



.NET - Framework und Mono

.NET - Framework und Mono

Plattformunabhängige Softwareentwicklung mit .Net - Framework / Mono

Wenn Entwickler mit einem neuen Software-Projekt beginnen, ist eine der ersten zu stellenden Fragen: „Wer ist eigentlich der Anwender dieser Software?“. Immer häufiger sind diese Anwender keine homogene Gruppe, sondern Benutzer, die mit unterschiedlichsten Systemen arbeiten. Daher sollen neue Softwarelösungen möglichst plattformunabhängig, also z.B. auf Windows und Linux laufen.

Die Erfahrung hat aber gezeigt, dass es enorm viel Arbeit macht, die jeweiligen Anwendungen auf andere Systeme und Plattformen zu portieren. Und wie in den meisten Fällen heißt es auch hier: viel Entwicklungsarbeit = hohe Kosten.

Dementsprechend definieren Softwareentwickler die Plattformunabhängigkeit als eines ihrer prioritären Ziele. Zur Realisierung dieser Ziele ist die Kombination von .NET - Framework und Mono genau die richtige Laufzeit- und Entwicklungsumgebung.

Was ist das „.NET - Framework“?

In den 90er Jahren stellte die Firma Sun Microsystems mit Java eine neue Sprache und Technologie zugleich vor. Java sollte es den Entwicklern ermöglichen, plattformunabhängige Anwendungen, damals vorwiegend für das Internet, zu entwickeln. Diese Idee und Technologie wurde zu einem großen Erfolg bei Entwickeln und auch bei den Anwendern.

Microsoft hatte mit solch einem Erfolg von Java nicht gerechnet (und somit kein ebenbürtiges Produkt im Angebot) und begann erst als Resultat des Java-Erfolges, hinter verschlossenen Türen an einer ähnlichen Technologie zu arbeiten. Das Resultat wurde 2002 in Form von .NET - Framework (gesprochen: „dot NET“) vorgestellt. Mit der Entwicklung des .NET - Framework verfolgte Microsoft zum einen die Absicht, einen Gegenpol zu Suns Java zu schaffen. Zum anderen soll wenn es nach Microsoft ginge, das .NET - Framework in den nächsten Jahren die Basis einer jeden Windowsanwendung bilden.



Autor:

Klaus Greisen (Dipl.-Ing.)
Leiter Softwareentwicklung

„Die Kombination aus .NET - Framework und Mono bietet eine zukunftsorientierte Basis für flexible Softwareentwicklungen und Plattformunabhängigkeit Ihrer Anwendungen“

In der von Microsoft eigens für .NET entwickelten Sprache C# (gesprochen: „C sharp“) spiegeln sich nicht nur Einflüsse von C++ und Java wieder, sondern man hat auch die besten Features anderer Programmiersprachen integriert. Das Entscheidende ist jedoch, dass Microsoft C# zur Standardisierung bei der European Computer Manufacturers Association (ECMA) und bei der International Organisation for Standardisation (ISO) eingereicht hat, wodurch es möglich wurde, eine Implementierung von C# für andere Softwareplattformen zu bewerkstelligen.

Woraus besteht das .NET - Framework?

Zur Beschreibung des .NET - Framework sind vier wesentliche Komponenten zu betrachten, die untenstehend näher erläutert werden:

- Common Type System
- Common Language Specification
- Common Language Runtime/Infrastructure
- Compiler

Common Type System

Um Sprachunabhängigkeit zu gewährleisten, ist eine gemeinsame Basis erforderlich, die definiert, wie mit Datentypen umgegangen wird und welche Datentypen überhaupt zur Verfügung stehen. Diese Basis wird gemeinhin als Common Type System (CTS) bezeichnet. Durch diese gemeinsame Basis greifen alle .NET - Programmiersprachen intern auf dieselben Datentypen zu.

Common Language Specification

Das Ziel der Sprachunabhängigkeit wirft das Problem auf, dass nicht alle Sprachen alle Datentypen unterstützen oder dass manche Sprachen case-sensitive sind und andere nicht. Um dieses Problem zu lösen, definiert die Common Language Specification (CLS) quasi einen kleinsten gemeinsamen Nenner, den alle .NET - Sprachen erfüllen müssen.

