

---

## **Plugins entwickeln für den VM Slide Explorer 2010**

---

## Inhaltsverzeichnis

1	Allgemeines.....	3
2	Ein Plugin schreiben.....	4
3	Ein Plugin registrieren.....	11
4	Aktivieren von Plugins.....	12
5	Plugin-Tutorials.....	13

## 1 Allgemeines

Durch eine Plugin-Schnittstelle ist es möglich dem VM Slide Explorer 2010 Funktionalität hinzuzufügen bzw. diese zu verändern.

In dieser Schnittstelle sind beinahe alle Funktionen und Daten des Slide Explorers in einem Plugin referenzier- und somit änderbar.

Der VM Slide Explorer 2010 ist eine Entwicklung basierend auf der .Net Plattform. Plugins müssen deshalb ebenfalls in einer .Net Sprache (vorzugsweise in C#) programmiert werden.

Die eigentliche Schnittstelle wird durch die Klasse SEPlugin angeboten (sie befindet sich in der VMscope.SEPluginBase.DLL im Slide Explorer Installationsverzeichnis). Um ein eigenes Plugin zu schreiben wird eine von dieser Klasse abgeleitete öffentliche Klasse implementiert und in einer .net Assembly (.exe oder .dll) bereitgestellt.

Damit ein Plugin im VM Slide Explorer ausgeführt wird muss es bei diesem registriert und aktiviert werden.

Im folgendem sollen die Möglichkeiten und Benutzung der Plugin-Schnittstelle durch Beispiele veranschaulicht werden.

## 2 Ein Plugin schreiben

Ein Plugin für den VM Slide Explorer 2010 ist eine .net Assembly mit einer öffentlichen von VMscope.SEPluginBase .SEPlugin abgeleiteten Klasse.

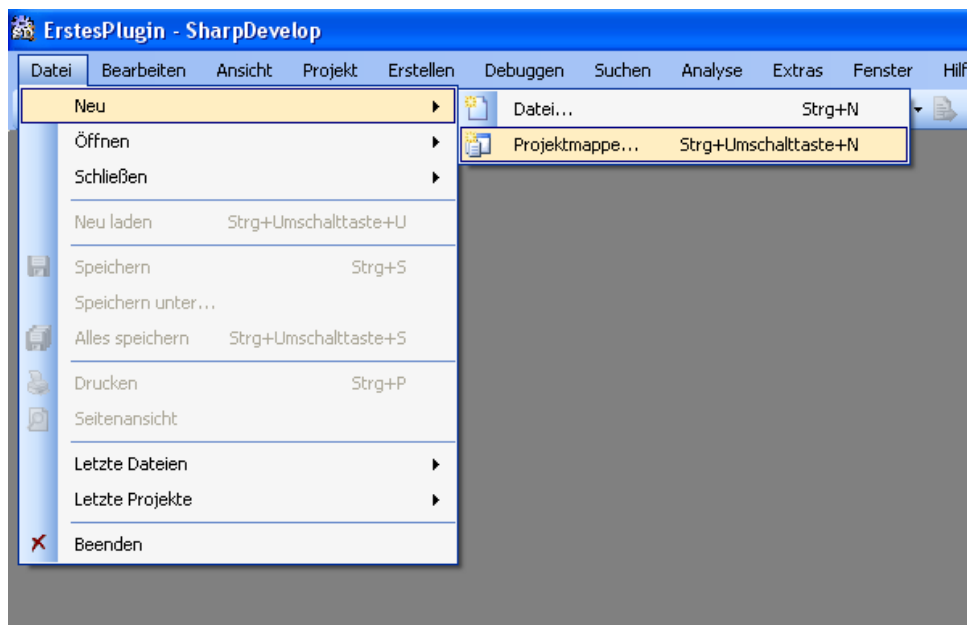
Um ein solches Plugin zu erstellen sind folgende Schritte nötig:

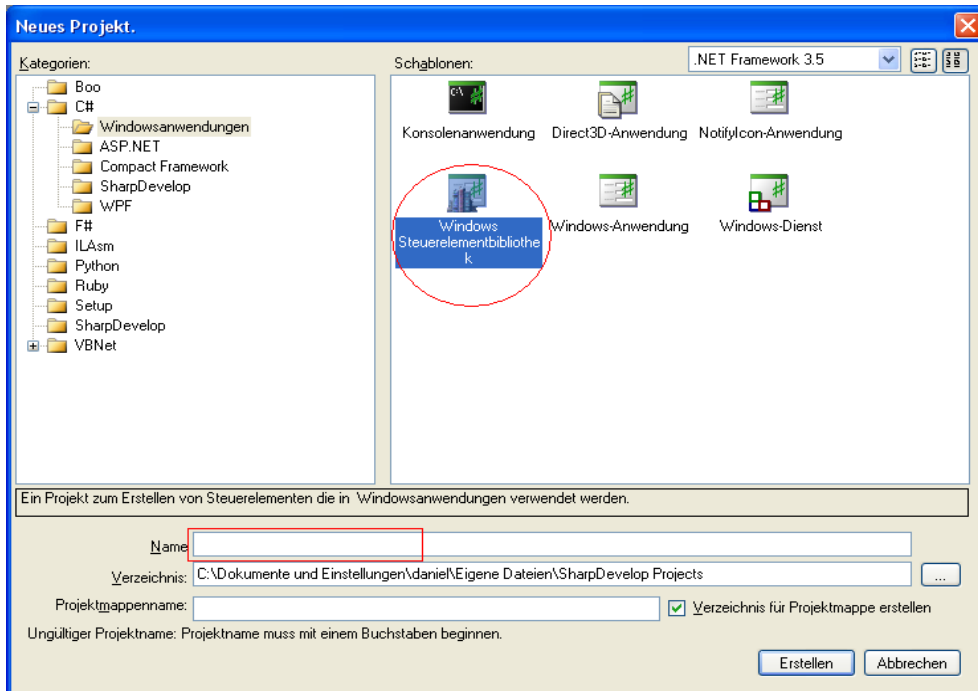
- Anlegen eines neuen Projektes
- Hinzufügen von Referenzen (insbesondere auf die Datei: „VMscope.SEPluginBase.dll“)
- Erstellen einer von SEPlugin abgeleiteten Klasse

