



iConnect e-commerce solutions GmbH  
Gustav-Meyer-Allee 25  
D-13355 Berlin  
(Germany)

TEL +49 (0) 30 46307-390  
FAX +49 (0) 30 46307-388  
info@iConnect.de

AG Charlottenburg  
HRB 72241  
Ust.-IdNr. DE203286333

Geschäftsführer  
Dipl.-Kfm. Stefan Hinz  
Dipl.-Kfm. Hans-Peter Ott

<http://www.iConnect.de>

Konto 2030016686  
Landesbank Berlin  
BLZ 10050000

## Kurs Webserver-Konfiguration

### Inhalt

#### Theoretischer Teil

- Was ist ein Webserver?
- Was ist ein Browser?
- Was ist eine URL?
- Was ist HTTP?
- Was ist HTTPS?
- Client-Server-Prinzip und wichtige Grundbegriffe
- Ressourcen und Weiterführendes

#### Praktischer Teil: Installation und Konfiguration eines Web- und FTP-Servers

- Der verwendete Server
- Verwaltete Verzeichnisstruktur
- Statusmeldungen des Webservers
- Zugriffsprotokollierung
- Hinweise zur Dateiverwaltung
- Webserver-Verwaltung
- Webserver-Konfiguration
- Virtuelle Hosts
- Die Server-Konsole
- Die Konfigurationsdateien

### Voraussetzungen der Teilnehmer

- Sichere Beherrschung eines Betriebssystems (vor allem Dateiverwaltung)
- Netzwerk-Grundkenntnisse (Arbeiten im Netzwerk, Rechteverwaltung usw.)
- Grundkenntnis verschiedener (Netzwerk-) Dienste und Protokolle

### Lernziele

- Installation eines Webservers beherrschen
- Konfiguration eines Webservers selbstständig vornehmen können (grafische Oberfläche, z. B. WBA, und Konfigurationsdatei)
- Kenntnis der wichtigsten Begriffe und Protokolle
- Einsatzmöglichkeiten in Intranets und im Internet kennen

### Zeitlicher Umfang

- 10 bis 20 Unterrichtseinheiten à 45 Minuten (je nach Umfang der Übungen und abhängig vom Stand der Vorkenntnisse)

## Theoretischer Teil

### **Was ist ein Webserver?**

Ein Webserver ist ein Programm, das auf einem – meist ebenfalls Server genannten – Rechner im Netzwerk dafür sorgt, dass jeder Mitarbeiter im Netzwerk die notwendigen Informationen auf Knopfdruck (oder Mausklick) erhält.

Webserver sind auch das Herz des WWW (World Wide Web), also des Internets. Diese Internet-Technik kann innerhalb des Hauses ebenso zur Informationsverteilung genutzt werden wie im Internet. Man spricht dann von einem Intranet.

Ein Webserver verwaltet HTML-Dokumente und andere Internet- / Intranet-Ressourcen und versendet diese über HTTP. Daher wird er auch HTTP-Server genannt.

Bei der Einordnung in die unterschiedlichen Server-Klassen und ihre spezifischen Anforderungen rücken die Aufgabengebiete, die abzudecken sind, in den Vordergrund der Betrachtung. Bei einer Klassifizierung nach dem Einsatzbereich ergeben sich sechs verschiedene Server-Klassen mit jeweils eigenem Anforderungsprofil:

- Ein **Dateiserver** (File-Server) stellt seinen Clients Dateien und Platz auf dem Dateisystem bereit. Zusätzlich übernimmt er die Sicherung der Benutzerdateien.
- Ein **Anwendungsserver** (Application-Server) ermöglicht den Anwendern den Zugriff auf Anwendungsprogramme.
- Auf einem **Datenbankserver** läuft eine mehr oder weniger große Datenbank. Die Aufgabe des Servers ist die Verwaltung und Organisation der Daten, die schnelle Suche, das Einfügen und das Sortieren von Datensätzen.
- Ein **Compute-Server** bietet möglichst viel Rechenleistung. Typische Beispiele für Compute-Server sind Supercomputer à la Cray in Kernforschungsanstalten.
- Ein **Internet-Server** stellt Internet- und Intranet-Dienste bereit. Typische Dienste umfassen das World Wide Web, den Domain Name-Service, FTP sowie E-Mail.
- (Streaming-) **Media-Server** stellen Multimedia-Daten (z.B. Audio- und Video-Clips) in Echtzeit und hoher Dienstqualität zur Verfügung.

Die verbreitetsten Webserver sind: Das Open-Source-Produkt Apache (Marktanteil im Internet ca. 90%) und Microsofts IIS (Internet Information Server). Daneben gibt es eine Fülle "kleiner" Webserver-Applikationen wie Xitami von iMatix und Omni-HTTP.

### **Was ist ein Browser?**

Ein Browser ist ein Programm zur Anzeige von HTML-Seiten, den Texten des Inter- und Intranets. Die verbreitetsten Browser sind Microsoft Internet Explorer, Netscape Navigator und Opera. Browser sind die Client-Applikationen, die auf Webserver über HTTP zugreifen.

### **Was ist eine URL?**

URL ist die Abkürzung für Unified Resource Locator (vereinheitlichte Ortsangabe für Ressourcen). Dies ist eine eindeutige Adresse im Inter- oder Intranet, ähnlich wie eine Telefonnummer. Eine URL-Adresse hat entweder einen sprechenden Namen (z.B. "iconnect.de") oder besteht aus der "nackten" IP-Adresse (z.B. 195.20.252.222). Vollständig betrachtet kann eine URL aus folgenden Komponenten bestehen:

1. Bezeichnung des Internet-Dienstes bzw. des entsprechenden Übertragungsprotokolls: HTTP für Web-Dokumente, FTP für Übertragung grosser und / oder vieler Dateien (File Transfer Protocol), NEWS für Newsgroups (das Usenet),

2. der Domain (z. B. iconnect) und der Landes-Kennung (Top Level Domain, TLD, z. B. de),
  3. optional aus Subdomains (so ist in "buero.iconnect.de" "buero" eine Subdomain),
  4. optional dem Port (meist 80),
  5. optional dem Pfad auf dem Server
  6. und dem Namen des Dokuments beziehungsweise allgemeiner der Ressource.
- Beispiel: <http://buero.iconnect.de/verwaltung/vertraege/agb.html>

### Was ist HTTP?

Abkürzung für "Hypertext Transfer Protocol". HTTP ist das Kommunikationsprotokoll zwischen Web-Server und Web-Browser zur Übertragung von HTML- und anderen Dateien. HTTP wurde 1990 am Genfer CERN von Tim Berners-Lee (s. Bild) entwickelt, der damit die Grundlage für das World Wide Web schuf. Das HTTP-Protokoll stellt die oberste von mehreren Protokoll-"Schichten" zur Verwaltung im Internet dar:



- Das Internet Protocol (IP) ist die Grundlage, die das Internet definiert,
- das Transport Control Protocol (TCP) fungiert als Zwischenschicht und richtet die Verbindungswege zur Datenübertragung ein,
- das HTTP-Protokoll teilt in der obersten Schicht die Daten in einzelne Pakete auf und legt fest, wie diese verschickt werden sollen.

