

Sitecore CMS 6

CMS のロー レベルのセキュリティ とカスタム プロバイダー

ロー レベルのセキュリティとカスタム セキュリティ プロバイダーの概念の紹介

目次

Chapter 1	イントロダクション	3
Chapter 2	セキュリティのデフォルト設定	4
2.1	メンバーシップ サービス	5
2.2	RoleManager サービス	7
2.3	プロファイル サービス	8
Chapter 3	複数プロバイダーの設定	10
3.1	スイッチング プロバイダー	11
3.1.1	スイッチング プロバイダーの設定	11
Chapter 4	カスタム プロバイダーの設定	14
4.1	イントロダクション	15
4.2	プロファイル プロバイダーの実装に関する推奨事項	17
4.3	独立設定	18

Chapter 1

イントロダクション

Sitecore CMS 6 のセキュリティ モデルは ASP.NET のセキュリティ プロバイダー モデルをベースにしており、ASP.NET の長所をすべて継承しています。このマニュアルでは、セキュリティ プロバイダーのデフォルト設定、およびカスタム プロバイダーを作成して Sitecore CMS のセキュリティに統合する方法について説明します。

このマニュアルは、ASP.NET 2.0 のプロバイダー モデルの基本概念を理解している読者を対象にしています。

このマニュアルには次の章があります。

- **セキュリティのデフォルト設定**
- **複数プロバイダーの設定**
- **カスタム プロバイダーの設定**

Chapter 2

セキュリティのデフォルト設定

Sitecore CMS 6 のデフォルト インストールでは、メンバーシップ、ロール、プロファイルの各サービスについて、事前作成済みの Microsoft SQL のプロバイダー セットを使用するように設定されます。

また、Sitecore CMS 6 のデフォルト インストールでは、SQLite バージョンのセキュリティおよびデータベース エンジンが含まれることもあります。次の説明は、SQLite にも同様に適用されます。

この章には次のセクションがあります。

- メンバーシップ サービス
- RoleManager サービス
- プロファイル サービス

2.1 メンバーシップ サービス

メンバーシップ サービスは、ユーザーの作成/更新/削除、ユーザー資格情報の検証、パターン別のユーザー検索、ユーザー パスワードのリセットなどを実行します。これらすべての動作は、基本クラス `MembershipProvider` に定義され、すべてのメンバーシップ プロバイダーがこの基本クラスを継承する必要があります。

`web.config` ファイルを開いて、`system.web/membership` エLEMENTに移動します。Sitecore CMS のクリーン インストールでは、このセクションは次のようになります。

```
<membership defaultProvider="sitecore">
  <providers>
    <clear />
    <add name="sitecore" type="Sitecore.Security.SitecoreMembershipProvider,
Sitecore.Kernel" realProviderName="sql" providerWildcard="%" raiseEvents="true" />
    <add name="sql" type="System.Web.Security.SqlMembershipProvider"
      connectionStringName="core" applicationName="sitecore"
      minRequiredPasswordLength="1" minRequiredNonalphanumericCharacters="0"
      requiresQuestionAndAnswer="false" requiresUniqueEmail="false" />
    <add name="switcher" type="Sitecore.Security.SwitchingMembershipProvider"
      applicationName="sitecore" mappings="switchingProviders/membership" />
  </providers>
</membership>
```

各プロバイダー定義は、`<membership>` ELEMENTの `<providers>` グループ内に、`<add/>` ELEMENTとして追加する必要があります。冒頭の `<clear/>` ELEMENTは、上位のコンフィギュレーション ファイル (例: `machine.config` ファイル) に定義されている古いプロバイダー定義を消去するように ASP.NET エンジンに指示します。

上のコードに示すように、このELEMENTにはプロバイダーが 3 つ定義されています。`sitecore` という名前のプロバイダーは CMS システム プロバイダーで、セキュリティ キャッシュを正しく処理するために使用されます。`switcher` という名前のプロバイダーは複数のプロバイダーの処理に使用されます。このプロバイダーについては後述します。はじめに `sql` プロバイダーについて説明します。

