

Best Practices in der Webpart-Entwicklung

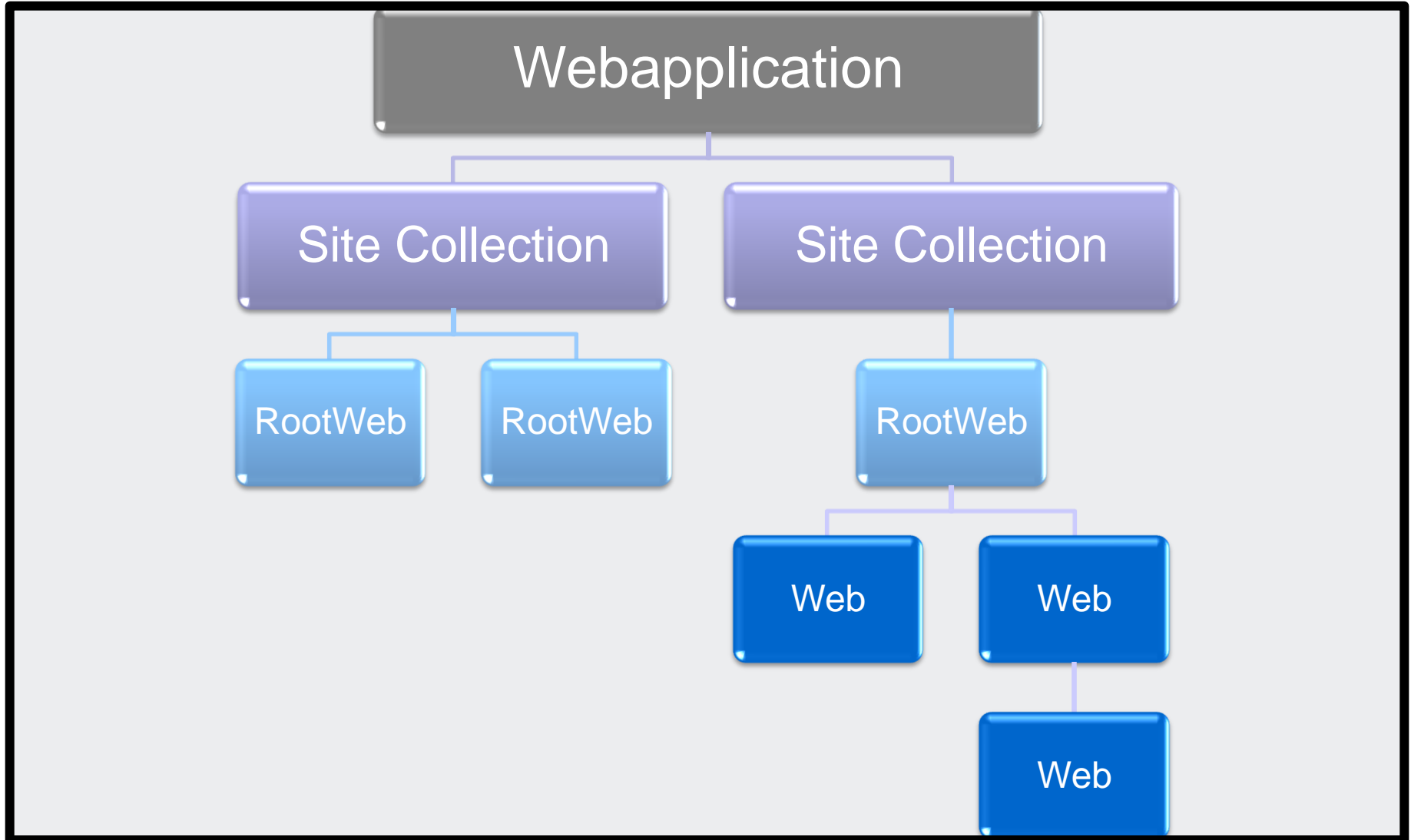
René Hézsér – Net at Work Netzwerkssysteme GmbH

