ASP.NET Infrastructure

Core Runtime Features

Overview

- Caching
- Internationalization and localization
- Diagnostics
- Membership & Security
- State Management

Output Caching

- **OutputCache action filter**
  - Use as attribute on action or controller
  - Specify Duration and VaryByParam
  - Configurable with cache profiles
- **Don’t use OuputCache directive on views!**

```
public class CachedController : Controller
{
    [OutputCache(Duration=60, VaryByParam="none")]
    public ActionResult Index()
    {
        return View();
    }

}
```

Output Cache Settings

- **VaryByParam**
  - Vary by “none” to always cache the same content
  - Vary by “*” to cache for every permutation of all parameters
  - Vary by “name” to cache for every value of name parameter
- **Location**
  - Cache on server, client, client and server
- **VaryByHeader**
  - Vary on an HTTP header, like Accept-Language
- **VaryByCustom**
  - Implement custom static method in global.asax
- **SqlDependency**
  - Cache until data in a SQL Server table changes

Cache Profiles

- Avoids repetition in cache attributes
- Easy to change during performance profiling

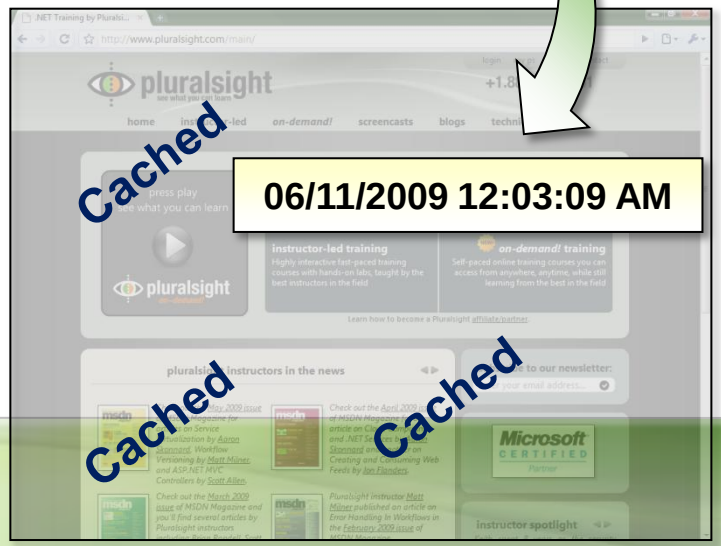
```
[OutputCache(CacheProfile="Aggresive", VaryByParam="firstName")]
public ActionResult Index(string firstName)
{
    // ...

    return View();
}
```

```
<caching>
  <outputCacheSettings>
    <outputCacheProfiles>
      <add name="Aggresive" duration="300" />
      <add name="Mild" duration="10"/>
    </outputCacheProfiles>
  </outputCacheSettings>
</caching>
```

Cache Substitution

- **Also known as “donut caching”**
  - When *almost* everything can be cached
  - Relies on the cache substitution API of ASP.NET
- **Wrap the WriteSubstitution call in a helper**

```
Response.WriteSubstitution(
    new HttpResponseSubstitutionCallback(
        ctx => DateTime.Now.ToLongTimeString()));
```



Data Caching

- **ASP.NET Cache API**
  - Cache object associated with HttpContext
  - Also accessed via HttpRuntime.Cache (better for unit tests)
  - Absolute and sliding expirations
  - Cache dependencies on file system and SQL Server data

```
List<Movie> topMovies = null;
if (HttpContext.Cache[TopMovieKey] == null)
{
        topMovies = (from m in ctx.Movies
                     orderby m.Reviews.Average(r => r.Rating) descending
                     select m).Take(10).ToList();
        HttpContext.Cache.Insert(TopMovieKey,
                                 topMovies,
                                 null,
                                 Cache.NoAbsoluteExpiration,
                                 TimeSpan.FromMinutes(10));
}
```

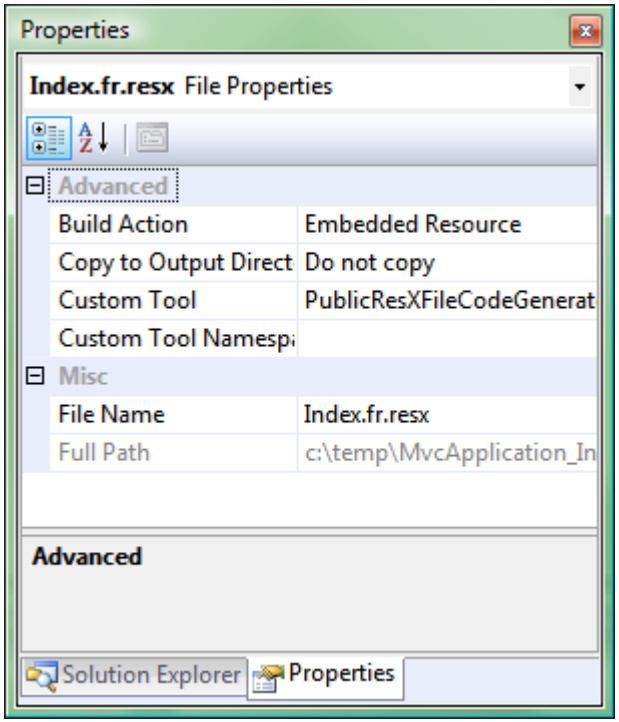
Localization & Culture

- **Thread.CurrentCulture property impacts formatting**
  - Example: DateTime.Now.ToString()
- **Thread.CurrentUICulture impacts resource loading**
- **ASP.NET can set cultures according to HTTP headers**
  - Accept-language header
  - Use <globalization> section in web.config

```
<system.web>
    <globalization culture="auto" uiCulture ="auto"/>
     …
<system.web />
```