Common Language Runtime/Infrastructure

Die Common Language Runtime (CLR) ist zusammen mit der Common Language Infrastructure (CLI) die Laufzeitumgebung, unter der alle in den .NET - Sprachen entwickelten Programme ausgeführt werden. Ein großer Vorteil dieser als „managed“ bezeichneten Programmausführungsumgebung ist die automatische Speicherverwaltung (garbage collection). Bei der automatischen Speicherverwaltung wird innerhalb der Laufzeitumgebung Speicherplatz dynamisch angefordert und automatisch wieder freigegeben. Somit ergibt sich ein großer Produktivitätsgewinn während der Programmentwicklung, da sich der Entwickler nicht mit dem Anfordern und Freigeben von Speicher auseinandersetzen muss. Die automatische Speicherverwaltung ist zwar marginal langsamer als die manuelle, aber in Zeiten hocheffektiver Prozessoren ist die kürzere Entwicklungszeit wesentlicher als eine um Millisekunden schnellere Ausführungszeit.

Compiler

Der Sourcecode jeder .NET-Sprache wird in den plattformunabhängigen Intermediat-Language-Code (IL-Code) überführt. Erst zur Laufzeit (also bei Aufruf des Programms) wird dieser Code in einen plattformspezifischen Code kompiliert. Dies übernimmt der Just-In-Time (JIT) Compiler des CLR, wodurch es beim Programmstart zu kürzeren Verzögerungen kommen kann. Um diese Startverzögerungen zu vermeiden, kann der IL-Code eines Programms durch den Ahead-Of-Time (AOT) Compiler in den Maschinen-Code der betreffenden Plattform übersetzt werden. Dieses Compilerergebnis ist dann verzögerungsfrei ausführbar, aber nicht mehr plattformunabhängig.

Was ist Mono?

Mono, das spanische Wort für Affe, ist ein von Miguel de Icaza initiiertes Open Source Projekt, welches von der Firma Novell gesponsert wird. Ziel von Mono ist es, das .NET – Framework mit all seinen Komponenten unter einer offenen Lizenz nachzubilden, um somit Plattform- und Betriebssystemgrenzen zu überwinden.

Bestandteil von Mono sind auf ECMA-Standards basierende Runtime-, Compiler- und Entwicklungswerkzeuge für die verschiedensten Programmiersprachen und eine umfangreiche Klassenbibliothek.

Mono unterstützt auch Features, die über die ECMA-Spezifikationen hinausgehen, z.B. können Anwendungen basierend auf ASP.NET und ADO.NET oder Windows-Forms auch unter Mono entwickelt und ausgeführt werden.

Hauptsprache unter Mono und .NET ist die von Microsoft entwickelte Sprache C#. Diese gemeinsame Programmiersprache ist der Kern der Plattformunabhängigkeit. Dabei deckt das ursprünglich für Linux entwickelte Mono, alle Linux-Plattformen ab, während .NET die Windows-Plattform bedient. Seit nicht allzu langer Zeit ist die Mono-Implementierung auch für die Windows-Plattform verfügbar. So ist es möglich, unter .NET entwickelte Anwendungen direkt unter Windows auf die Kompatibilität mit Mono zu testen.

Plattformunabhängige Softwareentwicklung macht ihre Anwendung flexibler und damit kostengünstiger

Die plattformunabhängige Laufzeitumgebung ermöglicht es Entwicklern, Lösungen zu finden, die in verschiedenen Systemen einsetzbar und dadurch kostenoptimiert sind. Da Novell und Microsoft gegenseitig auf die Geltendmachung von Patentansprüchen verzichten, ist der zukünftige Bestand und die Weiterentwicklung von Mono gewährleistet.

Das Augenmerk der heutigen Softwareentwicklung liegt u.a. auf wieder verwendbaren, portablen Codes, auf Plattformunabhängigkeit und kurzen Entwicklungszeiten. All dies bietet dem Entwickler die Kombination aus .NET - Framework, Mono und C#.

Bei dataschalt ist diese Entwicklungsumgebung eines der Werkzeuge, mit denen wir Ihre Anwendungen und Projekte realisieren. Sprechen Sie uns an, für eine flexible Lösung ihrer Vorstellungen.

dataschalt 

Klaus Greisen | Leiter Softwareentwicklung
An der Hülshorst 7 - 9 | D-23568 Lübeck
Phone: +49 451 29059-29 | Fax : +49 451 38812-69
eMail: klaus.greisen@dataschalt.com