### Ein neues Projekt anlegen

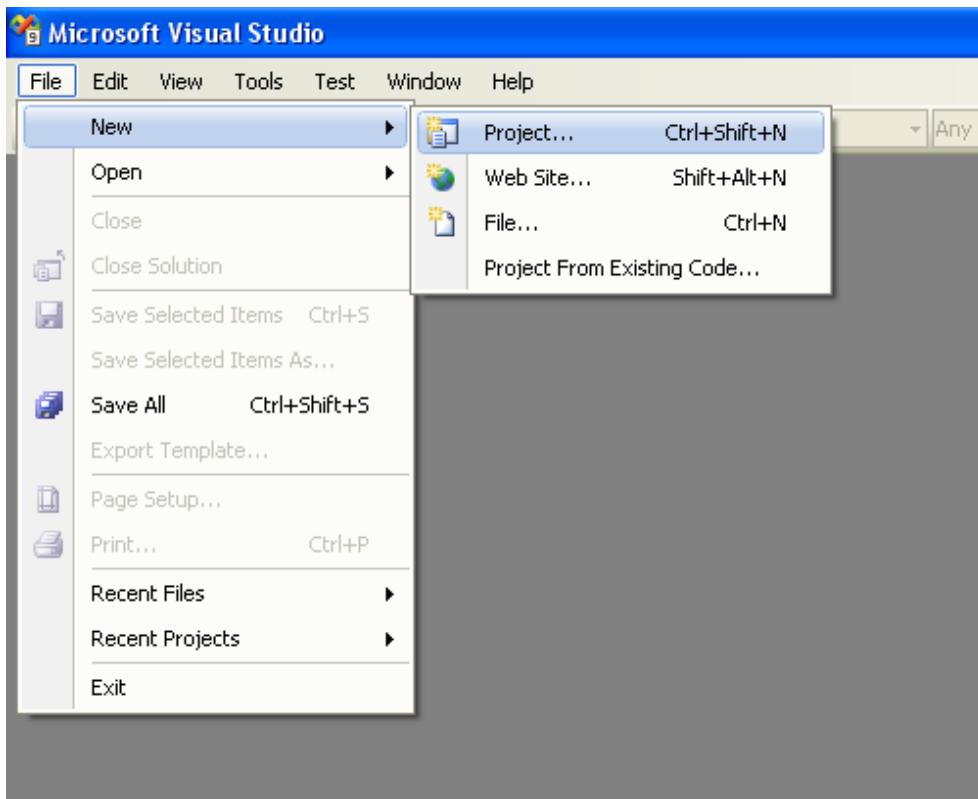
Legen Sie Ihrer Entwicklungs-Umgebung eine neue Projektmappe an, wählen sie als Template C# Klassen Bibliothek (Windows-Steuerelement-Bibliothek) aus und vergeben Sie einen Namen.

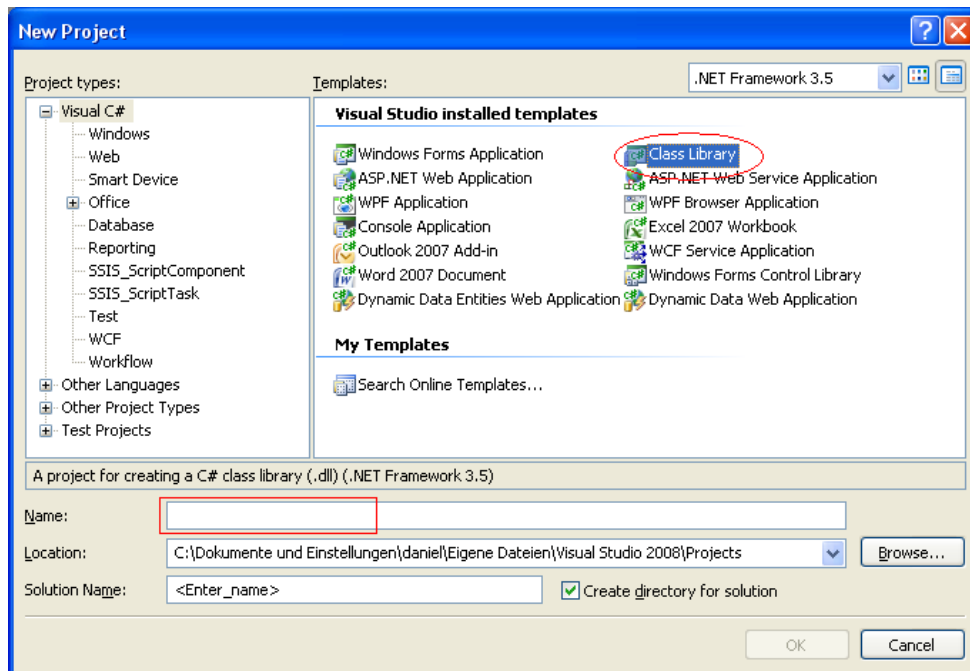
Entwicklungsumgebung SharpDevelop:





Entwicklungsumgebung Visual Studio 2008:





## Referenzen hinzufügen

Die für die Plugin Entwicklung nötigen Klassen befinden sich in den Dateien:

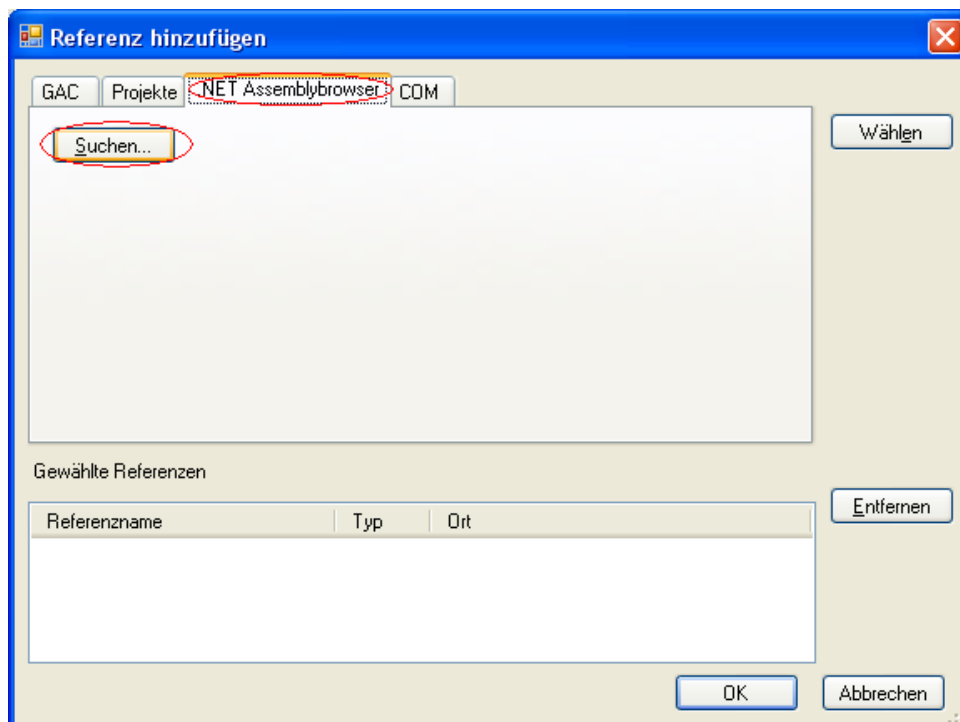
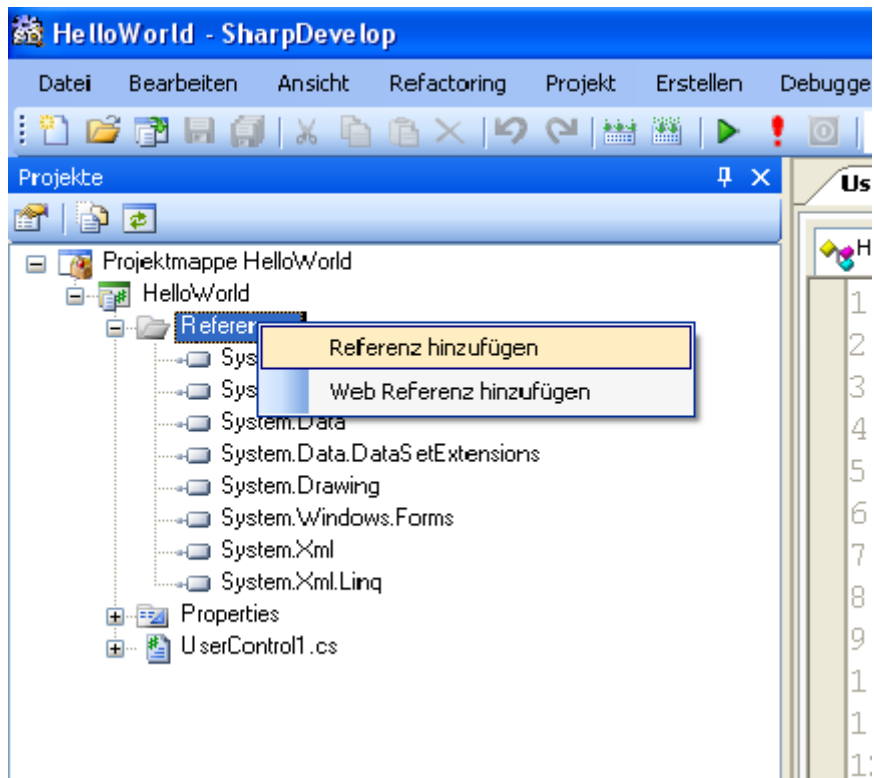
```
VMscope.FrameWork.Trustable.Resources.DLL  
VMscope.FrameWork.Trustable.Resources.Extended.DLL  
VMscope.InteropCore.DLL  
VMscope.SEPluginBase.DLL
```

