

Webseiten durchsuchen mit ht://Dig

[Martin Sommer](#)

Inhaltsverzeichnis

- [Was ist ht://Dig](#)
- [Voraussetzungen](#)
- [Installation](#)
- [Die einzelnen Programme](#)
- [Konfiguration](#)
 - [Das Suchformular](#)
 - [Die Konfigurationsdatei](#)
 - [Die wichtigsten Attribute](#)
 - [Die Ausgabemplates](#)
- [Start und Automatismus](#)
- [Trickreiches zum Abschluß](#)
- [Links](#)

Dieser Artikel bietet eine Einführung in die Benutzung der freien Suchmaschine ht://Dig. Aufgrund der Komplexität dieses Programmpaketes kann er nicht alle Features und Funktionen abhandeln, versucht aber, für den Alltag sinnvolle Optionen vorzustellen und zu erklären. Die in den Beispieldateien dargestellten Konfigurationen decken einen Großteil der Anforderungen an eine komfortable Suchmaschine für eigene Homepages und Intranets ab. Im folgenden sind mit "ht://Dig" das ganze Programmpaket und mit "*htdig*", "*htmerge*" usw. die einzelnen Programme bezeichnet.

Was ist ht://Dig

ht://Dig ist eine freie Suchmaschine, die für die Volltextsuche in Homepages bzw. Intranets programmiert wurde. Sie bietet eine Vielzahl von nützlichen Optionen, die relativ einfach zu konfigurieren und auf der [Homepage von htdig](#) alle hervorragend kommentiert sind. Die herausragenden Features sind die Fuzzy-Suche (ergänzt Substrings zu Wörtern, findet ähnlich klingende sowie alle möglichen Endungen), die Boole'sche Suchmethode (mit AND, OR und NOT) und die (fast) freie Editierbarkeit der Ausgabemplates. Auf der [Homepage](#) finden Sie eine Liste mit den wichtigsten Merkmalen.

Voraussetzungen

htdig läuft unter diversen Unix- und Linuxsystemen und ist unter der GNU GPL veröffentlicht. In dem Artikel wird davon ausgegangen, daß ht://Dig in SuSE Linux per *YaST* installiert wird. In anderen Distributionen ist natürlich die Installation als rpm-Paket möglich. Die Datei *htdig.rpm* liegt z.B. auf dem SuSE FTP-Server unter: <ftp://ftp.suse.com/pub/suse/i386/7.1/suse/n1/htdig.rpm> zum Download bereit. Die Installation erfolgt dann mit dem Aufruf als root:

```
rpm -Uhv htdig.rpm
```

Auf die Kompilierung der Originalquellen des Programmes und die dort zu treffenden Einstellungen

wird hier nicht eingegangen. Dafür bitte den Anweisungen von [ht://Dig](http://Dig) folgen. [Hier](#) finden Sie eine Liste, wo diese Quellen zu erhalten sind.

Außer einem relativ großen Plattenplatz bestehen keine weiteren Voraussetzungen zur Installation und zum erfolgreichen Laufen der Suchmaschine. Beim Plattenplatz für die Datenbanken ist das Programmpaket allerdings recht anspruchsvoll. Grob gerechnet benötigt man pro durchsuchtem Dokument etwa 12KB. Bei 1000 HTML-Seiten wären das also 12 MB.

Installation

Das ht://Dig-Paket wird per *YaST* installiert. Es befindet sich in der Serie *n* (Netzwerk-Support). Beim Installieren wird das Verzeichnis `/opt/www` erzeugt, in das alle nötigen Dateien entpackt werden. Damit ist die Installation eigentlich schon beendet. Zum durchsuchen von Webseiten ist es sinnvoll, das Suchprogramm (*htsearch*) ins `cgi-bin`-Verzeichnis des Webservers zu kopieren (also gewöhnlich in `/usr/local/httpd/cgi-bin`). Nach der Installation steht *htsearch* im Verzeichnis `/opt/www/htdig/bin`.

