

Einrichten eines virtuellen Hosts per Apache-HTTP-Server

Anstelle des HTTP-Webservers vom Internetprovider kann der eigene PC als Test-Server fungieren, um z.B. die Webseite so zu testen, als wäre sie gerade online auf dem Webserver.

Der Hobbyprogrammierer will u.a. kostengünstig testen, also bieten sich die kostenlosen HTTP-Server an, die einen virtuellen Host anhand eines beliebigen Ordners auf der Festplatte erzeugen können, wobei Local Host (127.0.0.1) als virtueller Host einrichtbar sein muss (der nicht von der Firewall-Software des PC allein verwaltet sein darf) und die zu testende Webseite im Browser per Domainnamen aktivierbar sein muss (anstelle der Eingabe von 127.0.0.1). Man google, um festzustellen, welche Produkte diesen Kriterien entsprechen. - Abkürzend fällt die Wahl nicht zufällig auf den HTTP-Server von Apache (www.apache.org), welcher ziemlich schwierig zu konfigurieren ist, wenn mehr als nur localhost genutzt werden soll (Für Hobbyzwecke reicht localhost aus).

Der Hobbyprogrammierer, welcher für den Microsoft Internet Explorer (ab IE 7 heißt der Windows Internet Explorer) in seinen Varianten je nach Windows-Version programmieren will, wird definitiv folgende Probleme bekommen:

Microsoft lässt u.a. die Installation des Internet Explorers in parallelen Versionen nicht zu, obwohl Browserversionen nachweislich nicht kompatibel sind und Microsoft Browserversionen innerhalb Windowsversionen supportet werden. Daher benötigt man pro Version des Internet Explorers eine Windows-Installation. Pro Windows-Installation wird eine Windows-Lizenz fällig, auch wenn auf anderer Festplatte am ansonsten identischen PC installiert wird (Windows-Online-Update erkennt solche Doppelversionen und verweigert den Support). Auch wer Windows unter VM emulieren will (oder auf Apple mit Intel-Technik Windows parallel mit Apple nutzen will), benötigt eine Lizenz. Mit anderen Worten: Auch wenn die Mehrfachinstallation nicht parallel nutzbar wäre, sondern immer nur genau 1, will Microsoft Geld haben - nicht umsonst ist Microsoft-Chef so beliebt wie reich und nicht umsonst migrieren immer mehr IT-Anwender zu Linux-Derivaten, die fast identische Browser bezüglich Windows haben. Ergo, der Hobbyprogrammierer wird wohl sämtliche Bekannte mit Testwünschen nerven, oder illegal testen, oder auf andere Browser-Hersteller und deren HTML- sowie Script-Versionen ausweichen, die nicht nur Parallelinstallationen des Browsers zulassen (z.B. Opera unter Windows), sondern auch noch ziemlich gut kompatibel sind (weil identische Scriptmaschine nutzend). Dass Microsoft eventuell keinen kostenlosen HTTP-Server anbietet (der ansonsten auch noch Javascript, Active-X-Control- Aufrufe per Script, DirectX-Zugriff kennen müsste), ist das kleinere - vorallem lösbarere Problem: Eben ohne Microsoft.

Pfade für Dateien im Script einer Webseite, die per HTTP-Server oder lokal von Festplatte gestartet wird:

Relative Pfade per './' sind unabhängig davon, wie die Wurzel (Root) der Webseite heißt.
Absolute Pfade sind abhängig davon, wie die Wurzel (Root) der Webseite heißt.

Wird die Webseite (z.B. www.testseite.de) von einem HTTP-Server gestartet, dann sind absolute Pfade bezüglich http://www.testseite.de möglich, wobei http://www.testseite.de die Root der Webseite darstellt.

(Achtung: Auf einem HTTP-Server des Internetproviders liegt die Webseite natürlich auch auf einer Festplatte, also dort in einem Ordner. Dieser Ordner muss also mit www.testseite.de logisch verbunden sein - wie, das teilt der Internet-Provider mit.

