド・クラス	idc.server.KeepaliveIdcServerThread

keepalive SSL 送信ソケット・プロバイダの設定には、次の Java クラスが使用されます。

プロバイダ・クラス	idc.provider.KeepaliveSocketOutgoingProvider
接続クラス	idc.provider.ssl.SSLSocketOutgoingConnection
リクエスト・クラス	idc.provider.KeepaliveServerRequest

追加構成

選択したセキュリティ・プロバイダのタイプに応じて、追加の構成が必要な場合があります。

- **keepalive および SSL 送信プロバイダ :** 「Add Provider」ページに「Num Connections」フィールドがあり、プールする接続を指定します。
- **SSL 受信プロバイダ :** 「Add Provider」ページに、次の 2 つのオプションがあります。
  - 「Request Client Auth」オプション : 接続を作成する際、可能な場合は、クライアントで認証を行います。
  - 「Require Client Auth」オプション : 接続を作成する場合、クライアントで認証を行う必要があります。

SSL プロバイダでは、「Request Client Auth」オプションの値、「Require Client Auth」オプションの値と、これらのオプションで処理される証明書に署名した認証局のタイプに応じて、クライアントとサーバーに対するキーストアとトラストストアの設定も必要です。キーストアとトラストストアの詳細は、3-5 ページの「キーストアとトラストストア」を参照してください。

キーストアとトラストストア

SSL プロバイダでは、キーストアの使用とトラストストアが必要な場合があります。キーストアは、SSL で使用する公開鍵と秘密鍵の情報が含まれるファイルです。トラストストアには、信頼できるとみなされた証明書が含まれます。サーバーとクライアントで使用される証明書が、VeriSign や Thawte などの既知の認証局（CA）によって署名されている場合、トラストストアは必要ありません。これは、デフォルトの JVM トラストストアにこれらの CA の証明書が含まれているためです。SSL プロバイダで使用される証明書が自己署名されているか、プライベート CA によって署名されている場合、トラストストアが必要です。SSL プロバイダでキーストアおよびトラストストアが必要な場合、これを作成して管理する必要があります。

次の項で、キーストアとトラストストアの概要について説明します。

- キーストアとトラストストアを使用する場合（3-5 ページ）
- キーストアとトラストストアの情報の指定（3-6 ページ）
- キーストアの生成（3-6 ページ）
- トラストストアの作成（3-7 ページ）

キーストアとトラストストアの詳細は、3-1 ページの「Security Providers について」にリストした情報源を参照してください。

キーストアとトラストストアを使用する場合

次に、キーストアとトラストストアを使用する状況の例を示します。

- SSL ソケットを作成するために、署名された SSL 証明書を含むキーストアがサーバーで必要です。
- クライアント認証がサーバーでリクエストされたか、必要です。この認証にトラストストアが必要です。クライアントの証明書が、既知の CA によって署名されていないため、CA の証明書を含むトラストストアがサーバーで必要です。
- クライアント認証がサーバーでリクエストされたか、必要です。この認証では、認証用にクライアントで使用する証明書を含むキーストアをクライアントが持っている必要があります。
- 既知の CA によって署名されていない証明書がサーバーで使用されるため、サーバーの証明書を含むトラストストアがクライアントに必要です。

キーストアとトラストストアの情報の指定

キーストアとトラストストアの情報を使用するには、SSL 受信 / 送信プロバイダで、sslconfig.hda という名前のファイルをプロバイダ・ディレクトリ（provider.hda ファイルの次）に設定する必要があります。sslconfig.hda ファイルには、キーストアおよびトラストストアに対して指定した構成情報が含まれます。次の例のような形式です。セキュリティ上の理由から、このファイルの編集に使用できる Web インタフェースはありません。テキスト・エディタを使用して、すべての編集を手動で行う必要があります。このファイルまたはその他の .hda ファイルの各行の最後に後続空白文字を含めないようにしてください。

```
@Properties LocalData
TruststoreFile=/servers/idc/data/providers/ssloutgoing1/truststore
KeystoreFile=/servers/idc/data/providers/ssloutgoing1/keystore
@end
```

構成名	値の説明
TruststoreFile	トラストストア・ファイルへのフルパス
KeystoreFile	キーストア・ファイルへのフルパス

キーストアの生成

この項では、キーストアの生成の基本的なプロセスについて説明します。特定の要件と、SSL プロバイダ用に作成するキーとキーストアの名前を決める必要があります。sslconfig.hda ファイルに KeystoreFile 構成設定のフルパスが含まれるため、キーストア・ファイルは任意の場所に格納できます。ただし、キーストア・ファイルは、*<install_dir>*/data/providers/*<provider_name>* ディレクトリ（provider.hda ファイルおよび sslconfig.hda ファイルの次）または *<install_dir>*/config/ ディレクトリに格納することをお薦めします。Content Server のプロバイダ・ページを使用して、別名とパスワードを設定します。

keytool ユーティリティを使用してキーストアを生成する方法については、Sun 社のサイト（www.sun.com）から、ドキュメント「keytool Key and Certification Management Tool」を参照してください。