Die einzelnen Programme

htdig

Der "Spider", der alle Dateien durchsucht und die Informationen darüber erfaßt und speichert. ht://Dig nennt das Programm den "Suchroboter"

htmerge

Der "Indexer", der aus den Informationen von *htdig* den Dokumentenindex und die Wortdatenbank generiert.

htfuzzy

Die Fuzzy-Suche, die u.a. aus allen gefundenen Wörtern mittels des Endungsskriptes und des Endungswörterbuchs eine Datenbank erzeugt, mit der später alle möglichen Formen eines Wortes in der jeweiligen Sprache erkannt und in die Suchergebnisse einbezogen werden. Ebenso ermöglicht *htfuzzy*, Substrings zu Wörtern zu ergänzen. *htfuzzy* muß zur Erzeugung der jeweiligen Datenbanken nur ein einziges Mal laufen, da diese Datenbanken von den Dokumenten unabhängig sind. Außer der Endungsdatenbank kann *htfuzzy* weitere Datenbanken erzeugen. Zum Generieren einer Synonym-Datenbank braucht man ein Synonym-Wörterbuch. Ein englischsprachiges wird mitinstalliert. Die Erzeugung von Soundex- und Metaphone-Datenbanken ist weniger sinnvoll. Damit werden neben dem Suchwort auch alle ähnlich klingenden Wörter gefunden. Dies geht jedoch so weit, daß die Ergebnisse meist nicht mehr viel mit dem Suchwort zu tun haben. Zum Aufruf [s.u.](#)

htnotify

Beim Durchlauf von *htnotify* werden die Dateien danach durchsucht, ob sie veraltet sind. Falls veraltete Dateien entdeckt werden, wird eine Email an den Zuständigen Betreuer gesandt. Die Email-Adresse, das Subject und das Datum, ab wann eine Datei als veraltet gelten soll, können mittels Metatags für jede Datei festgelegt werden. Die jeweiligen Metatags im Head einer HTML-Seite sehen dann etwa folgendermaßen aus:

```
<meta name="htdig-email" content="maintainer@bigpage.de">
<meta name="htdig-email-subject" content="Seite updaten!!!">
<meta name="htdig-notification-date" content="01/07/2001">
```

htsearch

Die eigentliche Suchfunktion, die durch Klicken des Submit-Knopfes in der Suchmaske aufgerufen wird. Sie kann sowohl mit der Methode "POST" als auch mit "GET" aufgerufen werden. "GET" sollte vorgezogen werden, da dann die übergebenen Variablen in der URL

auftauchen und dann z.B. die Suchworte mit dem Webalizer ausgewertet werden können. (Siehe dazu den [content/server/webalizer.html](http://www.webalizer.com/content/server/webalizer.html)>Artikel über den Webalizer). Auch *htsearch* greift auf viele Einstellungen in der Konfigurationsdatei zurück, so vor allem, wo die Datenbanken liegen, die es durchsuchen soll oder wie die Ergebnisseiten generiert werden und was für Bilder für die Ergebnisseiten benutzt werden sollen.

Konfiguration

Im folgenden wird beispielhaft die Konfiguration der Suchmaschine vorgestellt, die Einrichtung des Suchformulars auf der Webseite sowie die Bearbeitung der Ausgabemplates. Zu allen drei Themen finden sich nach der Installation von *htdig* sehr brauchbare Beispieldateien in den Verzeichnissen `/opt/www/htdig/conf` (Datei `htdig.conf`) und `/opt/www/htdig/common` (Dateien `search.html`, `header.html`, `footer.html`, `long.html`, `short.html`, `wrapper.html`, `nomatch.html` und `syntax.html`). Ein Großteil dieser Beispieldateien kann übernommen bzw. bei Bedarf einfach editiert werden.

