CoreForms

Run Web Forms applications on the latest .NET Core technologies

Your enterprise still uses Web Forms to run complex business applications? You want to update to .NET Core? Then CoreForms is your solution for migrating Web Forms applications to the latest technologies.

While Microsoft eventually ported most .NET technologies to .NET Core, it‘s very clear that Web Forms will not be available. This is a problem for many line of business applications that might still have a bright future if it were not for their dependency on Web Forms. This is in many cases not a matter of just adapting the application code: the Web Forms framework is deeply entrenched in the architecture and design of many applications, often to the point where it‘s inseparable from the actual application code. CoreForms now provides this critical technology for .NET Core applications. CoreForms enables both direct updates to .NET Core, and incremental migrations to new Core technologies such as ASP.NET Core MVC or Razor Pages.

CoreForms Features

CoreForms provides the System.Web namespace to .NET Core applications and includes the accompanying client components (JavaScript and other resources).

Access to all .NET Core features

- Modern frameworks and classes
- Current C# features

Mix Web Forms code with other UI technologies

- Use Web Forms code and MVC or Razor pages in a single application
- Seamlessly share in-memory state
- Incrementally migrate your application to a current web framework

Pick your third party libraries, Open Source or commercial

- Up-to-date .NET Core libraries
- Existing Web Forms libraries, including control collections

Host your application on Microsoft Windows in any of the standard ASP.NET Core hosting models...

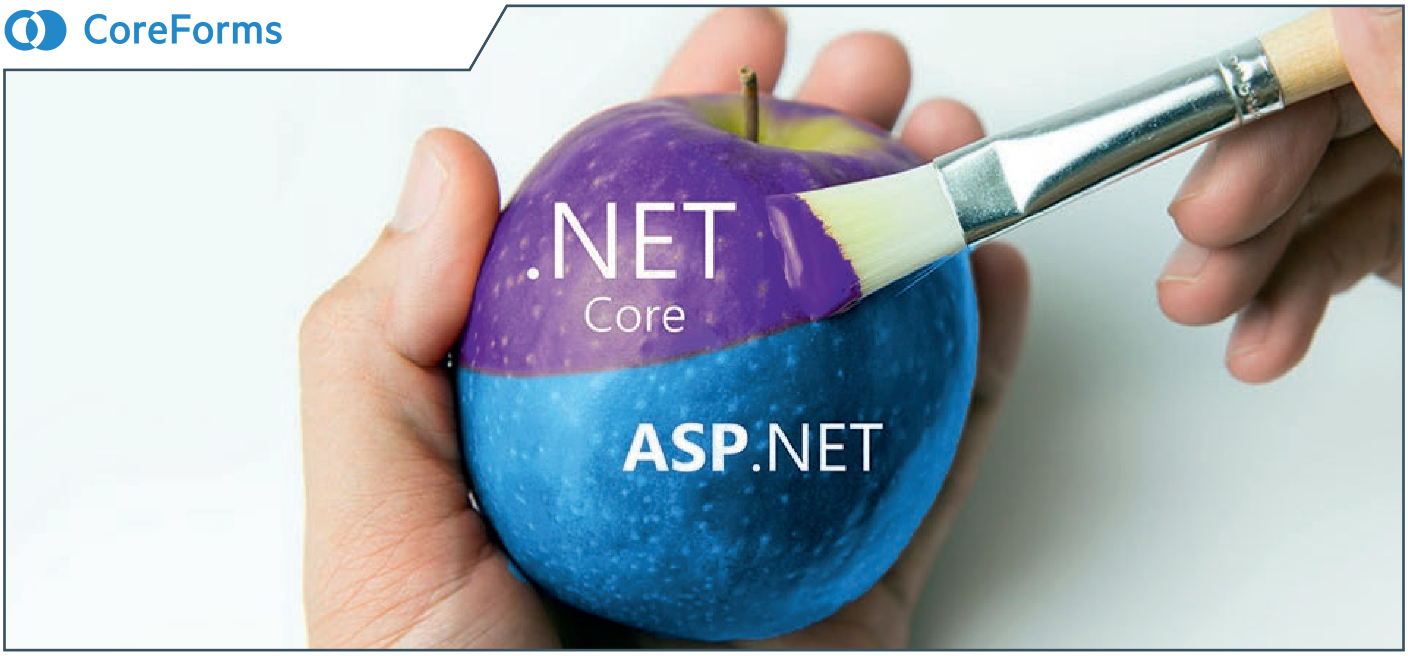
- Kestrel web server
- IIS using ASP.NET Core Module (in-process or out-of-process)
- HTTP.Sys web server implementation

ISO 27001:2013 .NET

RUBICON

Foundation: 2001, **HQ:** Vienna, **Offices:** Bern, Berlin

Employees: ~250, **Customers:** >2.500



Increasing pressure to port existing applications to .NET Core

Microsoft will support .NET Framework 4.8 for many years to come, but this support is limited to security issues and critical bugs. This presents many problems, including the following:

No support for current C# language versions

The last version of C# with full .NET Framework support was C# 7.3.
C# 8 through 12 are only partially available for .NET Framework.
This leads to the following consequences:

- Modern C# code cannot be used, including code snippets but also entire libraries.
- Development tools such as IDEs and refactoring tools might cease to support these limitations.
- Developers might be frustrated not be able to use the latest language features.
- It gets harder to hire new developers, and top developers might even consider quitting.

No library updates

Many libraries, especially those from the Open Source community, will no longer be maintained for .NET Framework.
This leads to the following consequences:

- You might have to pay premium support fees for commercial vendors.
- You might have to maintain old versions of Open Source libraries yourself.
- You might experience situations where critical bug fixes, or even worse, security fixes for your libraries are no longer available. This would force you to replace entire libraries and frameworks, and it would make your software vulnerable in the meantime.
- APIs and SDKs for other products that you depend on might no longer be available.

No new libraries and tools

Most libraries and tools created today are only provided for .NET Core.
This lead to the following consequence:

It will become increasingly hard, costly, and frustrating to support new technologies in existing applications.

Visit us
www.coreforms.net