Rene@Hezser.de

- Webpart Grundlagen
- Logging in das SharePoint Log
- Webpart Feature inkl. FeatureReceiver
- Extension Methods
- Webparts mit eigenen ToolParts
- SPContext – wo bin ich
- Code Ausführung mit erweiterten Berechtigungen
- Nutzung “.ToHtml()” Funktionen
- SharePoint Webcontrols
- Fallen umgehen und Probleme vermeiden
- Mehrsprachigkeit

- SPWebApplication – Web Application
- SPSite – Site Collection / Site
- SPWeb – Web
- GAC – Global Assembly Cache
C:\Windows\Microsoft.NET\Framework\vx.y.z
- Code Beispiel

Farm

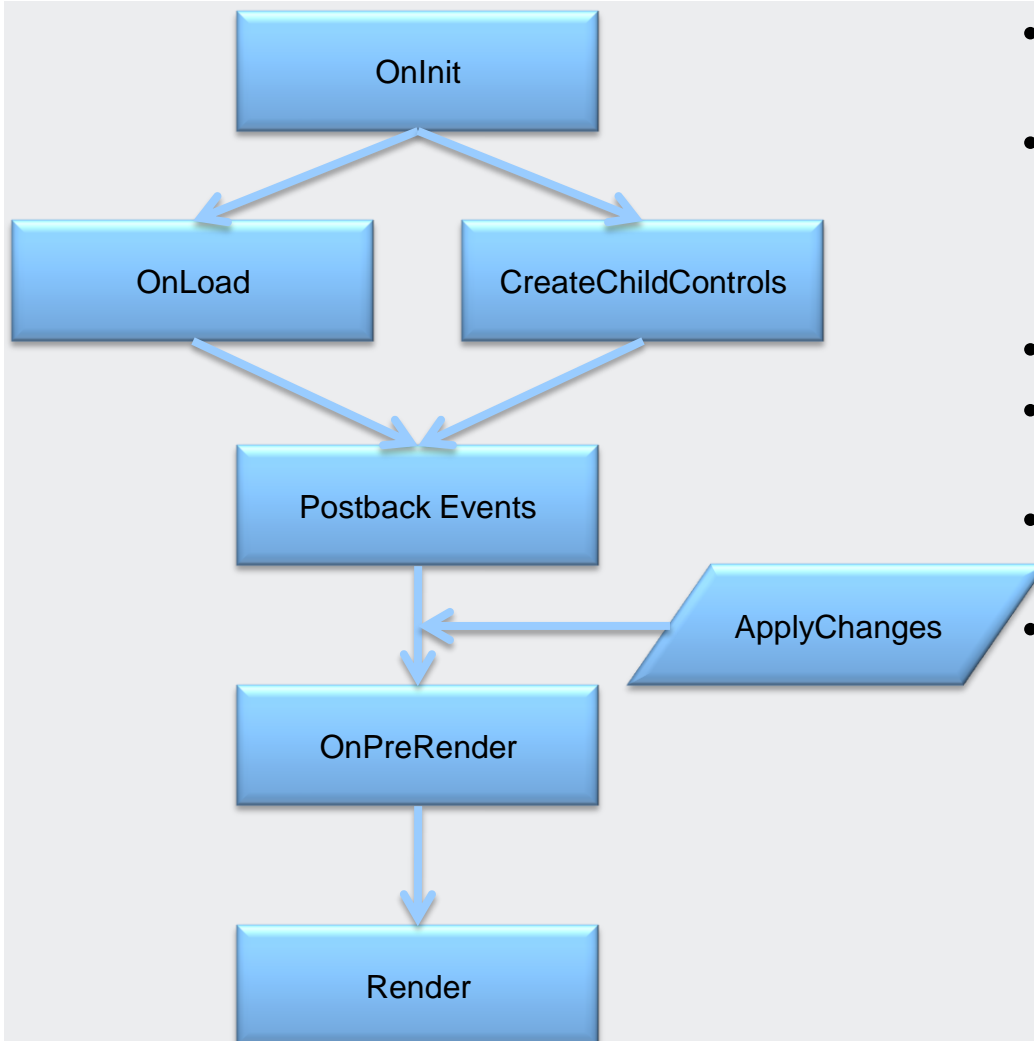


- Nach der Konferenz auf <http://www.hezser.de/blog>

```
1 using ...
4
5 namespace RH.SharePointKonferenz
6 {
7     public class ActivationWeb : SPFeatureReceiver
8     {
9         public override void FeatureActivated(SPFeatureReceiverProperties properties)...
10        public override void FeatureDeactivating(SPFeatureReceiverProperties properties)...
11        public override void FeatureInstalled(SPFeatureReceiverProperties properties)...
12        public override void FeatureUninstalling(SPFeatureReceiverProperties properties)...
13
14        private static void CreateCountryList(SPFeatureReceiverProperties properties)
15        {
16            try
17            {
18                var web = (SPWeb)properties.Feature.Parent;
19
20                // create a new List if it does not already exist
21                if (web.Lists.Cast<SPList>().Any(l => l.Title == "Countries") == false)
22                {
23                    // create a new list
24                    web.Lists.Add("Countries", "Countries", SPListTemplateType.GenericList);
25                }
26
27                // get the list to modify it
28                SPList list = web.Lists.Cast<SPList>().Where(l => l.Title == "Countries").First();
29
30                SPFieldCollection fields = list.Fields;
31                SPView view = list.DefaultView;
32                // change title field
33                var titleField = fields[SPBuiltInFieldId.Title];
34                titleField.Required = false;
35                titleField.Update();
