



NRW Conf 2013 | Das Community Event | 10.-11.10.2013 | Wuppertal

TOUCH TECHNOLOGY

WWW.NRWCONF.DE



SCRIPTCS

C# als Skriptsprache!?



Was ist scriptcs?

Vollständige *Engine* für C#-Code



Wer hat's erfunden?

Glenn Block (@gblock)

Justin Rusbatch (@jrusbatch)

Filip Wojcieszyn (@filip_woj)



Was will scriptcs?

C# mit einem *Texteditor* schreiben & ausführen



Ziele & Werte?

Öffentliche *Werte*, *Anwendungsfälle* & *Akzeptanzkriterien*



Werte

Einfachheit, Agilität, schnelle Entwicklung



Anwendungsfälle

Rapid Prototyping, Anwendungen, Erweiterungen, Skripte



Akzeptanzkriterien

Kein Handbuch, keine Abhängigkeiten, *lose Skripte*, *Pakete*


```
1 using System.IO;
2
3 // The dssp source file.
4 var file = @"DSSP\1a6e.dssp";
5 // The output file.
6 var output = "output.txt";
7
8 // Column positions
9 var aminoAcidsStart = 13;
10 var aminoAcidsLength = 1;
11 var structureStart = 16;
12 var structureLength = 1;
13
14 // Read the whole file and skip the first 28 lines.
15 var lines = File.ReadAllLines(file);
16
17 var aminoAcids = new StringBuilder("AA:");
18 var structures = new StringBuilder("S :");
19
20 for (var i = 28; i < lines.Length; i++)
21 {
22     aminoAcids.Append(" " + lines[i].Substring(aminoAcidsStart, aminoAcidsLength));
23     structures.Append(" " + lines[i].Substring(structureStart, structureLength));
24 }
25
26 // Write the sequences to an output file.
27 using (var writer = new StreamWriter(output))
28 {
29     writer.WriteLine(aminoAcids.ToString());
30     writer.WriteLine(structures.ToString());
31 }
```



Ein paar kurze Fakten

Open Source (*GitHub*), sehr aktiv, sehr beliebt



Technische Basis

Basiert hauptsächlich auf *Roslyn* und *NuGet*

The diagram consists of three stacked rectangular boxes. The top box is dark purple and contains the text 'scriptcs' in white. Below it are two boxes: a light blue one on the left containing 'NuGet' and an orange one on the right containing 'Roslyn'. This visualizes 'scriptcs' as a higher-level layer that depends on both 'NuGet' and 'Roslyn'.

