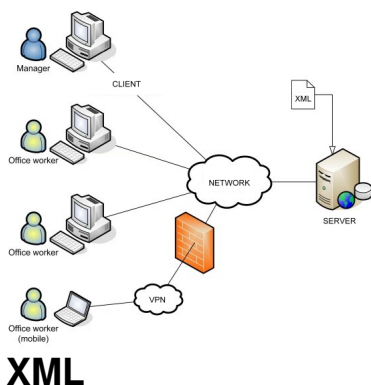


### Einführung

Dieser Artikel basiert im Wesentlichen auf dem Tutorial „[TCP/IP Socket-Programmierung in C#](#)“ und setzt die dort vermittelten Kenntnisse in ein fortschrittliches Client-Server-Modell um. Ziel ist es mithilfe eines Clienten Daten von einem entfernten Server abfragen zu können. Der Server soll diese Daten an einen oder mehrere Clienten gleichzeitig senden können und somit asynchron arbeiten. Die Daten liegen als XML-Dokument vor und werden vom Server nur bei Bedarf „on the fly“ übermittelt. Damit wird zum einen gezeigt, wie sich die Belastung für den Server im Upstream senken lässt und zum anderen eröffnet es die Möglichkeit die Funktion von einfachen Netzwerkprotokollen zu demonstrieren. Netzwerkprotokolle sind exakte Vereinbarungen, nach der Daten zwischen Computern bzw. Prozessen ausgetauscht werden, die durch ein Netz miteinander verbunden sind. Chatprogramme und auch Browser greifen auf Protokolle zurück, um im Internet zu kommunizieren.

Als Beispiel wurden in diesem Artikel Kontaktdaten von natürlichen Personen gewählt. Diese Daten werden in einem XML-Dokument gespeichert und vom Server an die Clienten übertragen. Da der Server im Zentrum des Netzwerks steht, kann er Ressourcen verwalten, die allen Usern gemeinsam sind, um Probleme der Redundanz und Widersprüchlichkeit zu vermeiden. Die Clienten können die Kontaktliste vom Server empfangen und auch mit verschiedenen Befehlen detaillierte Daten aus der Kontaktliste abfragen.

Das Projekt wurde mit [Microsoft Visual Studio](#) und dem .NET Framework 4.0 realisiert.



XML ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten. XML wird u. a. für den plattform- und implementationsunabhängigen Austausch von Daten zwischen Computersystemen eingesetzt, insbesondere über das Internet. Die Extensible Markup Language wurde in einem Standard vom [World Wide Web Consortium \(W3C\)](#) spezifiziert.

Die vom W3C herausgegebene XML-Spezifikation (Recommendation, erste Ausgabe vom 10. Februar 1998, aktuell ist die fünfte Ausgabe vom 26. November 2008) definiert dabei eine Metasprache, auf deren Basis durch strukturelle und inhaltliche Einschränkungen anwendungsspezifische Sprachen definiert werden. Diese Einschränkungen werden durch Schemasprachen wie DTD oder XML Schema ausgedrückt. Beispiele für XML-Sprachen sind: RSS, MathML, GraphML, XHTML, XAML, [Scalable Vector Graphics \(SVG\)](#), aber auch XML-Schema.

Die Webtechnologie Ajax ist heute im Internet oft anzutreffen und basiert ebenfalls auf XML. Ajax ist ein Apronym für die Wortfolge „Asynchronous JavaScript and XML“. Es bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Browser und dem Server. Dieses ermöglicht es, HTTP-Anfragen durchzuführen, während eine HTML-Seite angezeigt wird, und die Seite zu verändern, ohne sie komplett neu zu laden.

### Kontaktliste

In dem Client-Server-Modell soll der entfernte Server Kontaktdaten übertragen, die in einem XML-Dokument lokal abgespeichert sind. In diesem Dokument kann eine Stelle zu beliebigen Zeitpunkten Kontakte hinterlegen und die Clienten können anschließend diese Kontakte von überall auf der Welt abfragen. Wichtig für die Übertragung der Daten ist das sowohl Server als auch Client wissen welche Daten überhaupt vorhanden sind. Die Beschreibung der Kontaktliste mit den Elementen ist daher der erste logische Schritt bei dem Entwurf der Applikation.