36
37                // add fields
38                if (fields.FieldExists("Short") == false)
39                {
40                    // create the field, and add it to the defaultView
41                    string shortName = fields.Add("Short", SPFieldType.Text, false);
42                    view.ViewFields.Add(shortName);
43                }
44
45                view.Update();
46                list.OnQuickLaunch = true;
47                list.Update();
48            }
49            catch { }
50        }
51    }
52}
```

- Zusammenfassung von Server Controls
- SharePoint Webparts
 - Erweitern das ASP.Net Webpart Framework
 - Ermöglichen Webseitenübergreifende Verbindungen
 - Verbindungen zu Webparts außerhalb von Webpart Zonen
 - Caching u. A. in den Inhaltsdatenbanken
- Werden im Benutzer Kontext ausgeführt
- Viele Standardwebparts
- Personalisierbar

Lebenszyklus eines Webparts (ASP.Net Lifecycle)



- Erstes Laden → OnLoad und dann CreateChildControls
- Postback → CreateChildControls und dann OnLoad, sowie ApplyChanges (ToolParts)
- EnsureChildControls() möglich
- CreateChildControls: Erstellung von Controls mit Initialdaten
- OnPreRender: Datenbindung für Controls
- ASP.NET Page Life Cycle Overview → <http://msdn.microsoft.com/en-us/library/ms178472.aspx>

1. Erstellung

- `TextBox tb = new TextBox();`

2. Hinzufügen zu den Controls des Webparts

- `Controls.Add(tb);`

3. Zuweisen von Werten

- `tb.Text = „SharePoint Konferenz“;`

4. Reagieren auf Ereignisse

- `tb.TextChanged +=
new EventHandler(tb_TextChanged);`

5. Auslesen der Werte

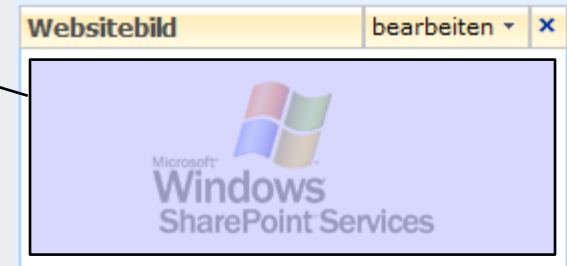
- `string wert = tb.Text;`

6. Rendern durch das Framework

- Visual Studio 2008 SP1 (Express)
- WSPBuilder - <http://codeplex.com/wspbuilder>
- Reflector - <http://www.red-gate.com/products/reflector>
- ReSharper - <http://www.jetbrains.com/resharper>
- SharePoint SDKs
 - <http://msdn.microsoft.com/en-us/library/ms441339.aspx>
 - <http://msdn.microsoft.com/en-us/library/ms550992.aspx>
- SharePoint Solution Installer -
<http://www.codeplex.com/sharepointinstaller>