Tipp: Namen des Ordners auf der Festplatte des Internet-Providers kann man genauso nennen wie den des Ordners auf der lokalen Festplatte des PC. Ordernamen mit Pfadzeichen wie ':' oder '/' sind natürlich nicht erlaubt)

Wird die Webseite (z.B. www.testseite.de) von lokaler Festplatte aus gestartet, dann sind absolute Pfade bezüglich http://www.testseite.de nicht möglich, da es solchen Ordernamen nicht geben kann.
Festplattenordner möglich z.B. c:\testseite\

Will man die Webseite identisch verwalten, egal ob man die Webseite von einem HTTP-Server oder von lokaler Festplatte aus startet, dann schaue man sich folgendes Beispiel für www.testseite.de an, das allerdings mit JavaScript oder JScript realisiert wird. Der Festplattenordner ist c:\testseite.

```
var BrowserAufOnlinePruefen=true;           // false für Browser nicht auf online prüfen
                                           // online: Webseite wurde auf HTTP-Server aktiviert
var DomainOhneHTTP='www.testseite.de'; // Host der Webseite ohne http:// und ohne Port
var BrowserIstOnline=false;                // Annahme: Browser ist nicht online
var PfadDerDateien="";                     // Annahme: Browser ist nicht online also
                                           // alle Pfade unterhalb von c:\testseite\
                                           // wobei die Startdatei index.html der
                                           // Webseite eben in diesem Ordner liegt

if(BrowserAufOnlinePruefen)                // wenn auf online geprüft werden soll
{if(window.location.hostname!=null);      // aktuell gefundener Host
  BrowserIstOnline=(window.location.hostname== DomainOhneHTTP);}
                                           // Host www.xxx.yyy prüfen auf aktuelle gefundenen Host
                                           // true so Browser online
```



```
}  
  
if(BrowserIstOnline) {PfadDerDateien='http://' + DomainOhneHTTP;}  
// Pfad wenn Browser online ist: Alle Pfade unterhalb von  
// http://www.testseite.de  
// Für den HTTP-Server liegt die Webseite natürlich  
// auf einer Festplatte, also dort in einem Ordner.  
// Jeder Pfad in der Webseite wird in den Ordnerpfad  
// der Festplatte automatisch konvertiert (HTTP zu  
// Festplatte per Dienst: Daher der Name HTTP-  
// Server)
```

Nachfolgend die beispielhafte Einrichtung einer Webseite per Apache-HTTP-Server 2.2.2 bis 2.2.4. unter Windows XP ab SP 1:

Webseitendomain heißt www.testseite.de mit index.html als Startdatei
Festplattenordner der Domain, die als virtueller Host über localhost (also 127.0.0.1) laufen soll
 c:\testseite\
Apache wurde installiert unter e:\wxp\apache\
wobei gelten muss
 DNS-Dienst von Windows muss aktiv sein (in der Regel ist der Dienst automatisch aktiv)
Installationstyp All Users, on Port 80, as a Service - Recommended
Domain und Servername 127.0.0.1 (nicht localhost kodieren)
Email beliebig@localhost
Mit der Installation wird der Dienst-Monitor von Apache bei Windowsstart ebenfalls starten.
Es wird Apache als permanenter Dienst eingerichtet, der über den Dienst-Monitor von Apache
aktivierbar / deaktivierbar ist. Es sind zwar mehrere Apache-Dienste einrichtbar, aber genau 1
kann nur immer aktiv sein.

Apache-Software einrichten:

Dienst-Monitor von Apache: siehe oben
aber: Da in Windows für jeden Aufruf der Webseite www.testseite.de unter Apache eine
Einstellung getroffen werden muss, empfiehlt es sich, den Apache-Dienst
grundsätzlich manuell zu starten und zu stoppen anhand nachfolgend
vorgestellter BAT-Dateien.
Dafür muss aber einmalig folgendes eingestellt werden:
Der Apache-Dienst-Name ist per Dienst-Monitor zu sehen (und der Status ob
aktiv oder deaktiv).
Dienste sind unter Windows per Systemsteuerung-Verwaltung-Dienste
verwaltbar (auch per Apache-Monitor ist die Dienstverwaltung
aktivierbar).
Der Apache-Dienst muss zuerst deaktiviert werden, dann auf Starttyp manuell
gesetzt werden: Mit Windows-Start startet der Dienst nicht automatisch.

Windows anpassen an den Virtuellen Host

Die Anpassung erfolgt so, dass Apache per Batch-File, die unten erklärt werden, gestartet
und deaktiviert wird (Batch-Files sind passend zur Anpassung von Windows).

Unter c:\windows\system32\drivers\etc\ liegt die Datei hosts

Copyright (c) 1993-1999 Microsoft Corp.

```
#  
# Dies ist eine HOSTS-Beispieldatei, die von Microsoft TCP/IP  
# für Windows 2000 verwendet wird.  
#  
# Diese Datei enthält die Zuordnungen der IP-Adressen zu Hostnamen.  
# Jeder Eintrag muss in einer eigenen Zeile stehen. Die IP-  
# Adresse sollte in der ersten Spalte gefolgt vom zugehörigen  
# Hostnamen stehen.  
# Die IP-Adresse und der Hostname müssen durch mindestens ein  
# Leerzeichen getrennt sein.  
#  
# Zusätzliche Kommentare (so wie in dieser Datei) können in  
# einzelnen Zeilen oder hinter dem Computernamen eingefügt werden,  
# aber müssen mit dem Zeichen '#' eingegeben werden.  
#  
# Zum Beispiel:  
#
```



```
#      102.54.94.97      rhino.acme.com      # Quellserver
#      38.25.63.10      x.acme.com          # x-Clienthost

127.0.0.1      localhost
```

Diese Datei verwaltet -wie man sieht - auch den localhost.

Standardgemäß ist localhost auf 127.0.0.1 gelegt (nur deswegen sind localhost und 127.0.0.1 synonym)

Soll aber die Webseite www.testseite.de über 127.0.0.1 getestet werden, muss also die Zeile

```
127.0.0.1      localhost
```

ersetzt werden durch

```
127.0.0.1      www.testseite.de
```

Nach dem Test muss localhost wieder für 127.0.01 verfügbar gemacht werden, also der Standard gesetzt werden.

Genau dieses Ersetzen machen die Batch-Files siehe unten. Dafür benötigen sie nur 2 neue Ordner, die einmalig mit Inhalt manuell angelegt werden müssen in c:\windows\system32\drivers\etc\

Schritt 1: Ordner c:\windows\system32\drivers\etc_hosts_standard\

manuell erzeugen, und dorthin die bisher unveränderte, also originale hosts-Datei kopieren (Datei enthält 127.0.0.1 localhost)

```
# Die IP-Adresse und der Hostname müssen durch mindestens ein
# Leerzeichen getrennt sein.
#
# Zusätzliche Kommentare (so wie in dieser Datei) können in
# einzelnen Zeilen oder hinter dem Computernamen eingefügt werden,
# aber müssen mit dem Zeichen '#' eingegeben werden.
#
# Zum Beispiel:
#
#      102.54.94.97      rhino.acme.com      # Quellserver
#      38.25.63.10      x.acme.com          # x-Clienthost

127.0.0.1      localhost
```

Schritt 2 Ordner c:\windows\system32\drivers\etc_hosts_mit_www_testseite_de\ manuell erzeugen, und dorthin die bisher unveränderte, also originale hosts-Datei kopieren (Datei enthält 127.0.0.1 localhost) dann diese Datei in diesem Ordner per notepad.exe (nicht Word etc.) auf 127.0.0.1 www.testseite.de

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# Dies ist eine HOSTS-Beispieldatei, die von Microsoft TCP/IP
# für Windows 2000 verwendet wird.
#
# Diese Datei enthält die Zuordnungen der IP-Adressen zu Hostnamen.
# Jeder Eintrag muss in einer eigenen Zeile stehen. Die IP-
# Adresse sollte in der ersten Spalte gefolgt vom zugehörigen
# Hostnamen stehen.
# Die IP-Adresse und der Hostname müssen durch mindestens ein
# Leerzeichen getrennt sein.
#
# Zusätzliche Kommentare (so wie in dieser Datei) können in
# einzelnen Zeilen oder hinter dem Computernamen eingefügt werden,
# aber müssen mit dem Zeichen '#' eingegeben werden.
```



```
#  
# Zum Beispiel:  
#  
#      102.54.94.97      rhino.acme.com      # Quellserver  
#      38.25.63.10      x.acme.com        # x-Clienthost  
  
127.0.0.1      www.testseite.de
```

Virtual-Host genau 1x einrichten in e:\wpx\apache\conf\ dort in der Datei httpd.conf

erst httpd.conf kopieren in einen Sicherungsordner freier Wahl

dann per notepad.exe (nicht per Word etc.) am Ende der httpd.conf folgenden Text einfügen

Achtung: Es muss natürlich dann

www.testseite.de ersetzt werden durch zu

testende Domain

c:/testseite

ersetzt werden durch den

wirkliche Pfad

