



René Hézser | Net at Work Netzwerksysteme GmbH

Webpart-Entwicklung - Tipps und Tricks

Agenda

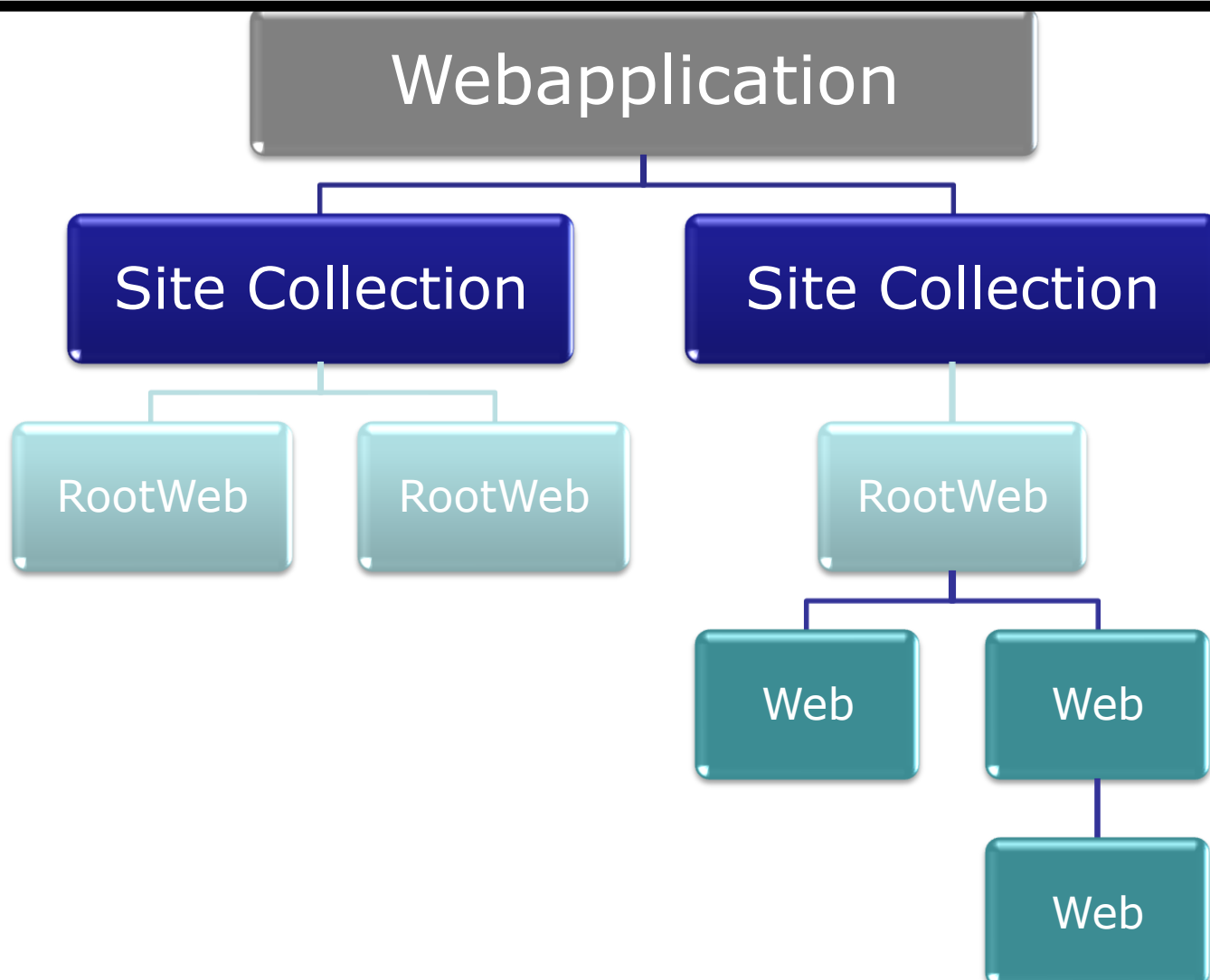
- Webpart Grundlagen
- Logging in das SharePoint Log
- Webpart Feature
- Extension Methods
- Webparts mit eigenen ToolParts
- SPContext – wo bin ich
- Code Ausführung mit erweiterten Berechtigungen
- Nutzung “.ToHtml()” Funktionen
- SharePoint Webcontrols
- Fallen umgehen und Probleme vermeiden
- Mehrsprachigkeit