- [Design Guidelines for Class Library Developers](#)
 - Assembly Name und Default Namespace sind identisch
 - Kommentare, Variablen, Klassen und Methoden haben sprechende Namen
 - Sprechende Namen in Englisch können von jedem gelesen werden
- Try/Catch Blöcke verwenden!
- `Url/default.aspx?Contents=1` bringt Webpart Wartungsseite
- .NET 3.5 ohne Probleme nutzbar

- Methode wird überschrieben und die base Methode aufgerufen
- Vorteile gegenüber Render()
 - SharePoint Objektmodell bleibt intakt
 - Variablen für den Zugriff auf Eigenschaften vom Client aus
 - ID für Webpart Tabelle und DIV



- <http://andrewconnell.com/blog/archive/2008/02/18/Understanding-how-Web-Parts-are-rendered-why-to-never-use.aspx>

Dispose() or not Dispose()

- Garbage Collector des Frameworks kümmert sich nicht um alle SharePoint Objekte
- SPSite und SPWeb Objekte müssen häufig manuell nach deren Erstellung wieder zerstört werden
 - Mehr Infos im ULS Log →
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shared Tools\Web Server Extensions\HeapSettings\SPRequestStackTrace:
DWORD = 1
- Viele Beispiele und Infos: <http://blogs.msdn.com/rogerla>
- Aus dem SPContext erstellte Objekte werden nicht Disposed! Sonst funktioniert eine SharePoint Seite nicht mehr korrekt
- “SPDisposeCheck tool for SharePoint Developers” angekündigt -
<http://blogs.msdn.com/sharepoint/archive/2008/11/12/announcing-spdisposecheck-tool-for-sharepoint-developers.aspx>

Dispose() or not Dispose()

- Dieses SPSSite Objekt wird nicht nach dem Gebrauch zerlegt

```
SPSSite site = new SPSSite(siteUrl);
```



- Besser

```
using (SPSSite site = new SPSSite(siteUrl))  
{  
}
```

- Oder

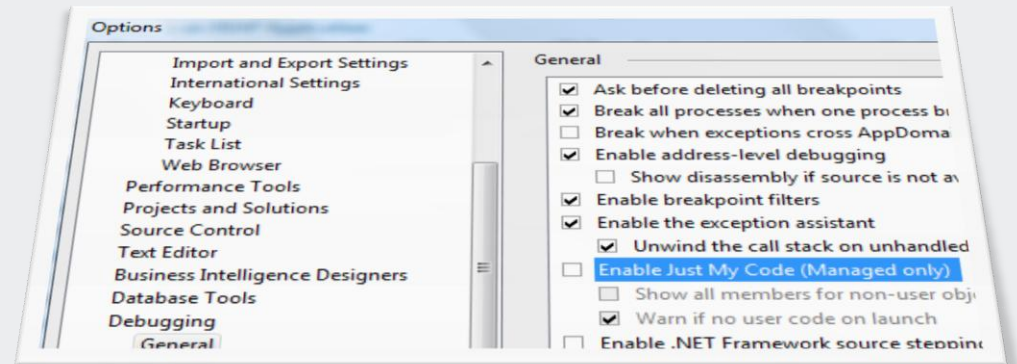
```
SPSSite site = null;  
try {  
    site = new SPSSite(siteUrl);  
}  
finally {  
    if (site != null) site.Dispose();  
}
```