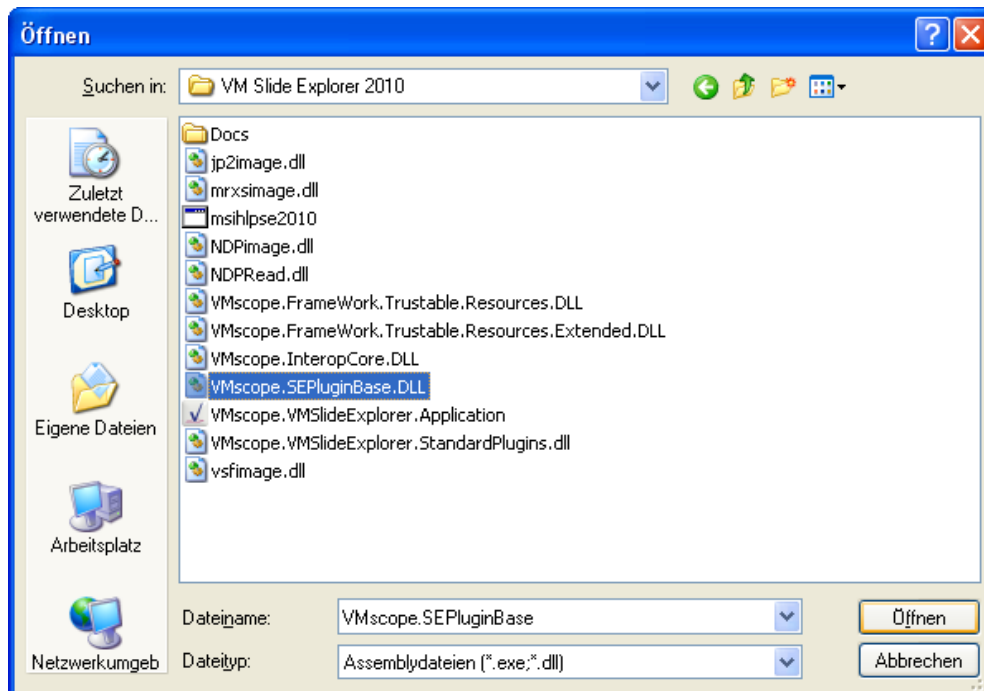
Sie befinden sich im Slide Explorer 2010 Installationsverzeichnis (üblicherweise: C:\Programme\VMscope\VM Slide Explorer 2010).

Fügen Sie Ihrem Projekt Referenzen zu diesen Dateien hinzu.

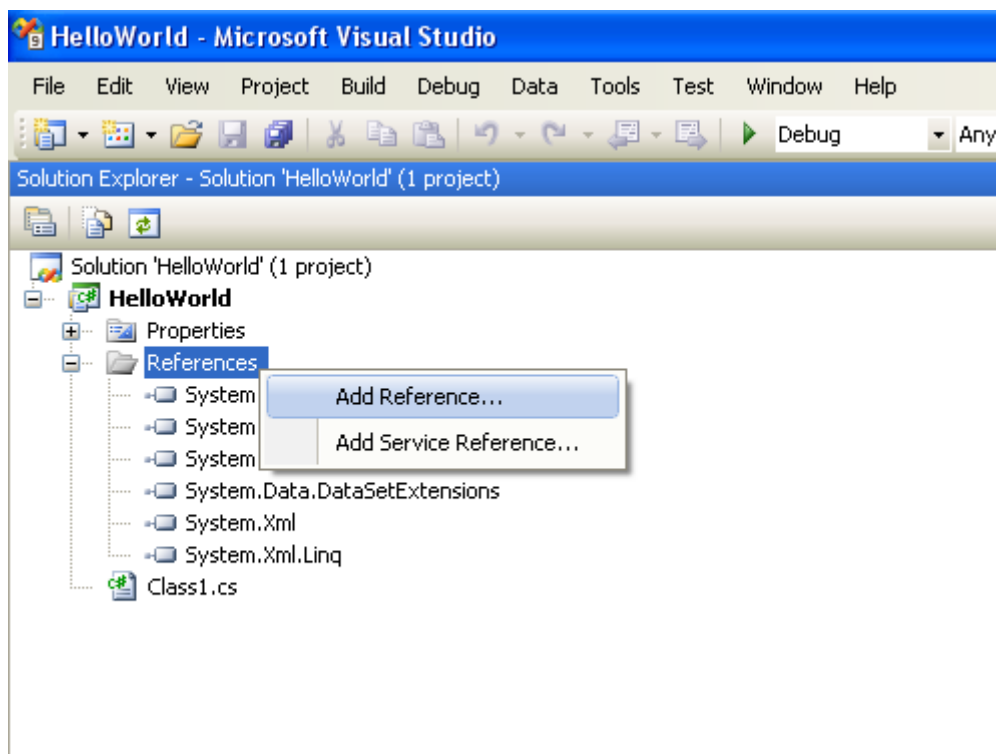
In den folgenden Abbildungen wird nur eine einzelne Referenz hinzugefügt, Sie können auch mehrere Dateien auf einmal auswählen, die referenziert werden sollen.