Begriffe

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

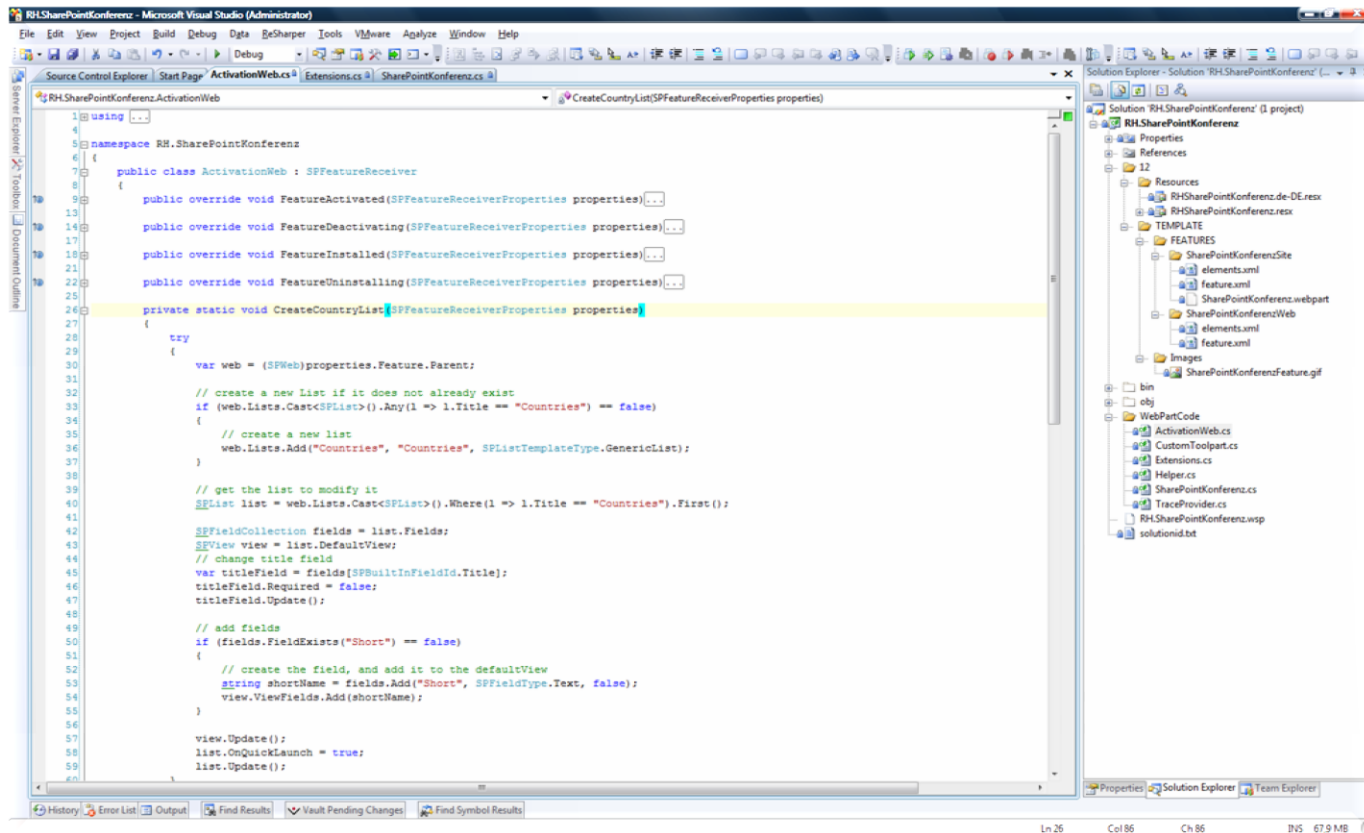
SharePoint Objekte

Farm



Beispiel Code

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

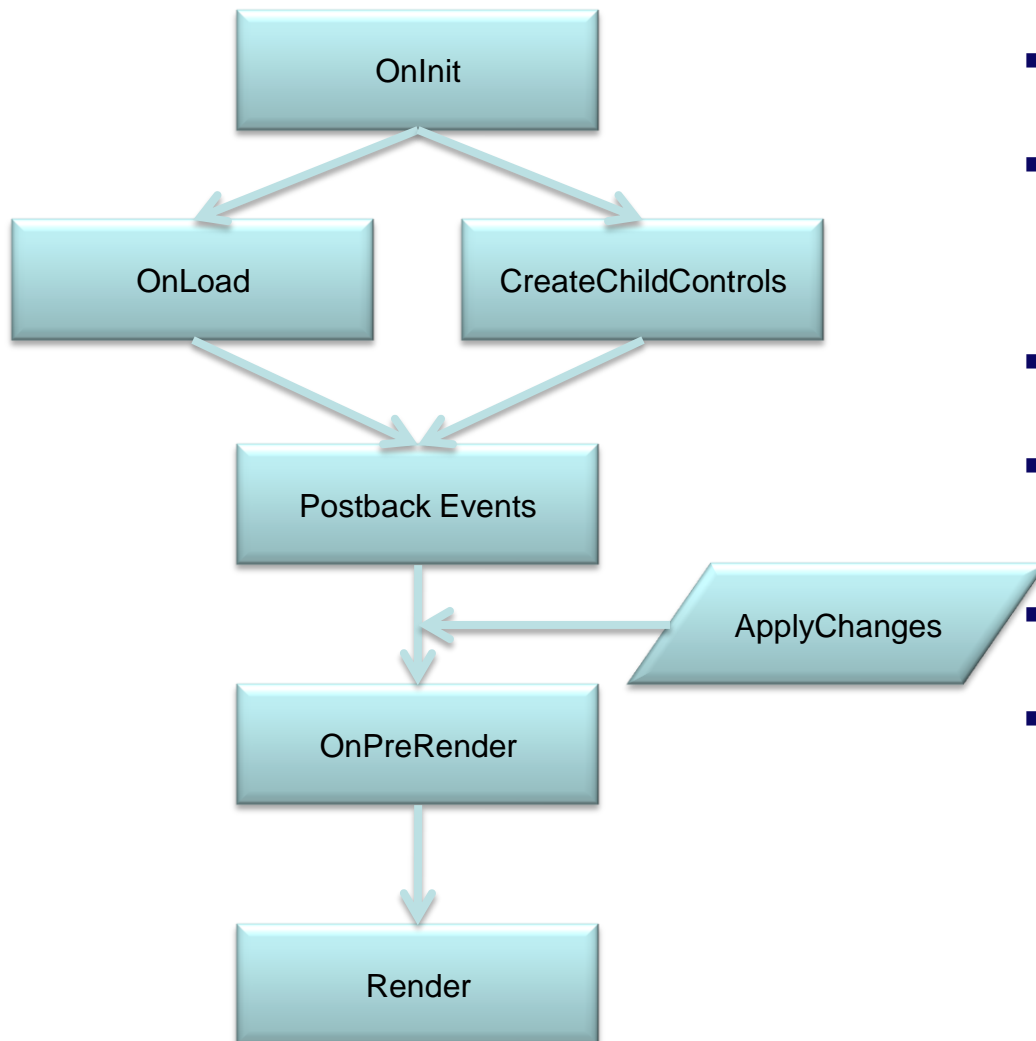


```
1 using ...
2
3
4 namespace RH.SharePointKonferenz
5 {
6     public class ActivationWeb : SPFeatureReceiver
7     {
8
9         public override void FeatureActivated(SPFeatureReceiverProperties properties)...
10
11         public override void FeatureDeactivating(SPFeatureReceiverProperties properties)...
12
13         public override void FeatureInstalled(SPFeatureReceiverProperties properties)...
14
15         public override void FeatureUninstalling(SPFeatureReceiverProperties properties)...
16
17         private static void CreateCountryList(SPFeatureReceiverProperties properties)
18         {
19             try
20             {
21                 var web = (SPWeb)properties.Feature.Parent;
22
23                 // create a new List if it does not already exist
24                 if (web.Lists.Cast<SPList>().Any(l => l.Title == "Countries") == false)
25                 {
26                     // create a new list
27                     web.Lists.Add("Countries", "Countries", SPListTemplateType.GenericList);
28                 }
29
30                 // get the list to modify it
31                 SPList list = web.Lists.Cast<SPList>().Where(l => l.Title == "Countries").First();
32
33                 SPFieldCollection fields = list.Fields;
34                 SPView view = list.DefaultView;
35                 // change title field
36                 var titleField = fields[SPBuiltInFieldId.Title];
37                 titleField.Required = false;
38                 titleField.Update();
39
40                 // add fields
41                 if (fields.FieldExists("Short") == false)
42                 {
43                     // create the field, and add it to the defaultView
44                     string shortName = fields.Add("Short", SPFieldType.Text, false);
45                     view.ViewFields.Add(shortName);
46                 }
47
48                 view.Update();
49                 list.OnQuickLaunch = true;
50                 list.Update();
51             }
52             catch { }
53         }
54     }
55 }
```


