

Web-Server tunen, Bandbreite sparen

HTTP-Kompression mit Apache und IIS

Außer immer nur mehr Speicher und CPU-Leistung zu implementieren, gibt es noch ganz andere Möglichkeiten, um mehr aus seinem Server herauszuholen.

THOMAS WÖLFER

Es mag für Sie eine Überraschung sein, aber Webseiten können in komprimierter Form ausgeliefert werden. Nachdem es sich bei HTML um eine Textdatei handelt, eignen sich Dateien in diesem Format sogar besonders gut dazu, komprimiert zu werden. Praktisch alle HTML-Seiten sind mit den immer wiederkehrenden gleichen Tags angefüllt. Das macht HTML zu einem besonders guten Kandidaten für die Komprimierung, denn gleiche Byte-Sequenzen lassen sich nicht nur sehr gut, sondern auch sehr leicht komprimieren.

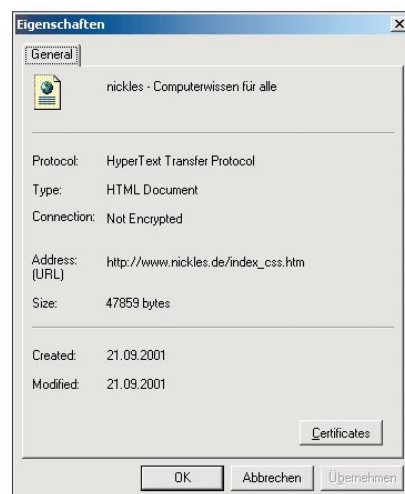
■ Erster Vorteil: Sparen an der Bandbreite

Als Beispiel soll an dieser Stelle die Startseite der bekannten Site www.nickles.de dienen. Diese hat in der CSS-Variante zwar keine feste Größe, da die Seite dynamisch aus einer Datenbank erzeugt wird, aber eine einmal heruntergeladene Variante hatte eine Größe von 47.548 Byte. Wenn Sie diese Datei mit gzip komprimieren, kommt eine Datei mit einer Größe von 10610 Byte heraus. Es ist ohne besondere Mühe möglich, die Seite so stark zu komprimieren, dass das Resultat nur noch weniger als ein Viertel des Umfangs der Originalseite umfasst. Beim Ausliefern der komprimierten Seite werden also circa 37 Kilobyte eingespart - das macht bei über 600.000 Abrufen pro Monat eine Einsparung von 22 Gigabyte Traffic. Der pro GByte abrechnende ISP verdient zwar weniger, aber der Betreiber der Site kann wirklich deutlich an der Traffic sparen, und

das einfach nur durch das komprimieren von Seiten.

■ Zweiter Vorteil: Schneller surfen

Eine Seite, die komprimiert ausgeliefert wird, ist deutlich kleiner. Nachdem aber weniger Daten transportiert werden müssen, kommt die Seite auch beim Sur-



DIE EIGENSCHAFTEN EINER WEBSEITE: Bei einigen Browsern – zum Beispiel dem IE 5.5 – kann man die komprimierte und die unkomprimierte Größe sehen. Beim IE 6 geht das leider nicht mehr.

fer schneller an. Bei der Beispielseite musste weniger als ein Viertel der Originalgröße ausgeliefert werden – die Seite erscheint beim Surfer also auch viermal schneller.

Das freut nicht nur den Surfer, sondern auch Sie als Site-Administrator haben einen Vorteil. Wird eine Seite

schneller ausgeliefert, wird auch der zugehörige HTTP-Dämon (oder unter NT der zugehörige Thread) schneller beendet und die belegten Sockets schneller wieder frei. Das bedeutet letzten Endes, dass der den Web-Server betreibende Rechner, auch weniger Last zu verkraften hat. Beim Ausliefern komprimierter Webseiten handelt es sich um eine Tuning-Maßnahme für Web-Server, die dem Betreiber nebenher noch Geld für die belegte Bandbreite spart.

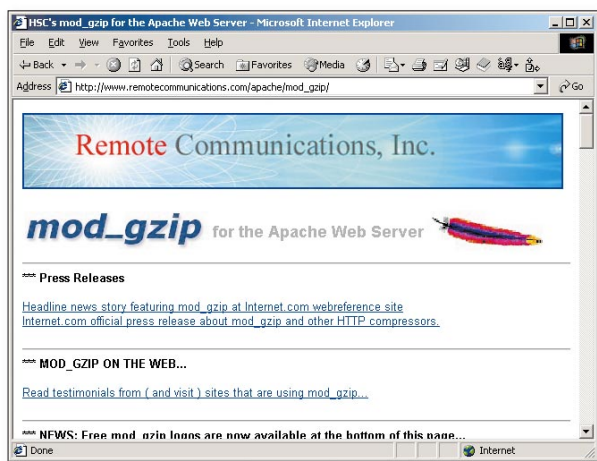
Ob Sie in der Praxis durch die Auslieferung komprimierter Seite tatsächlich die Server-Last verringern, ist von vielen Faktoren abhängig. Aber selbst wenn Sie nur Geld für Bandbreite sparen und dem Surfer ein besseres Surf-Verhalten bieten, das Komprimieren lohnt sich auf jeden Fall. Damit wäre auch der erste Faktor angesprochen, von dem die tatsächliche Last-Ersparnis abhängt. Man darf nicht vergessen, dass die Seiten erstmal komprimiert werden müssen und das erzeugt seinerseits erst einmal Last. Wie viel, ist davon abhängig, ob einmal komprimierte Seiten zwischengespeichert werden können und nicht wieder komprimiert werden brauchen. Ob das überhaupt geht ist vom verwendeten Server beziehungsweise den verwendeten Zusatzprodukten abhängig.

Im Rahmen der verschiedenen vom Autor betreuten Websites konnte dies sowohl auf Apache-Servern als auch mit IIS-Installationen getestet werden, und zwar mit Sites unterschiedlichster Größe. Dabei wurden Sites untersucht, die zwischen 1500 und 5 Millionen Seitenabrufe, sowohl dynamischer als auch statischer Seiten verarbeiten. Bei praktisch allen Websites wurde dabei ein all-

gemeiner Rückgang der Server-Last festgestellt. : Ob das wirklich immer der Fall ist, ist von vielen Faktoren abhängig und im Vorhinein nicht ohne weiteres zu sagen.

■ Seite komprimiert ausliefern

Das Einrichten von komprimierten Webseiten setzt einige Vorarbeit voraus.



DIE WEBSEITE VON MOD_GZIP: Hier finden Sie unter anderem eine sehr hilfreiche Mailing-Liste zu diesem Programm,

Nur gzip oder WinZip aufrufen, und Dateien vor dem Hochladen zum Web-Server komprimieren reicht nicht aus. Leider werfen komprimiert ausgelieferte Seiten auch das ein oder andere Problem auf.

Zunächst sollten Sie wissen, dass die Auslieferung von HTML-Seiten in kodierter Form von Haus aus kein Problem darstellt. Dabei kann es sich um eine beliebige Kodierung handeln. Die Kompression ist nur eine Spezialform.

Wird eine Webseite nicht in ihrer Originalform als Text transportiert, spricht man bei dem angewendeten Verfahren von „Transfer Encoding“. Beim Transfer Encoding ist es gleichgültig, ob das Encoding eine Verschlüsselung, eine Komprimierung oder irgend ein anderes Verfahren darstellt. Wichtig ist allerdings, dass der Client (also der Browser des Surfers) im Zuge des Requests mitgeteilt hat, dass er eine bestimmte Form des Encodings verarbeiten kann.

Im Rahmen dieses Beitrages liegt das besondere Augenmerk auf der Komprimierung. Dafür gibt es ein extra standardisiertes Kompressionsverfahren – eigentlich sogar zwei – die die Namen „gzip“ bzw. „deflate“ tragen. Beide sind Teil des HTTP-1.1-Standards und dieser Standard wird von allen gängigen

Browsern unterstützt. Zumindest unterstützen alle gängigen Browser Transfer-Encoding mit gzip über HTTP 1.1. Bei einigen älteren Browser gibt es das Problem, dass sie melden mit HTTP 1.1 zurechtzukommen, das aber dann nicht tun. Normalerweise unterstützen alle Microsoft Browser ab dem Internet Explorer 4.0 und alle Netscapes ab der Version 3.0 das Protokoll. Auch etwas ausgefallene Browser wie etwa der auf

Text basierende Lynx-Browser ist kompatibel zum Transfer-Encoding in HTTP-1.1-Verbindungen.

Meldet sich hingegen ein Browser als HTTP-1.0-kompatibler Client am Server an, so muss der Server einfach nur weniger tun. Der Komprimierungsschritt muss entfallen, sodass die Seite in ihrer normalen Form ausgeliefert wird.

Im Großen und Ganzen macht die Verwendung von Komprimierung mit HTTP 1.1 keine Schwierigkeiten. Es tauchen hin und wieder veraltete Proxy-Server auf,

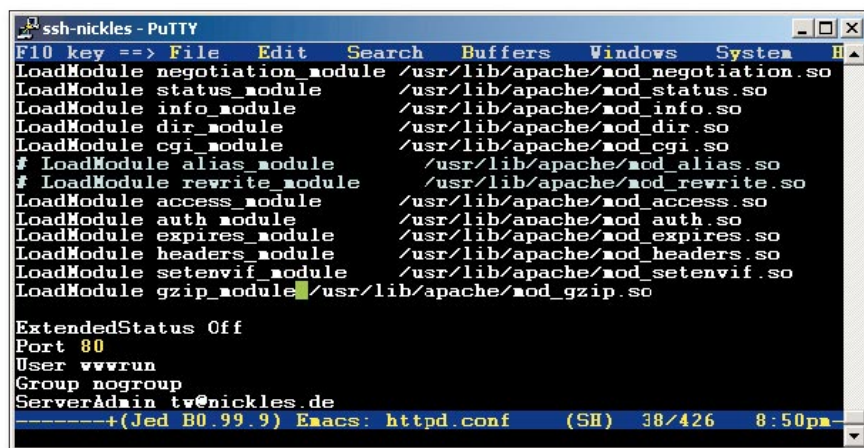
in HTTP 1.1 im Wege stehen, gibt es nicht.

Es drängt sich natürlich die Frage auf, wie man die Komprimierung richtig einrichtet. Diese Frage hat mehrere Antworten. Welche Antwort auf Sie zutrifft, ist davon abhängig, welchen Web-Server Sie oder Ihr ISP verwendet. Im Folgenden finden Sie zwei Erklärungen: eine für den IIS von Microsoft, und eine zweite für den Apache.

■ Transfer Encoding mit Apache

Der Apache beherrscht von Haus aus keine Komprimierung. Das ist nicht so schlimm, denn beim Apache wird diese Möglichkeit mit einem separaten Modul geboten. Auch hier gilt es einige Vorarbeiten zu leisten. Ein einfacher Eintrag in der httpd.conf reicht nicht aus, damit Sie Dateien komprimiert übertragen können.

Statt dessen benötigen Sie ein Apache Module, und zwar das Modul mod_gzip. Mod_gzip ist eine Open-Source-Lösung, sodass Sie auch Zugriff auf den Quell-Code des Moduls haben. Wenn Sie dies nicht möchten, müssen Sie den Quell-Code nicht beachten, denn es gibt auch eine fertig kompilierte Variante, die in Form eines Binaries heruntergeladen werden kann. Die Fir-



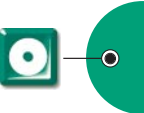
DAS MOD_GZIP muss ganz normal in der Apache-Konfigurationsdatei geladen werden.

die nicht kompatibel sind. Dann gibt es die bereits erwähnten älteren Browser-Builds, die HTTP 1.1 nicht unterstützen. Beides sind jedoch Probleme, die nur selten auftreten und über kurz oder lang verschwinden, wenn die „fehlerhaften“ Browser und die veralteten Proxies nicht mehr vorhanden sind. Schwierigere Probleme, die der Verwendung von Transfer Encoding

ma Hyperspace Communications stellt mod_gzip her. Die zu mod_gzip gehörende Webseite finden Sie bei dieser Firma unter der URL:

http://www.remotecomunications.com/apache/mod_gzip/.

Im einfachsten Fall laden Sie das mod_gzip Binary für Ihre Plattform herunter und kopieren es in das Verzeichnis, in dem Sie auch Ihre anderen mods



```
ssh-nickles - PuTTY
F10 key ==> File Edit Search Buffers Windows System H
# MOD_GZIP configuration
<Directory "/usr/home/nickles/public_html">
  mod_gzip_on Yes
  mod_gzip_can_negotiate yes

  mod_gzip_minimum_file_size 500
  mod_gzip_maximum_file_size 0
  mod_gzip_maximum_inmem_size 100000
  mod_gzip_item_include mime "application/x-httpd-php"
  mod_gzip_item_include mime text/*
  mod_gzip_item_include mime "httpd/unix-directory"
  mod_gzip_dechunk yes
  mod_gzip_temp_dir "/tmp"
  mod_gzip_keep_workfiles No
  mod_gzip_item_include file "\.php3$"
  mod_gzip_item_include file "\.php$"
  mod_gzip_item_include file "\.html$"
  mod_gzip_item_include file "\.htm$"

+ (Jed B0.99.9) Emacs: httpd.conf (SH) 318/426 8:51pm
```

MOD_GZIP hat eine ganze Reihe von Konfigurationsparametern. Unter anderem können Sie festlegen, welche Dateien komprimiert werden sollen und welche nicht.

untergebracht haben. Das ist fast immer das Verzeichnis `/usr/lib/apache`.

Nachdem es sich bei `mod_gzip` um ein ganz normales Apache-Modul handelt, müssen Sie das Modul per Eintragung in der `httpd.conf` laden. Dazu suchen Sie die Stelle in Ihrer `httpd.conf`, in der auch die anderen Module geladen werden.

Wenn Sie eine SuSE-Distribution verwenden, ist die Sache ein bisschen schwierig. Unverständlicherweise verteilt SuSE eine hoffnungslos überkonfektionierte HTTPD-Konfigurationsdatei. Suchen Sie in diesem Fall nach einem „LoadModule“-Statement - und das am besten mehrfach, denn wenn Sie die Original Suse-HTTPD.conf verwenden, müssen Sie das Modul mehrfach eintragen. (Kurzer Tipp: Löschen Sie die SuSE-Variante, und installieren Sie einen vernünftigen Apache von Apache.org - das macht nicht nur das Konfigurieren deutlich einfacher.)

In einer normalen Apache Konfigurationsdatei sollten Sie die richtige Stelle zum Laden des mods relativ einfach finden.

Fügen Sie am Ende der bisher mit „LoadModule“-Konfigurationsanweisungen geladenen Module eine neue Zeile hinzu. Diese hat, sofern sich der Pfad bei Ihren mods nicht unterscheidet, folgendes Aussehen:

```
LoadModule gzip_module
  /usr/lib/apache/mod_gzip.so
```

Nun müssen Sie das Modul noch konfigurieren. Sie können `mod_gzip` mit Directory-Statements innerhalb der `httpd.conf` konfigurieren. Das bedeutet, dass Sie die Komprimierung bei Bedarf auf bestimmte Verzeichnisse einschränken können. `Mod_gzip` kann nicht nur statische HTML-Inhalte komprimieren, sondern auch solche, die von Skripten oder PHP dynamisch erzeugt werden.

Das bedeutet, dass Sie im Normalfall eine Einschränkung auf bestimmte Verzeichnisse gar nicht durchführen müssen, abgesehen von der Testphase. Schließlich ist es am effektivsten, wenn alle Ihre Inhalte in komprimierter Form ausgeliefert werden, ganz gleich woher diese stammen.

Wenn Sie zum Beispiel alle Inhalte Ihrer Site, die aus HTML-Seiten oder

```
mod_gzip_on Yes
mod_gzip_minimum_file_size
  1002
mod_gzip_maximum_file_size 0
mod_gzip_maximum_inmem_size
  60000
mod_gzip_item_include mime
  "application/x-httpd-php"
mod_gzip_item_include mime
  text/*
mod_gzip_item_include mime
  "httpd/unix-directory"
mod_gzip_dechunk Yes
mod_gzip_keep_workfiles No
mod_gzip_item_include file
  "\.php3$"
mod_gzip_item_include file
  "\.html$"
mod_gzip_item_include file
  "\.htm$"
mod_gzip_item_exclude file
  "\.css$"
mod_gzip_item_exclude file
  "\.js$"
</Directory>
```

Wenn Sie diesen Block in Ihre `httpd.conf` eingetragen haben, müssen Sie den Apache neustarten. Danach wird `mod_gzip` aufgefordert PHP- und HTML-Dateien vor der Auslieferung zu komprimieren. Ob das tatsächlich funktioniert, können Sie verhältnismäßig einfach testen. Das geht deshalb so einfach, weil `mod_gzip` Informatio-

```
ssh-nickles - PuTTY
F10 key ==> File Edit Search Buffers Windows System Help
ContentLookups Off
ErrorLog /var/log/httpd/error_log
# LogLevel warn
LogLevel crit

LogFormat "%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\" combined"
LogFormat "%h %l %u %t \"%r\" %s %b common"
LogFormat "%h %l %u %t \"%r\" %s %b mod_gzip: %{mod_gzip_result}n %{mod_gzip_compression_ratio}npt." common_gzip
LogFormat "%{Referer}i -> %U referer"
LogFormat "%{User-Agent}i agent"

CustomLog /var/log/httpd/access_log common_gzip

# using SetHandler
+ (Jed B0.99.9) Emacs: httpd.conf (SH) 367/426 8:53pm
```

DAMIT SIE AUCH die `mod_gzip`-Statistik im Logfile sehen können, müssen Sie das Log-Format entsprechend erweitern.

PHP-Programmen stammen, komprimieren möchten, können Sie die folgenden Anweisungen für die Konfiguration von `mod_gzip` verwenden. In den Beispiel-Konfigurations-Anweisungen bei Hyperspace Communications gibt es noch ein paar mehr Anweisungen. Allerdings sind die dort vorliegenden Dokumentationen nicht immer ganz auf der Höhe der Zeit. Wenn mal eine Anweisung nicht ganz so funktioniert, wie Sie sich das vorstellen, ist vermutlich diese Anweisung in Ihrer Version von `mod_gzip` nicht mehr enthalten.

```
<Directory "HierStehtDerPfadZu
  IhremRootVerzeichnis">
```

nen in Ihre HTTP-Log-Dateien eintragen kann. Dazu müssen Sie eine weitere Änderung an der Konfigurationsdatei für Ihr Apache vornehmen.

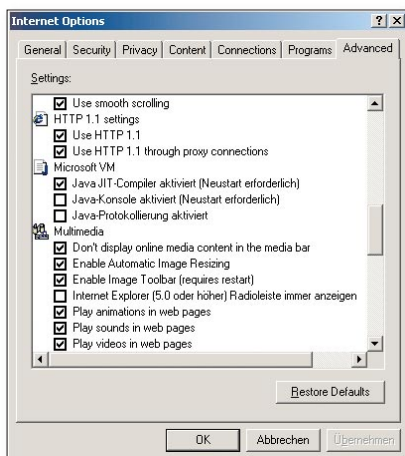
Im Wesentlichen müssen Sie das „LogFormat“ in Ihrer `httpd.conf` verändern. Normalerweise sieht die Format-Definition des „common“ Logfile-Formates in etwa folgendermaßen aus:

```
LogFormat "%h %l %u %t \"%r\" %s %b" common
```