Entwicklungsumgebung SharpDevelop:

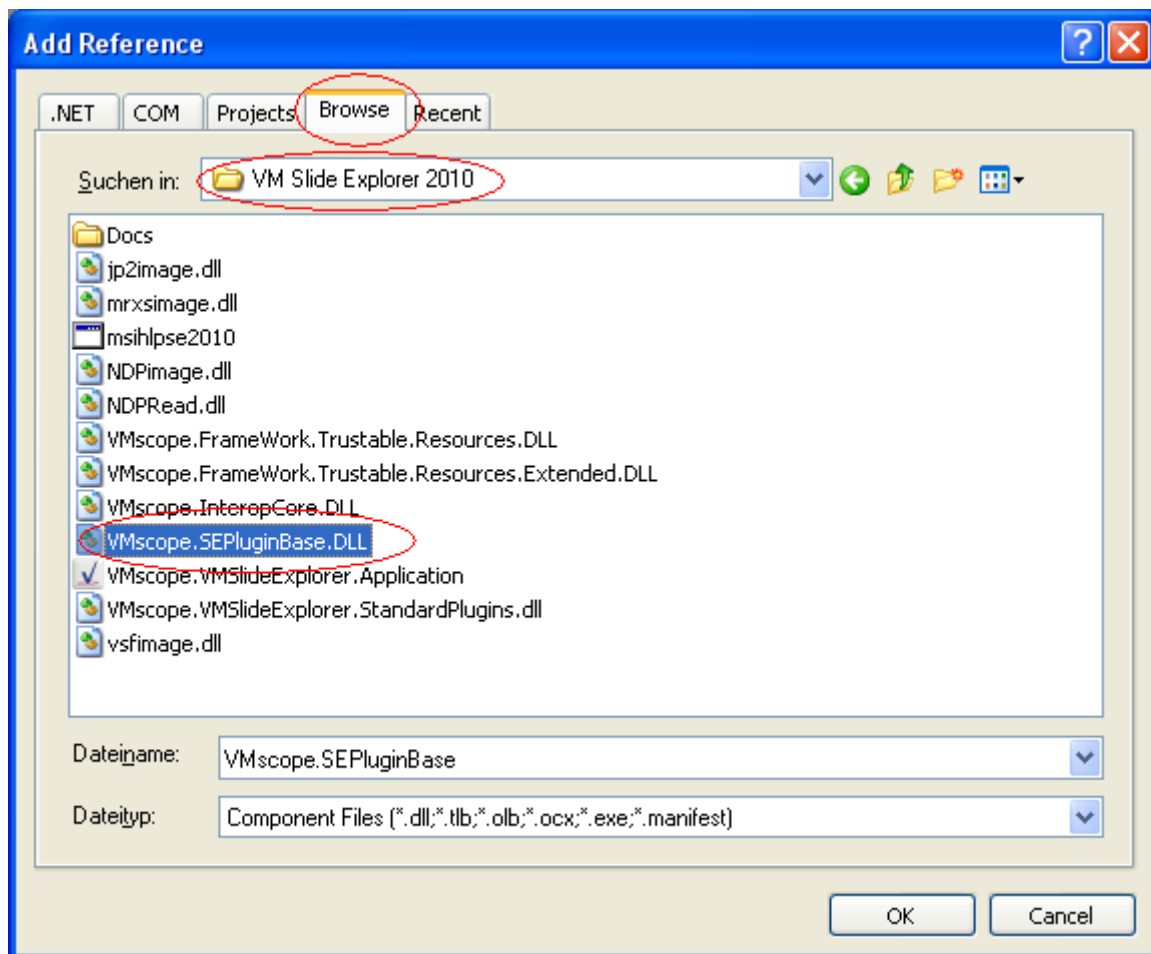




Entwicklungsumgebung Visual Studio 2008:







## Erstellen einer von SEPlugin abgeleiteten Klasse

Legen Sie nun eine neue Klasse (Sie können auch eine evtl. bereits durch das Template angelegte verändern) mit folgendem Inhalt an, die von SEPlugin abgeleitet ist.

```
using System;
using System.Windows.Forms;
using VMscope.SEPluginBase;
using VMscope.SEPluginBase.ComponentModel;

namespace HelloWorld
{
    public abstract class HelloWorld:SEPlugin
    {
    }
}
```

Beachten Sie bitte, dass die Methode Prepare implementiert werden muss, da sie ein abstraktes Member von SEPlugin ist.

```

using System;
using System.Windows.Forms;
using VMscope.SEPluginBase;
using VMscope.SEPluginBase.ComponentModel;

namespace HelloWorld
{
    public class HelloWorld:SEPlugin
    {
        protected override PluginProperties Prepare(HostProperties p)
        {
            /*** Variablen-Deklaration ***/
            PluginProperties pp=new PluginProperties();

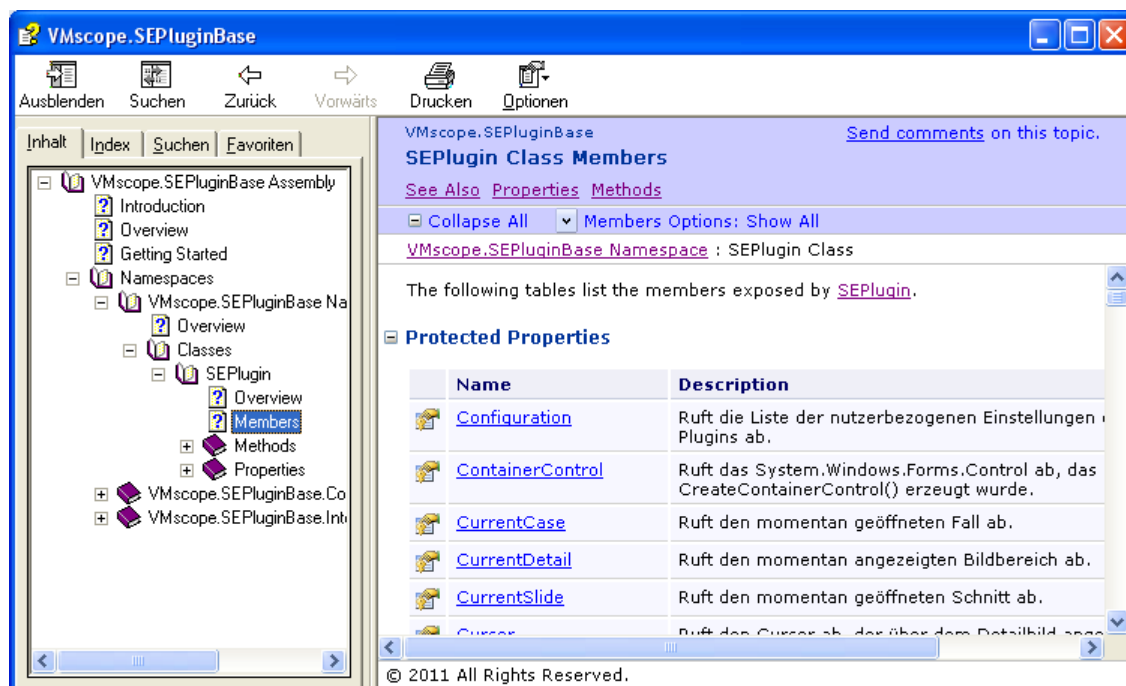
            /*** Hallo sagen ***/
            MessageBox.Show("Hello World!");

            /*** Plugin Eigenschaften zurückgeben ***/
            return pp;
        }
    }
}

```

Dieses Plugin hat als Funktionalität lediglich in einer MessageBox „Hello World“ zu sagen.

Eine genaue Beschreibung (Api-Doc) der Plugin Klasse finden Sie, nach Installation unseres SDKs unter <C:\Programme\VMscope\VMscope Platform SDK 3.0\Docs\VMscope.SEPluginBase.chm>.



VMscope.SEPluginBase

SEPlugin Class Members

See Also: [Properties](#) [Methods](#)

Collapse All  Members Options: Show All

VMscope.SEPluginBase Namespace : SEPlugin Class

The following tables list the members exposed by [SEPlugin](#).

Protected Properties	
Name	Description
<a href="#">Configuration</a>	Ruft die Liste der nutzerbezogenen Einstellungen der Plugins ab.
<a href="#">ContainerControl</a>	Ruft das System.Windows.Forms.Control ab, das CreateContainerControl() erzeugt wurde.
<a href="#">CurrentCase</a>	Ruft den momentan geöffneten Fall ab.
<a href="#">CurrentDetail</a>	Ruft den momentan angezeigten Bildbereich ab.
<a href="#">CurrentSlide</a>	Ruft den momentan geöffneten Schnitt ab.
<a href="#">Cursor</a>	Ruft den Cursor ab, der über dem Detailbild angezeigt wird.

© 2011 All Rights Reserved.

