

dataschalt scientific paper



.NET - Framework and Mono

.NET - Framework and Mono

Platform independent software development with .Net - Framework / Mono

When developers begin a new software project, one of the first questions asked is: “Who are the actual users of this software?” Increasingly, these users are not a homogeneous group, but users with different systems. Therefore, new software solutions should be as platform independent as possible, e.g. run on Windows and Linux.

Experience has shown that it takes a tremendous amount of work to port applications on different systems and platforms. And as in most cases: a lot of development work = high costs.

Accordingly, software developers define platform independence as one of their priority objectives. The combination of .NET - Framework and Mono offers just the right runtime and development environment to realize this objective.

What is this „.NET - Framework“?

In the 90's, with Java, Sun Microsystems introduced a new language and technology at the same time. Java was meant to enable developers to develop platform independent applications, then mainly for the Internet. This idea and technology has become a big success in developing and also among users.

Microsoft had not expected Java to be such a success (and thus had no comparable product to offer) and as a result, began work on a similar technology behind closed doors. The result was introduced in 2002 in the form of .NET - Framework (pronounced “dot NET). With the development of the .NET - Framework, it was Microsoft's intention to create a competitive alternative to Sun's Java. Also, if it were up to Microsoft, the .NET – Framework would form the basis of every single Windows application in the years to come.

The Microsoft language C# (pronounced “C sharp”), developed specifically for .NET, reflects



Author:

Klaus Greisen (Dipl.-Ing.)

Head of Software Development

„The combination of .NET framework and Mono offers a forward looking basis for flexible software development and platform independence“

not only influences of C++ and Java again, but it also integrates the best features of other languages. The crucial point is that Microsoft C# became standardized by the European Computer Manufacturers Association (ECMA) and the International Organisation for Standardisation (ISO), making it possible to implement C# for other software platforms.

What does the .NET – Framework consist of?

The following four main components describe the .NET - Framework:

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

Common Type System

In order to ensure language independence, it is necessary to have a common base which defines how data types are handled and what types of data are even available. This base is commonly known as Common Type System (CTS). All .NET - programming languages can access the same data types through this common base.

Common Language Specification

The goal of language independence has the problem that not all languages support all data types or that some languages are case-sensitive and others are not. In order to solve this problem, the Common Language Specification (CLS) defines a kind of lowest common denominator which all .NET - languages must meet.

Common Language Runtime / Infrastructure

The common language runtime (CLR), together with the Common Language Infrastructure (CLI), form the runtime environment, in which all in the .NET - languages developed programs are carried out. One major advantage of this so called “managed” program execution environment is the automatic memory management (garbage collection). With automatic memory management, disk space is dynamically and automatically released within the runtime environment. The result is large productivity gains during program development, as the developers do not have to deal with requirements for and clearing of memory. Automatic memory management is marginally slower than manual, but in times of high-speed processors, shorter development time is more significant than execution time which is milliseconds faster.

Compiler

The source code for every .NET language is transferred into the platform independent Intermediate-Language Code (IL-code). This code is first compiled in a platform specific code during runtime (i.e. program call). This is taken over by the just-in-time (JIT) compiler of the CLR, whereby short delays can be caused at startup. To avoid delays, the program's IL-code can be translated into the machine code of the relevant platform by the Ahead-Of-Time (AOT) compiler. This compiler result is executed without delay, but is no longer platform independent.

What is Mono?

Mono, the Spanish word for monkey, is one of Miguel de Icaza's open source projects, and is sponsored by Novell. The goal of Mono is to replicate the .NET - Framework with all its components under an open license in order to overcome platform and operating system limits.

ECMA standards-based runtime, compilers and development tools for a variety of programming languages and an extensive class library are elements of Mono.

Mono also supports features that go beyond the ECMA specifications, for example, applications can be based on ASP.NET and ADO.NET or Windows Forms can also be developed and executed under Mono.

The main language under Mono and .NET is the Microsoft-developed C#.

This joint programming language is the core of the platform independence. Mono, originally developed for Linux, covers all Linux platforms, while .NET handles the Windows platform. Just recently the Mono implementation became available for the Windows platform as well. It is possible to test compatibility between Mono and .Net developed applications directly in Windows.

Platform independent software development makes application more flexible and thus more cost-effective

The platform independent runtime environment enables developers to find solutions which are applicable on various systems and are thus cost-optimized. As Novell and Microsoft are mutually refraining from asserting patent claims, the future existence and further development of Mono is safeguarded.

Part of the focus of today's software development is on re-usable, portable codes, platform-independence and short development times. The .NET - Framework, Mono and C# combination offers the developer all of this.

At dataschalt, this development environment is one of the tools with which your applications and projects are realized. Contact us for a flexible solution to your ideas.

dataschalt 

Klaus Greisen | Head of Software Development
An der Huelshorst 7 – 9 | 23568 Luebeck | Germany
phone: +49 451 29059-29 | fax: +49 451 38812-69
email: klaus.greisen@dataschalt.com