Was ist ein Webpart?

- Wiederverwendbarer Funktionsblock
- 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
 - Da meist das SharePoint Objektmodell genutzt wird, kann von `Microsoft.SharePoint.WebPartPages` abgeleitet werden und nicht von `System.Web.UI.WebControls.WebParts`
- 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>

Lebenszyklus eines Controls in einem Webpart

1. Erstellung

```
- TextBox tb = new TextBox(){ID=„textbox“};
```

2. Hinzufügen zu den Controls des Webparts

```
- Controls.Add(tb);
```

3. Zuweisen von Werten

```
- tb.Text = „ShareConnect 2009“;
```

4. Reagieren auf Ereignisse

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

5. Auslesen der Werte

```
- string wert = tb.Text;
```

6. Rendern durch das Framework (automatisch)

Tools für die Webpart Entwicklung

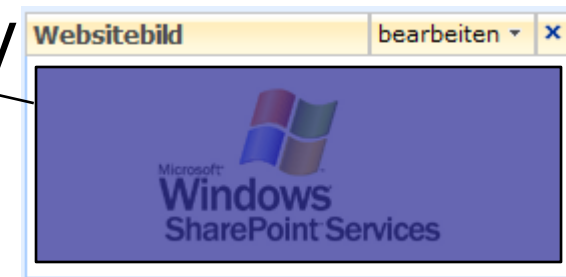
- Visual Studio 2008 SP1 (Express)
- WSPBuilder - <http://codeplex.com/wspbuilder>
- Reflector - <http://www.red-gate.com/products/reflector>
- ReSharper - <http://www.jetbrains.com/resharper>
- SharePoint SDKs (WSS / MOSS)
 - <http://msdn.microsoft.com/en-us/library/ms441339.aspx>
 - <http://msdn.microsoft.com/en-us/library/ms550992.aspx>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

Verschiedenes

- [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 SP1 ohne Probleme nutzbar

RenderContents()

- Methode wird überschrieben und die Basis 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()

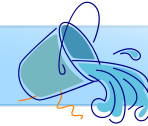
- Garbage Collector des Frameworks kümmert sich nicht um alle SharePoint Objekte
- SPSite und SPWeb Objekte müssen häufig manuell nach deren Erstellung wieder zerstört werden
 - Mehr Infos im ULS Log →
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shared Tools\Web Server Extensions\HeapSettings\SPRequestStackTrace: 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 (SPContext.Current.Web / -Site)
- SharePoint Dispose Checker Tool
<http://code.msdn.microsoft.com/SPDisposeCheck>
 - Integration in das VS
<http://vspug.com/teameli/2009/02/20/using-spdisposecheck-in-visual-studio>

```
Total Found: 0
-----
Modules Checked: 1
RH.Sharepoint.Share
-----
Modules Ignored: 0
-----
Methods Ignored: 0
```

