

ASP.NET 2.0 Provider Model 概要

Agenda

- ASP.NET 2.0 Provider Model とは
- カスタムプロバイダの実装
- まとめ

ASP.NET 2.0 Provider Model とは

- ASP.NET 2.0 のインフラストラクチャ
- データストアへのアクセスを
アプリケーションロジックから分離
→データストアの変更に柔軟に対応。
- Strategy パターン
→デザインパターンによる意識の共通化

プロバイダモデル

ASP.NET Application

ASP.NET Services

Providers

Data Stores

プロバイダモデル

ASP.NET Application

ASP.NET Services

Provider
A

Provider
B

Provider
C

Data
Store A

Data
Store B

Data
Store C



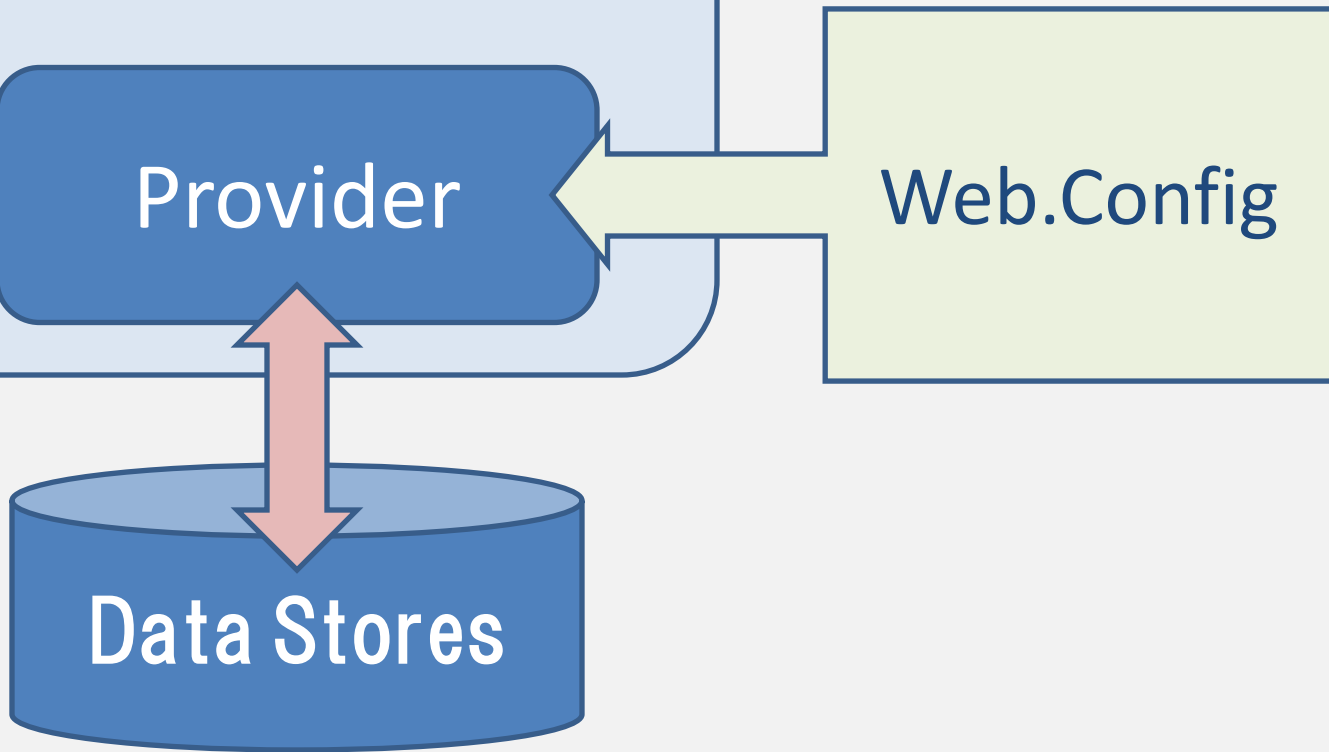
プロバイダモデル

ASP.NET Services

Provider

Web.Config

Data Stores



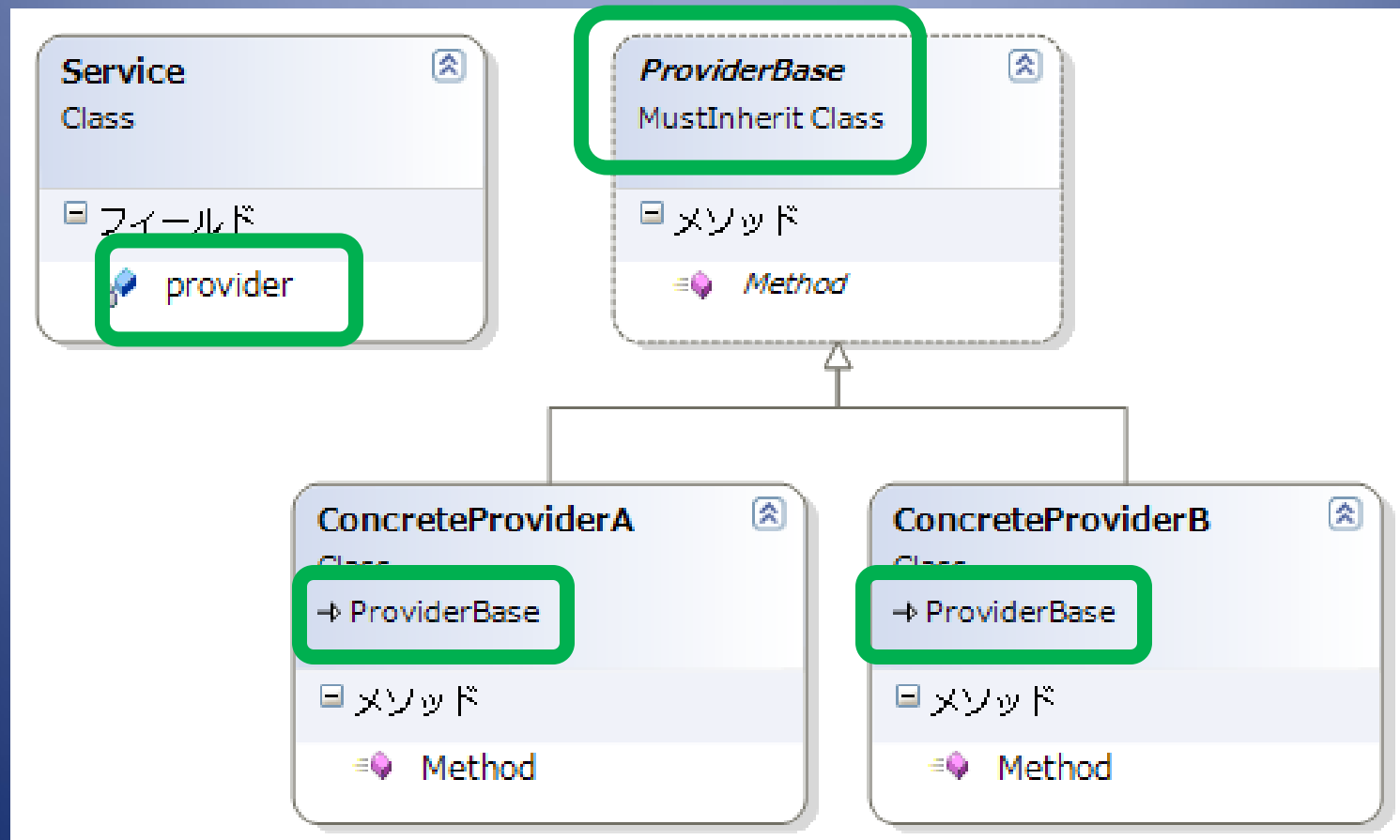
ASP.NET 2.0 Provider Model と Strategy

- Strategy パターンとは



ASP.NET 2.0 Provider Model と Strategy

- ASP.NET 2.0 Provider Model



ASP.NET 2.0 Provider Model と Strategy

- 本来の Strategy パターンは Interface で実装。
- ASP.NET 2.0 Provider Model は基本クラスによる実装を採用。
 - 理由はよくわからない。

機能と提供されるプロバイダ

- Membership
 - System.Web.Security.MembershipProvider
 - System.Web.Security.ActiveDirectoryMembershipProvider
 - System.Web.Security.SqlMembershipProvider
- Role management
 - System.Web.Security.RoleProvider
 - System.Web.Security.AuthorizationStoreRoleProvider
 - System.Web.Security.SqlRoleProvider
 - System.Web.Security.WindowsTokenRoleProvider
- Site map
 - System.Web.SiteMapProvider
 - System.Web.XmlSiteMapProvider