- Erweiterungen für SharePoint
 - z.B. Webpart, eine angepasste Liste oder eine Aktion auf einer Liste
- Installation über eine Solution in der gesamten Farm
 - Mehrere Features in einer Solution möglich – wsp Datei
- Zielgerichtet auf Bereiche
 - Aktivierung auf Geltungsbereiche
 - Bereich: Farm, WebApplication, Site, Web
 - Webparts auf Site Ebene, da sie im Webpart Katalog auf der Site aufgenommen werden
 - Aktionen auf Listen können auf der Site oder auf dem Web hinzugefügt werden

- Neues WSPBuilder Projekt
- Webpart Feature erstellen
- Installation
- Deployment
- Feature Aktivierung
- Webpart auf Seite hinzufügen

Demo

- Fehlersuche
- Debugging von Assemblies im GAC
- „Attach to process“ im Visual Studio und mit dem w3wp Prozess verbinden
- Auflistung der Application Pools mit „iisapp.vbs“
- Web.config Einstellungen
 1. `<SafeMode ... CallStack="true" ... />`
 2. `<customErrors mode="Off" />`
 3. `<compilation batch="false" debug="true">`



- Schreiben in die SharePoint Logs
 - ULS (Unified Logging Service)
- C:\Programme\Gemeinsame Dateien\Microsoft Shared\Web Server Extensions\12\LOGS
- MSDN: Writing to the Trace Log -
[http://msdn.microsoft.com/hi-in/library/aa979595\(en-us\).aspx](http://msdn.microsoft.com/hi-in/library/aa979595(en-us).aspx)
- Konfiguration des Schweregrades
- Eigene Kategorien möglich
- „Allow Unsafe Code“ Einstellung in den Projekt Build Eigenschaften nötig

Kategorie auswählen

Allgemein

Unwichtigstes, im Ereignisprotokoll aufzuzeichnendes Ereignis

Fehler

Unwichtigstes, im Ablaufverfolgungsprotokoll aufzuzeichnendes Ereignis

Mittel

- Eintrag im SharePoint Log:

01/10/2009 10:40:05.60 RH.FileserverAccess.dll
RH.FileserverAccess General Verbose Getting
current user from web 'http://www.hezser.de/blog'

- Assembly Name

- Produkt Name

- Kategorie

- Level

- Text

- TraceProvider Klasse
- Write Methoden für verschiedene Trace Level

```
internal static void WriteULSError(string text)
{
    try {
        TraceProvider.RegisterTraceProvider();
        TraceProvider.WriteTrace(
            0, TraceProvider.TraceSeverity.Exception, Guid.Empty,
            "RH.SharePointKonferenz.dll",
            "RH.SharePointKonferenz", "General", text);
    }
    catch (Exception) {}
    finally
    {
        TraceProvider.UnregisterTraceProvider();
    }
}
```

- Aktionen für Feature Ereignisse definieren
- Keine UI
 - Meldungen ins Log oder Abbruch
- Assembly im GAC bedeutet Full Trust

Reagieren auf Ereignisse

- Activation
- Deactivation
- Installation
- Uninstallation

Zugriff auf Parent Objekte

- Abhängig vom Scope
- SPSTite
- SPWeb

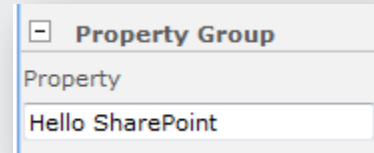
Zugriff auf die Feature Definition

- Name
- ID
- ActivationDependency
- ...

- Vereinfachung von häufigen Aufgaben
- Leichte Lesbarkeit
- Weniger Code
- Wiederverwendbar

```
public static bool GetList(  
    this SPWeb web, string listTitle, out SPList list)  
{  
    list = web.Lists.Cast<SPList>().Where(  
        l => l.Title == listTitle).FirstOrDefault();  
    return list != null;  
}
```

- Eigenschaften anpassen



- „Aufpeppen“ mit Web Controls und zusätzlichen Funktionen



- ToolPart[] definiert was in den Eigenschaften angezeigt wird
- Datentypen für Standardeigenschaften: String, Boolean, Integer, DateTime und Enum