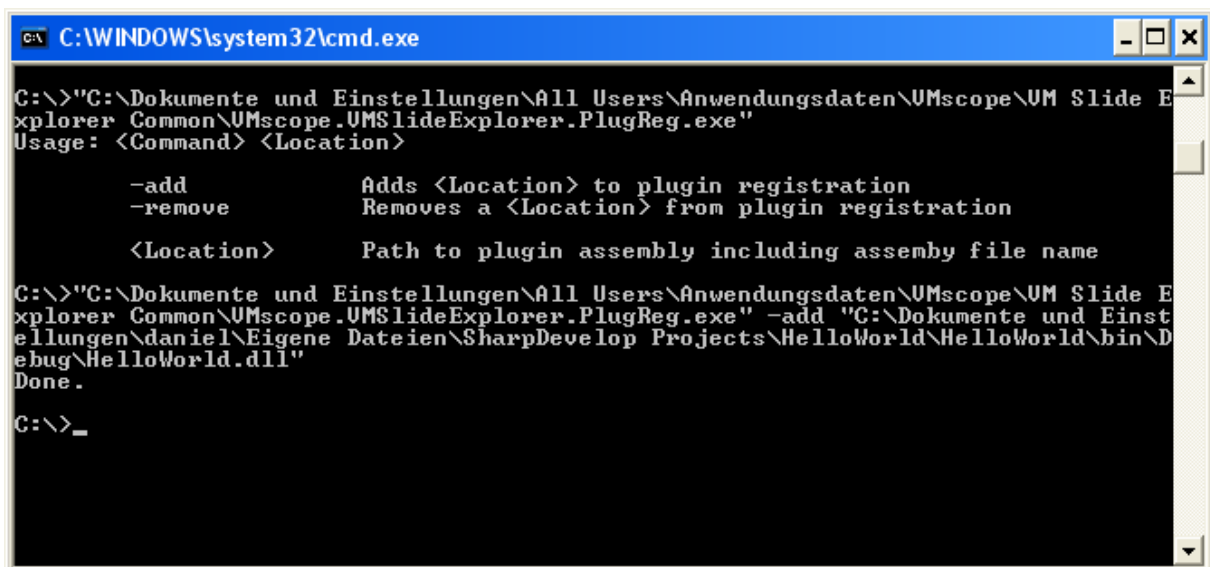
### 3 Ein Plugin registrieren

Damit ein Plugin ausgeführt werden kann, muss die erzeugte Assembly Datei einmalig beim Slide Explorer registriert werden.

Im Verzeichnis C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\VMscope\VM Slide Explorer Common (Windows XP) oder C:\ProgramData\VMscope\VM Slide Explorer Common (Windows Vista oder höher; versteckter Ordner) finden Sie die Datei „VMscope.VMSlideExplorer.PlugReg.exe“.

Diese führen sie einfach mit den Parametern -add <Pfad Ihrer Assembly> aus.

Hinweis: Mit Drag and Drop können Sie Pfadangaben auf das Kommandozeilenfenster ziehen)



```
C:\WINDOWS\system32\cmd.exe
C:\>"C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\VMscope\VM Slide Explorer Common\VMscope.VMSlideExplorer.PlugReg.exe"
Usage: <Command> <Location>

    -add          Adds <Location> to plugin registration
    -remove       Removes a <Location> from plugin registration

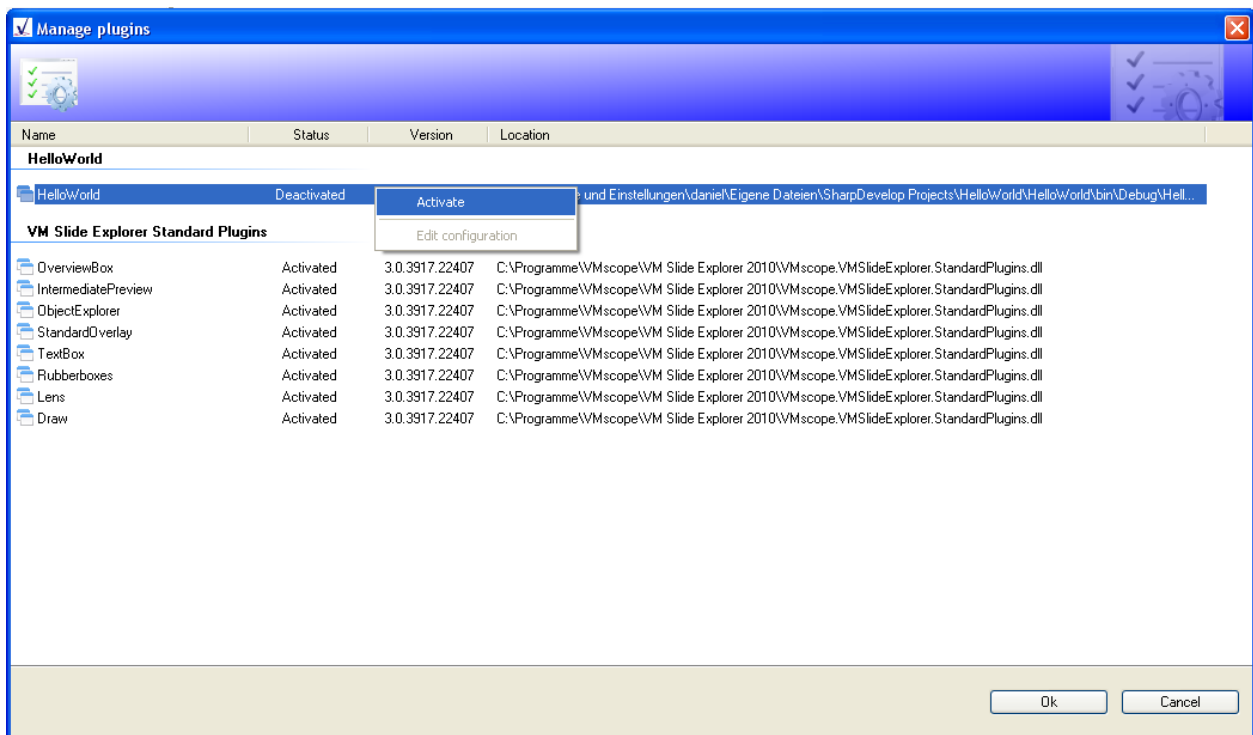
    <Location>   Path to plugin assembly including assembly file name

C:\>"C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\VMscope\VM Slide Explorer Common\VMscope.VMSlideExplorer.PlugReg.exe" -add "C:\Dokumente und Einstellungen\daniel\Eigene Dateien\SharpDevelop Projects\HelloWorld\HelloWorld\bin\Debug\HelloWorld.dll"
Done.
C:\>_
```