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();  
}
```


SharePoint Feature

- 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

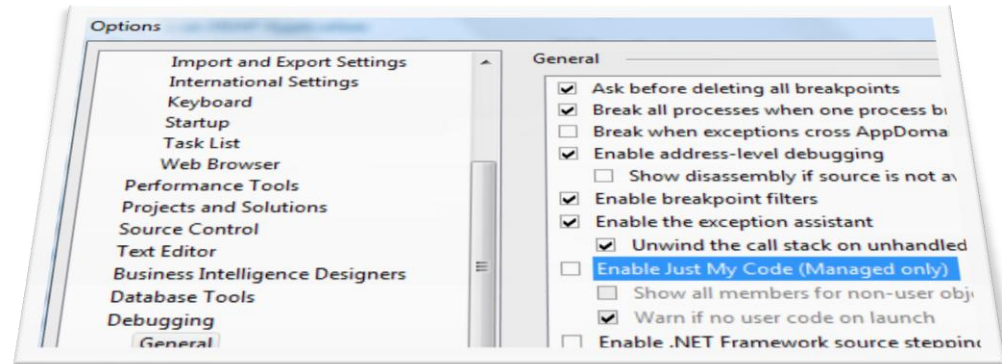
Demo - Webpart erstellen und veröffentlichen

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

Demo

Debugging

- 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“ für den IIS 6
 - „appcmd.exe“ für den IIS7
- Web.config Einstellungen
 1. `<SafeMode ... CallStack="true" ... />`
 2. `<customErrors mode="Off" />`
 3. `<compilation batch="false" debug="true">`



Internet Information Server 7

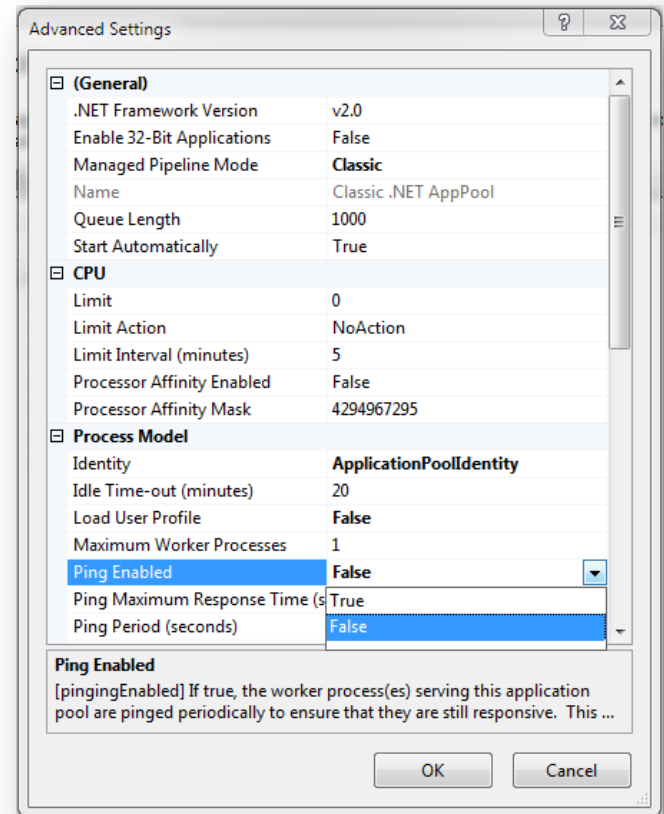
- **Maximale Upload Größe**

