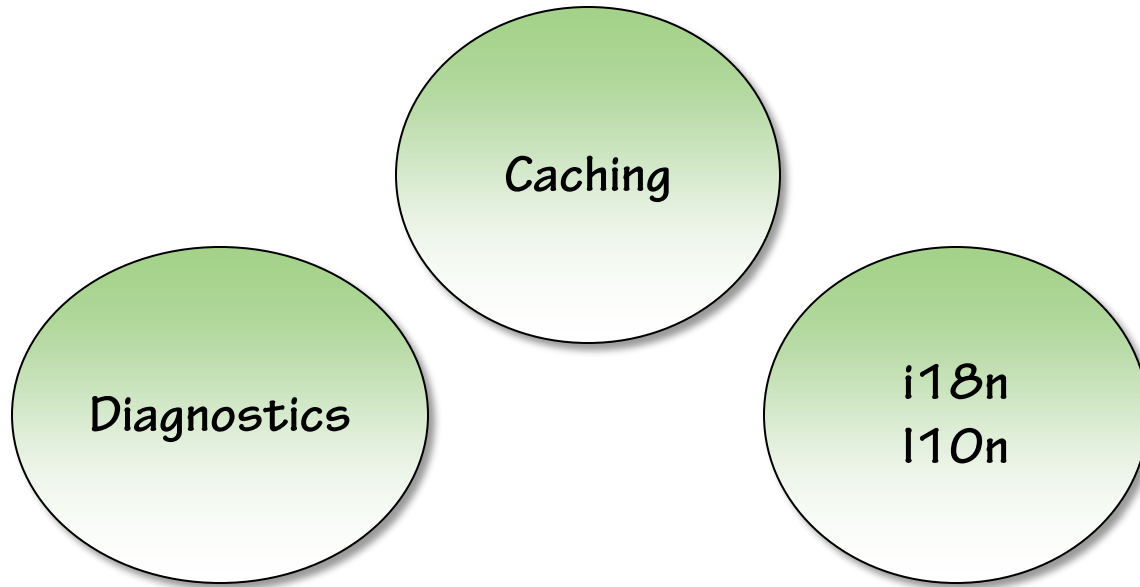


# ASP.NET MVC & ASP.NET Infrastructure

K. Scott Allen



# Overview



# Output Caching

- [OutputCache] action filter
- Can also use on child actions

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

# Output Cache Settings

- **VaryByParam**

- Vary by "\*" to cache for every permutation of all parameters
- Vary by "none" to always cache the same content
- 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 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="Aggressive", VaryByParam="firstName")]  
public ActionResult Index(string firstName)  
{  
    // ...  
  
    return View();  
}
```

```
<キャッシング>  
  <outputCacheSettings>  
    <outputCacheProfiles>  
      <add name="Aggressive" duration="300" />  
      <add name="Mild" duration="10"/>  
    </outputCacheProfiles>  
  </outputCacheSettings>  
</キャッシング>
```



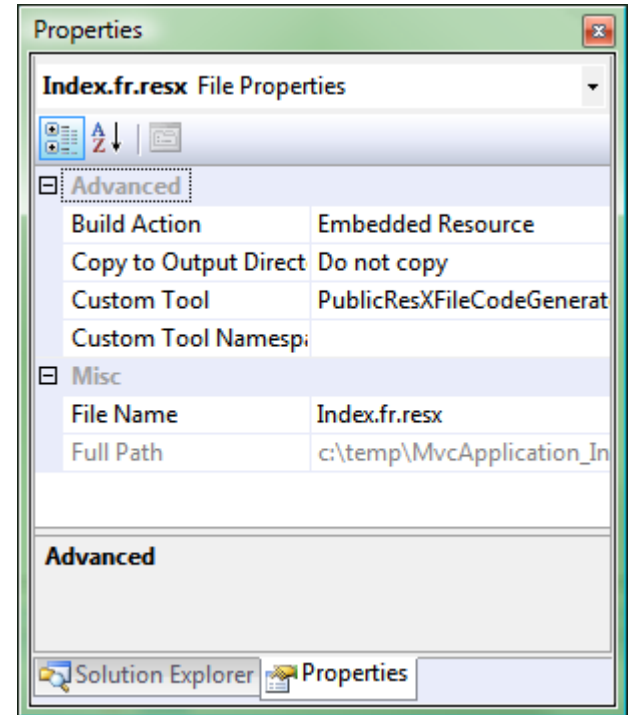
# Localization & Culture

- **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



`@Resources.Strings.Greeting`

```
[Required(ErrorMessageResourceName="BodyRequiredError",  
           ErrorMessageResourceType=typeof(Resources.Strings))]  
public virtual string Body { get; set; }
```



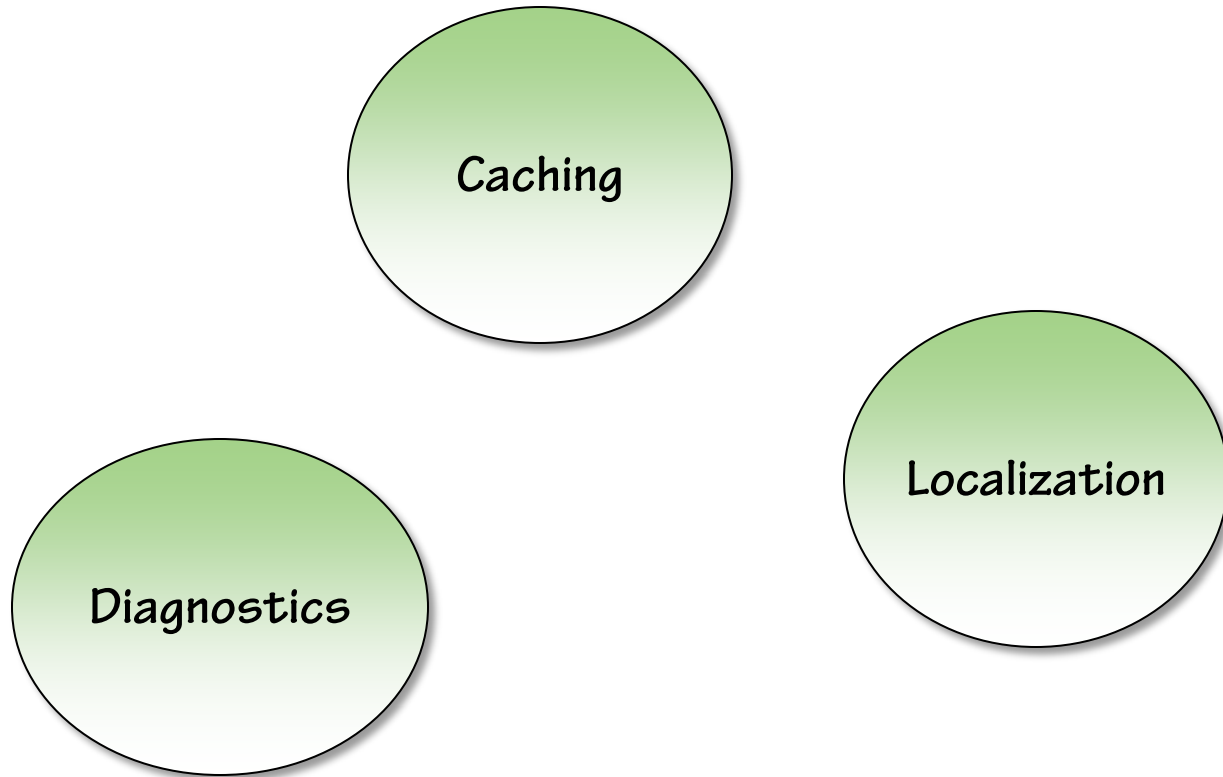
# Diagnostics

## ■ 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>
```

# Summary



## ASP.NET will now see IE indicate a preference for French in the Accept-Language HTTP header

Request headers	Request body	Response headers	Response body	Cookies	Initiator	Timings
Key	Value					
Request	GET / HTTP/1.1					
Accept	text/html, application/xhtml+xml, */*					
Accept-Language	fr-FR;fr;q=0.8,en-US;q=0.7,en;q=0.5,es-AR;q=0.3,es;q=0.2					
User-Agent	Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Trident/6.0)					
Accept-Encoding	gzip, deflate					
Host	localhost:56470					
DNT	1					
Connection	Keep-Alive					