機能と提供されるプロバイダ

- Profile
 - System.Web.Profile.ProfileProvider
 - System.Web.Profile.SqlProfileProvider

- Session state
 - System.Web.SessionState
 - SessionStateStoreProviderBase
 - System.Web.SessionState.InProcSessionStateStore
 - System.Web.SessionState.OutOfProcSessionStateStore
 - System.Web.SessionState.SqlSessionStateStore

機能と提供されるプロバイダ

- Web events
 - System.Diagnostics.Eventing.EventProvider
 - System.Web.Management.EventLogWebEventProvider
 - System.Web.Management.SimpleMailWebEventProvider
 - System.Web.Management.TemplatedMailWebEventProvider
 - System.Web.Management.SqlWebEventProvider
 - System.Web.Management.TraceWebEventProvider
 - System.Web.Management.WmiWebEventProvider

カスタムプロバイダの実装

- ビルトイン プロバイダ以外にも独自のカスタムプロバイダを実装できる。



シンプルなフレームワーク

提供されるクラス

- System.Configuration.Provider 名前空間
 - ProviderBase クラス
 - すべての Provider クラスの基本クラス
 - ProviderCollection クラス
 - ProviderBaseを継承するクラスを格納するコレクションクラス
 - ProviderException クラス
 - Provider Model の実装時に使用する例外クラス

ProviderBase クラス

Namespace System.Configuration.Provider

Public MustInherit Class ProviderBase

Private _name As String

Private _description As String

Private _initialized As Boolean

Public Overridable ReadOnly Property Name() As String

Public Overridable ReadOnly Property Description() As String

Public Overridable Sub Initialize

(ByVal name As String,

ByVal config As System.Collections.Specialized.NameValueCollection)