```
appcmd set config /section:requestfiltering  
/requestlimits.maxallowedcontentlength:30000000
```

- **Anwendungspool beendet sich beim Debuggen**

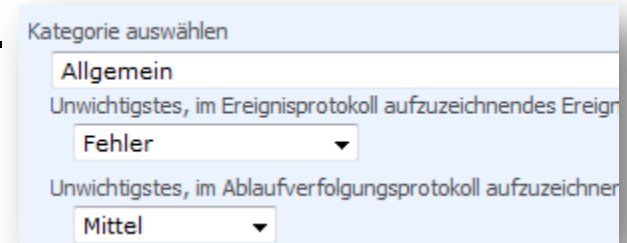
- **Installation SharePoint auf Windows Vista / 7**

<http://community.bamboosolutions.com/media/p/193.aspx>



Logging

- Schreiben in die SharePoint Logs
 - ULS (Unified Logging Service)
- C:\Program Files\Common Files\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)
- System.Diagnostics.Debug.Assert() löschen
- 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

Mittel
Unwichtigstes, im Ablaufverfolgungsprotokoll aufzuzeichnendes Ereignis

Logging - Beispiel Eintrag

- 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

Code

- 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.Sharepoint.Shareconnect.dll",
            "RH.Sharepoint.Shareconnect", "General", text);
    }
    catch (Exception) {}
    finally
    {
        TraceProvider.UnregisterTraceProvider();
    }
}
```

Feature Receiver

- 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
- SPSite
- SPWeb
- ...

Zugriff auf die Feature Definition

- Name
- ID
- ActivationDependency
- ...

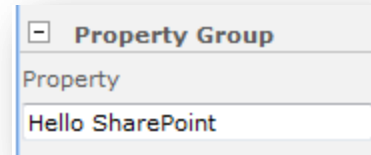
Extension Methods

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

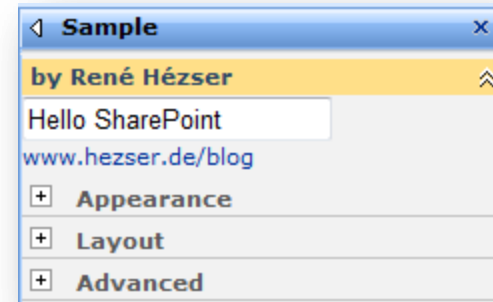
```
public static bool GetList(  
    this SPWeb web, string listTitle, out SPList list)  
{  
    list = web.Lists.Cast<SPList>().SingleOrDefault(  
        l => l.Title == listTitle);  
    return list != null;  
}
```

Custom Toolpart

- 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

SPContext - der SharePoint Kontext

- 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)`

Elevated Privileges - Erweiterte Berechtigungen

- 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

Elevated Privileges

- Zugriff auf Objekte als Account des Anwendungspools

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

Elevated Privileges

- 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!

RenderAsHtml()

- 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()

Demo - RenderAsHtml()


- User Information List anzeigen

Demo

SharePoint Webcontrols

- 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>

simpleWebpart		SPControlMode.Display	SPControlMode.Edit
Field			
MultiLinePlainText	Multiple lines, but simple text		Multiple lines, but simple text
MultiLineRichText	RichText with multiple lines		RichText with multiple lines
MultiLineEnhancedRichText	RichText with multiple lines enhanced		RichText with multiple lines enhanced
ChoiceDropDown	Dropdown Choice #1		Dropdown Choice #1 <input checked="" type="radio"/> Radio Choice #1 <input type="radio"/> Radio Choice #2 <input type="radio"/> Radio Choice #3
ChoiceRadio	Radio Choice #1		<input type="text" value="123,456"/>
Number	123,456		<input type="text" value="321,647"/>
Currency	\$321,647.00		
DateTimeDateOnly			
DateTime	4/29/2007 1:00 PM		4/29/2007 1 PM 00
LookupSingleItem	Get Started with Windows SharePoint Services!		Get Started with Windows SharePoint Services!
LookupMultipleItems	Get Started with Windows SharePoint Services!; Testsite for Control Test created		Get Started with Windows SharePoint Services!; Testsite for Control Test created
Yes/No	Yes		<input checked="" type="checkbox"/>
PersonOrGroup	Administrator		Administrator
HyperLink	Site		Type the Web address: (Click here to test) <input type="text" value="http://sharepoint"/> /spscontrols/Lists/Co Type the description: <input type="text" value="Site"/>
Picture			Type the Web address: (Click here to test) <input type="text" value="http://sharepoint"/> /_layouts/images/titlegraphic.gif Type the description: <input type="text" value="Image"/>