HTTP 1.1 ist in RFC 2068 definiert, die Vorgängerversion HTTP 1.0 in RFC 1945. RFC steht für "Request for Comment" und stellt quasi das Amtsblatt des Internets dar.

Eine internationale Organisation zur Festlegung von Standards hat im Juli 1999 das Internet-Protokoll HTTP 1.1 zu einem vorläufigen Standard erklärt. Mit der neuen Version werden nach Angaben der Entwickler des World Wide Web Consortium (W3C) und der Internet Engineering Task Force (IETF), die dem Status als vorläufiger Standard zugestimmt hat, einige der Designfehler von HTTP 1.0 behoben und die Geschwindigkeit der Informationsübertragung erhöht. Während HTTP 1.0 die Datenübertragung dadurch verlangsamte, dass für jedes versandte Datenpaket ein neuer Verbindungsweg eingerichtet werden musste, ist die neue Version in der Lage, mehrere Pakete über denselben Weg zu schicken. Darüber hinaus ermöglicht es HTTP 1.1 den Anbietern von Inhalten, genauer festzulegen, welche Inhalte zwischengespeichert werden können. Auf diese Weise können Internet-Dienstleister Kopien häufig abgefragter Daten so speichern, dass sie dem Kunden leichter zugänglich sind. So kann der Weg reduziert werden, den die Informationen im Internet zurücklegen müssen.



## Was ist HTTPS?

Die verschlüsselte Variante von HTTP heisst HTTPS (secure HTTP). Hierzu wird das HTTP-Protokoll um eine zusätzliche Schicht (englisch layer) ergänzt, den Secure Socket Layer (SSL).

Die Stärke der Verschlüsselung hängt von der Länge des verwendeten Schlüssels ab, die in Bit angegeben wird. Ein gängiger Wert ist 128 Bit.

Der "Gegenpart" des Webserver, nämlich der Browser, muss die Verschlüsselung ebenfalls unterstützen. Hier gilt, dass das schwächste Glied in der Kette die Stärke der Verschlüsselung bestimmt. Unterstützt der Browser z. B. nur 40 Bit, ist die Verschlüsselung 40 Bit stark, selbst wenn der Webserver 128 Bit bietet.

Die meisten Browser zeigen verschlüsselte Verbindungen auf zweierlei Arten an: Zum einen in der Adressleiste (location bar), zum anderen durch ein Schloss-Symbol in der Statusleiste (status bar). Beides ist aus obenstehender Abbildung ersichtlich.

## Client-Server-Prinzip und wichtige Grundbegriffe

Das Client-Server-Prinzip sollte zum Verständnis der Arbeit mit dem Webserver (und dem FTP-Server) „verinnerlicht“ worden sein. Der Zusammenhang mit den entsprechenden Netzwerk-Protokollen sollte klar sein.

### Wichtige Begriffe

#### Benutzer (User)

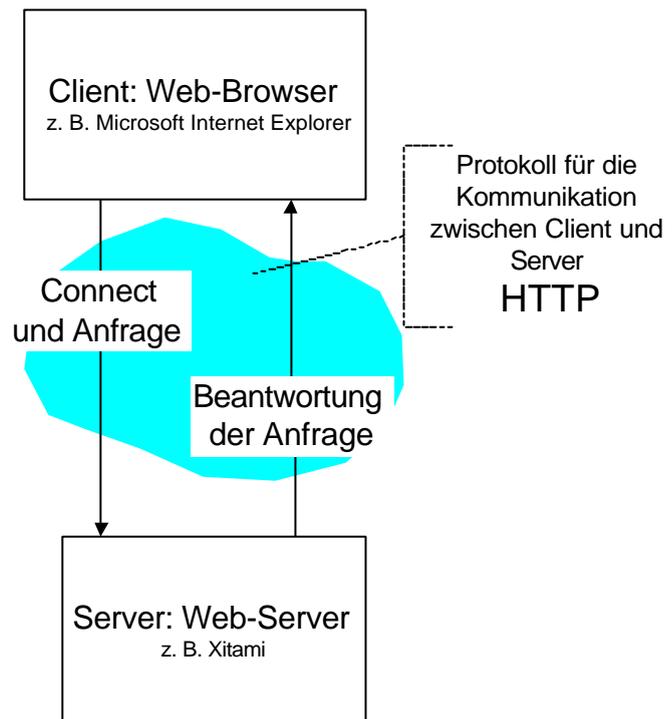
Das ist nicht (direkt) der Mensch an der Konsole, sondern das Programm, mit dem dieser Mensch auf den Server zugreift. Das kann z. B. ein Web-Browser sein, der sich über HTTPS mit einem Webserver verbindet.

#### Client und Server

Häufig wird das Wort Server etwas unzutreffend für eine physikalische Maschine benutzt, das Wort Client für eine andere physikalische Maschine. Richtig ist, dass ein Server ein Prozess ist, der es anderen Prozessen (nämlich den Clients) erlaubt, auf ihn zuzugreifen. Den Clients stellt er Dienste (englisch: services) zur Verfügung – daher auch der Name. Solch ein Client-Prozess kann auf der selben physikalischen Maschine ablaufen wie der Server-Prozess; er kann aber auch von einer anderen physikalischen Maschine kommen. Hinsichtlich Sicherheitsaspekten heisst das, dass man Clients bestimmte Rechte einräumt, auf den Server zugreifen zu können.

#### Host

Der Host (englisch für „Gastgeber“) ist derjenige physikalische oder virtuelle Rechner, auf dem der Prozess läuft, den man momentan meint. Sprechen wir also vom Client(-prozess), der mittels Browser auf den Webserver zugreift, ist der relevante Host der Client-Rechner. Dieser Host kann z. B. buero.icconnect.de heissen. Der Webserver kann auf einem ganz anderen Host laufen, z. B. auf dokumente.icconnect.de, aber prinzipiell auch auf demselben Host.



Oft wird es der Fall sein, dass Client-Host und Server-Host auf unterschiedlichen physikalischen Maschinen laufen. So könnte z. B. buero.iconnect.de ein Linux-Rechner in Berlin, Deutschland sein, und calendar.vielefirmen.com ein Solaris-Rechner in Newark, NJ, USA. Es gibt aber auch den Fall der virtuellen Hosts, von denen mehrere auf ein und der selben physikalischen Maschine ablaufen. Das beinhaltet im Extremfall, dass Client-Host und Server-Host auf der selben Maschine laufen.