```
# Virtual Host für www.testseite.de auf 127.0.0.1:80 gehostet
```

```
# ERST Virtual Host für den Server, der damit alle Servernamen ungleich  
www.testseite.de abfängt
```

```
# also Angaben aus Serverinstalltion verwendet
```

```
NameVirtualHost 127.0.0.1
```

```
<VirtualHost 127.0.0.1>
```

```
    ServerName localhost
```

```
    ServerAlias 127.0.0.1
```

```
    DocumentRoot e:/wpx/apache/htdocs
```

```
    ErrorLog e:/wpx/apache/logs/error.log
```

```
    TransferLog e:/wpx/apache/logs/access.log
```

```
    ScriptAlias /cgi-bin/ e:/wpx/apache/cgi-bin/
```

```
</VirtualHost>
```

```
# Virtual Host für www.testseite.de
```

```
<VirtualHost 127.0.0.1>
```

```
    ServerName www.testseite.de
```

```
    ServerAlias 127.0.0.1
```

```
    DocumentRoot c:/testseite
```

```
    DirectoryIndex index.html
```

```
    ErrorLog c:/testseite/apache_error.log
```

```
    TransferLog c:/testseite/apache_access.log
```

```
    <Directory c:/testseite>
```

```
        Options Indexes FollowSymLinks MultiViews
```

```
        AllowOverride None
```

```
        Order allow,deny
```

```
        Allow from all
```

```
    </Directory>
```

```
</VirtualHost>
```

Apache-Start und Stop per folgender BAT-Dateien, die z.B. in C:\ liegen (egal wo auf Festplatte):

Achtung: Falls der Start von Apache wegen nicht vorhandenem Dienst nicht erfolgen kann, dann gilt:

Mit der Apache-Installation wurde ein Dienst eingerichtet: siehe oben. Nur leider, der Apache-Start verlangt einen weiteren Dienst mit anderem Namen. Da bekannt ist, dass aber nur 1 Dienst zu jedem Zeitpunkt aktiv sein kann, ist es sehr verwunderlich, wieso Apache-Start den bereits vorhandenen Dienst nicht nimmt.

Anstelle von wundern bitte folgendes genau 1 mal ausführen:

```
e:\wpx\apache\bin\ httpd.exe -k install
```



im bin-Verzeichnis liegen die ausführbaren Dateien von Apache.
httpd.exe ist die Hauptkomponente von Apache.
-k install installiert einen weiteren Apache-Dienst.

Logischerweise ist dieser Dienst analog zum Dienst, der bei Apache-Installation erzeugt wurde,
ebenfalls per Dienste-Verwaltung zu deaktivieren und auf Starttyp manuell zu setzen.
Dieser neue Dienst hat aber den passenden Namen, der vom Start von Apache akzeptiert wird.

Hier ein Beispiel für den Dienste-Wirrwarr:

Apache-Installation erzeugt Dienst 'Apache2.2'.
Apache-Start will aber den Dienst 'Apache 2'

Batch-File 'ApacheStarten.bat' für Start des Apache, wobei der Dienst 'Apache2' aktiviert wird

```
@echo off
cls
echo Apache als Anwendung starten bei inaktivem Apache-Dienst 'Apache2'
echo          (Apache-Dienst darf nicht bei Windows-Start aktiv sein)
echo          (Apache stoppen immer per ApacheStoppen.bat)
echo hosts-Datei mit www.testseite.de bereitstellen
echo.
copy /V /Y C:\WINDOWS\system32\drivers\etc\_hosts_mit_www_testseite_de\hosts
C:\WINDOWS\system32\drivers\etc\hosts > NUL
e:\wxp\apache\bin\httpd.exe -k start
echo.
echo.
echo          ..... Apache-Anwendung wurde nur gestartet, wenn oben keine
echo                   Fehlermeldung angezeigt wurde !
echo                   Falls Fehlermeldung dann im Monitor Apache Servers
echo                   zu allen Eintraege STOP-Button klicken (falls klickbar)
echo                   und Batch-File neu starten
echo.
echo.
echo          Apache kann keine ActiveX durchreichen (z.B. Logox WebSpeech)
echo                   per JScript-erzeugte Popups nicht immer korrekt rendern
echo.
echo.
pause
echo.
echo.
echo aktuelle hosts-Datei lautet
echo.
type C:\WINDOWS\system32\drivers\etc\hosts
echo.
pause
```

Batch-File 'ApacheStoppen.bat' für Start des Apache, wobei der Dienst 'Apache2' de-aktiviert wird