	注意 : Java keytool ユーティリティには、秘密鍵との直接対話を防ぐ機能があります。つまり、keytool を使用して生成される証明書はキーストアに格納され、証明書の秘密鍵の部分を取得する方法はありません。反対に、keytool を使用して既存の証明書を Java キーストアにインポートする方法もありません。 Portecle Java キーストア・ツールでは、Java キーストアからの秘密鍵のインポートとエクスポートを行えます。Portecle の詳細は、portecle.sourceforge.net を参照してください。

keytool を使用するには、コマンドを入力する際、パスにユーティリティが含まれている必要があります。

1. キーストアに鍵を作成します。次のコマンドラインの例では、*keystore* という名前のキーストアに *alias* という名前の鍵のエントリを作成する方法を示します。このコマンドでは、キーストアのパスワード、鍵の生成に使用される情報および鍵自体のパスワードが要求されます。鍵のパスワードがキーストアのパスワードと異なる場合、鍵の取得に KeystoreAlias と KeystoreAliasPassword の値が必要です。

```
keytool -genkey -v -alias <alias> -keystore <keystore>
```

2. 証明書の署名リクエストを生成します。次のコマンドラインの例では、*keystore* という名前のキーストアの *alias* という名前の鍵のエントリに対して証明書の署名リクエストを生成し、*csr_file* という名前のファイルに格納する方法を示します。このファイルを CA に送り、署名できます。

```
keytool -certreq -v -alias <alias> -keystore <keystore> -file <csr_file>
```

3. CA の証明書をキーストアにインポートします。インポート時に、ユーザーの証明書のトラスト・チェーンが、keytool で確認されます。証明書の署名が既知ではない CA によって行われ、CA に関する情報が keytool にない場合、証明書は拒否されます。つまり、既知ではない CA からの証明書は、まず、キーストアにインポートして、次のステップでユーザーの証明書を正常にインポートできるようにします。次のコマンドラインの例では、*cert_file* という名前のファイルの証明書を、*keystore* という名前のキーストアにインポートする方法を示します。

```
keytool -import -v -alias <ca_alias> -keystore <keystore> -file <cert_file>
```

4. 署名された証明書をキーストアに再度インポートします。証明書の署名リクエストが CA で受信され、署名された証明書が CA から送信されると、*alias* で指定されたキーストア・エントリに証明書が読み込まれます。次のコマンドラインの例では、署名された証明書のインポート方法を示します。

```
keytool -import -v -alias <alias> -keystore <keystore_name> -file <csr_file>
```

5. キーストアの内容を確認します。

```
keytool -list -v -keystore <keystore_name>
```

トラストストアの作成

この項では、トラストストアの生成の基本的なプロセスについて説明します。既知の認証局によって署名されていない鍵を、SSL プロバイダで使用する場合、トラストストアが必要です。トラストストアには、コンテンツ・サーバーのトラストストアの管理者（トラスト・マネージャ）によって証明される公開証明書のみが含まれます。特定の要件と、作成するトラストストアの名前を決める必要があります。sslconfig.hda ファイルに TruststoreFile 構成設定のフルパスが含まれるため、トラストストア・ファイルは任意の場所に格納できます。ただし、トラストストア・ファイルは、*<stellent_dir>*/data/providers/*<provider_name>* ディレクトリ（provider.hda ファイルおよび sslconfig.hda ファイルの次）または *<stellent_dir>*/config/ ディレクトリに格納することをお薦めします。

keytool ユーティリティを使用してトラストストアを生成する方法については、Sun 社のサイト（www.sun.com）から、ドキュメント「keytool Key and Certification Management Tool」を参照してください。

keytool を使用するには、コマンドを入力する際、パスにユーティリティが含まれている必要があります。

トラストストアを作成するには、次のコマンドを入力します。

```
keytool -import -v -alias <alias> -keystore <keystore> -file <cert_files>
```