Eine Besonderheit ist der **Localhost**. Dieser verfügt über eine IP-Nummer (IP = internet protocol), die ausserhalb der physikalischen Maschine keine Gültigkeit hat. Hierfür findet meist die nur auf der Maschine gültige IP-Nummer 127.0.0.1 Verwendung.

### Connect und Login

Unter Connect versteht man das Herstellen einer Verbindung von einem Client zu einem Server. Oft wird es ein Server nicht zulassen, dass man sich anonym verbindet (Ausnahmen sind z. B. Guest Account, anonymes FTP oder Zugriffe auf öffentliche HTML-Seiten, die ein Webserver verwaltet). Daher ist nach erfolgtem Connect ein Login notwendig, zu meist mit einer Kombination Benutzername (user) und Passwort (password).

- **Benutzername**, meist in Kleinschreibung. Achtung: Unix-Systeme sind im Gegensatz zu Windows-Systemen case-sensitive, unterscheiden also Gross- und Kleinschreibung! Daher ist „john“ nicht dasselbe wie „John“. Im Internet, zunehmend aber auch in Intranets, kommen für Webserver und FTP-Server zumeist Unix-Maschinen zum Einsatz. So laufen z. B. mehr als 50% aller Internet-Server unter Linux, weitere 20% unter Sun Solaris.
- **Passwort**: Dieses sollte den üblichen Sicherheitsregeln entsprechen, was letztlich heisst, dass es nicht leicht zu erraten sein darf. Um auch Brute-Force-Angriffen gegenüber gewappnet zu sein, sollte es aus einer Kombination von Klein- und Grossbuchstaben sowie Zahlen bestehen. Das erschwert natürlich den Login-Prozess. Für Intranet-Accounts wird man daher häufig weniger sichere Passwörter wählen. (Nebenbemerkung: Brute-Force-Angriffe („rohe Gewalt“) verwenden z. B. Lexikon- und / oder Namen-Listen, um Passwörter zu erraten. Dieser werden dann einfach der Reihe nach durchprobiert, natürlich nicht durch Menschen, sondern durch entsprechende Programme. Man kann einen Linux-Server so einstellen, dass er bei falscher Passwort-Eingabe z. B. 3 Sekunden wartet, bis er die nächste Login-Aufforderung gibt. Damit erreicht man zumindest, dass ein Angreifer-Programm nicht Hunderte von Passwort-Versuchen pro Sekunde ausführen kann.)

### Ressourcen und Weiterführendes

- Anlaufadresse für den am meist verbreiteten Webserver Apache: [www.apache.org](http://www.apache.org)
- Zum Thema Webserver-Sicherheit: [www.koehntopp.de/kris/artikel/websec/](http://www.koehntopp.de/kris/artikel/websec/)
- Interessanter Webserver für den Einstieg: [www.xitami.com](http://www.xitami.com)

## Praktischer Teil: Installation und Konfiguration eines Web- und FTP-Servers

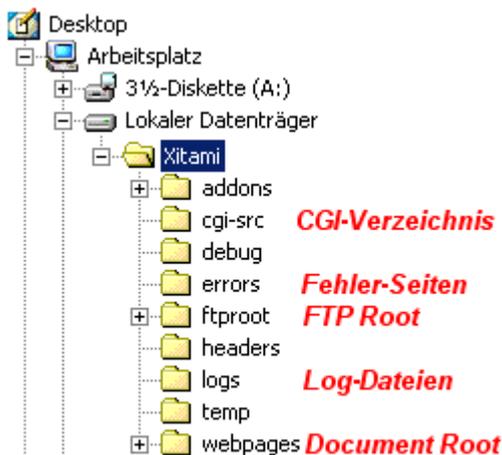
### Der verwendete Server

Im Rahmen des Seminars wird als Webserver (und FTP-Server) das Open-Source-Produkt Xitami der Firma iMatix verwendet. Xitami läuft auf diversen Betriebssystemen (Windows, Unix und OS/2). Die Installationsdatei ist sehr klein (< 1 MB) und auf Windows-Systemen extrem einfach zu installieren. Dennoch bietet das Produkt volle Web- und FTP-Server-Funktionalität. Es ist einfacher zu erlernen (also letztlich zu konfigurieren) als der "grosse Bruder" Apache, dessen Konfigurationsdatei mit stolzen 1700 Zeilen Länge den angehenden Webserver-Administrator etwas abschrecken dürfte.

Während des Installationsprozesses mit dem Setup-Assistenten gibt es nur eine wichtige Seite, die nicht leichtfertig durchgeklickt werden sollte. Auf dieser Seite werden **Benutzername und Passwort** für den Server-Administrator festgelegt. Diese Kombination sollte man besser nicht vergessen! ☺

### Verwaltete Verzeichnisstruktur

Nach erfolgreicher Installation hat Xitami eine Reihe von Unterverzeichnissen angelegt, die kurz erläutert werden sollen:



- Ins CGI-Verzeichnis gehören Skripte, die von HTML-Dokumenten aus aufgerufen werden, z. B. PERL-Skripte.
- Die Fehler-Seiten können vom Webserver-Administrator selbst gestaltet werden, um z. B. dem Corporate Design und der restlichen Website Genüge zu tun. Wird z. B. eine vom Browser angeforderte Web-Seite nicht gefunden, meldet der Webserver den Fehler 404 (not found) – siehe Auflistung der Statuscodes weiter unten.
- FTP Root: In diesem Verzeichnis – bzw. seinen Unterverzeichnissen – landet man, wenn man sich mit dem FTP-Server verbindet. Hier können Dateien zum Download für die interessierte Öffentlichkeit bereit gestellt werden (public) und Upload-Möglichkeiten für Gäste (guest) zur Verfügung gestellt werden.

- Log-Dateien: Der Webserver protokolliert jede Verbindung minutiös mit. Die wichtigsten Log-Dateien sind das ACCESS.LOG (Dokumentenforderungen von Browsern – Beispiel siehe unten) und das ERROR(S).LOG (Fehlermeldungen des Webserver).
- Document Root: Das ist das Wurzelverzeichnis, ab dem der Webserver das Dateisystem verwaltet ("exportiert"). Der Webserver hat keinen Zugriff auf Dateien oberhalb dieses Verzeichnisses. Während der Webserver-Administrator weiss, dass sich eine Datei "index.htm" z. B. in C:\Xitami\webpages\ befindet, sieht ein Browser dieselbe Datei nur in "Root" (also "/").