Mit diesem Format erhalten Sie alle relevanten Daten im http-Log. Um diese um die `mod_gzip`-Informationen zu erweitern, müssen Sie die Definition des „common“-Formates folgendermaßen

```
ssh-nickles - PuTTY
00:00:38 +0200] GET /php_jobs/build_thread_cache.php?tid=537021985 HTTP/1.0* 200 17720 mod_gzip: DECLINED: NO_ACCEPT_ENCODING 0pct.
00:00:38 +0200] GET /thread_cache/537021985.html HTTP/1.1* 200 4828 mod_gzip: DECLINED: NO_ACCEPT_ENCODING 73pct.
00:00:39 +0200] GET /thread_cache/537022235.html HTTP/1.1* 200 4267 mod_gzip: OK 76pct.
00:00:40 +0200] GET /at/faq/index.php?cid=12 HTTP/1.1* 200 2653 mod_gzip: DECLINED: NO_ACCEPT_ENCODING 72pct.
00:00:40 +0200] GET /c/c/1846.htm HTTP/1.1* 304 - mod_gzip: SEND_AS_IS: NO 200 0pct.
00:00:40 +0200] GET /forum2/index.php?this_category=32 HTTP/1.1* 200 368 mod_gzip: DECLINED: NO_ACCEPT_ENCODING 41pct.
00:00:50 +0200] GET /c/c/30-0001-150-3.htm HTTP/1.1* 304 - mod_gzip: SEND_AS_IS: NO 200 0pct.
00:00:51 +0200] GET /thread_cache/537021845.html HTTP/1.1* 200 4403 mod_gzip: OK 74pct.
00:00:51 +0200] GET /c/c/000.htm HTTP/1.1* 200 4460 mod_gzip: OK 76pct.
00:00:51 +0200] GET /php_jobs/build_thread_cache.php?tid=537022184 HTTP/1.0* 200 15717 mod_gzip: DECLINED: NO_ACCEPT_ENCODING 0pct.
00:00:51 +0200] GET /thread_cache/537022184.html HTTP/1.1* 200 4188 mod_gzip: DECLINED: NO_ACCEPT_ENCODING 74pct.
00:00:51 +0200] GET /c/c/1/32.htm HTTP/1.1* 200 9123 mod_gzip: OK 89pct.
00:00:52 +0200] GET /at/faq/index.php?cid=12 HTTP/1.1* 200 2653 mod_gzip: DECLINED: NO_ACCEPT_ENCODING 72pct.
00:00:59 +0200] GET /c/c/a.php?i=3&b=0.39145626944492245 HTTP/1.1* 200 5 mod_gzip: SEND_AS_IS: RESPONSE_CONTENT_TYPE_EXCLUDED 0pct.
00:00:59 +0200] GET /c/c/30-0001-150-4.htm HTTP/1.1* 304 - mod_gzip: SEND_AS_IS: NO 200 0pct.
00:01:00 +0200] GET /c/c/30-0001-150-4.htm HTTP/1.1* 200 5419 mod_gzip: OK 75pct.
/var/log/httpd/access_log lines 1-16/13400 OK
```

MOD_GZIP KANN EINE ganze Reihe von Informationen in Ihrem Logfile eintragen.



QB: DAMIT DER CLIENT die komprimierten Dateien auch empfangen kann, muss er das http-1.1-Protokoll verwenden. Beim Internet Explorer ist dies in einigen Versionen extra einzuschalten.

verändern. Am besten bearbeiten Sie eine Kopie der ursprünglichen Definition:

```
LogFormat "%h %l %u %t \"%r\"%"
>>s %b mod_gzip: %{mod_gzip_
compression_ratio}npct."
>>common_gzip
```

Dadurch definieren Sie eine neue Format-Angabe, die unter der Bezeichnung `mod_gzip` verwendet werden kann. Zusätzlich zu den bisherigen Informationen enthält das Logfile eine Information von `mod_gzip` und zwar die Komprimierungsrate der abgerufenen Datei.

Allerdings reicht die Definition eines neuen Formates nicht aus, Sie müssen dem Apache noch mitteilen, dass er diese Format-Definition benutzen soll. Das

```
0 Sep 21 19:19 a.html
43086 Mai 29 15:15 category1.html
8211 Sep 21 19:03 category1.html.gz
41554 Jun 8 23:25 index.php3
41747 Jun 8 23:19 index.php3~
66 Mai 29 14:40 t.php3
8211 Sep 21 19:24 x
```

WENN SIE DIE HTML-DATEIEN „vorkomprimieren“, reicht `mod_gzip` bei der richtigen Konfiguration einfach die komprimierte Dateien weiter, statt die HTML-Datei selbst zu komprimieren und weiterzugeben.

geht über den „CustomLog“-Eintrag in Ihrer Konfigurationsdatei. Angenommen Sie haben das Format von oben – und zwar einschließlich des Namens – verwendet, muss Ihr neuer CustomLog-Eintrag wie folgt aussehen:

```
CustomLog PfadZuIhrerLogDatei
>>common_gzip
```

Starten Sie den Apache erneut. Danach können Sie die Log-Datei mit „tail

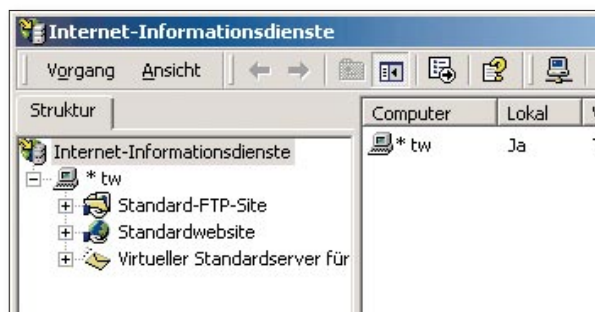
-f“ betrachten. Dort sollten sofort Einträge mit dem folgendem Aussehen erscheinen:

```
212.124.226.3 - -
[18/Jul/2001:00:00:44 +0200]
"GET /index_nocss.htm HTTP/1.1"
200 9521 mod_gzip: 85pct.
```