Resources

- **Resx files can store localized text and binary assets.**
  - Strings.resx will store default resources
  - Strings.es.resx will store resource for Spanish culture
- **Resource manager will load appropriate file**
- **Visual Studio generates strongly-typed class**
  - Use “PublicResXFileCodeGenerator” as tool outside of App_GlobalResources
  - Set Build Action to “Embedded Resource”
  - Don’t use App_GlobalResouces from a controller if you want to unit test




```
<%= Html.Encode(Resources.Strings.Greeting) %>
```

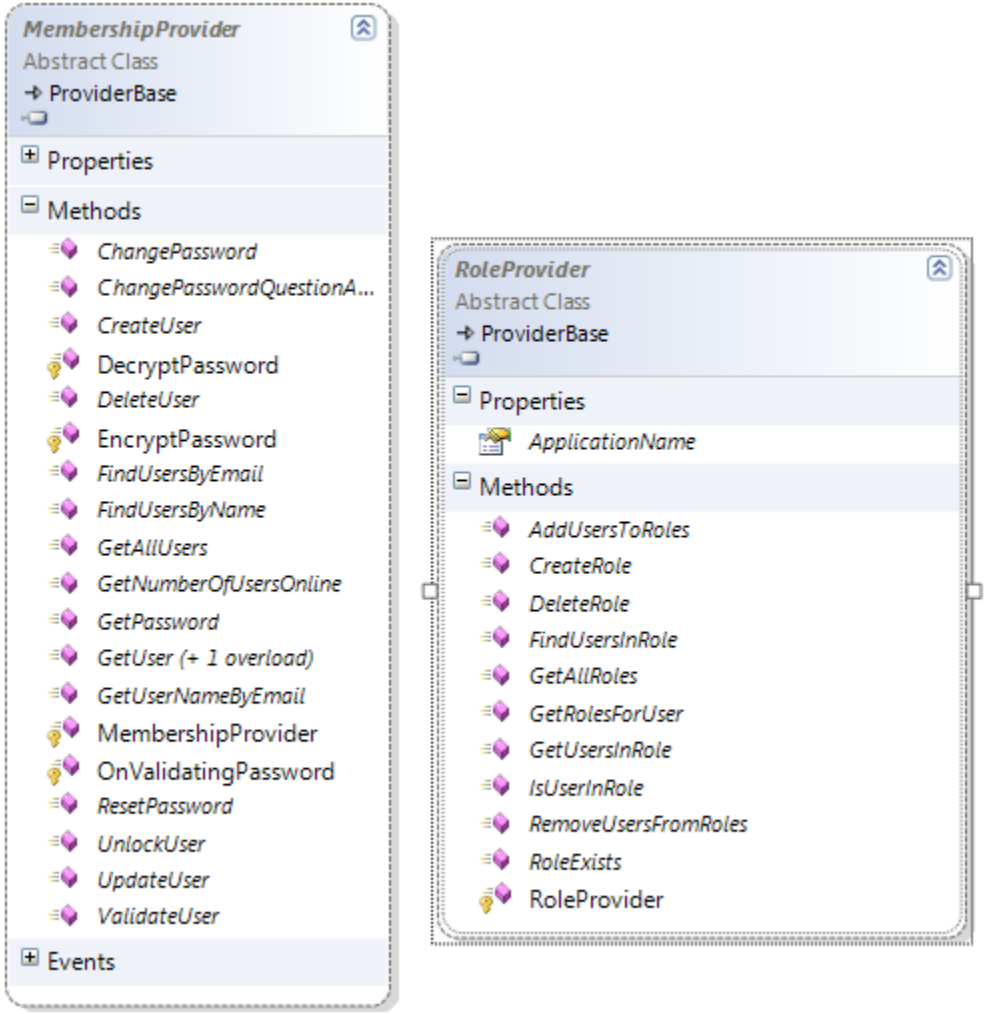
Diagnostics

- **Traditional ASP.NET trace feature is geared for Web Forms**
  - Traces page lifecycle events only
- **Logging options**
  - ASP.NET Health Monitoring
  - log4net (http://logging.apache.org/log4net/index.html)
  - elmah (http://code.google.com/p/elmah/)
  - P&P Application Logging Block

```
<healthMonitoring enabled="true">
  <rules>
    <add name="All Events" eventName="All Events“
         provider="SqlWebEventProvider"/>
  </rules>
</healthMonitoring>
```

Security

- **Membership provider**
  - Authenticates users
  - SQL provider for forms authentication
  - Active Directory provider for integrated authentication
- **Role provider**
  - Authorizes users
  - SQL provider stores roles in SQL Server
  - AD provider uses roles in Active Directory



State Management

- **ASP.NET State Management**
  - Session & Application
  - Cookies
  - Cache
- **ASP.NET MVC State Management**
  - Model State
  - TempData

Summary

- **Build on the ASP.NET Platform**
  - Caching
  - Diagnostics
  - Resource localization
  - Security
  - State Management