変数	説明
<alias>	鍵の別名
<keystore>	キーストアの名前
<cert_files>	認証局の証明書へのパス

SSL プロバイダと KEEPALIVE プロバイダの管理

SSL セキュリティ・プロバイダと keepalive セキュリティ・プロバイダの管理には、次のタスクがあります。

- 受信セキュリティ・プロバイダの追加（3-8 ページ）
- 送信セキュリティ・プロバイダの追加（3-9 ページ）

コンテンツ・サーバーのプロバイダの管理と、プロバイダの編集と削除の方法については、『システム設定およびプロセスの管理』を参照してください。

受信セキュリティ・プロバイダの追加

受信セキュリティ・プロバイダを追加するには、次の手順に従います。

1. 「Providers」ページ（3-11 ページ参照）を表示します。
2. 「Create a New provider」表で、受信セキュリティ・プロバイダ・タイプの「Action」列の**「Add」**をクリックします。

   受信セキュリティ・プロバイダのページが表示されます。

3. 次のフィールドを完成させます。

   必須フィールド

   - Provider Name
   - Provider Description
   - Provider Class（入力済）
   - Server Port

   オプションのフィールド

   - Connection Class（入力済）
   - Configuration Class
   - Server Thread Class（入力済）

   オプションのチェック・ボックス（sslincoming プロバイダのみ）

   - Request Client Authentication
   - Require Client Authentication

4. 「**Add**」をクリックします。

   「Providers」表に新規プロバイダが追加された「Providers」ページが表示されます。

5. コンテンツ・サーバーを再起動します。

送信セキュリティ・プロバイダの追加

送信セキュリティ・プロバイダを追加するには、次の手順に従います。

1. 「Providers」ページ（3-11 ページ参照）を表示します。
2. 「Create a New provider」表で、送信セキュリティ・プロバイダ・タイプの「Action」列の「**Add**」をクリックします。

   送信セキュリティ・プロバイダのページが表示されます。

3. 次のフィールドを完成させます。

   必須フィールド

   - Provider Name
   - Provider Description
   - Provider Class（入力済）
   - Server Host Name（入力済）
   - Server Port
   - Instance Name
   - Relative Web Root

オプションのフィールド

- Connection Class（入力済）
- Configuration Class
- Request Class（入力済）
- Number of Connections（入力済）
- HTTP Server Address
- Proxied（チェック・ボックス）
- Notify Target（チェック・ボックス）
- Users（チェック・ボックス）
- Released Documents（チェック・ボックス）
- Enterprise Searchable（チェック・ボックス）
- Required Roles
- Account Filter

4. 「**Add**」をクリックします。

   「Providers」表に新規プロバイダが追加された「Providers」ページが表示されます。

5. コンテンツ・サーバーを再起動します。

セキュリティ・プロバイダ・インタフェース・ページ

Security Providers コンポーネントでは、次のプロバイダ・インタフェース画面を使用して、セキュリティ・プロバイダを管理できます。

❖ keepaliveincoming プロバイダの追加ページ（3-12 ページ）

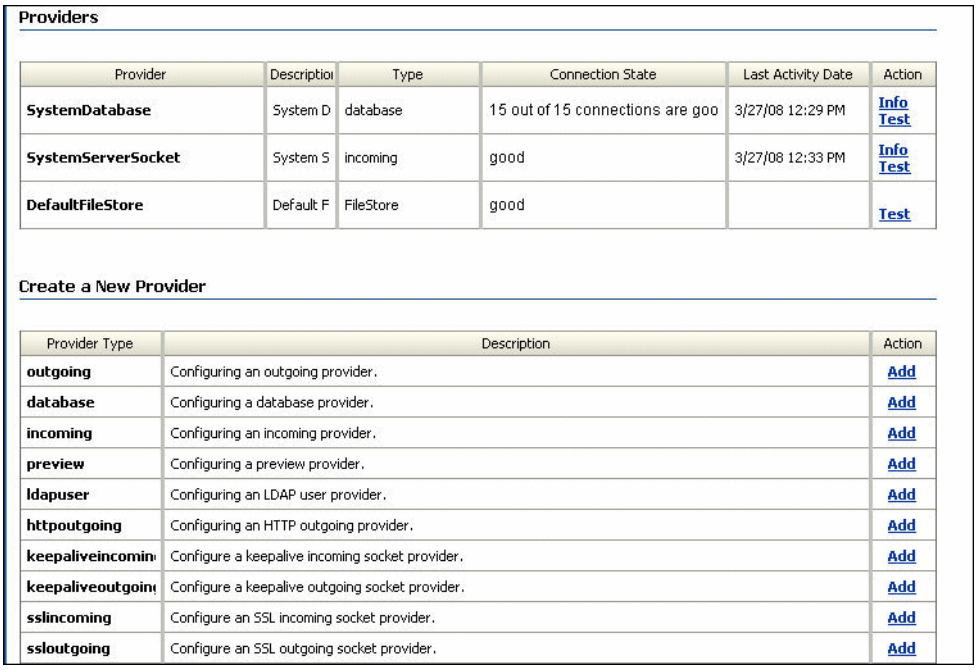
❖ keepaliveoutgoing プロバイダの追加ページ（3-14 ページ）