Demo - SharePoint Webcontrols

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

Demo

Fallen

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



Fallen

- 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

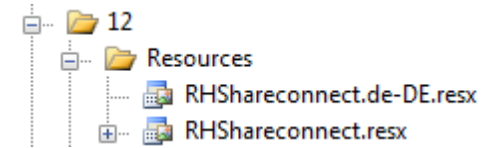


Mehrsprachigkeit / Multilanguage

- Lokalisierung von strings in Webparts
- Ressourcen
 - SharePoint Ressourcen nutzen
 - Eigene Ressource für jede Sprache erstellen
- Umschalten anhand der RegionalSettings des Webs oder des Benutzers
- LocaleID zur Identifizierung der Sprache
- `SPUtility.GetLocalizedString()` verwenden
- `stsadm -o copyappbincontent` nach der Installation um Ressourcen nutzen zu können

Mehrsprachigkeit - eigene Ressourcen nutzen

- Eigene Ressourcen Datei pro Sprache



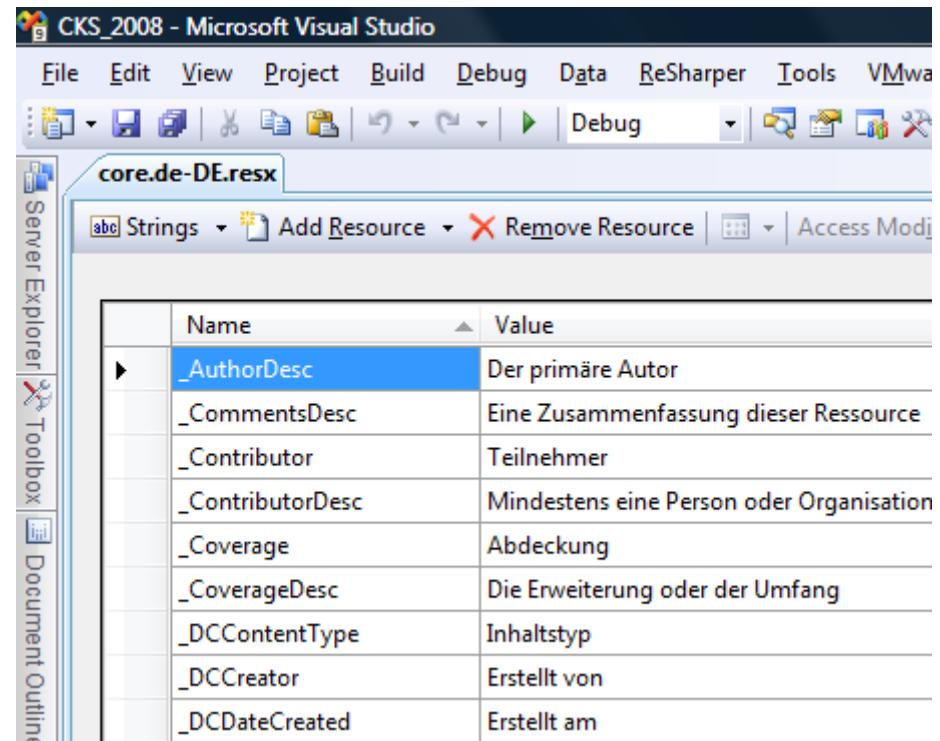
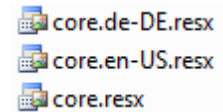
- LCID aus

- web.Language - ursprüngliche Sprache des Webs
- web.RegionalSettings.LocaleId
- SPContext.Current.RegionalSettings.LocaleId

