



# .NET Framework

**Windows bringt schon seit etlichen Versionen das .NET Framework mit. Normalerweise verrichtet es so lautlos seinen Dienst, dass es vielen Anwendern gar nicht im Bewusstsein ist. Trotzdem tauchen einige Fragen dazu immer mal wieder auf.**

Von Hajo Schulz

## Definition

? Was ist überhaupt das .NET Framework?

! Das .NET Framework ist eine Sammlung von Programmierschnittstellen und Hilfsprogrammen, die Entwickler zum Erstellen verschiedenster Anwendungen verwenden können. Der Einsatzbereich reicht von Universal Apps über hergebrachte Windows-Programme bis hin zu Server-Anwendungen.

Zentraler Bestandteil von .NET ist eine „Common Language Runtime“ (CLR) genannte Laufzeitumgebung. Ihren Namen trägt sie, weil man zum Erstellen von .NET-Anwendungen verschiedene Programmiersprachen benutzen kann, auch mehrere innerhalb eines Projekts. Zu den populäreren .NET-Sprachen gehören C# (gesprochen „Ssie scharp“, wie der Ton Cis auf Englisch), VisualBasic.NET und F#. Die dazugehörigen Compiler übersetzen den Quelltext nicht direkt in Maschinencode, sondern in einen Bytecode, der „Common Intermediate Language“ (CIL) heißt. Erst zur Laufzeit übersetzt ein Just-in-Time-Compiler (JIT) diesen in Maschinencode, den die CPU direkt ausführen kann. Zu den Vorteilen, die dieser „Managed Code“ bringt, gehört unter anderem, dass er auf allen Plattformen ausführbar ist, für die es eine passende Laufzeitumgebung gibt. Außerdem verhindert er verbreitete Programmierfehler wie Pufferüberläufe oder ungültige Zeiger.

## Versionitis

? Wieso ist das .NET Framework auf meinem Rechner in mehreren Versionen installiert? Brauche ich die alle?

! Es hat das .NET Framework mit verschiedenen Versionsnummern gege-

ben. Wenn ein Programm .NET benutzt, enthält es immer auch die Information, welche Version es benötigt. So kommt es, dass unterschiedlich alte Anwendungen nach verschiedenen .NET-Versionen verlangen.

Nicht alle .NET-Versionen sind rückwärtskompatibel; in der Entwicklung hat es zwei größere Brüche gegeben. Zur ersten Phase gehören die Versionen 1.0 und 1.1, die aber heutzutage praktisch keine Rolle mehr spielen und auch nicht mehr von Microsoft gepflegt werden. Die zweite Gruppe bilden die Versionen 2.0 bis 3.5.1. Der dritte Entwicklungsstrang beginnt mit der Versionsnummer 4.0; aktuell ist die 4.7.

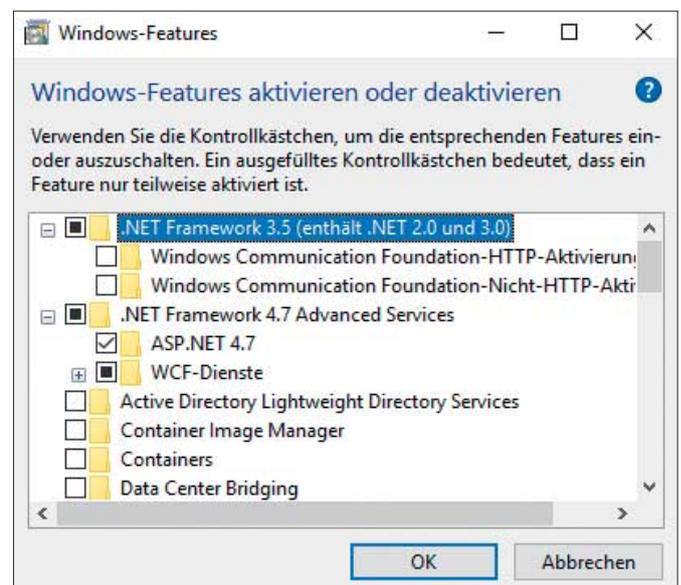
Innerhalb einer Gruppe kann die .NET-Ausgabe mit der höchsten Versionsnummer jeweils auch für ihre Vorgänger einspringen. Um alle .NET-Anwendungen ausführen zu können, benötigen Sie das .NET Framework also in den Versionen 3.5.1 und (derzeit) 4.7.

.NET 3.5 ist seit Windows 7 in allen Windows-Ausgaben vorinstalliert und wird durch Windows Update auf dem Laufenden gehalten. Ab Windows 8 ist außer-

dem ein .NET Framework der 4er-Schiene vorhanden; auch um dessen Pflege kümmert sich Windows Update. Unter Windows 7 gehört .NET 4.x zu den empfohlenen Updates; wenn Sie die regelmäßig installieren, sollten Sie also auch hier auf alle Bedürfnisse von .NET-Anwendungen vorbereitet sein.

