

# !Tocs! Tutorials – RE Plugins 1.0

Dieses Tutorial ist ein Standardtutorial von RGSS Editor

## Vorwort:

Herzlich Willkommen zu !Tocs! Tutorials – RE Plugins Version 1.0 !

In diesem Tutorial werde ich alle notwendigen Grundlagen zur Erstellung eines eigenen Plugins für RGSS Editor mit Visual Basic 8.0 (Visual Studio 2005) behandeln. Dabei werde ich auf gewisse Details wie den Aufbau und die Funktionsweise eines Plugins in RGSS Editor näher eingehen. Andere, allgemeinere Aspekte zu Plugins werde ich aber nicht näher erläutern, da ich diese als bekannt voraussetzte.

Aus diesem Grund sollte jeder, der beabsichtigt ein eigenes Plugin für RGSS Editor zu erstellen, gewisse Grundkenntnisse mitbringen. Neulingen in Visual Basic .Net rate ich, sich erst einmal mit anderen, grundlegenden Projekten zu befassen und ein wenig Fachlektüre zu lesen.

Aber nun genug, fangen wir lieber an.

## Plugin?:

Bevor wir uns jedoch an ein eigenes Plugin setzen müssen wir erst noch ein wenig Theorie über uns ergehen lassen. Um zu verstehen, wie wir ein Plugin für RGSS Editor erstellen, sollten wir uns zuerst einmal klar machen, was ein Plugin ist und wie es in RGSS Editor funktioniert!

Nun, was genau ist ein Plugin?!

Diese Frage scheint für einen Laien sicherlich kompliziert, doch ist die Antwort darauf weniger kompliziert als man wohl zu Anfang denkt.

Im Grunde ist ein Plugin nichts anderes als Programmcode, der nicht ursprünglich im Programm enthalten ist und zur Laufzeit des Programms hinzugefügt wird (d.h. während das Programm läuft/ausgeführt wird). Dieser Programmcode wird normalerweise in Form einer „**Dynamic Link Library**“, sprich einer „**dll**“ gespeichert. Eingeführt wurde dieses System, um Entwicklern die Möglichkeit zu geben ein Programm auch nach der Fertigstellung zu erweitern, ohne das Ganze Programm erneut kompilieren zu müssen (DLLs haben zwar auch noch eine andere Funktion, diese ist für uns aber nicht weiter von Bedeutung). Das ist eigentlich auch schon alles, was ein Plugin ausmacht!

Als nächstes sollten wir uns nun mit der Funktionsweise eines Plugins in RGSS Editor beschäftigen.

Zunächst einmal ist es wichtig zu verstehen, dass unser Plugin in irgendeiner Art und Weise mit unserer Hauptanwendung „**kommunizieren**“ muss. In Visual Basic .Net geschieht dies mit Hilfe eines **Interfaces** welches sich in der Datei „**RE\_Pluginloader.dll**“ befindet. Dieses Interface oder diese Interfaces enthält bzw.

enthalten eine Art „**Schattensubs**“ und „**Schattenfunctions**“, welche keinerlei Programmcode enthalten.

Diese „Schattensubs“ und „Schattenfunctions“ sind sowohl in RGSS Editor enthalten als später auch in unserem Plugin. Dort jedoch nicht mehr als „Schattensubs“ und „Schattenfunction“ sondern als ganz „normale“ Subs und Functions.

Wenn wir nun in der Hauptanwendung (sprich RGSS Editor) ein Plugin aufrufen, also quasi unserem Programm den Befehl geben das Plugin zu starten, wird unser Befehl bildlich gesprochen durch das Interface geschoben und von dort aus weiter zum Plugin geleitet. Umgekehrt funktioniert dies genauso. Somit ist eine **Zweiwegekommunikation** möglich. Damit dies dann später auch problemlos funktioniert muss unser Plugin auch einen Verweis auf die Datei „**RE\_Pluginloader.dll**“ erhalten.

Zuletzt sollten wir uns noch klar machen, dass **jedes** Plugin den gleichen **Rootnamespace** besitzen muss nämlich „**clsPlugin**“. Unserer **Hauptklasse** in unserem Plugin geben wir ebenfalls **immer** diesen Namen.

Unsere Hauptklasse muss also folgendermaßen aufgebaut sein:

Abb. 1

```
Eine Beispielklasse:  
1: Public Class clsPlugin  
2: End Class
```

Da wir dies nun geklärt haben, haben wir den eigentlichen theoretischen Teil dieses Tutorials auch schon hinter uns.

