# Implementierung firmenspezifischer PlugIns innerhalb von LEEGOO.NET

|  |  |  |
| --- | --- | --- |
| Erstellung | 2015-04-15, Thilo Vitt / EAS |  |
| Änderung | 2015-04-23, Thilo Vitt / EAS | Entfernen der TFS-Einträge (s.u.) |
|  | 2015-06-18, Thilo Vitt / EAS | Ausgabepfad in den Unterordner PlugIns\DemoPlugIn verschoben  div. EventHandler eingebaut, die dem PlugIn ermöglichen auf div. Aktionen in LEEGOO BUILDER.NET zu reagieren |
|  | 2019-04-24, Thilo Vitt / EAS | Kleinere Anpassungen und Korrekturen |

## **Beispiel-PlugIn kopieren**

Es gibt eine Implementierung eines Demo-PlugIns.  
Darin sind bereits viele benötigten Funktionalitäten enthalten.  
Dieses Beispiel kann als Template für eigene PlugIns dienen.   
Bitte kopieren Sie das entsprechende Verzeichnis auf Ihren Rechner.

Verweise anpassen

Folgende Verweise auf LEEGOO BUILDER.NET-Assemblies müssen ggf. manuell angepasst werden:  
<Binaries> steht für das Verzeichnis mit den LEEGOO BUILDER.NET-Dateien.  
Microsoft Prism muss nicht installiert sein. Es genügt auf die im <Binaries>-Ordner enthaltenen Assemblies zu verweisen.

* <Binaries>\EAS.LeegooBuilder.Client.Common.ToolsAndUtilities.dll
* <Binaries>\EAS.LeegooBuilder.Client.ServerProxy.BusinessServiceClientBase.dll
* <Binaries>\EAS.LeegooBuilder.Common.CommonTypes.dll
* <Binaries>\EAS.LeegooBuilder.Server.DataAccess.Core.dll
* <Binaries>\EAS.LeegooBuilder.Client.GUI.Modules.MainModule.dll
* <Binaries>\ToolsAndUtilities.dll
* <Binaries>\Microsoft.Practices.Prism.Composition.dll
* <Binaries>\Microsoft.Practices.Prism.Interactivity.dll
* <Binaries>\Microsoft.Practices.Prism.MefExtensions.dll
* <Binaries>\Microsoft.Practices.Prism.Mvvm.dll
* <Binaries>\Microsoft.Practices.Prism.Mvvm.Desktop.dll
* <Binaries>\Microsoft.Practices.Prism.PubSubEvents.dll
* <Binaries>\Microsoft.Practices.Prism.SharedInterfaces.dll
* <Binaries>\Microsoft.Practices.ServiceLocation.dll

Falls es gewünscht ist, z.B. von der in LEEGOO BUILDER.NET verwendeten Version von DevExpress unabhängig zu sein, muss die Eigenschaft ‚Lokale Kopie‘ bei allen entsprechenden Verweisen auf true gesetzt werden. Dadurch werden die verwendeten Assemblies mit in den Ausgabepfad kopiert.

Umbenennen von Dateien

Folgende Dateien sollten umbenannt werden:

* DemoPlugIn.csproj
* DemoPlugInMainModuleController.cs
* DemoPlugInModuleInit.cs
* DemoPlugInViewModel.cs
* DemoPlugInViewModelCommands.cs
* DemoPluginView.xaml

Die Namespaces sollten entsprechend angepasst werden.  
Die Datei Themes\Generic.xaml darf nicht umbenannt werden!

Entfernen der TFS-Einträge

EAS verwendet für die Entwicklung den TeamFoundationServer. Das gilt auch für das DemoPlugin.  
Dadurch wird beim Öffnen der Solution auf fremden Rechnern versucht, auf unseren TFS zu verbinden. Dieser ist jedoch nicht öffentlich zugänglich.   
Um die Warnungen dauerhaft zu unterdrücken, müssen folgende Anpassungen durchgeführt werden:

**Datei DemoPlugIn.dsprj**  
Unterhalb von <Project>|<PropertyGroup> finden sich folgende Einträge:  
<SccProjectName>SAK</SccProjectName>  
<SccLocalPath>SAK</SccLocalPath>  
<SccAuxPath>SAK</SccAuxPath>  
<SccProvider>SAK</SccProvider>  
Diese Zeilen müssen gelöscht werden

**Datei DemoPlugIn.sln**  
Unterhalb von Global| GlobalSection(TeamFoundationVersionControl) = preSolution findet sich folgender Eintrag:  
SccTeamFoundationServer = <https://eas.dnsalias.net:8010/tfs/defaultcollection>  
Diese Zeile muss gelöscht werden.

Sonstige Anpassungen

Datei Properties\AssemblyInfo.cs  
Atribute AssembyTite() umbenennen  
Attribut AssemblyProduct() umbenennen

Datei DemoPlugInMainModuleController.cs  
Konstante DemoPlugInRegionName umbenennen

Datei Datei DemoPlugInModuleInit.cs  
Variable demoPlugInModuleController umbenennen  
Attribut ModuleExport() umbenennen

Datei Images\NavigationBar\plugin\_32x32.png  
Ersetzen durch ein eigenes Glyph in der Größe 32x32  
Falls diese Datei auch umbenannt wird, muss dies in der Datei DemoPluginMainModuleController beim Aufruf von RegisterViewModel() angepasst werden

Projekteigenschaften anpassen

In den Projekteigenschaften müssen folgende Änderungen durchgeführt werden:

* AssemblyName und Standardnamespace müssen angepasst werden
* Der Ausgabepfad muss auf denUnterordner PlugIns\DemoPlugIn (o.ä.) unterhalb des lokalen Bin-Verzeichnis des LEEGOO BUILDER.NET zeigen
* Als Startaktion muss ‚Externes Programm starten‘ angewählt werden. Programm muss auf <Binaries>EAS.LeegooBuilder.Client.GUI.Shell.exe gestellt werden.

Integration in LEEGOO BUILDER.NET

In der Datei EAS.LeegooBuilder.Client.GUI.Shell.exe.config muss unterhalb von <Modules> Folgendes ergänzt werden:

    <module startupLoaded="true" assemblyFile="PlugIns\DemoPlugIn\DemoPlugIn.dll" moduleType="EAS.LeegooBuilder.Client.GUI.Modules.DemoPluginModule.ModuleInit, EAS.LeegooBuilder.Client.GUI.Modules.DemoPluginModule" moduleName="DemoPlugInModule">

      <dependencies>

        <dependency moduleName="MainModule" />

      </dependencies>

    </module>

Die dort angegebenen Bezeichner von assemblyFile, moduleType und ModuleName dll, ModuleName müssen angepasst werden

In DemoPlugInMainModuleController.RegisterNavBarItem () kann über die Konstante ‚position‘ die Position des Glyphs innerhalb der NavigationBar angepasst werden.  
An gleicher Stelle können diverse EventHandler des LEEGOO BUILDER.NET abonniert werden, um auf verschiedene Programmereignisse reagieren zu können (z.B. Speichern von Bewegungsdaten).

ViewModel anpassen

Im DemoPlugInViewModel sollten folgende Anpassungen durchgeführt werden:

* Property Caption
* In SetupUI() die Variable ‚titel‘
* In SetUpRibbonViewModel() das Ribbon neu aufbauen

View anpassen

* Pfad ‚/DemoPlugIn‘ im ‚SpecialisedDescriptionTemplate‘ anpassen
* Pfad ‚/DemoPlugIn‘ im ‚SpecialisedLongTextTemplate‘ anpassen