Loswerden können Sie die vorinstallierten Ausgaben des .NET Framework bei Bedarf über das Fenster „Windows-Features aktivieren oder deaktivieren“. Unter Windows 7 und 8.1 steckt es in der Rubrik „Programme“ in der Systemsteuerung; unter Windows 10 öffnen Sie es am einfachsten, indem Sie die ersten Zeichen von „Features“ ins Suchfeld eintippen. Ändern sollten Sie hier aber nur etwas, wenn sich aus irgendeinem Grund Windows Update weigert, Ihre .NET-Installation zu aktualisieren. Dann können Sie versuchen, sie hier zu deaktivieren und mithilfe eines heruntergeladenen Pakets (siehe [ct.de/ys7n](http://ct.de/ys7n)) von Hand zu installieren. Die Unter-Features der „Windows Communication Foundation“ (WCF) können Sie auf Desktop-Systemen ausge-

Die älteren .NET-Versionen kann man bei Bedarf ausschalten. Sinnvoll ist das aber nur in Ausnahmefällen.



schaltet lassen; Sie brauchen sie nur im Zusammenhang mit Web-Anwendungen, die auf dem Windows-eigenen Webserver laufen.

### Aktivität nach Updates

? Jedes Mal, wenn Windows Update einen Patch für das .NET Framework einspielt, ist mein Rechner danach längere Zeit beschäftigt. Im Task-Manager fällt dann ein Prozess namens ngen mit hoher CPU- und Festplattenauslastung auf. Was tut der?

! .NET-Code – so auch der in den Updates enthaltene – besteht grundsätzlich aus einem virtuellen Bytecode (CIL – siehe oben). Normalerweise wird der erst kurz vor der Ausführung im Speicher in Maschinencode übersetzt. Für seine eigenen Kernkomponenten erledigt das .NET Framework diese Übersetzung aber im Voraus und speichert den Maschinencode im sogenannten Global Assembly Cache (GAC) auf der Festplatte, um ihn bei Bedarf sofort laden zu können. Außerdem passt es den Code bei dieser Übersetzung auf das aktuelle System, also etwa das konkrete CPU-Modell oder den vorhandenen Arbeitsspeicher an. Diesen Übersetzungsschritt erledigt das von Ihnen genannte Programm. Es handelt sich also um eine Art Compiler, der den virtuellen .NET-Zwischencode in echten Maschinencode übersetzt und speichert.

### Speicherplatz

? Wo liegt das .NET Framework auf der Festplatte? Kann man da womöglich Platz sparen?

! Die Programme und Bibliotheken, aus denen .NET besteht, liegen im Windows-Ordner, normalerweise unter C:\Windows\Microsoft.NET\Framework; auf 64-Bit-Systemen gibt es unter Microsoft.NET auch noch den Unterordner Framework64. Der GAC (siehe oben) steckt in C:\Windows\Microsoft.NET\assembly. Er enthält den vorübersetzten .NET-Code sowohl als CIL als auch als Maschinencode.

Endanwender sollten in diesen Ordnern tunlichst nichts ändern oder löschen. Programmierer können im GAC eigene Bibliotheken hinzufügen, aktualisieren oder

entfernen. Dazu dient das Programm gacutil, das im Lieferumfang der Entwicklungsumgebung Visual Studio enthalten ist.

### Core

? Ich habe von einer .NET-Ausgabe namens „.NET Core“ gehört. Was ist das?

! Dabei handelt es sich um eine komplette Neuentwicklung des .NET Framework als Open Source. Koordiniert wird die Entwicklung von der .NET Foundation unter maßgeblicher Beteiligung von Microsoft. Im Unterschied zum klassischen .NET Framework ist .NET Core wesentlich modularer aufgebaut und hat sich von Anfang an echte Plattformunabhängigkeit auf die Fahnen geschrieben; aktuell gibt es Implementierungen für Windows, Linux und macOS.

Derzeit liegt .NET Core in Version 2.2 vor. Geeignet ist das Framework bislang nur für Konsolen- und Server-Anwendungen. Eine GUI-Bibliothek fehlt noch, soll aber in der noch für dieses Jahr angepeilten Version 3.0 enthalten sein. Quelltexte und das SDK sind auf GitHub gehostet (siehe ct.de/ys7n).

### Selber machen

? Was brauche ich, um selbst in die .NET-Programmierung einzusteigen?

! Die empfohlene Entwicklungsumgebung für .NET-Anwendungen aller Art ist Microsofts Visual Studio. Das gibt es in verschiedenen Ausgaben zu sehr unterschiedlichen Preisen. Die kostenlose Community-Edition (siehe ct.de/ys7n) entspricht vom Lieferumfang her der Professional-Ausgabe und ist nur in der Lizenz beschränkt: Benutzen dürfen es Privatanwender sowie Teams bis zu fünf Entwicklern in kleinen Unternehmen, außerdem darf es für beliebig große Open-Source-Projekte verwendet werden.

Die gesamte Dokumentation zu Visual Studio, zum .NET Framework und zu den Microsoftschen Programmiersprachen ist ebenfalls kostenlos online verfügbar; einige Einstiegspunkte haben wir unter ct.de/ys7n verlinkt. (hos@ct.de)

**Dokumentation und Tools: ct.de/ys7n**

Anzeige