War doch gar nicht so schwer oder?!

Wenn doch, dann lest euch den letzten Abschnitt noch ein- bis zweimal durch! Habt ihr es dann immer noch nicht verstanden rate ich euch dringend, vorher mit leichteren Dingen zu beginnen.

## Die Praxis...?!:

Jetzt geht es endlich los! Startet Visual Basic und... Stop! Zu früh gefreut.

Auch hier kommen wir nicht ganz ohne ein kleines Bisschen Denksport aus.

Bevor wir auch nur daran denken Visual Basic zu starten müssen wir erst mal Wissen, was unser Plugin überhaupt können bzw. machen soll!

Dazu wiederum ist es notwendig zu Wissen auf welche Programmteile uns RGSS Editor den Zugriff erlaubt. Dies sind um genau zu sein **drei** Programmteile! Klingt zunächst nach ziemlich wenig, bietet aber eine große Fülle an Möglichkeiten mit RGSS Editor zu arbeiten.

Auf Folgende drei Programmteile erlaubt RGSS Editor den Zugriff!

1. Das Hauptfenster von RGSS Editor
2. Die Childforms (Skriptformulare) von RGSS Editor
3. Die mit den Childforms verbundenen Codeboxen (der Bereich, in dem man den Skriptcode eingibt)

In unserem doch sehr einfachen Beispielplugin wollen wir lediglich, dass unser Plugin die Zeilen des aktuellen Skriptformulars bzw. der aktuellen Codebox zählt und diese Zahl danach in die letzte Zeile des Skriptes schreibt.

## Das Beispielplugin:

Jetzt aber genug! Startet nun endlich Visual Basic (wenn ihr es nicht vor lauter Ungeduld und Vorfreude bereits getan habt) und startet ein neues „Class Library“ Projekt. Stellt nun als erstes den Rootnamespace und den Klassennamen der Hauptklasse so ein, wie oben bereits beschrieben. Danach fügt noch einen Verweis auf die Datei „**RE\_Pluginloader.dll**“ hinzu.

Jetzt müssen wir unsere Klasse noch für die weitere Arbeit mit RGSS Editor vorbereiten. Fügt dazu einfach folgenden Code **anstatt** des bereits vorhandenen Codes in unsere Hauptklasse ein. Er sollte sich eigentlich mit Hilfe der Anmerkungen selbst erklären.

Abb. 2

```
Das PlugIn
1: <Serializable(> Public Class clsPlugin 'Plugin zur Erstellung während
2:                               'Der Laufzeit vorbereiten
3:     Inherits MarshalByRefObject ' Plugin zum Kapseln vorbereiten
4:
5:     'Das Interface implementieren
6:     Implements clsRE_Pluginloader.clsRE_Pluginloader.i_Plugin
7:
8: End Class
```

Nun müssen wir noch eine Variable für den später Folgenden Link zur Hauptanwendung festlegen! Dazu fügen wir einfach die folgende Variablendeklaration über dem „**End Class**“ Statement hinzu:

Abb. 3

```
Die Hauptanwendung
1: Dim m_Host As clsRE_Pluginloader.clsRE_Pluginloader.i_Host
```

Im nächsten Schritt müssen wir den Gültigkeitsbereich des Plugins festlegen. Normalerweise ist dieser unendlich. Bei unserem Plugin ist der gültigkeitsbereich ebenfalls unendlich, weshalb es genügt folgenden Code unter der letzten Variablendeklaration einzufügen:

Abb. 4

```
Der Gültigkeitsbereich
1: 'Gültigkeitsbereich festlegen
2: Public Overrides Function InitializeLifeTimeService() As Object
3:     Return Nothing
4: End Function
```

Nachdem wir dies alles erledigt haben, ist die komplizierteste Arbeit bereits geleistet. Wir haben zwar noch einiges an Arbeit zu erledigen aber wie ich sagte ist das komplizierteste bereits vollbracht. Die restliche Arbeit besteht nun nur noch aus ganz „normalen“ Subs und Functions.

Es folgt nun ein ganzer Batzen Code! Lasst euch aber nicht davon abschrecken! Ich habe wie immer alles mit Kommentaren versehen und das Meiste dürfte sich somit von selbst erklären.

Zu erwähnen ist noch, dass sich die Namen der Subs und Functions ruhig von den hier aufgeführten unterscheiden können. Die jeweiligen „Implements“ Statements dahinter müssen aber in jedem Fall die gleichen sein.