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

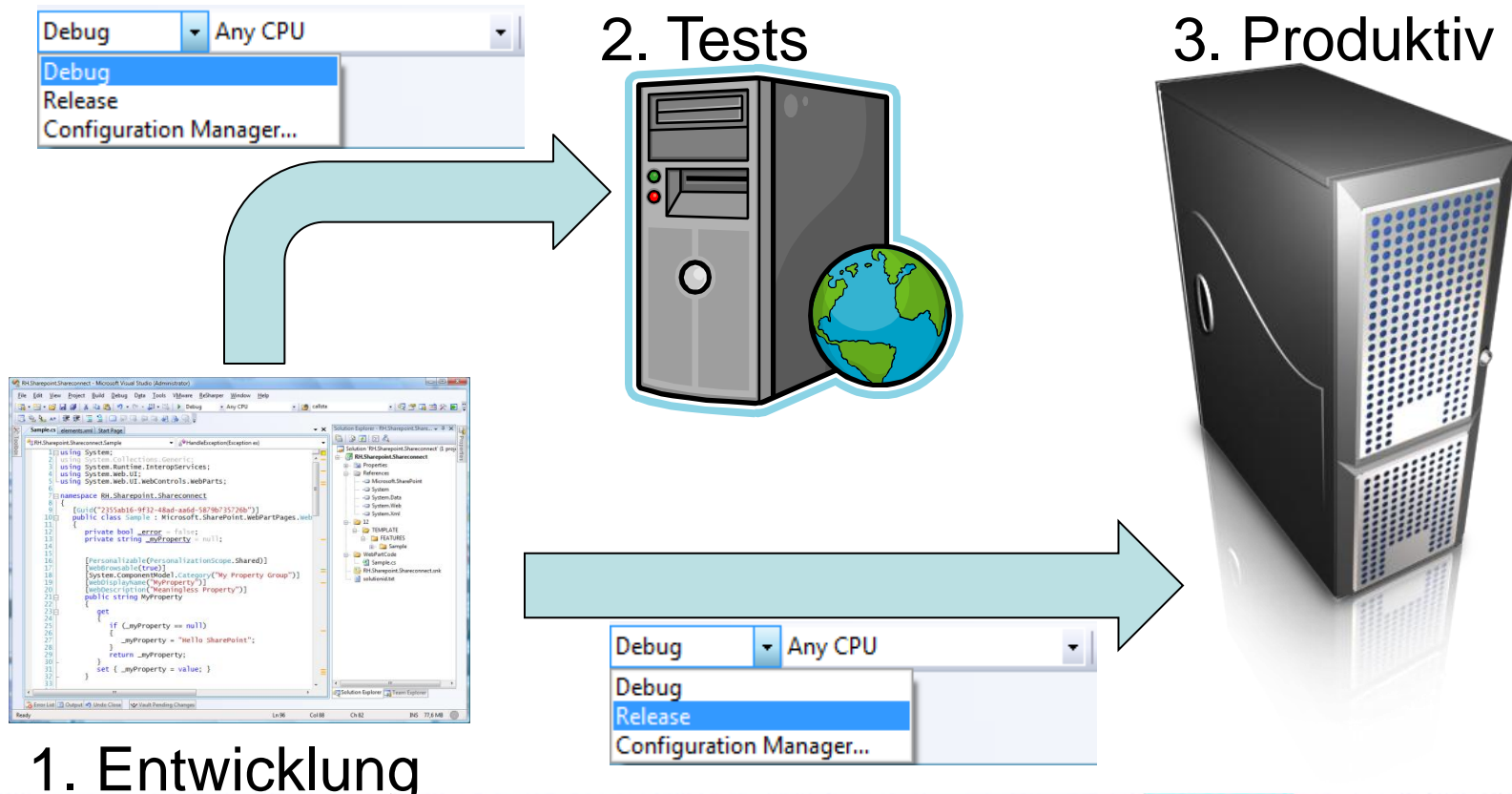
Mehrsprachigkeit - SharePoint Ressourcen nutzen

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



Deployment

- Immer zuerst auf einem Testserver!



Weitere Links

- 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>
- Don't use the System.Diagnostics Assert() methods unless... - <http://mark.michaelis.net/Blog/DontUseTheSystemDiagnosticsAssertMethodsUnless.aspx>
- Iisapp.vbs: IIS application query script <http://technet.microsoft.com/en-us/library/cc755905%28WS.10%29.aspx>

Noch Fragen?

- Jetzt ist die Zeit dafür!



Ich freue mich auf Ihr Feedback

DANKE!