```
@echo off
cls
echo Apache als Anwendung stoppen bei aktiven Apache-Dienst 'Apache2'
echo          (Apache-Dienst aktiviert per ApacheStarten.bat)
echo hosts-Datei ohne www.testseite.de bereitstellen
echo.
copy /V /Y C:\WINDOWS\system32\drivers\etc\_hosts_standard\hosts
C:\WINDOWS\system32\drivers\etc\hosts >NUL
e:\wxp\apache\bin\httpd.exe -k stop
echo.
echo.
echo          ..... Apache-Anwendung wurde nur gestoppt, wenn oben keine
echo                   Fehlermeldung angezeigt wurde !
```



```
echo                Falls Fehlermeldung dann im Monitor Apache Servers
echo                dann war kein Apache-Server aktiv.
echo.
echo.
pause
echo.
echo.
echo aktuelle hosts-Datei lautet
echo.
type C:\WINDOWS\system32\drivers\etc\hosts
echo.
pause
```

Batch-File 'ApacheHostsDateiAnzeigen.bat' für Anzeige der hosts-Datei

```
@echo off
cls
echo aktuelle hosts-Datei anzeigen
echo.
type C:\WINDOWS\system32\drivers\etc\hosts
echo.
echo.
echo.
pause
```

Webseite aktivieren per Apache:

Da die Webseite nur per o.g. Batch-Dateien gestartet bzw. gestoppt werden kann, kann der Apache-Dienst-Monitor nur noch zur Kontrolle benutzt werden, ob eine Apache-Dienst aktiv ist.

Schritt 1: Vor dem Start der Webseite darf kein Apache-Dienst aktiv sein

Schritt 2 ApacheHostsDateiAnzeigen.bat aktivieren. Die hosts-Datei muss enthalten

```
127.0.0.1 localhost
```

Schritt 3: ApacheStarten.bat aktivieren. Apache startet.

Wer will kann jetzt Schritt 2 wiederholen und sieht dann

```
127.0.01. www.testseite.de
```

Wenn jetzt die Firewall-Software sich meldet, dann Apache erlauben am Port 80 (also HTTP) von 127.0.0.1

Achtung: Nutzt die zu testende Webseite andere Ports z.B. die von Plugins, dann können die Plugins selbst einen Firewall eintrag verlangen, aber Apache kann natürlich nur HTTP (Port 80) und muss nur zwischen Webseite und den Plugins vermitteln (eben über Port 80)

Schritt 4: Browser der Wahl aktivieren
Dort die Domain www.testseite.de eintippen die Webseite öffnet sich anhand c:\testseite\index.html

Tipp: Sollte die Webseite bereits auf einem Webserver online sein, dann würde ohne Apache-Start natürlich die online-Variante aktiviert werden und nicht die Webseite von der Festplatte aus dem Ordner c:\testseite !

Wichtig: Sollte die Webseite im Ordner c:\testseite\ geändert worden sein, so muss der Browser-Cache gelöscht werden, bevor die die Webseite erneut aktiviert wird. Apache lässt den Browser genauso agieren, als ob der online die Webseite laden würde, also über den Browser-Cache. Manche Browser oder Webseiten unterbinden dann das Neuladen einer Datei, die im Browser-Cache dem Namen nach vorhanden ist, aber in c:\testseite verändert vorliegt also zwingend neu geladen werden muss.... Ergo Browser-Cache vorher löschen.

Browser-Cache im löschen im



Internet Explorer per Internetoptionen, die
im Browser per Menüpunkt Extra oder per Systemsteuerung
aktivierbar sind (ev. Verknüpfung auf Desktop manuell
erzeugen)
Opera per Menüpunkt Extra-Einstellungen-Erweitert

Schritt 5: Webseite getestet (man hat jetzt ein Smily-Gesicht oder graue Haare mehr, oder den Chef,
Kunden, Frau am Hals *g)
also Zeit zum Deaktivieren von Apache per ApacheStoppen.bat

Schritt 6: Webseitendaten aus c:\testseite sichern.