### **Statusmeldungen des Webservers**

Den Zustand (Status) der Anfrage meldet der Webserver mittels eines normierten Status-Codes. Der Webserver kann so konfiguriert werden, dass er daraufhin ggf. bestimmte Seiten herausgibt (zum Beispiel 404 – nicht gefunden). Leider wird diese Einstellung von einigen Browsern überschrieben; so gibt Internet Explorer beispielsweise eigene 404-Seiten heraus.

Statuscode	Bedeutung
200	Anforderung war verständlich und eine korrekte Antwort wurde zurückgesendet
201	Resource wurde erzeugt (Verzeichnispfad wird übermittelt)
202	Anforderung wurde angenommen und wird noch bearbeitet
204	Anforderung war verständlich, jedoch kein Inhalt zum zurücksenden
301	das angeforderte Dokument wurde dauerhaft verschoben (URL im Location-Header)
302	das angeforderte Dokument wurde vorübergehend verschoben (URL im Location-Header)
304	Datei wurde angefordert, falls diese neueren Datums ist - dies war jedoch nicht der Fall
400	Anforderung war nicht verständlich (Browserproblem)
401	eine Datei wurde angefordert, für die sich der Client ausweisen muß
403	Anforderung war verständlich, der Server weigert sich jedoch das Dokument zu senden, da dieser oder der Client keine Berechtigung haben
404	Datei wurde nicht gefunden
500	interner Fehler, CGI-Programm ist entweder fehlerhaft geschrieben oder konfiguriert
501	Anforderung ist nicht beim Server implementiert
502	Proxy- oder Gateway-Server erhielt eine ungültige Antwort von einem anderen Server
503	Server ist ausgelastet und kann keinen weiteren Anforderungen nachkommen

### **Zugriffsprotokollierung**

Jede Anforderung an den Webserver ("sende mir eine HTML-Seite", "sende mir ein Bild") wird protokolliert. Jeder dieser Zugriffe findet sich in der Datei ACCESS.LOG in einer eigenen Zeile wieder. Hier ein Beispiel mit Erläuterung:

```
192.168.10.41 - - [28/Jun/2001:16:58:53 +0100] "GET /images/xitami1.gif HTTP/1.1" 200 5104 "http://mephisto/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"
```

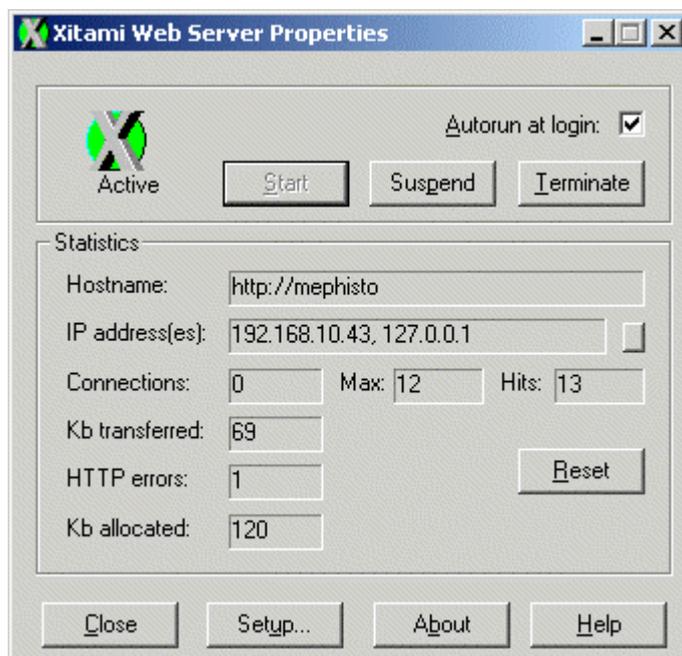
- 192.168.10.41: Die IP-Nummer des anfragenden Hosts
- [28/Jun/2001:16:58:53 +0100]: Datum, Uhrzeit und Zeitzone der Anforderung
- "GET /images/xitami1.gif HTTP/1.1": Anforderungsart, angeforderte Datei und Übermittlungsprotokoll
- 200: Statuscode (s. o.)
- 5104: Anzahl der übermittelten Bytes
- "http://mephisto/": Referer (= von wo die Datei angefordert wurde). Wenn der Referer "" ist, heisst das, dass die Datei direkt vom Browser angefordert wurde.
- "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)": Der anfordernde Client (Browser, hier: Microsoft Internet Explorer 5.01 [5.5] auf Windows 2000)

### Hinweise zur Dateiverwaltung

Wenn der Webserver auf einer Windows-Maschine läuft, wird Gross-/Kleinschreibung in der Regel keine Rolle spielen. Für Webserver-"Anwender" ist wichtig zu wissen, wie man Problemen mit Dateien vorbeugt, die möglicherweise auf unterschiedlichen Betriebssystem-Umgebungen eingesetzt werden (z. B. HTML-Dokumente, Bilder). Hier die beiden wichtigsten Praxis-Tipps:

- Leerzeichen und Sonderzeichen in Dateinamen vermeiden: Um Problemen vorzubeugen, sollte man sich Dateinamen wie "Mein größtes und schönstes Bild.JPG" abgewöhnen. Lange Dateinamen dagegen sind okay (mein\_groesstes\_und\_schoenstes\_bild.jpg).
- Kleinschreibung in Dateinamen verwenden: Um sich selbst den Überblick zu erleichtern, sollte man Dateinamen immer klein schreiben. Es ist schon ärgerlich, wenn man mein\_nettes\_bild.jpeg sucht und es nicht findet, weil es Mein\_nettes\_Bild.JPEG heisst.

### Webserver-Verwaltung



Nachdem Xitami gestartet wurde, befindet sich eine X-förmiges Symbol im System Tray. Per Doppelklick auf dieses Symbol lässt sich das Eigenschaften-Fenster des Webserver aufgerufen (Properties). Folgende Einstellungen lassen sich vornehmen:

- Autorun at login: Wenn aktiviert, wird Xitami beim Hochfahren von Windows automatisch geladen und gestartet.
- Start: Webserver starten, wenn er im Zustand "angehalten" ist.
- Suspend: Webserver anhalten.
- Terminate: Webserver anhalten und beenden.