次の表に、デフォルトで定義される `sql` プロバイダーの属性を示します。

属性	説明
<code>name</code>	プロバイダーの名前。定義するプロバイダー セット全体で、一意である必要があります。Microsoft プロバイダーでは、このELEMENTのデフォルト値は <code>sql</code> です。
<code>type</code>	省略しないクラスの名前で、プロバイダー コントラクトの実装を含みます。Sitecore CMS がデフォルトで使用する Microsoft SQL プロバイダーは、クラス <code>System.Web.Security.SqlMembershipProvider</code> により実装されます。
<code>connectionStringName</code>	セキュリティ情報が保存されるデータベースの接続文字列名。Microsoft SQL のプロバイダー セットは、セキュリティの保存に使用する多数のテーブルをデータベース内に定義します。これらのテーブルは、デフォルトでは Sitecore CMS の <code>core</code> データベースに格納されます。

属性	説明
applicationName	プロバイダーのデータ認識を示す領域。Sitecore のセキュリティ ツールがカスタム セキュリティ データ (ユーザー/ロール) を認識できるようにするには、この属性に値 <code>sitecore</code> (デフォルト) を指定する必要があります。「独立設定」のトピックについては後述します。
minRequiredPasswordLength	各ユーザー パスワードに必要な最小文字数。SQL プロバイダーのデフォルト値は 1 です。このため、デフォルトの Admin アカウントのパスワードは、空白ではなく "b" です。
minRequiredNonalphanumericCharacters	ユーザー パスワードに必要な非英数字の最小文字数。非英数字とは、英数字以外の記号です。この属性のデフォルト値は "0" です。
requiresQuestionAndAnswer	この属性は、"ユーザー作成" 機能で質問と答えを表示するかどうかを定義します。デフォルト値は <code>false</code> です。
requiresUniqueEmail	この属性は、各ユーザーに一意的電子メール アドレスが必要かどうかを定義します。デフォルト値は <code>false</code> です。

[ASP.NET のプロバイダー モデルの記事 \(英語版\)](#) には、上記の属性を含むメンバーシップ プロバイダーの属性の詳細情報があります。

ここで注意が必要な項目は、`<membership>` エレメントの `defaultProvider` 属性です。このプロバイダーは、ASP.NET のログイン コントロールで使用されます。多数の静的クラスがあり (例: `System.Web.Security.Membership`)、これらの静的クラスも `defaultProvider` 属性に指定されるプロバイダーに依存します。

Sitecore CMS 6 の場合、この属性には値 `sitecore` を指定する必要があります。セキュリティ キャッシュを正しく処理するには、この指定が必須です。ただし、サービスをカスタマイズするコードを作成するときには、`sitecore` プロバイダーの `realProviderName` 属性に指定した値が、サービスの `defaultProvider` 属性の値と見なされると考える必要があります。

`realProviderName` 属性のデフォルト値は、クリーン インストールで定義される唯一のプロバイダーである `sql` です。この属性の要件として、その名前を持つプロバイダーが `<providers>` セクションに存在する必要があります。

Sitecore の上位セキュリティ API も、デフォルトのプロバイダーを介して動作します。

たとえば、`shell` サイトのログイン ページには、デフォルトのメンバーシップ プロバイダーの `ValidateUser` メソッド (入力したユーザー資格情報を確認) を呼び出す、標準の ASP.NET の Login コントロールが含まれます。

別の例として、特定のユーザーを編集する目的でリボンの該当ボタンをクリックすると、システムが一連の動作を実行し、その一部としてユーザーが属するロールをすべて取得します。このロールを取得する動作を実行するために、Sitecore API は、`System.Web.Security` 名前空間に定義された `Roles` クラスを指定して、メソッド `GetRolesForUser` を呼び出します。この呼び出しは、`RoleManager` のデフォルト プロバイダーも経由します。