❖ sslincoming プロバイダの追加ページ（3-17 ページ）

❖ ssloutgoing プロバイダの追加ページ（3-19 ページ）

「Providers」ページ

Security Providers コンポーネントをインストールして有効にすると、管理者は、「Providers」ページを使用して SSL プロバイダおよび keepalive プロバイダを作成できます。このページにアクセスするには、Content Server の「Administration」メニューで「Providers」リンクをクリックします。



Providers

Provider	Descriptio	Type	Connection State	Last Activity Date	Action
SystemDatabase	System D	database	15 out of 15 connections are goo	3/27/08 12:29 PM	Info Test
SystemServerSocket	System S	incoming	good	3/27/08 12:33 PM	Info Test
DefaultFileStore	Default F	FileStore	good		Test

Create a New Provider

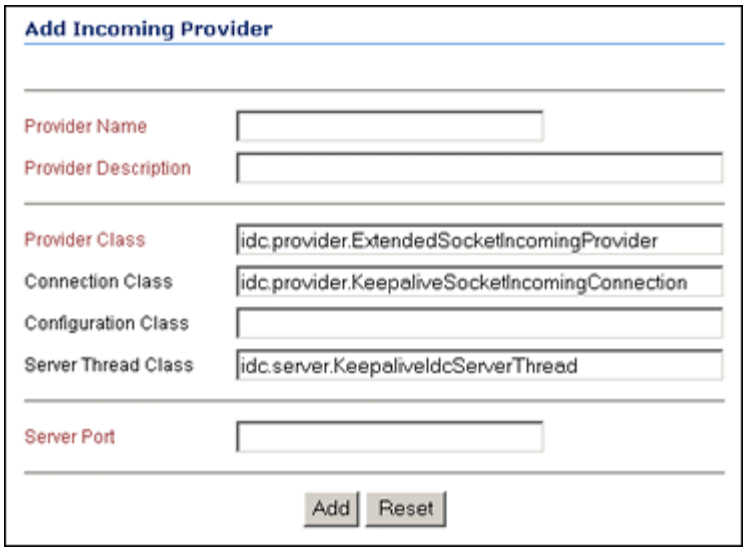
Provider Type	Description	Action
outgoing	Configuring an outgoing provider.	Add
database	Configuring a database provider.	Add
incoming	Configuring an incoming provider.	Add
preview	Configuring a preview provider.	Add
ldapuser	Configuring an LDAP user provider.	Add
httpoutgoing	Configuring an HTTP outgoing provider.	Add
keepaliveincomin	Configure a keepalive incoming socket provider.	Add
keepaliveoutgoin	Configure a keepalive outgoing socket provider.	Add
sslincoming	Configure an SSL incoming socket provider.	Add
ssloutgoing	Configure an SSL outgoing socket provider.	Add

機能	説明
「Providers」表	
「Providers」列	外部エンティティへの接続を確立するプロバイダの名前。
「Description」列	プロバイダの説明。
「Provider Type」列	プロバイダのタイプ。incoming、database など。
「Connection State」列	次のステータスが表示されます。 • misconfigured • good • down • requires restart

機能	説明
「Last Activity Date」列	プロバイダがアクティブだった最終日時。
「Actions」列	• 「Info」リンクでは、プロバイダの「Provider Information」ページが表示されます。 • 「Test」リンクでは、プロバイダの「Connection State」列と「Last Activity Date」列がリフレッシュされます。
「Create a New Provider」表	
「Provider Type」列	プロバイダのタイプ。
「Description」列	プロバイダ・タイプの説明。
「Action」列	「Add」ボタンをクリックすると、そのタイプのプロバイダの「Add/Edit Provider」ページが表示されます。

keepaliveincoming プロバイダの追加ページ

keepalive 機能の「Add Incoming Provider」ページは、keepalive ソケット受信プロバイダの作成に使用されます。このページにアクセスするには、Content Server の「Administration」メニューで「Providers」リンクをクリックし、keepaliveincoming プロバイダ・タイプの「Action」列で**「Add」**をクリックします。



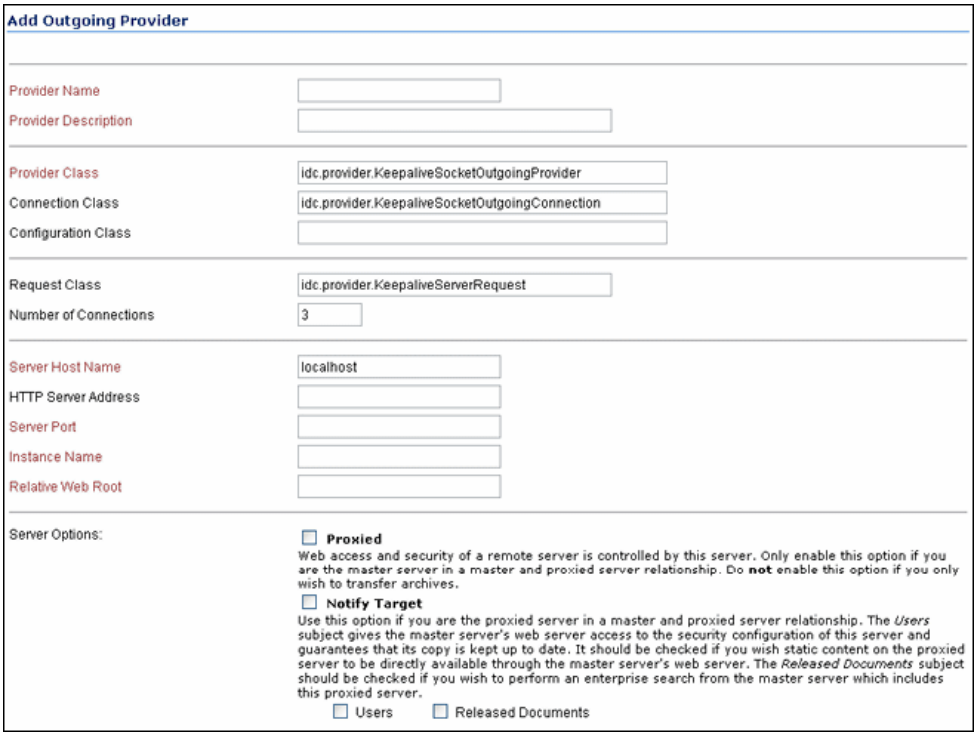
機能	説明
「Provider Name」フィールド *	プロバイダの名前。
「Provider Description」フィールド *	プロバイダの説明。
「Provider Class」フィールド *	プロバイダの Java クラスの名前。idc.provider.ExtendedSocketIncomingProvider など。
「Connection Class」フィールド	プロバイダ接続を実装する Java クラスの名前。idc.provider.KeepaliveSocketIncomingConnection など。
「Configuration Class」フィールド	追加構成を行う Java クラスの名前。このクラスは、接続クラスがすでにプロバイダであるデータベース・プロバイダに非常に有益です。
「Server Thread」フィールド	受信接続のサーバー・スレッドの名前。idc.provider.KeepaliveIdcServerThread など。
「Server Port」フィールド *	プロバイダが受信接続をリスニングするポート。たとえば、受信システム・プロバイダでは、デフォルトで、ポート 4444 をリスニングします。
「Add」ボタン	プロバイダ情報を保存します。
「Reset」ボタン	最後に保存した値に、プロバイダ情報をリセットします。

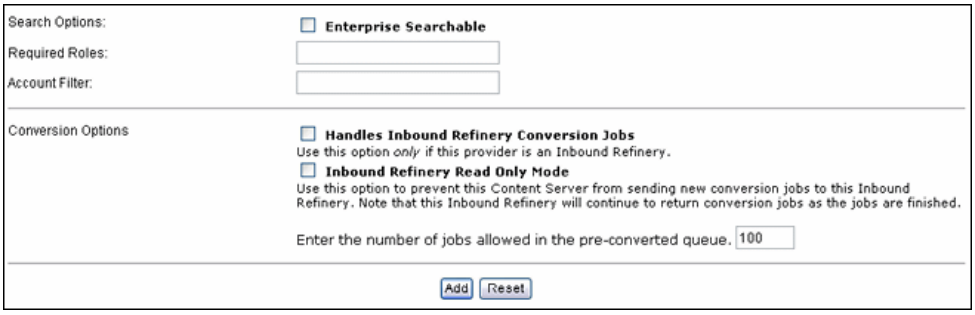
* 必須メタデータ・フィールド。

keepaliveoutgoing プロバイダの追加ページ

管理者は、keepalive 機能の「Add Outgoing Provider」ページを使用して、keepalive ソケット送信プロバイダを作成できます。このページにアクセスするには、Content Server の「Administration」メニューから「Providers」リンクを選択し、keepaliveoutgoing プロバイダ・タイプの「Action」列で「**Add**」を選択します。