Damit ist das Plugin registriert und Sie können den Slide Explorer starten, um es zu aktivieren und zu testen.

## 4 Aktivieren von Plugins

Wenn Sie den Slide Explorer starten finden Sie im Menüpunkt Extras->Manage Plugins einen Eintrag für die Registrierten Plugins. Durch einen Rechtsklick auf einen Eintrag öffnet sich ein Kontextmenü durch das Sie das Plugin aktivieren können. Die Änderung ist beim nächsten Start des Slide Explorers Wirksam.



Haben Sie die Schritte 2-4 durchgeführt sollte beim nächsten Start des Slide Explorers die MessageBox erscheinen.



Glückwunsch! - Ihr erstes Slide Explorer 2010 Plugin wird erfolgreich ausgeführt.

Eine weitere Erklärung der Funktionalität der Plugin-Schnittstelle finden Sie in den Beispiel-Plugins.

## 5 Beispiel-Plugins

In der Projektmappe Tutorial2010 finden Sie zwölf Klassen, die jeweils ein Plugin für den VM Slide Explorer 2010 sind. Anhand der Quelltexte und der hier folgenden Beschreibungen können Sie nachvollziehen, welche Möglichkeiten Sie als Plugin-Schreiber haben und, wie Sie diese nutzen.

### **HelloWorld**

Sagt beim Start des Plugins Hallo Welt.

### **Configuration**

Legt einen Hauptmenü-Eintrag *Konfiguration-Plugin-Container* an durch den die Hintergrundfarbe eines Containers gesetzt wird. Diese Einstellung wird durch das Configuration-Feld des Plugins und ist so nach Neustart des Slide Explorers noch wirksam.

### **Case**

Beim Öffnen eines Falles (Sammlung mehrerer Schnitte) werden die Übersichtsbilder aller Schnitte aus dem Fall in einem Container als Treeview dargestellt.

### **Annotationssimple**

Legt einen Kontextmenüeintrag an  
Durch diesen wird eine Pfeilannotation angelegt

### **Annotations**

Legt eine Pfeil-Annotation auf jede manuell erzeugte Annotation an  
Legt eine Text-Annotation zu jeder bearbeiteten Annotation an  
Löscht automatisch alle automatisch erzeugten Annotation zu einer manuell gelöschten Annotation

## **BinaryObjects**

Über einen Kontextmenüeintrag kann der aktuelle Bildausschnitt über die BinaryObject Schnittstelle als Metainformation zum aktuellen virtuellen Schnitt gespeichert werden. Binäre Objekte werden in einem Container als Listview angezeigt und können durch Auswahl + „Entfernen“ gelöscht werden.

*Die BinaryObject-Schnittstelle nimmt es ihnen ab Metadaten zu einem Schnitt selbst speichern zu müssen.*

## **Navigation**

Startet eine automatische Navigation durch alle Zoomstufen und Annotationen sobald ein Schnitt geöffnet wird.

## **IconsAndRenderer**

Erzeugt einen Container mit Buttons. Auf den Buttons werden Symbole im Stile des Slide Explorer Renderers gezeichnet. Durch die Buttons kann die Position des aktuellen Bildausschnittes gespeichert werden. Zwischen gespeicherten Positionen kann durch weitere Buttons navigiert werden.

## **MouseInteraction**

Durch „Strg+M“ kann durch die Maus ein Auswahl-Rechteck aufgezogen werden.

## **MenuAndControl**

Legt exemplarisch Menüeinträge in allen Menüs an, in welchen dies einem Plugin möglich ist. Durch den Bereich Invertieren Eintrag kann mit der Maus ein Bereich aufgezogen werden, dessen Bilddaten invertiert dargestellt werden.

*Beachten Sie, dass durch den Slide Explorer (und dessen Plugins) niemals Bilddaten geschrieben werden. Alle Änderungen von Bilddaten sind temporär.*

## **Modus**

Überschreibt den Modus Rechteck-Zeichnen, wenn „Strg“ gedrückt wurde und invertiert den ausgewählten Bereich anstatt eine Rechteck-Annotation anzulegen.

## **RenderMark**

Zeichnet geometrische Figuren links oben in den Virtuellen Schnitt.

Ein Bereich rechts unterhalb des letzten Links-Klicks wird invertiert.

*Hier soll ein Überblick über die verschiedenen Koordinatensysteme geben werden, mit welchen sich ein Plugin-Entwickler ggf. befassen muss.*