- HttpContext des aktuellen Request
 - Dadurch nur im Web verfügbar, nicht in einer Konsolenapplikation oder Workflows
- Der Kontext ermöglicht Zugriff auf aktuelle Objekte
- Rechte des Benutzers werden übernommen
- Zugriff über SPContext.Current
 - Site, Web, Web.CurrentUser, List, ListItem
- Setzen des Kontext möglich
 - `SPContext context = SPContext.GetContext(HttpContext.Current, ListItemID, ListID, SPWeb)`

- Zugriff auf Objekte mit erweiterten Berechtigungen
- Ausführen als
 - Benutzer des Anwendungspools
 - Impersonation – Ausführung als jeder beliebige Benutzer
- Immer ein neues Objekt erzeugen und nicht aus einem anderen Kontext zuweisen!
- Sparsam nutzen und gut testen

Elevated Privileges – Vergleich

	RunWithElevatedPrivileges	New SPSite
Ausführen als	Anwendungspool Benutzer	Jeder Benutzer
Innerhalb SharePoint	Ja	Ja
Außerhalb SharePoint	Ja	Nein

- **RunWithElevatedPrivileges** ist mächtiger
- **New SPSite** ist flexibler

- Zugriff auf Objekte als Account des Anwendungspools

```
SPSite elevatedSite;  
SPSecurity.RunWithElevatedPrivileges(delegate  
{  
    elevatedSite = new  
        SPSite(SPContext.Current.Web.Url);  
});
```

- Zugriff auf Objekte als System Account Benutzer

```
SPSite elevatedSite;  
elevatedSite = new SPSite(  
    SPContext.Current.Web.Url,  
    SPContext.Current.Site.SystemAccount.UserToken);
```

- Statt des System Account kann auf jeden beliebigen Benutzer gewechselt werden!

- Views als HTML ausgeben
- Änderung der Caml Abfrage möglich
- Postback „vergisst“ z.B. Filterungen
- [Searching SharePoint lists and document libraries with RenderAsHtml](#)
- Ähnlich
 - `SPField.GetFieldValueAsHtml()`

- User Information List anzeigen

Demo

- Jede Spalte ist durch ihren FieldType einem Webcontrol zugeordnet
- Webcontrol muss den Kontext der Liste/des Items kennen
- ControlMode: New, Edit, Display
- Beispiel auf



<http://www.hezser.de/blog/Lists/Beitraege/Post.aspx?ID=38>

- „Name“ und „Über mich“ aus „Alle Personen“ anzeigen

Demo



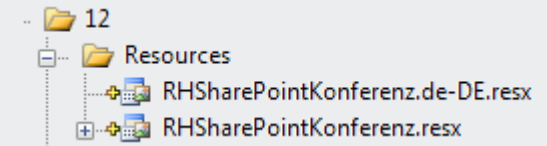
- Feature
 - „Leeres“ Feature wird nicht unterstützt
- Klassen
 - Als „public“ deklarieren! (Webparts, Feature Receiver...)
- Methoden
 - `base.BasisMethode()` vergessen aufzurufen



- SPList
 - `SPList.ItemCount != SPList.Items.Count`
 - DateTime Spalten
 - Indexer liefert lokale Zeit
 - `list.Items[0][„Time“] != list.GetItemById(1)[„Time“]`
- Objekte
 - ID setzen nicht vergessen