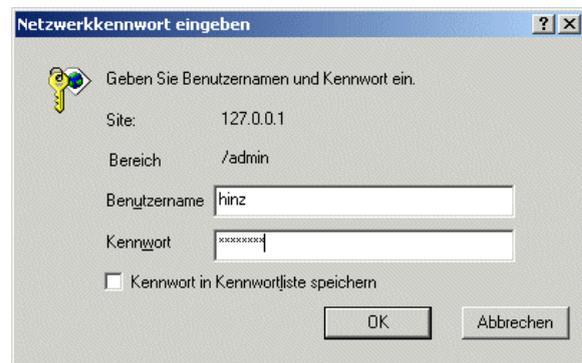
- **Reset:** Statistik-Informationen zurücksetzen.
- **Close:** Das Properties-Fenster schliessen.
- **Setup...:** Zum Web Based Admin (WBA, s. u.) wechseln.
- **About:** Lizenzinformationen anzeigen.

Daneben zeigt der Webserver unter "Statistics" noch einige nützliche Informationen an:

- **Hostname:** Der Name des Rechners, auf dem der Webserver läuft. Unter diesem Namen lässt sich der Webserver auch von anderen Rechnern im Netzwerk aus erreichen.
- **IP address(es):** IP-Adresse(n), unter der der Webserver erreichbar ist. Unter 127.0.0.1 (localhost) ist er nur von der Maschine aus erreichbar, wo der Webserver läuft, unter der anderen IP-Adresse auch von anderen Rechnern im Netzwerk aus.
- **Connections:** Derzeit bestehende Verbindungen zu Clients (Browsern).
- **Hits:** Anzahl der angeforderten Dateien (aktuelle Verbindungen).
- **Kb transferred:** Übermittelte Datenmenge in Kilobytes (aktuelle Verbindungen).
- **HTTP errors:** Anzahl der bei der aktuellen Übermittlung aufgetretenen Fehler.

### Webserver-Konfiguration

Mit Klick auf Setup... im Eigenschaften-Fenster gelangt man in den grafischen Administrationsmodus von Xitami, den WBA (Web Based Admin). Diese findet per Browser statt. Wenn der Browser-Aufruf nicht klappen sollte, kann man auch durch Eingabe von <http://localhost/admin> oder <http://127.0.0.1/admin> in den WBA gelangen. In jedem Fall müssen jetzt Benutzername und Passwort des Webserver-Administrators eingetragen werden.



Unter "**Configuration**" gelangt man zu den wichtigsten Konfigurationsmöglichkeiten des Webserver, den Server Properties (Abbildung auf der Folgeseite).

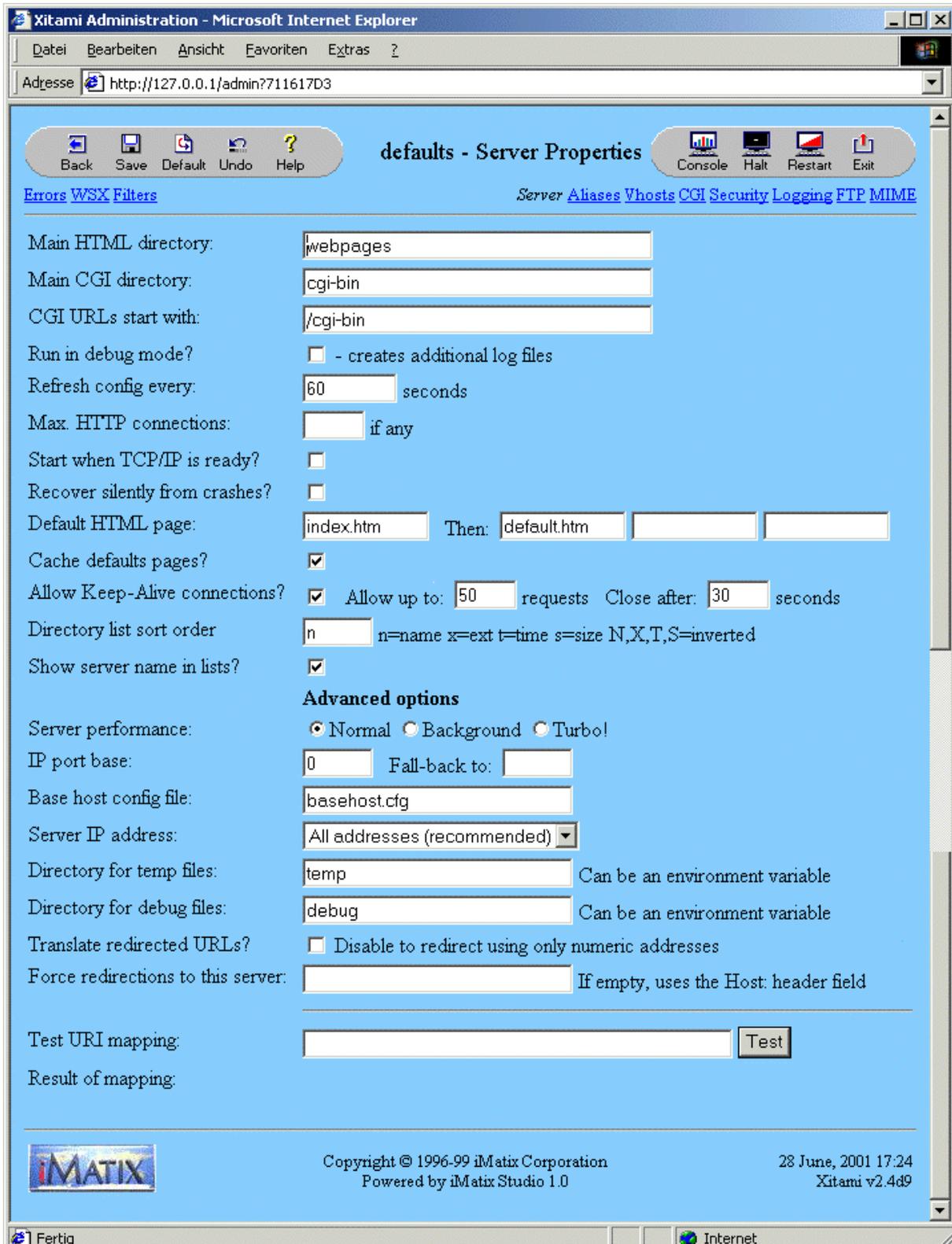
Die Schaltflächen oben links und rechts haben zwei wichtige Symbole:

- **Save:** Hiermit werden Änderungen gespeichert.
- **Restart:** Hiermit wird der Webserver manuell neu gestartet und die Konfigurationsdateien (und damit die gemachten Änderungen) werden neu eingelesen.