次の 2 つの画像に、このページを示します。





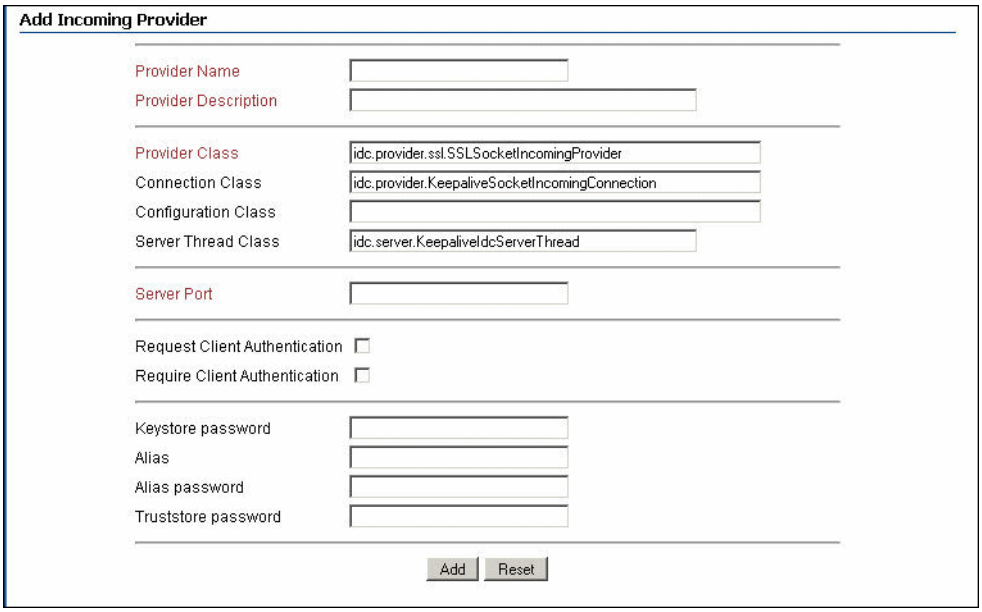
機能	説明
「Provider Name」フィールド *	プロバイダの名前。
「Provider Description」フィールド *	プロバイダの説明。
「Provider Class」フィールド *	プロバイダの Java クラスの名前。idc.provider.KeepaliveSocketOutgoingProvider など。
「Connection Class」フィールド	プロバイダ接続を実装する Java クラスの名前。idc.provider.KeepaliveSocketOutgoingConnection など。
「Configuration Class」フィールド	追加構成を行う Java クラスの名前。
「Request Class」フィールド	サーバー・リクエストを実装する Java クラスの名前。idc.provider.KeepaliveServerRequest など。
「Number of Connections」フィールド	最大接続数。3 など。
「Server Host Name」フィールド *	他のコンテンツ・サーバー・インスタンスのサーバー・ホスト名。localhost など。
「HTTP Server Address」フィールド	他のコンテンツ・サーバー・インスタンスの HTTP アドレス。
「Server Port」フィールド *	プロバイダが他のコンテンツ・サーバーと通信するポート。
「Instance Name」フィールド *	他のコンテンツ・サーバー・インスタンスのインスタンス名。
「Relative Web Root」フィールド *	他のコンテンツ・サーバー・インスタンスの相対 Web ルート。
「Proxied」チェック・ボックス	プロバイダが、現在のインスタンスによって制御されるコンテンツ・サーバーと接続する場合、このオプションを有効にします。

機能	説明
「Notify Target」チェック・ボックス	プロバイダが、制御インスタンスとして機能するコンテンツ・サーバーと接続し、ユーザー情報やコンテンツ・アイテム情報の変更時に、このコンテンツ・サーバーから制御インスタンスへの通知を行う場合、このオプションを有効にします。
「Users」チェック・ボックス	ユーザー情報の変更時に、このコンテンツ・サーバーから制御インスタンスへの通知を行う場合、このオプションを有効にします。
「Released Documents」チェック・ボックス	コンテンツ・アイテム情報の変更時に、このコンテンツ・サーバーから制御インスタンスへの通知を行う場合、このオプションを有効にします。
「Enterprise Searchable」チェック・ボックス	Enterprise Search が有効で、このコンテンツ・サーバー・インスタンスを検索可能にする場合、このオプションを有効にします。詳細は、『Enterprise Search Administration and User Guide』を参照してください。
「Required Roles」フィールド	Enterprise Search を使用してこのコンテンツ・サーバー・インスタンスを検索する権限を持つロールを入力します。ロールが入力されない場合、すべてのユーザーが権限を持ちます。
「Account Filter」フィールド	Enterprise Search を使用してこのコンテンツ・サーバー・インスタンスを検索する権限を持つアカウントを入力します。アカウントが入力されない場合、すべてのユーザーが権限を持ちます。
「Conversion」オプション	プロバイダが Inbound Refinery かどうかの指定に使用します。
「Add」ボタン	プロバイダ情報を保存します。
「Reset」ボタン	最後に保存した値に、プロバイダ情報をリセットします。