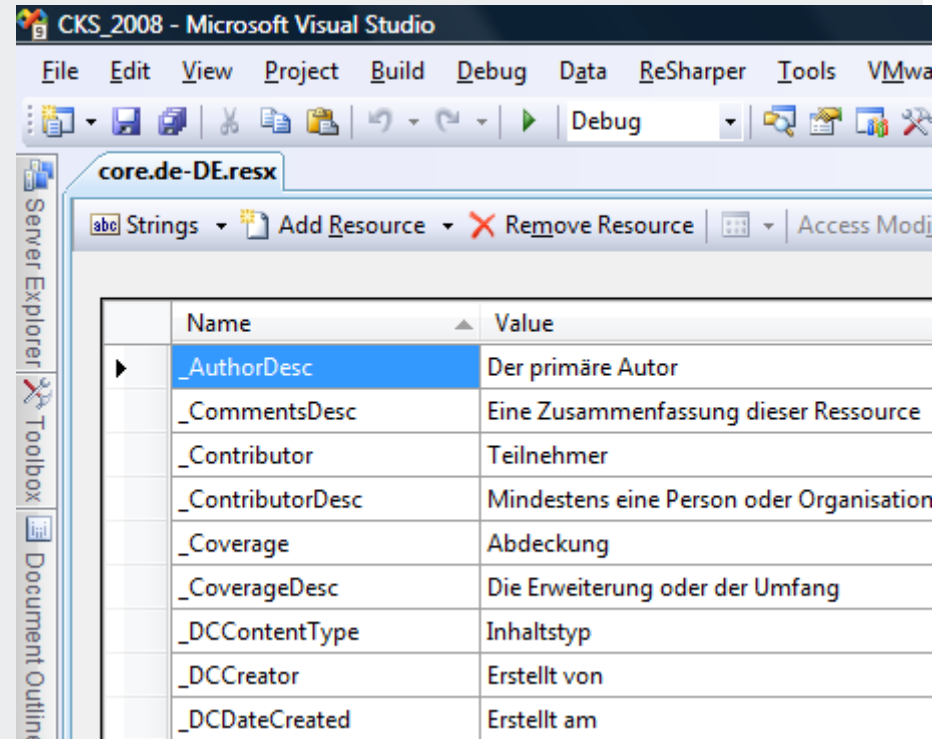
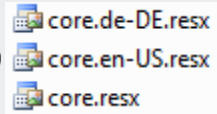
- Lokalisierung von strings in Webparts
- Ressourcen
 - SharePoint Ressourcen nutzen
 - Eigene Ressourcen für jede Sprache erstellen
- Umschalten anhand der RegionalSettings des Webs oder des Benutzers
- LocaleID zur Identifizierung der Sprache
- `SPUtility.GetLocalizedString()` verwenden

- Eigene Ressourcen Datei pro Sprache
- LCID aus
 - web.Language
 - web.RegionalSettings.LocaleId
 - web.CurrentUser.RegionalSettings.LocaleId



```
uint lcid = web.RegionalSettings.LocaleId;  
  
string welcome = SPUtility.GetLocalizedString(  
"$Resources:Welcome", "RHSharePointKonferenz",  
lcid);
```

- Öffnen der core Resource mit dem Visual Studio
- „\$Resources:“ + Name



- C# Debate: When Should You Use var? - <http://www.infoq.com/news/2008/05/CSharp-var>
- Don't use the System.Diagnostics Assert() methods unless - <http://mark.michaelis.net/Blog/DontUseTheSystemDiagnosticsAssertMethodsUnless.aspx>
- Web Part Caching – A simple approach - <http://www.zimmergren.net/archive/2008/10/07/web-part-caching-%E2%80%93-a-simple-approach.aspx>
- SPListItem Class - <http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.splistitem.aspx>
- Best Practices: SharePoint Object Model for Performance Tuning - <http://blogs.msdn.com/sowmyancs/archive/2008/10/26/best-practices-sharepoint-object-model-for-performance-tuning.aspx>

Noch Fragen?

- Jetzt ist die Zeit dafür!
- ...oder Frage per E-Mail an rene@hezser.de



Ich freue mich auf Ihr Feedback

DANKE!

Wir sehen uns wieder:



**Schneller zum .NET 3.5
Developer**

23.-27. März 2009 in Burghausen
04.-08. Mai 2009 in Köln

www.DevTrain.de/Camp



**Advanced Developers
Conference**

Oktober 2009

www.ADC09.de



**In 5 Tagen zum
SharePoint Profi**

23.-27. Februar 2009 in Köln
09.-13. März 2009 in München

www.SharePointCamp.de

Vielen Dank!

René Hézszer

Goldsponsor:



Partner:



Silbersponsoren:



Veranstalter:

