

Source Code Konverter

Thomas Reinwart

Welcher Entwickler stand noch nicht vor dem Problem, den Sourcecode einer Entwicklungssprache in eine andere zu übernehmen?

Die Gründe dafür können vielseitig sein:

- Organisatorisch bedingt: es werden mehrere Entwicklungssprachen parallel eingesetzt.
- Historisch bedingte Relikte
- Keine ausreichenden Kenntnisse des Syntax einer Sprache
- Sample Code nur in einer Sprache verfügbar
- Der Sourcecode muss debuggt werden, dabei durchgängig in einer Sprache

Durch die .net runtime von Microsoft ist ein konvertieren des Sourcecodes wesentlich einfacher geworden. Entsprechende Tools gibt es am Markt.

Man kann allerdings nicht immer davon ausgehen, dass es „sauber“ ist, ein Program 1 zu 1 zu konvertieren und damit weiter zu arbeiten. Jede Entwicklungssprache hat seine Eigenheiten, die in einer Sprache anderes oder optimaler gelöst werden können. Manche Code Konverter markieren die relevanten Stellen im Code mit Kommentaren, ein Nachbearbeiten ist also auf jeden Fall notwendig.

Beispiele dieser Stellen sind der eigene VB Namespace in der .net runtime, der zwar in einer C# Umgebung auch eingebunden werden kann, aber sehr unschön ist, da selbige Funktionen in der .net runtime ohnehin (unter anderem Namen und Parameter) abgedeckt sind. (Den VB Namespace in der .net runtime enthält viele gleichnamige VB 6.0 Methoden)

Oder auch Arrays, die in einer Sprache mit dem Index 0, in einer anderen mit dem Index 1 beginnen.

Eine Durchsicht des konvertierten Codes macht also auf jeden Fall Sinn, besonders dann, wenn der Code nicht von einem selber stammt. Hilfreich sind solche Code Konverter für einzelne Klassen oder Bereiche. Bei einer kompletten Anwendung würde ich das Grundgerüst in der neuen Sprache erstellen, und einzelne konvertierte Klassen Schritt für Schritt übernehmen.

Bei den Code Konvertern selber lassen es manche zu, eine .net Solution bzw. das Projekt File anzugeben und damit alles inkludiert zu konvertieren, anderen beschränken sich auf einzelne Files bzw. Klassen.

Wenn bei der Konvertierung folgende Punkte beachtet werden, wird sichergestellt, dass eine Anwendung am Ende zumindest gleichwertig wenn nicht besser wird:

- Das Grundgerüst entspricht der aktuelle Vorgabe (Template und .net runtime) eines Projektes
- Durch die Codedurchsicht wird am Ende nochmals überprüft. (z.B. Objektorientiertheit)
- Markierte Codestellen werden nachbearbeitet
- Testen der übernommenen Klassen mit Unit Tests
- Code refactoring

- Gedankenlose 1 zu 1 Übernahme ausschließen

Online: (VB.net <-> C#)

<http://www.carlosag.net/Tools/CodeTranslator/Default.aspx>

<http://labs.developerfusion.co.uk/convert/csharp-to-vb.aspx>

<http://codeconverter.sharpdevelop.net/Convert.aspx>

<http://converter.telerik.com/>

Kommerzielle Produkte (VB, C#, C++, Java)

<http://tangiblesoftware.com/>

- Instant C# converts VB code to C#
- Instant VB converts C# code to VB
- C++ to C# Converter converts C++ code to C#
- C++ to VB Converter converts C++ code to VB
- C++ to Java Converter converts C++ code to Java
- Instant C++ (C# Edition) converts C# code to C++/CLI
- Instant C++ (VB Edition) converts VB code to C++/CLI
- Instant C++ (Java Edition) converts Java code to C++/CLI
- Java to VB & C# Converter converts Java code to VB or C#
- Clear VB formats and standardizes VB.NET source code

Code Nachbearbeitung in der Praxis

Je nach Komplexität und Umfang des Projekts gibt es mit Code Konvertern unterschiedliche Ergebnisse, also mehr oder weniger gut übersetzte Stellen, die manuell nachbearbeitet werden müssen.

Anbei ein Beispiel einer Code Konvertierung (VB.net -> C#) mit der „Tangible Software Solutions“ Anwendung, bei dem die Warnings nach der erfolgreichen Konvertierung sowohl als Zusammenfassung im Tool selber dargestellt als auch im Sourcecode als Kommentar eingefügt werden: (Bild unten)

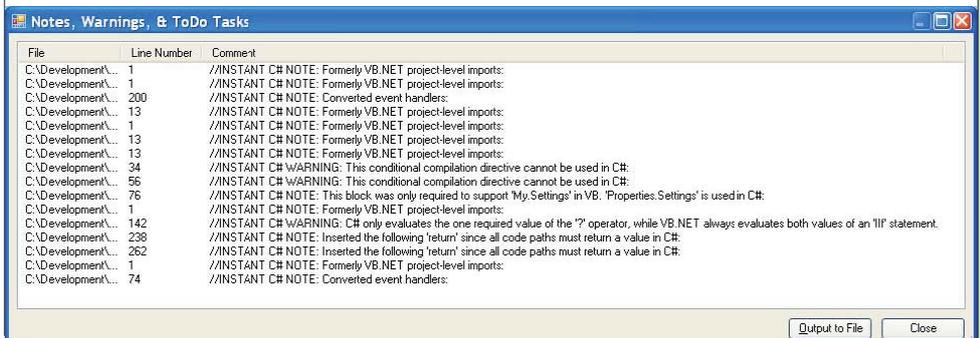
Der nächste Schritt für den Entwickler ist es, diese Stellen zu überarbeiten. Der direkte Codevergleich zwischen VB.net und dem neu erstellten C# ist zwecks Übersichtlichkeit möglich. Dabei wird alter (VB.net) und neu Code (C#) als Gegenüberstellung in zwei Fenstern dargestellt.

Hier einige häufig auftretende Beispiele:

1) Form Designer File

Das Form Designer File muss nachbearbeitet werden, das Attribut von VB muss entfernt werden.

```
namespace Sample {
    [Microsoft.VisualBasic.CompilerServices.DesignerGenerated()]
```



```
public partial class Report :
System.Windows.Forms.Form
{
```

2) Event Handler

```
//INSTANT C# NOTE: Converted event handlers:
this.KeyUp += new
System.Windows.Forms.KeyEventHandler (ReportView
er_KeyUp); ReportViewer.Paint += new
System.Windows.Forms.PaintEventHandler (ReportVi
ewer_Paint); this.Paint += new
System.Windows.Forms.PaintEventHandler (Report_P
aint);
```

Hier wurde der Code automatisch ergänzt, das Handling hinter den neuen Events muss nachbearbeitet werden.

3) MySettings/Properties.Settings

```
//INSTANT C# NOTE: This block was only required
to support 'My.Settings' in VB.
'Properties.Settings' is used in C#:
```

'My Settings' wird in VB.net verwendet, um eigene Einstellungen der Anwendung zu speichern. Im Kommentar steht, was zu tun ist.

4) VB.net Functions – C# Methoden/Properties

Wenn in VB.net Funktionen verwendet wurden, muss die übersetzte C# Methode auf Rückgabewerte hin überprüft werden. Eine ausgedachte Businesslogik des Entwicklers bezüglich Rückgabewerte kann natürlich kein Konverter erkennen.

```
//INSTANT C# NOTE: Inserted the following
'return' since all code paths must return a
value in C#:
return false; // Dieser Wert wurde automatisch
eingefügt.
```

Ein Vergleich der Syntaxunterschiede zwischen VB.net / C# und Java / C# ist hier zu finden und hilft, manche Details der Sprachunterschiede besser zu verstehen.

Java (J2SE 5.0) and C# Comparison

http://www.harding.edu/fmccown/java1_5_csharp_comparison.html

VB.NET and C# Comparison

http://www.harding.edu/fmccown/vbnet_csharp_comparison.html

Zusammenfassung Code Konvertierung

- Nützlich in der Praxis
- Spart Zeit, damit wird das Rad nicht nochmals neu erfunden
- Nachbearbeitung sinnvoll bzw. notwendig (je nach Komplexität und Zustand des Codes)
- Code Konverter sind nicht dazu da, Code Richtlinien oder Formatierung (Lesbarkeit) zu verbessern oder einzuführen. Ein schlechter, unlesbarer Spagetti- oder fehlerhafter Code wird also nicht von alleine besser.