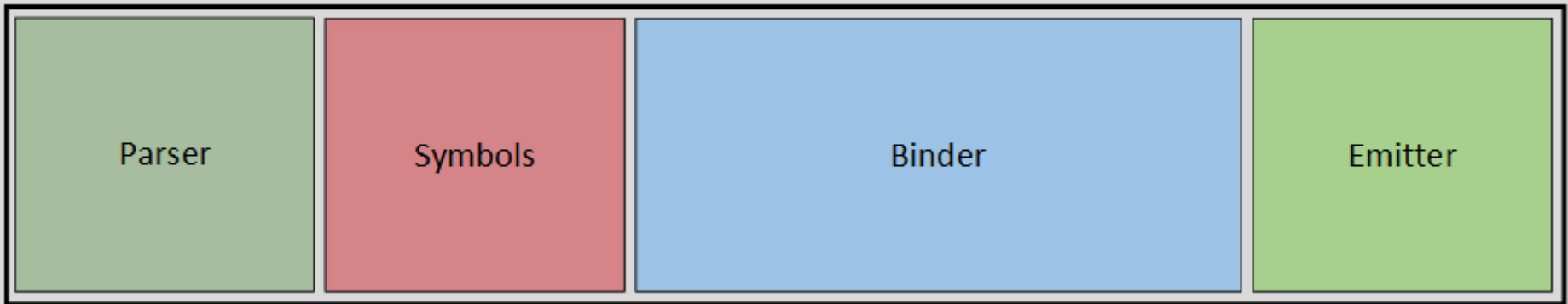
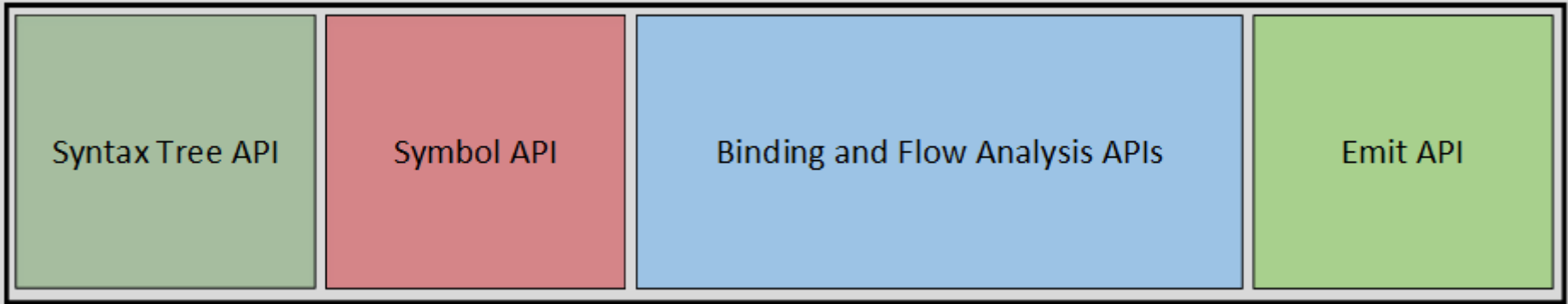
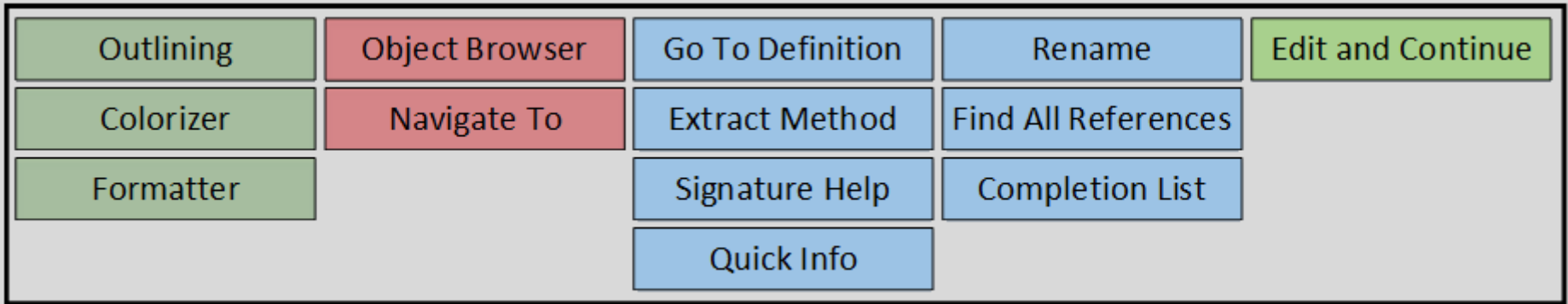
scriptcs

NuGet

Roslyn

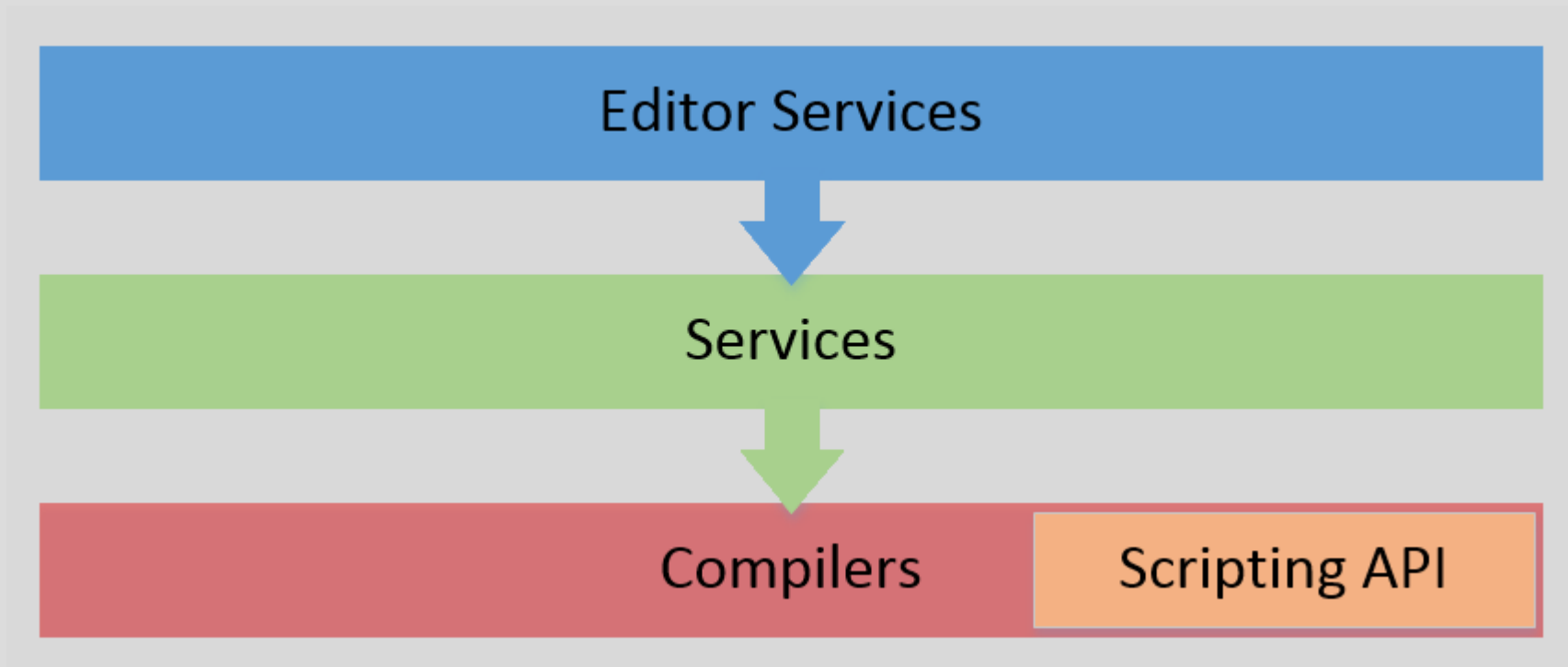
Roslyn?