Typisch für ein XML-Dokument ist das es aus Textzeichen besteht, im einfachsten Fall ASCII, so dass es damit menschenlesbar – Binärdaten enthält es per Definition nicht - wird. XML-Dokumente besitzen einen physischen und einen logischen Aufbau. Der logische Aufbau entspricht einer Baumstruktur und ist damit hierarchisch organisiert. Als Baumknoten gibt es Elemente, deren physische Auszeichnung mittels einem passenden Paar aus Start-Tag (< *Tag-Name* >) und End-Tag (</ *Tag-Name* >) erfolgen kann, aus Attributen, aus Verarbeitungsanweisungen (<? *Ziel-Name* *Parameter* ?>, engl. Processing Instruction), aus Kommentaren (<!-- *Kommentar-Text* -->) und aus Text.

## Client-Server-Modell

Geschrieben von: Kristian

Dienstag, den 24. August 2010 um 11:20 Uhr - Aktualisiert Mittwoch, den 28. März 2012 um 18:02 Uhr

---

Ein einzelner Kontakt in dem Programm besteht aus der Identifikationsnummer des Mitarbeiters, kurz Id, dem Namen, der Firma bei der der Mitarbeiter angestellt ist, seiner Adresse, der E-Mail, der Telefon-, Fax- und Handynummer, der Anrede, dem Titel und der Position innerhalb des Unternehmens. Nachfolgend ist so eine Kontaktliste mit drei Beispielkontakten aufgelistet.

```
[code xml:lang="csharp"]
1      Stefan Müller      Lufthansa AG
      Elfenbeinstrasse 25, 80935 München      stefanmueller@mail.de      98362550
      01746352423      98362551      Herr      Dipl.-Ing. (FH)      Abteilungsleiter
2      Barbara Bauer      BMW      Burgstrasse 2, 14935 Berlin      barbara@bmw.de
      82272550      017463764254      0892272573      Frau      Dr.      Mitarbeiterin
3      John Doe      Doe Inc.      Evergreen 345, 10101 New York
john@doe.de      10001      10002      10003      Herr      Prof.      Dozent
[/code]
```

Damit wären die Daten beschrieben, die im Client-Server-Modell ausgetauscht werden sollen.

## Netzwerkprotokoll

Ein Netzwerkprotokoll im Prinzip eine Vereinbarung aus einem Satz von Regeln und Formaten (Syntax), die das Kommunikationsverhalten der kommunizierenden Instanzen in den Computern bestimmen (Semantik). Das Internet basiert auf dem OSI-Schichtenmodell, das die Aufgaben der Kommunikation in sieben aufeinander aufbauende Schichten unterteilt. Für jede Schicht existiert eine Beschreibung, in welcher steht, was diese Schicht zu leisten hat. Diese Anforderungen müssen von den Kommunikationsprotokollen realisiert werden. Die konkrete Umsetzung wird dabei nicht vorgegeben und kann daher sehr unterschiedlich sein. Somit existieren mittlerweile für jede der sieben Schichten zahlreiche solcher Protokolle.

Das hier vorgestellte Client-Server-Modell nutzt das Transmission Control Protocol (TCP) für den Datenaustausch als Transportprotokoll. TCP ist Bestandteil der Transportschicht im OSI-Schichtenmodell. Anwendungsprogramme nutzen Transportprotokolle und definieren zusätzlich ein eigenes Protokoll, das in der sogenannten Anwendungsschicht verankert ist. So muss das Programm beim Datenaustausch die über TCP gesendeten Daten interpretieren können. Das Hypertext Transfer Protocol ist ein bekanntes Protokoll aus der Anwendungsschicht und wird von allen Browsern unterstützt. Alle Programme im Internet basieren auf dem Internet Protocol. In diesem Artikel wird das neue [Internet Protocol Version 6 \(IPv6\)](#) genutzt.

Die Entwicklung von eigenen Kommunikationsprotokollen kann sehr komplex und zeitaufwändig sein. Instant Messenger, wie ICQ oder Skype, basieren auf großen Protokollen, wie dem