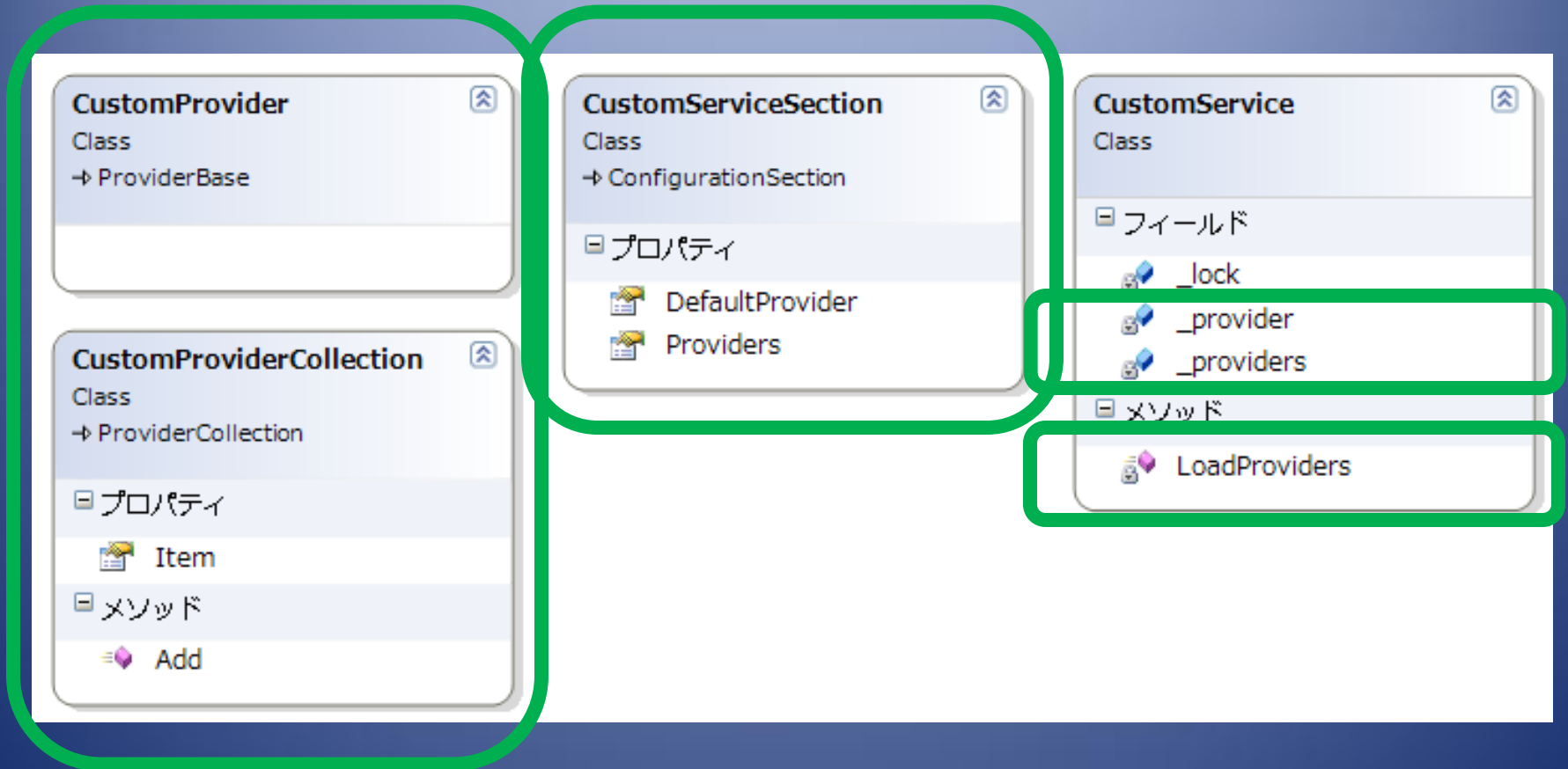
End Class

End Namespace

ProviderBase.Initialize メソッド

```
Public Overridable Sub Initialize(ByVal name As String, ByVal config As NameValueCollection)
    SyncLock Me
        If _initialized Then
            Throw New InvalidOperationException("...")
        End If
        _initialized = True
    End SyncLock
    If (name = Nothing) Then
        Throw New ArgumentNullException("name")
    End If
    If (name.Length = 0) Then
        Throw New ArgumentException("...", "name")
    End If
    _name = name
    If config IsNot Nothing Then
        _description = config("description")
        config.Remove("description")
    End If
End Sub
```

クラスレイアウト



Web.Config

```
<configSections>  
  <sectionGroup name="system.web">  
    <section name="CustomService"  
      type="CustomServiceSection, CustomSections"  
      allowDefinition="MachineToApplication"  
      restartOnExternalChanges="true" />  
  </sectionGroup>  
</configSections>
```

Web.Config

```
<CustomService defaultProvider="SqlCustomProvider">  
  <providers>  
    <add name="SqlCustomProvider"  
          type="SqlCustomProvider"  
          connectionStringName= "..."/>  
    <add name="XmlCustomProvider"  
          type="XmlCustomProvider"  
          directoryPath= "..."/>  
  </providers>  
</CustomService>
```

LoadProviders メソッド

```
Private Shared Sub LoadProviders()
```

```
    If _provider Is Nothing Then
```

```
        SyncLock _lock
```

```
            If _provider Is Nothing Then
```

```
                Dim section As CustomServiceSection = _
```

```
                    CType(WebConfigurationManager.GetSection _  
                        ("system.web/customService"), CustomServiceSection)
```

```
                _providers = New CustomProviderCollection()
```

```
                ProvidersHelper.InstantiateProviders _  
                    (section.Providers, _providers, GetType(CustomProvider))
```

```
                _provider = _providers(section.DefaultProvider)
```

```
            If _provider Is Nothing Then
```

```
                Throw New ProviderException _  
                    ("Unable to load default CustomProvider")
```

```
            End If
```

```
        End If
```

```
    End SyncLock
```

```
End If
```

```
End Sub
```

まとめ

- シンプルな設計。
 - ベースクラスを継承するだけ。
- データストアの変更に柔軟に対応。
 - データストアの数だけプロバイダを作っても。
- デザインパターンによる意識の共通化。
 - Strategy パターンというよりは
ASP.NET Provider Model という認識でOK。

参考文献

MSDN : Provider Model, Visual Basic (Visual Basic)

<http://msdn.microsoft.com/en-us/library/aa530801.aspx>

MSDN : ASP.NET 2.0 Provider Model : Introduction to the Provider Model (C#)

<http://msdn.microsoft.com/en-us/library/aa479030.aspx>

ASP.NET 2.0 Provider Model.pdf (C#)

<http://download.microsoft.com/download/2/a/e/2aeabd28-3171-4b95-9363-22150625a6a5/ASP.NET%20Provider%20Model.pdf>

MSDN : AProvider Tool Kit

<http://msdn.microsoft.com/en-us/asp.net/aa336558.aspx>