Einige wichtige Konfigurationsmöglichkeiten:

- **Main HTML directory:** Legt fest, wo Document Root sein soll. Das kann prinzipiell jedes Verzeichnis unterhalb des Xitami-Installationsverzeichnis sein. Voreingestellt ist "webpages".
- **Refresh config every ... seconds:** Legt fest, wie oft die Konfigurationsdateien automatisch neu eingelesen werden.
- **Default HTML page:** Legt bis zu 4 Dateinamen fest, nach denen im vom Browser angewählten Verzeichnis gesucht wird, bevor der Index (die Liste der im Verzeichnis befindlichen Dateien) gezeigt wird. Die Dateinamen werden hierarchisch von links nach rechts ausgewertet.
- **Server IP address:** Hier lässt sich einstellen, ob nur localhost (127.0.0.1), nur die extern gültige IP-Nummer der Maschine oder beide für Connects zum Webserver zur

Verfügung stehen. Eine super-sichere Einstellungen, zum Beispiel für Tests, ist nur 127.0.0.1; allerdings ist der Webserver dann von anderen Rechnern aus nicht erreichbar.

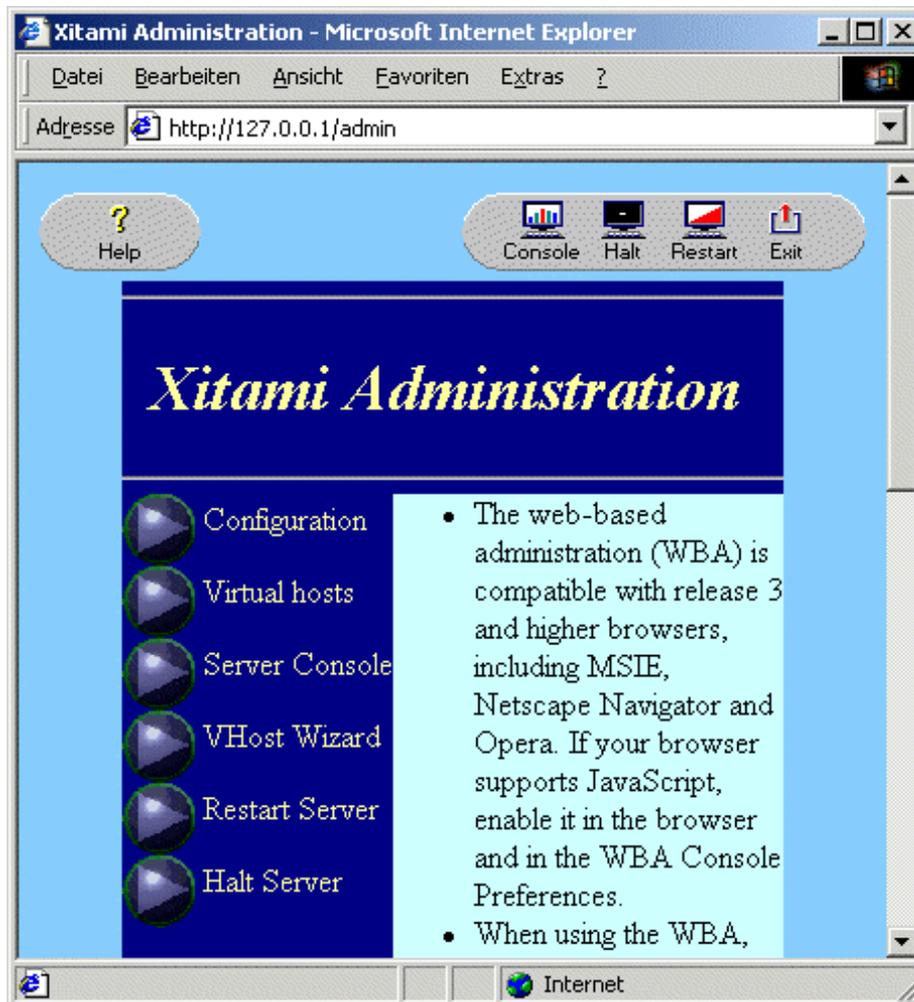


The screenshot shows the Xitami Administration interface in Microsoft Internet Explorer. The browser window title is "Xitami Administration - Microsoft Internet Explorer". The address bar shows "http://127.0.0.1/admin?711617D3". The main content area is titled "defaults - Server Properties" and contains various configuration options for the web server. The interface is light blue with a navigation bar at the top containing icons for Back, Save, Default, Undo, Help, Console, Halt, Restart, and Exit. Below the navigation bar are links for Errors, WSX Filters, Server, Aliases, Vhosts, CGI, Security, Logging, FTP, and MIME. The configuration options include:

- Main HTML directory: webpages
- Main CGI directory: cgi-bin
- CGI URLs start with: /cgi-bin
- Run in debug mode?  - creates additional log files
- Refresh config every: 60 seconds
- Max. HTTP connections: if any
- Start when TCP/IP is ready?
- Recover silently from crashes?
- Default HTML page: index.htm Then: default.htm
- Cache defaults pages?
- Allow Keep-Alive connections?  Allow up to: 50 requests Close after: 30 seconds
- Directory list sort order: n n=name x=ext t=time s=size N,X,T,S=inverted
- Show server name in lists?
- Advanced options**
- Server performance:  Normal  Background  Turbo!
- IP port base: 0 Fall-back to:
- Base host config file: basehost.cfg
- Server IP address: All addresses (recommended)
- Directory for temp files: temp Can be an environment variable
- Directory for debug files: debug Can be an environment variable
- Translate redirected URLs?  Disable to redirect using only numeric addresses
- Force redirections to this server: If empty, uses the Host: header field
- Test URI mapping: Test
- Result of mapping:

At the bottom of the page, there is a copyright notice: "Copyright © 1996-99 iMatix Corporation Powered by iMatix Studio 1.0" and a date/time stamp: "28 June, 2001 17:24 Xitami v2.4d9". The browser status bar shows "Fertig" and "Internet".

Das Hauptmenü "Configuration" bietet weitere Optionen – etwas verwirrend – mehrfach an:



- **Console** bzw. **Server Console** zeigt wichtige Status-Informationen zu Web- und FTP-Server an (Beschreibung unten).
- **Halt** bzw. **Halt Server** macht dasselbe wie "Suspend" im Properties-Fenster: Es hält den Server an.
- **Restart** bzw. **Restart Server** macht dasselbe wie "Start" im Properties-Fenster: Es startet den Server neu.
- **Virtual Hosts** verwaltet die virtuellen Server. Mit **VHost Wizard** lassen sich neue virtuelle Server einfach

anlegen. (Zu virtuellen Hosts siehe unten.)

- **Exit** beendet den WBA.

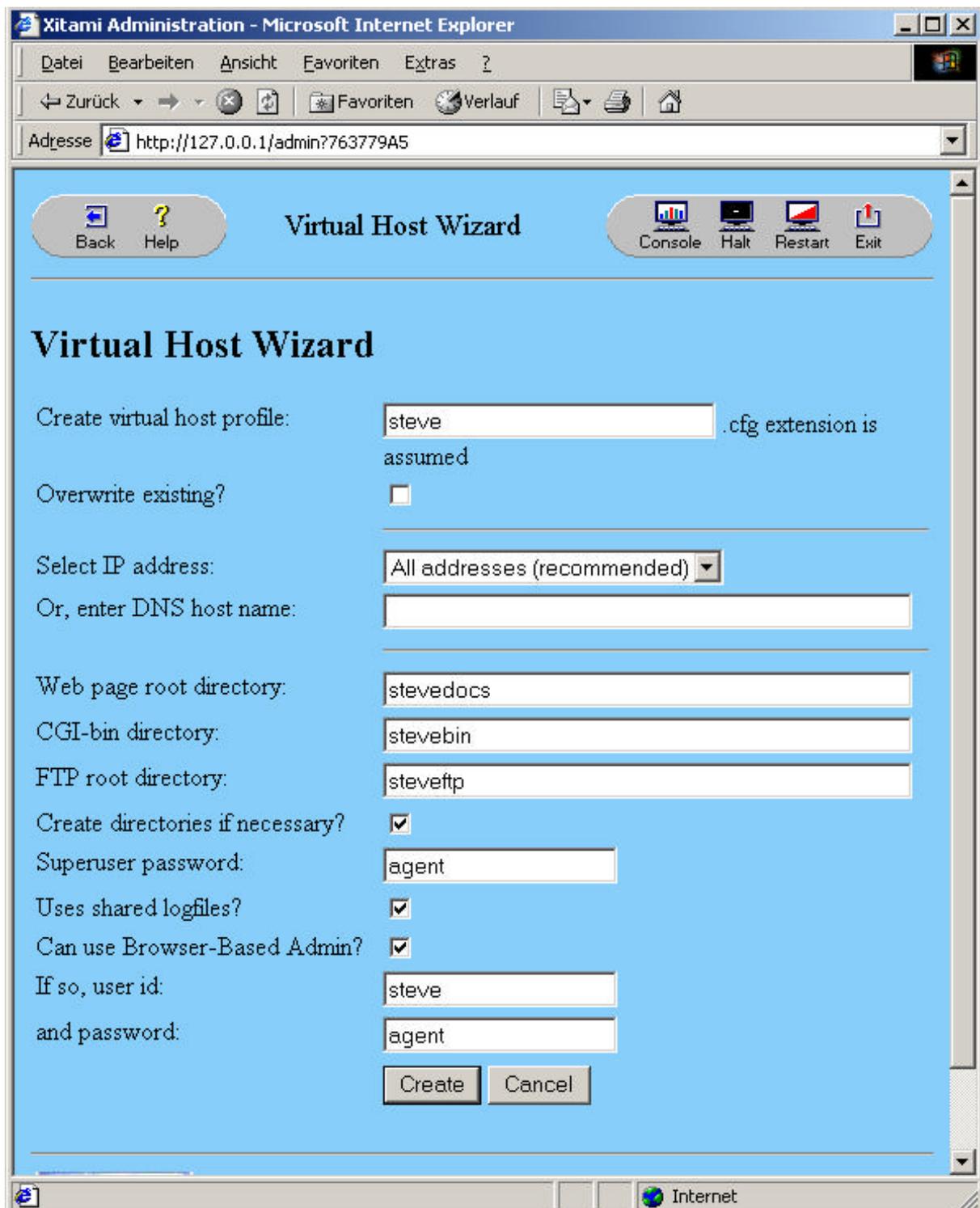
### Virtuelle Hosts

Virtuelle Hosts unterscheiden sich von echten (physikalischen) Hosts dadurch, dass scheinbar mehrere Maschinen (die aber eben nur virtuell sind) unter derselben IP-Nummer erreichbar sind. Aus Gründen der Exaktheit: Prinzipiell kann jeder virtuelle Host eine eigene IP-Nummer besitzen, oft wird jedoch ein und dieselbe IP-Nummer für viele Hosts verwendet. Ein Extremfall ist beim Internet Service Provider Strato gegeben; hier laufen mehrere Millionen virtuelle Hosts ("Domains") unter einer einzigen IP-Nummer!

Xitami legt bei der Installation bereits einen ersten "virtuellen" Host an, mit dem Namen "defaults". (Streng genommen ist dieser Host solange noch "echt" – also physikalisch, bis der nächste virtuelle Host angelegt wurde.)

Jeder virtuelle Host erhält eine eigene Konfigurationsdatei, optional eigene Logdateien (diese können auch mit anderen Hosts geteilt werden) und eigene Startverzeichnisse (Document Root, FTP-Root, CGI-Verzeichnis). Hierdurch zeigt sich ein virtueller Host von aussen betrachtet wie eine eigene physikalische Maschine. Lediglich anhand der IP-Nummer kann festgestellt werden, dass mehrere Hosts auf einer einzigen Maschine laufen – und auch nur dann, wenn die virtuellen Hosts sich eine IP-Nummer teilen.

Damit ein virtueller Host unter seinem Namen erreichbar ist, muss entweder einem DNS-Server oder der lokalen Maschine mitgeteilt werden, unter welcher IP-Nummer der Name des Hosts zu finden ist. Auf Windows-Maschine ist in letzterem Fall die Datei **hosts** hierfür zuständig. Diese befindet sich – abhängig von der Windows-Version – zum Beispiel in C:\winnt\system32\drivers\etc, bei Unix-Systemen meist in /etc.