C#- und VB.NET-Compiler als *Dienste*



Und was genau?

Ein bestimmter Aspekt von *Roslyn*



NuGet?

Paketverwaltung für die Microsoft-Plattform & .NET

Installation

Zwei Varianten. Eine schnell, die andere schneller

Variante 1

Quellen von *GitHub* besorgen
Die Datei *build.cmd* ausführen

Variante 2

Chocolatey installieren

Den Befehl *cinst scriptcs* ausführen

```
@powershell
```

```
-NoProfile
```

```
-ExecutionPolicy Unrestricted
```

```
-Command "iex ((New-Object Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))"  
&& SET PATH=%PATH%;%systemdrive%\chocolatey\bin
```



Alles startklar?

Auf die Plätze, fertig, *los!*



Erste Gehversuche mit REPL

Read-Eval-Print-Loop. Schnell, unkompliziert, für zwischendurch



REPL-Features

DEMO



REPL-Features

Direkte Ausführung, *multiline*, nette Gimmicks



C# Skriptdateien

REPL ist schön, aber schnell *zu eingeschränkt*



C# Skriptdateien

Datei auf, relevanter *Code rein*, speichern, *fertig*



Editoren?

Vollkommen egal! Ich bevorzuge *Sublime Text 2*



C# Skriptdateien

Keine Namespaces, keine Klassen, *nur wichtige Dinge*



Features

#load, #r, Parameter, *Standards*, Pakete



C# Skriptdateien

DEMO

C# Skriptdateien

```
1 using System.IO;
2
3 var fileName = ScriptArgs[0];
4 var data = ScriptArgs[1];
5
6 Console.WriteLine("Start writing!");
7
8 using (StreamWriter sw = new StreamWriter(fileName, true))
9 {
10     sw.WriteLine(data);
11 }
12
13 Console.WriteLine("Finished writing!");
```



Standards

- System
- System.Core
- System.Data
- System.Data.DataSetExtensions
- System.Xml
- System.Xml.Linq
- System
- System.Collections.Generic
- System.Linq
- System.Text
- System.Threading.Tasks



Pakete

scriptcs *Script Packs* & „normale“ *Pakete*



Pakete

Verwaltung läuft über *NuGet*
scriptcs -install <Paketname>



Script Packs

Simplere Anwendung, *Automatismen*, Factory-Methode



Weitere Pakete

Identische Installation, „normale“ NuGet-Pakete

Pakete

DEMO

Debugging

Auch bei Skriptdateien tauchen *Fehler* auf...

Debugging

Läuft über *Visual Studio*

Debugging

DEMO



Was können wir?

Installation, *REPL*, *Skripte*, Referenzen, Pakete, Debugging



That's it!

Wir kennen *scriptcs*



Anwendungsbeispiele

Und was macht man jetzt damit?



PDF-Datei manipulieren

Ich brauche Seite **2** und **3**



Konfigurationen

Können auch als *C# Skriptdateien* abgelegt werden



scriptcs ebnet den Weg

Die *Anwendungsszenarien* kommen von uns!



Zukunft

Export als *exe-Datei* & *Visual Studio Projekt*



Vielen Dank für die Aufmerksamkeit!



Fabian Deitelhoff

Informatikstudent Master of Science
Schwerpunkt Biomedizinische Informatik

Autor, Speaker, Trainer, Softwareentwickler



Kontaktinformationen

E-Mail: Fabian@FabianDeitelhoff.de

Twitter: @FDeitelhoff

Blog: <http://www.fabiandeitelhoff.de>