2.2 RoleManager サービス

ロール プロバイダーは、ロールの作成と削除、ロールへのユーザーの追加や削除、特定ロールのユーザーを取得するリクエストの処理、および特定ユーザーのロールの取得などを実行します。メンバーシップ サービスと同様に、これらすべての動作は単一の基本クラス `RoleProvider` のメソッドとして定義され、すべてのロール プロバイダーがこの基本クラスを継承する必要があります。

RoleManager サービスの定義エレメントは、`web.config` ファイルのメンバーシップ エレメントの直後にあります。`web.config` ファイルを開いて、`<system.web>` セクションの `<roleManager>` エレメントを表示します。このエレメントのデフォルト定義は次のようになります。

```
<roleManager defaultProvider="sitecore" enabled="true">
  <providers>
    <clear />
    <add name="sitecore" type="Sitecore.Security.SitecoreRoleProvider, Sitecore.Kernel"
realProviderName="sql" raiseEvents="true" />
    <add name="sql" type="System.Web.Security.SqlRoleProvider" connectionStringName="core"
applicationName="sitecore" />
    <add name="switcher" type="Sitecore.Security.SwitchingRoleProvider, Sitecore.Kernel"
applicationName="sitecore" mappings="switchingProviders/roleManager" />
  </providers>
</roleManager>
```

このセクションに含まれるすべての属性の意味は、メンバーシップ サービスのセクションで前述したものと同じです。ただし、`<roleManager>` エレメントの `enabled` 属性のみ異なります。この属性は単に、サービスが有効であるかどうかを示します。デフォルト値は `true` です。この属性を `false` に設定する必要があるのは、Sitecore CMS のロール管理に関する機能をすべて無効にする場合のみです。

このサービスは、ロール管理に関する動作が必要とされるときにはいつでも呼び出されます。その例として、RoleManager のセキュリティ ツールでロールを作成するときや、ユーザーのメンバーシップを編集するときなどがあります。

2.3 プロファイル サービス

セキュリティ システム内の各ユーザーは、環境設定、設定、およびステータスを格納する一連のプロパティを持っています。これらのプロパティにより、ユーザー プロファイルが構成されます。ASP.NET 2.0 の機能では、ユーザーとそのプロファイルを個別に保存できます。

プロファイル サービスは、web.config ファイルの `<system.web>` セクションに定義されます。デフォルトのプロファイル定義は、次のようになります。

```
<profile defaultProvider="sql" enabled="true" inherits="Sitecore.Security.UserProfile,
Sitecore.Kernel">
  <providers>
    <clear />
    <add name="sql" type="System.Web.Profile.SqlProfileProvider" connectionStringName="core"
applicationName="sitecore" />
    <add name="switcher" type="Sitecore.Security.SwitchingProfileProvider"
applicationName="sitecore" mappings="switchingProviders/profile" />
  </providers>
  <properties>
    <clear />
    <add type="System.String" name="SC_UserData" />
  </properties>
</profile>
```

前述の 2 つのサービスとは異なり、プロファイル定義には `<providers>` と `<properties>` の 2 つのグループがあります。`<providers>` グループには、メンバーシップやロールのセクションと同様に、プロファイル プロバイダーの定義があります。`<properties>` セクションには、ユーザー プロファイルに保持する必要があるその他のプロパティがあります。

各ユーザーのプロファイル オブジェクトは、profile セクションに指定された情報に基づいて、実行時に自動生成されます。inherits 属性は、ユーザー プロファイルの基本クラスを定義します。したがって、各プロファイル オブジェクトは、そのクラスに実装されたプロパティ (コア プロパティ) のセットを含みます。さらに、このコア セットにカスタム プロパティを必要な数だけ追加できます。このようなカスタム プロパティは、`<properties>` セクションに定義する必要があります。次の表に、カスタム プロパティの属性をいくつか示します。

属性	説明
name	プロパティの名前。プロパティ セット全体 (コアとカスタム) で一意である必要があります。一意でない場合、ハンドルされない例外がスローされます。
type	このプロパティがコードから取得する .NET の型。たとえば、値として System.String が入力された場合、プロパティは ASP.NET エンジンの string (文字列) 型になります。
provider	プロパティが関連するプロファイル プロバイダーの名前。この属性を指定した場合、他のプロファイル プロバイダーはこのプロパティをまったく認識しません。この属性を指定しない場合、デフォルトのプロファイル プロバイダー名が使用されます。この属性は、後述する独立設定の手法で重要な役割を果たします。

属性	説明
customProviderData	この属性の値が、このプロパティを処理するプロファイル プロバイダーに渡されます。この属性のフォーマットは厳密には定義されていないので、この値を解析して必要な情報を抽出するのはデベロッパーの責任です。使用方法の 1 つとして、.NET の実装とは異なるカスタム プロパティのネイティブ型の定義があります。
readOnly	これはオプションの属性で、ユーザーがプロパティを変更できるかどうかを定義します。デフォルト値は <code>false</code> で、ユーザーはこのプロパティの読み取りと書き込みができます。

`readOnly` 属性を使用して、プロファイル設定を安全に保持できます。たとえば、セキュリティ ドメインの各ユーザーに対してプロパティを表示するが、ユーザーによるそのプロパティの変更を拒否する場合があります。これを実現するには、そのようなプロパティについて `readOnly` 属性の値を `true` に設定します。作成するコードでこれらの状況进行处理する必要はありません。設定レベルですべてが解決されます。

Chapter 3

複数プロバイダーの設定

デフォルトのセキュリティ設定には、ソリューションのインフラストラクチャを構築するために必要な基本オプションがすべてあります。ただし、カスタム セキュリティ ストレージの接続が必要な場合が多数あります。会社で既に総合セキュリティ モデル (Active Directory ドメイン、カスタム データベースやフリー スキーマ、xml ファイルなど) を導入していて、このストレージ、および対応するコードを再使用する場合がその例です。

このセクションでは、これらのシナリオをサポートする Sitecore CMS のセキュリティ モデルのオプションについて説明します。

この章には次のセクションがあります。

- スイッチング プロバイダー

3.1 スイッチング プロバイダー

複数プロバイダーの設定における主要な概念は、「スイッチング プロバイダー」という仕組みです。ここでは、この仕組みの機能について説明します。

- メンバーシップ、ロール、およびプロファイルのそれぞれについて、スイッチング プロバイダーが 1 つ定義されます。
- 各スイッチング プロバイダーは、各サービスについて適切に同じ基本クラスを継承する共通のプロバイダーです。
- 個々のスイッチング プロバイダーの主な目的は、上位アプリケーションからのセキュリティ リクエストをすべて取得し、リクエストを処理するプロバイダーを定義して、リクエストをそのプロバイダーに転送することです。

いくつかの例を使用して、この状況を説明します。

すべてのロール (Sitecore CMS のデフォルト ロールとカスタム ロール) を、サイト シェル内にある RoleManager セキュリティ ツールから認識とアクセスを可能にする場合があります。これを実現するには、スイッチング プロバイダーがリクエストを受信し、その GetAllRoles () メソッドが、使用できるプロバイダーの繰り返し適用と、受信したロールの集合の連結を行う必要があります。

また、ユーザー資格情報を検証するときに、ユーザー名とパスワードが正しいメンバーシップ プロバイダーを使用して検証されているかどうかを確認する場合があります。これを実現するには、スイッチング プロバイダーは、リクエストを受信し、リクエストを処理するプロバイダーを解決して、そのプロバイダーにリクエストを転送する必要があります。

最後に、ユーザーのプロファイルを編集するときに、デフォルトの SQL データベースに保存されている基本プロパティが確実に元の場所に保存され、Active Directory ドメインに保存されているカスタム プロパティが適宜、確実に更新されるようにする場合があります。スイッチング プロバイダーは最初のリクエストを受信し、このリクエストとともに渡されるプロパティを繰り返し適用して、プロパティを取得元のプロバイダーに転送します。

3.1.1 スイッチング プロバイダーの設定

スイッチング プロバイダーの目的を明らかにしたところで、この機能を有効にする方法を説明します。

メモ

スイッチング プロバイダー機能は、デフォルトではオフになっています。このセクションでは、この機能をオンにする方法を説明します。

web.config 内にある各サービスのセクションには、switcher という名前のプロバイダーの定義が含まれます。複数プロバイダーの環境を有効にするには、まずスイッチング プロバイダーをオンにする必要があります。このためには、各サービス (membership および roleManager) に含まれる sitecore プロバイダーの realProviderName 属性を switcher に設定します。

スイッチング プロバイダーの設定には、該当する種類のその他のプロバイダーと同じ属性セットが含まれます。唯一追加される属性は、mappings です。この属性には、Sitecore CMS のドメインと ASP.NET のプロバイダーのマッピングが定義されている、web.config 内のセクションへの相対パスが格納されます。この属性のデフォルト値は、メンバーシップ サービスの場合は

switchingProviders/membership、ロール マネージャー サービスの場合は switchingProviders/roleManager、プロフィール サービスの場合は switchingProviders/profile です。このパスは、sitecore 親セクションから記述する必要があります。

マッピング定義を表示して、内部でどのように動作しているのかを調べます。web.config ファイルを開いて、sitecore セクションの switchingProviders エlementを表示します。次のように表示されます。

```
<!-- SWITCHING PROVIDERS -->
<switchingProviders>
  <membership>
    <provider providerName="sql" storeFullNames="true" wildcard="%" domains="*" />
  </membership>
  <roleManager>
    <provider providerName="sql" storeFullNames="true" wildcard="%" domains="*" />
  </roleManager>
  <profile>
    <provider providerName="sql" storeFullNames="true" wildcard="%" domains="*" />
  </profile>
</switchingProviders>
```

このElementには、メンバーシップ、roleManager、プロフィールのサービスごとに 1 つ、計 3 つのマッピングのグループがあります。各グループには、次のフォーマットで <provider/> Elementがあります。

属性	説明
name	system.web セクションに定義されたプロバイダーの名前。システムがこの設定セクションを解析するときに、プロバイダー定義が存在するかどうかを確認されます。プロバイダー定義が存在しない場合、例外がスローされます。デフォルトではマッピングが 1 つのみ定義され、そのターゲットは使用できる唯一のプロバイダーである sql です。
storeFullNames	この属性は、プロバイダーが提示するストレージが、完全名 (ドメイン名前) と短い名前 (名前) のいずれを使用するかを定義します。完全名の場合、リクエストを適切なプロバイダーに渡す際のドメイン名の除去 (ローカライズ) と、結果を返す際のエンティティ名の前へのドメインの追加 (グローバル化) は、スイッチング プロバイダーの役割です。sql プロバイダーは完全名を保持し、この属性は true に設定されます。ただし、Sitecore CMS のドメインを認識しないカスタム ストレージ (Active Directory など) は短いエンティティ名のみを保持するので、属性を false に設定する必要があります。

属性	説明
wildcard	検索リクエストのパターンの一部を構成するワイルドカードとして使用される記号。プロバイダーごとに、この目的で使用される記号が異なる場合があります。たとえば、sql プロバイダーでは文字 "%" を使用しますが、ad プロバイダーはアスタリスク ("*") のみを受け取ります。このため、この設定は、プロバイダー レベルの個別の属性に移動されました。sql プロバイダーのマッピングのデフォルト値は "%" です。
domains	特定プロバイダーが処理するドメインのリストを、ここに定義する必要があります。たとえば、カスタムセキュリティ ストレージを接続して、このストレージ用のドメイン (例: ad) を Sitecore CMS に作成するとします。次のステップとして、domains 属性の値にこのドメインを指定する必要があります。使用するストレージが複数ドメインのユーザー/ロールを処理する場合、すべてのドメインをカンマで区切ってここに指定する必要があります。複数のプロバイダーに対して同一のドメインを指定できないことに注意してください。デフォルトでは、sql プロバイダーはエンティティをすべて処理するので、"*" が指定されます。つまり、Sitecore CMS のすべてのドメインからのリクエストが、sql プロバイダーに送信される必要があるということです。

まとめを示します。

- domains 属性の値が複数ドメインのリスト、またはアスタリスク ("*") である場合、storeFullNames 属性を true に設定する必要があります。
- 1 つのプロバイダーは複数のドメインを処理できますが、1 つのドメインを処理できるのは 1 つのプロバイダーのみです。このため、"N 個のドメインに対して 1 つのプロバイダー" の関係があり、N は 1 以上で上限はありません。
- 特定のプロバイダーがどのドメインを処理するかをスイッチング プロバイダーが認識できるように、domains 属性が常に存在する必要があります。

重要

Sitecore CMS のデフォルト インストールでは、ストレージごとに異なるユーザー プロファイルのプロパティ セットを持つオプションはサポートされません。このオプションを使用するには、Active Directory モジュールをインストールして、適切にモジュールを設定する必要があります (Active Directory モジュールのマニュアルを参照)。

Chapter 4

カスタム プロバイダーの設定

この章では、カスタム プロバイダーの設定方法について説明します。

この章には次のセクションがあります。

- インTRODクション
- プロファイル プロバイダーの実装に関する推奨事項
- 独立設定

4.1 イントロダクション

前述のセクションで説明したスイッチング プロバイダーの仕組みは、プロバイダーを 1 つ定義しただけでは意味がありません。この仕組みをフルに活用するために、各サービスにカスタム プロバイダーをプラグインして、どのように連携するかを調べます。

はじめに、カスタム プロバイダーの設定には 2 種類あります。最初の設定は、カスタム プロバイダーが独立のときに使用します。これは、カスタム ストレージが Sitecore CMS のセキュリティ全体 (Sitecore Client のロール、定義済みのユーザーなど) を保持し、自己完結していることを意味します。このようなカスタム ソリューションは通常、標準のユーザー プロファイル全体を置き換えるために開発されます。たとえば、Sitecore CMS のセキュリティ全体を、Oracle やその他の RDBMS に移動する場合などです。

このようなソリューションには、スイッチング プロバイダーの機能はまったく不要です。web.config にある標準の sql のプロバイダー定義を、カスタム プロバイダー定義に置換するだけで済みます。この設定は特別なものではないので、本書では説明しません。

既存のモデルを置き換えるのではなく、拡張する場合は、別の方法を使用する必要があります。次に示すステップで、この場合に実行する必要がある手順を説明します。

- メンバーシップ プロバイダーを開発します (さらに web.config の system.web セクションで設定します)。メンバーシップ プロバイダーは System.Web.Security 名前空間の MembershipProvider クラスを継承し、すべての仮想メソッド/プロパティを上書きする必要があります。このプロバイダーは Sitecore CMS を認識せず、また認識する必要もありません。このプロバイダーはカスタム ストレージに対して特定の方法で動作する必要があり、どこで使用されるかを考慮する必要はありません。
- ロール プロバイダーを開発します (さらに web.config の system.web セクションで設定します)。ロール プロバイダーは System.Web.Security 名前空間の RoleProvider クラスを継承し、すべての仮想メソッド/プロパティを上書きする必要があります。ここでも同様に、Sitecore CMS を考慮する必要はありません。
- (オプション) プロファイル プロバイダーを開発します (さらに web.config の system.web セクションで設定します)。プロファイル プロバイダーは ProfileProvider クラスを継承し、実装と設定の特殊なルールに従う必要があります (後述)。Active Directory モジュールがインストールされている場合にのみ、部分プロファイルの機能を使用できることに注意してください。
- カスタム ユーザー/プロファイルに関連する Sitecore CMS のドメインを定義します。使用するカスタム プロバイダーのセットが短い名前のみを保持する (storeFullNames="false") 場合は、Sitecore のドメインを作成してそのドメイン名を選択する必要があります。
- ユーザーの完全名が保持される場合、ドメイン名は、外部ストレージ内にあるユーザーの完全名のプレフィックスに対応する必要があります。異なるプレフィックスを持つユーザーやロールがある場合は、対応するドメインをすべて、Sitecore CMS に作成する必要があります。

- ドメインとプロバイダーのマッピングを編集します。メンバーシップ プロバイダーのマッピングを `<membership>` グループ、ロール プロバイダーのマッピングを `<roleManager>` グループに追加します。マッピングの順序は重要ではありません。該当する Sitecore CMS のセキュリティ アプリケーションで処理されるユーザー/ロールのデフォルトの順序にのみ影響します。
- (オプション) ドメインとプロバイダーのマッピングを編集します。プロファイル プロバイダーのマッピングを `<profile>` グループに追加します。この行は、すべてのリクエストを処理できるプロバイダー (*) より前に置く必要があります。後に置いた場合、カスタム プロパティが共通ストレージに保存され、カスタム ストレージとの関係が失われます。

4.2 プロファイル プロバイダーの実装に関する推奨事項

他のサービスと比較して、プロファイル プロバイダーの実装には独特な点がいくつかあります。部分プロファイルをサポートするプロファイル プロバイダーを開発するには、次に示す推奨事項に従ってください。

- 設定

このプロバイダーが扱う各プロパティの `customProviderData` 属性には、このプロバイダーが理解可能な値を指定する必要があります。たとえば、Active Directory のプロパティの場合は、属性値を `ad|` で始める必要があります。

プロバイダーのマッピングは、定義されたプロパティをすべて処理できる全般プロファイル プロバイダーより前に置く必要があります。

- 実装

すべてのプロバイダーが、プロパティ全体の集合を受け取ります。各プロファイル プロバイダーは、リクエストを処理した後に、対応するプロパティ（例： `ad|` で始まるプロパティ）をフィルター処理して全体集合から削除する必要があります。これにより、他のプロバイダーは対応しないプロパティを受け取らなくなり、別のストレージに保存しません。

4.3 独立設定

Sitecore CMS ソリューションに外部アプリケーションをプラグインする場合があります。この外部アプリケーションはモジュールの一種で、独自の方法で ASP.NET のセキュリティ プロバイダーを既に使用しているとします。さらに、Sitecore CMS がこのモジュールをまったく認識しないようにしたいと考えます。このような場合は競合が発生する可能性があり、競合を解決するには外部コンポーネントを少し変更する必要があります。

重要

外部コンポーネントのソースにアクセスできない場合、ここで説明する独立設定のシナリオで、この外部コンポーネントを使用できないことがあります。

一般的な推奨事項をいくつか示します。

- 外部コンポーネントがそのプロバイダーのアドレスを直接ポイントするようにします。よく知られた静的クラス (Membership, Roles) 経由ではなく、必要なプロバイダーを明示的に呼び出します (Membership.Providers["providername"])
- プロバイダー定義の applicationName 属性を、Sitecore CMS のプロバイダー (一般的な場合は sitecore) にしないでください。
- この外部コンポーネントが定義するユーザー プロファイルのカスタム プロパティの provider 属性は、他の (Sitecore CMS の) プロバイダーが認識できないように、対応するプロバイダーに設定する必要があります。
- 最後に、この外部プロバイダーのマッピングを、Sitecore CMS のドメインとプロバイダーのマップに追加しないでください。

上記の作業が完了すると、Sitecore CMS ソリューションとこの外部コンポーネントが干渉しなくなります。