Bevor ihr diesen Codeblock aber einfügt solltet ihr noch einen Verweis auf folgende Datei hinzufügen: „**mkc\_CodeBox (Edited by !Tocs!).dll**“. Sie befindet sich im Hauptverzeichnis von RGSS Editor.

Fügt nun den Code unter dem letzten „End Function“ Statement ein.

Abb. 5

#### Subs und Functions

```
1: 'Linkzur Hauptanwendung setzen
2: Public Sub SetHost(ByVal mHost As _
3: clsRE_Pluginloader.clsRE_Pluginloader.i_Host) Implements _
4: clsRE_Pluginloader.clsRE_Pluginloader.i_Plugin.SetHost
5: 'Dieser Sub setzt einen Link zur Hauptanwendung
6:     m_Host = mHost'
7:
8: End Sub
9:
10:
11: 'Name des Plugins ausgeben
12: Public Function GetName()As String Implements _
13: clsRE_Pluginloader.clsRE_Pluginloader.i_Plugin.GetPIName
14:     'Dieser Sub gibt den späteren Namen des Plugins aus
15:     Return "MeinPlugIn"
16: End Function
17:
18:
19: 'Informationenüber das Plugin ausgeben
20: Public Function GetInfo()As String Implements _
21: clsRE_Pluginloader.clsRE_Pluginloader.i_Plugin.GetInfo
22:     ' Die einzelnen Informationen werden durch diese Zeichenfolge
23:     '<|> von einander getrennt
24:     Return "MeinPlugIn<|>1.0.0<|>Autor<|>01.01.06<|>Das ist die" & _
25:         "Beschreibung"
26: End Function
27:
28:
29: 'Dieser Sub wird beim Laden von RGSS Editor ausgeführt
30: Public Sub Load()Implements _
31: clsRE_Pluginloader.clsRE_Pluginloader.i_Plugin.Load
32: 'Da wir aber beim Start von RGSS Editor nichts ausführen möchten
33: 'lassen wir ihn einfach leer! Löschen dürfen wir ihn allerdings nicht.
34: End Sub
35:
36: 'Dieser Sub wird beim Beenden von RGSS Editor ausgeführt
37: Public Sub Unload()Implements _
38: clsRE_Pluginloader.clsRE_Pluginloader.i_Plugin.Unload
39: 'Da wir aber beim Beenden von RGSS Editor nichts ausführen möchten
40: 'lassen wir ihn einfach leer! Löschen dürfen wir ihn allerdings nicht.
41: End Sub
42:
43: 'Dieser Sub wird ausgeführt, wenn das Plugin gestartet wird
44: Public Sub Run()Implements _
45: clsRE_Pluginloader.clsRE_Pluginloader.i_Plugin.Run
46: 'Hier greifen wir nun auf das mkc_Codebox Objekt oder auch
47: 'Childcontrol oder auch Codebox Objekt zurück! Wir zählen hier
48: 'also die Zeilen der Codebox und fügen sie dem Skript hinzu.
49: M_Host.ChildControl.Text += vbCrLf & m_Host.ChildControl.LineCount
50: End Sub
```

Wenn ihr bis hierhin alle Schritte richtig befolgt habt, haben wir unser erstes Plugin erfolgreich erstellt! Kompiliert die DLL und kopiert sie in den „Plugins“ Ordner von RGSS Editor. Es werden sich noch einige andere Dateien und DLLs im Verzeichnis ihrer kompilierten DLL befinden, wichtig ist aber, dass nur unser Plugin in den „Plugins“ Ordner von RGSS Editor kopiert werden muss. Startet jetzt RGSS Editor und überprüft, ob alles Ordnungsgemäß funktioniert.

### **Schlusswort:**

Wie wir gesehen haben ist es (hat man die Theorie erst einmal verstanden) nicht sonderlich schwer ein Plugin für RGSS Editor in Visual Basic zu erstellen. Wenn ihr an einigen Stellen Probleme hattet empfehle ich euch wie immer, den Betroffenen Abschnitt noch einmal zu lesen.

Wenn ihr noch Fragen oder Anmerkungen habt, bin ich im Thread zu RGSS Editor oder per PN des Forums von [www.rpgxp.de](http://www.rpgxp.de) zu erreichen.

Ich wünsche euch noch viel Spaß und Erfolg!  
Bis zum nächsten Tutorial

!Tocs!