* 必須メタデータ・フィールド。

sslincoming プロバイダの追加ページ

管理者は、sslincoming 機能の「Add Incoming Provider」ページを使用して、SSL ソケット受信プロバイダを作成できます。ページにアクセスするには、Content Server の「Administration」メニューから「Providers」リンクをクリックし、sslincoming プロバイダ・タイプの「Action」列から「**Add**」を選択します。



機能	説明
「Provider Name」フィールド *	プロバイダの名前。
「Provider Description」フィールド *	プロバイダの説明。
「Provider Class」フィールド *	プロバイダの Java クラスの名前。idc.provider.ssl.SSLSocketIncomingProvider など。
「Connection Class」フィールド	プロバイダ接続を実装する Java クラスの名前。idc.provider.KeepaliveSocketIncomingConnection など。
「Configuration Class」フィールド	追加構成を行う Java クラスの名前。このクラスは、接続クラスがすでにプロバイダであるデータベース・プロバイダに非常に有益です。

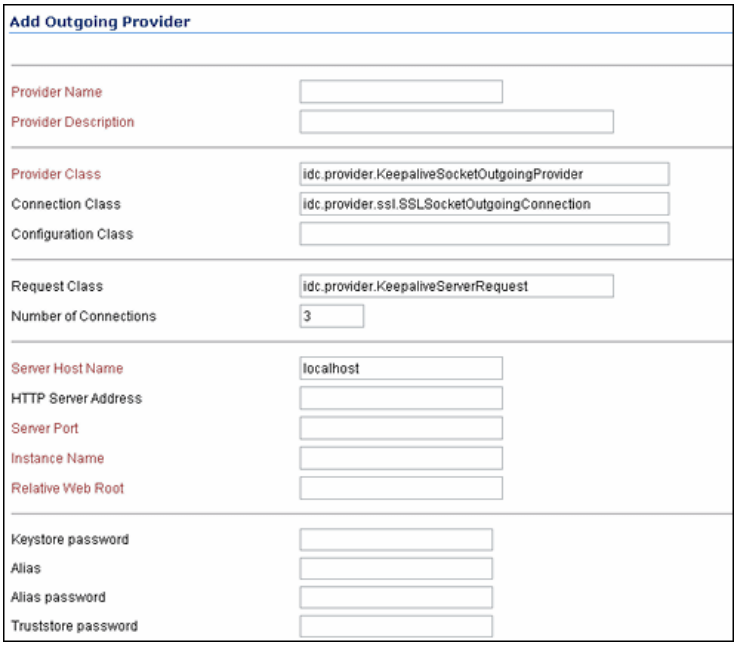
機能	説明
「Server Thread」フィールド	受信接続のサーバー・スレッドの名前。idc.provider.KeepaliveIdcServerThread など。
「Server Port」フィールド *	プロバイダが受信接続をリスニングするポート。たとえば、受信システム・プロバイダでは、デフォルトで、ポート 4444 をリスニングします。
「Request Client Authentication」チェック・ボックス	プロバイダで、受信接続からクライアント認証をリクエストする場合、このオプションを有効にします。
「Require Client Authentication」チェック・ボックス	プロバイダで、受信接続からのクライアント認証を必須とする場合、このオプションを有効にします。
キーストア / 別名 / トラストストアの情報	必要に応じて、キーストアのパスワード名、別名、別名のパスワード、トラストストアのパスワードを入力します。
「Add」ボタン	プロバイダ情報を保存します。
「Reset」ボタン	最後に保存した値に、プロバイダ情報をリセットします。