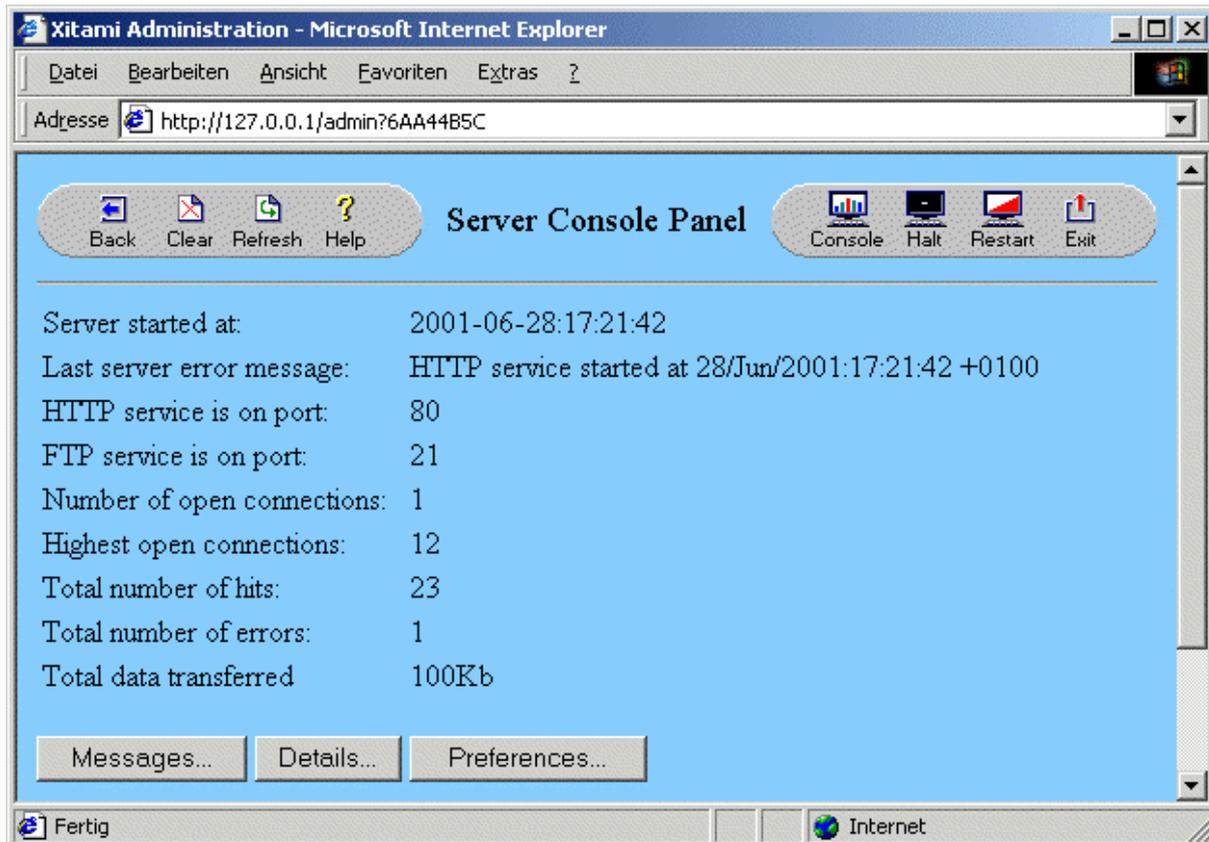


Sobald ein virtueller Host eingerichtet wurde und die Konfigurationsdateien durch den Webserver neu eingelesen wurden, ist dieser dann unter seinem Namen erreichbar. Die fol-

gende Abbildung zeigt beispielhaft die Einrichtung des virtuellen Hosts "steve", der dann unter <http://steve> angewählt werden kann.

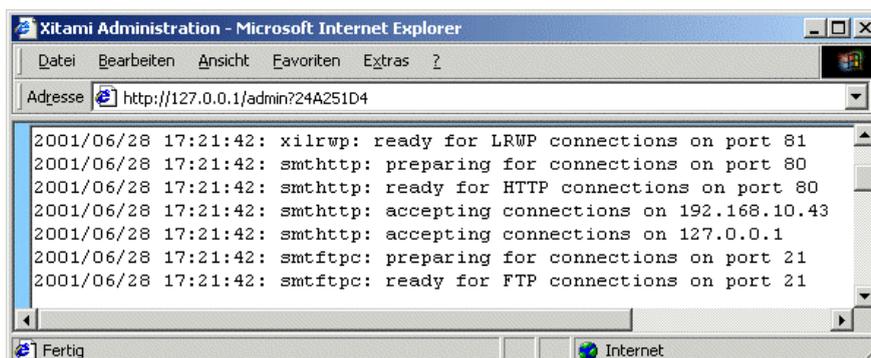
### Die Server-Konsole

Xitami WBA bietet unter "Console" das "Server Console Panel". Dieses enthält einige wichtige Informationen zum Zustand von Web- und FTP-Server:



- Uptime des Servers (wann er gestartet wurde),
- den HTTP-Port, unter dem der Webserver erreichbar ist (allgemeiner Standard hierfür ist Port 80),
- den FTP-Port, unter dem der FTP-Server erreichbar ist (hier ist der allgemeine Standard Port 21).

Unter **Messages...** können die Meldungen von Xitami eingesehen werden (Console Messages). Hier sind unter anderem noch einmal die Ports zu sehen. Ausserdem lässt sich feststellen, unter welchen IP-Adressen die Server (Web und FTP) erreichbar sind.



## Die Konfigurationsdateien

Bei der Installation von Xitami werden die ersten beiden Konfigurationsdateien erzeugt:

- xitami.cfg
- defaults.cfg

Für jeden virtuellen Host, der angelegt wird, wird eine weitere Konfigurationsdatei erzeugt. Diese erhält typischerweise den Namen des virtuellen Host; so wird z. B. der Host "steve" die Konfigurationsdatei "steve.cfg" erhalten.

Die Konfigurationsdateien werden von Xitami beim Start – und dann vorgabemässig alle 60 Sekunden – in folgender Reihenfolge eingelesen:

1. xitami.cfg
2. defaults.cfg
3. Konfigurationsdateien für die virtuellen Hosts

Einträge und Änderungen, die der Webserver-Administrator im WBA vornimmt, werden beim Klick auf "Save" in die jeweiligen Konfigurationsdateien geschrieben. Sie werden beim nächsten Laden der Konfigurationsdateien wirksam.

Man kann die Konfigurationsdateien alternativ zum WBA auch händisch editieren, wobei iMatix davon abrät, Änderungen in der Hauptdatei xitami.cfg vorzunehmen.

Beispiel für den Aufbau der Konfigurationsdateien (aus **xitami.cfg**):

```
[Server]
    ipaddress=*                # * = use any IP address on host
    webpages=webpages         # Relative or full path
    defaults=defaults.cfg     # Local config file
```

Abschnitte (sections) werden in eckigen Klammern geschrieben. Die Raute (#) dient als Anfangszeichen für einen Kommentar (alles ab # bis zum Zeilenende wird vom Server nicht beachtet).

Wenn sich Einträge in den Konfigurationsdateien "widersprechen", wird der Wert der zuletzt eingelesenen Datei genommen. So lässt sich beispielsweise der Eintrag für "webpages" (also Document Root) in der Datei **defaults.cfg** wie folgt ändern:

```
[Server]
    webpages=htdocs           # Relative or full path
```

Für den virtuellen Host "steve" würde der entsprechende Eintrag in der **steve.cfg** dann wie folgt lauten:

```
[Server]
    webpages=stevedocs        # Relative or full path
```

Somit gelangt man bei Anwahl des ersten "echten" Hosts (nennen wir ihn "mephisto", also mit <http://mephisto>) nach C:\Xitami\htdocs (Windows-Systeme), bei Anwahl des virtuellen Hosts "steve" (mit <http://steve>) nach C:\Xitami\stevedocs.