Wie Sie sehen, enthält die Log-Datei Einträge, die durch Angaben von `mod_gzip` erweitert wurden. Dabei handelt es sich im Wesentlichen um eine Angabe in Prozent. Diese gibt den Komprimierungsfaktor an. Sie beschreibt, wie gut `mod_gzip` die Datei vor dem Transfer komprimieren konnte.

■ HTML komprimieren mit dem IIS

Beim IIS können Sie das Transfer-Encoding deutlich einfacher einschalten.



BEIM IIS KÖNNEN SIE das Transfer-Encoding mit dem Internet-Dienste-Manager einschalten.

Hier müssen Sie nur ein paar Optionen aktivieren – schon werden Ihre Seiten komprimiert ausgeliefert. Führen Sie folgende Schritte aus:

- Starten Sie den Internet-Dienste Manager und öffnen Sie die *Eigenschaften* des Computers.
- Es erscheint der Dialog zur Einstellung der Eigenschaften des IIS. Wechseln Sie auf die Seite *Internet-Informationen-Dienste*.
- Wählen Sie dort unter *Haupteigenschaft* den *WWW-Dienst* aus, und klicken Sie auf *Bearbeiten*.
- Nun erscheint der Dialog *Haupteigenschaften des WWW-Dienstes*. Wechseln Sie hier auf den Reiter *Dienste*.
- Hier finden Sie im Bereich *http-Komprimierung* zwei Optionen. Die eine trägt den Namen *Anwendungsdateien komprimieren*, die

andere *Statische Dateien komprimieren*. Schalten Sie die Option ein, die Sie benötigen.

Für die „statischen“ Dateien können Sie noch ein paar Parameter angeben, etwa ein Pfad zu einem Verzeichnis für die zwischengelagerten Dateien sowie die maximale Größe dafür. Der IIS legt dann einmal komprimierte Versionen von statischen Dateien in diesem Verzeichnis ab. Ändert sich die Original-Datei, kümmert sich der IIS von selbst darum, dass die komprimierte Variante ebenfalls erneuert wird.

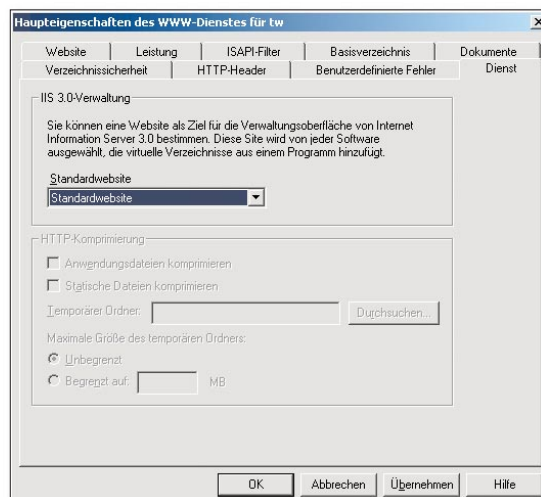
■ Letzte Hinweise

Mit komprimierten HTML-Seiten sparen Sie nicht nur Geld, sondern bieten dem Surfer auch eine schnellere Seite. Das können Sie noch verbessern. Sie brauchen nur die ursprünglichen

HTML-Seiten kompakter gestalten, denn diese sind meist großzügiger, als sie sein müssten. Das gilt besonders dann, wenn Sie HTML-Editoren verwenden.

In diesem Beitrag haben Sie erfahren wie Sie bares Geld sparen können, indem Sie eine praktisch überall zur Verfügung stehende, aber weitgehend unbekannte Methode des Dateitransfers bei HTTP nutzen. Das hilft nicht in allen Fällen weiter, aber sowohl Ihr Rechner als auch Ihre

Anbindung wird dadurch auf jeden Fall besser genutzt. ✔ UR



IE KOMPRIMIERUNG beim IIS: Dabei wird zwischen statischen und dynamischen Seiten unterschieden.