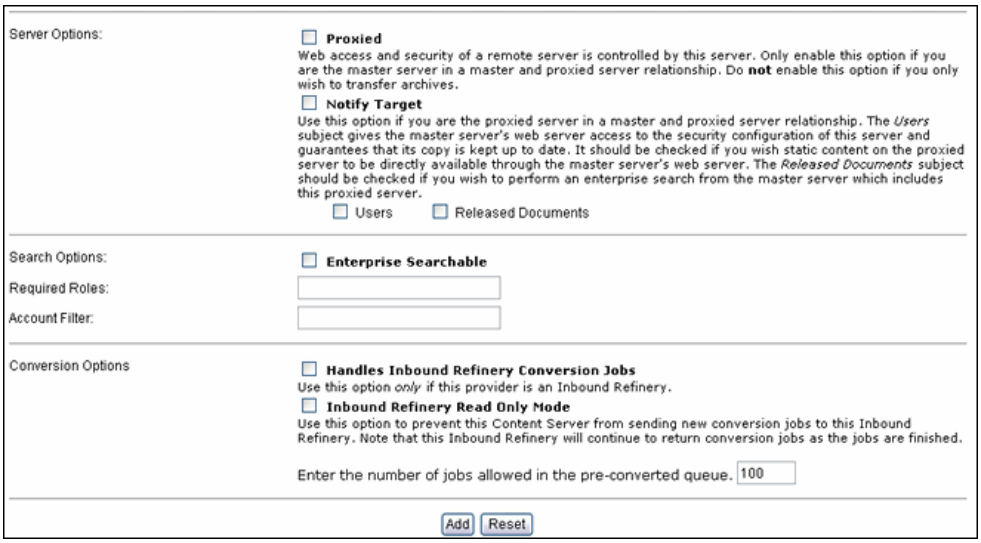
* 必須メタデータ・フィールド。

ssloutgoing プロバイダの追加ページ

管理者は、ssloutgoing 機能の「Add Outgoing Provider」ページを使用して、SSL ソケット送信プロバイダを作成できます。ページにアクセスするには、Content Server の「Administration」メニューから「Providers」リンクをクリックし、ssloutgoing プロバイダ・タイプの「Action」列から**「Add」**を選択します。

次の 2 つの画像に、このページを示します。





機能	説明
「Provider Name」フィールド *	プロバイダの名前。
「Provider Description」フィールド *	プロバイダの説明。
「Provider Class」フィールド *	プロバイダの Java クラスの名前。idc.provider.KeepaliveSocketOutgoingProvider など。
「Connection Class」フィールド	プロバイダ接続を実装する Java クラスの名前。idc.provider.KeepaliveSocketOutgoingConnection など。
「Configuration Class」フィールド	追加構成を行う Java クラスの名前。
「Request Class」フィールド	サーバー・リクエストを実装する Java クラスの名前。idc.provider.KeepaliveServerRequest など。
「Number of Connections」フィールド	最大接続数。3 など。
「Server Host Name」フィールド *	他のコンテンツ・サーバー・インスタンスのサーバー・ホスト名。localhost など。
「HTTP Server Address」フィールド	他のコンテンツ・サーバー・インスタンスの HTTP アドレス。
「Server Port」フィールド *	プロバイダが他のコンテンツ・サーバーと通信するポート。
「Instance Name」フィールド *	他のコンテンツ・サーバー・インスタンスのインスタンス名。
「Relative Web Root」フィールド *	他のコンテンツ・サーバー・インスタンスの相対 Web ルート。
キーストア / 別名 / トラストストアの情報	必要に応じて、キーストアのパスワード名、別名、別名のパスワード、トラストストアのパスワードを入力します。

機能	説明
「Proxied」チェック・ボックス	プロバイダが、現在のインスタンスによって制御されるコンテンツ・サーバーと接続する場合、このオプションを有効にします。
「Notify Target」チェック・ボックス	プロバイダが、制御インスタンスとして機能するコンテンツ・サーバーと接続し、ユーザー情報やコンテンツ・アイテム情報の変更時に、このコンテンツ・サーバーから制御インスタンスへの通知を行う場合、このオプションを有効にします。
「Users」チェック・ボックス	ユーザー情報の変更時に、このコンテンツ・サーバーから制御インスタンスへの通知を行う場合、このオプションを有効にします。
「Released Documents」チェック・ボックス	コンテンツ・アイテム情報の変更時に、このコンテンツ・サーバーから制御インスタンスへの通知を行う場合、このオプションを有効にします。
「Enterprise Searchable」チェック・ボックス	Enterprise Search が有効で、このコンテンツ・サーバー・インスタンスを検索可能にする場合、このオプションを有効にします。詳細は、『Enterprise Search Administration and User Guide』を参照してください。
「Required Roles」フィールド	Enterprise Search を使用してこのコンテンツ・サーバー・インスタンスを検索する権限を持つロールを入力します。ロールが入力されない場合、すべてのユーザーが権限を持ちます。
「Account Filter」フィールド	Enterprise Search を使用してこのコンテンツ・サーバー・インスタンスを検索する権限を持つアカウントを入力します。アカウントが入力されない場合、すべてのユーザーが権限を持ちます。
「Conversion」オプション	プロバイダが Inbound Refinery かどうかの指定に使用します。
「Add」ボタン	プロバイダ情報を保存します。
「Reset」ボタン	最後に保存した値に、プロバイダ情報をリセットします。

* 必須フィールド

索引