Das Suchformular

Es existieren eine Anzahl Variablen, die mit dem Suchformular als "hidden"-Values übergeben werden können. Notwendig ist jedoch lediglich die Variable `words`, die mit dem Textfeld der Suchmaske übergeben wird und in der für *htsearch* die Suchbegriffe gespeichert werden (s.u.). Die meisten Variablen müssen hier nicht definiert werden, da eine Default-Einstellung existiert. Die wichtigste ist der Name der Konfigurationsdatei (also im Normalfall `htdig.conf` (default)). Falls man die Konfigurationsdatei umbenannt hat oder mehrere in verschiedenen Suchmasken benutzt, muß man sie im Formular angeben. Steht die Konfiguration z.B. in der Datei `myfiles.conf`, so ist im Formularquelltext folgende Zeile zu addieren:

```
<input value="myfiles" type="hidden" name="config">
```

Ein einfaches, vollständiges (und für die meisten Fälle ausreichendes) Formular braucht bei Benutzung der Default-Einstellungen lediglich folgende Zeilen:

```
<form method="get" action="/cgi-bin/htsearch">
<input value="" type="text" size="12" name="words">
<input type="submit" value="Suche starten">
</form>
```

Alle anderen im Suchformular benutzbaren Variablen stehen gut beschrieben auf den [ht://Dig-Seiten](http://www.dig-seiten.de).

Die Konfigurationsdatei

Die Konfigurationsdatei ist das wichtigste editierbare Element von `ht://Dig`. Im folgenden wird der Einfachheit halber davon ausgegangen, daß nur eine Konfigurationsdatei verwendet wird. Sie soll wie im Default `htdig.conf` genannt werden. Da die Auflistung und Erläuterung der wichtigen Optionen in `htdig.conf` den vorliegenden Text zu stark aufweiten würde, ist [hier eine Beispieldatei](#) mit ausführlichen Kommentaren zu den wichtigsten Einstellungen. Die wichtigsten Variablen sind unten noch einmal kurz erläutert.

In der Konfigurationsdatei werden `ht://Dig`-eigene Variablen benutzt, die Verzeichnisse für bestimmte Dateien kennzeichnen. Die Pfade können allerdings auch absolut angegeben werden. Hier einige der Pfadvariablen:

- `${CONFIG_DIR}` : das Verzeichnis, in dem die `.conf`-Dateien liegen, defaultmäßig `/opt/www/htdig/conf`

- `#{COMMON_DIR}` : das Verzeichnis, in dem Templates, die von *htfuzzy* erzeugten Datenbanken, die zugehörigen Wörterbücher und die "bad words" (dazu weiter unten mehr) liegen, defaultmäßig `/opt/www/htdig/common`
- `#{DATABASE_DIR}` : das Verzeichnis, in dem die generierten Wortdatenbanken liegen, defaultmäßig `/opt/www/htdig/db`

Es gibt noch weitere mögliche: `#{IMAGE_DIR}` und `#{BIN_DIR}` z.B. für das Verzeichnis mit den Bildern oder das mit den Skripten. Sie müssen nicht benutzt werden, wenn die Bilder und die Programme in den dafür vorgesehenen Verzeichnissen liegen. Dann reicht für ein Bild nur der Dateiname, der Default-Pfad davor (also `#{IMAGE_DIR}`) wird automatisch ergänzt.

Die wichtigsten Attribute in der Konfigurationsdatei

`database_dir`

Das Datenbank-Verzeichnis. Hier werden die Datenbanken, die beim durchsuchen und indizieren mit *htdig* und *htmerge* erzeugt werden, abgelegt.

`start_url`

Die URL(s), ab denen die Seiten abwärts durchsucht werden

`exclude_urls`

Seiten, die definitiv nicht durchsucht werden sollen

`search_algorithm`

Mögliche Attribute: `exact`, `endings`, `substring`, `synonyms`, `prefix`, `metaphone`, `soundex`. Diesen Attributen werden Werte zwischen 0 und 1 zugeordnet. So gewichtet *htsearch* die Suche. Sinnvoll sind hier eigentlich nur `exact`, `endings` und `substring`. Das Ganze ist ziemlich kryptisch, da auch bei der Gewichtung `endings 0` das gesuchte Wort mit allen möglichen Endungen gesucht und gefunden wird bzw. bei Gewichtung `substring 0` der gesuchte String trotzdem zu Wörtern vervollständigt wird

`keywords_meta_tag_names`

Die Metatags, die auch durchsucht werden sollen, können hier festgelegt werden.

`locale`

Ganz wichtig vor allem im nicht englischen sprachigen Raum: die Sprachlokalisierung. Will man, daß z.B. Umlaute in HTML-Dokumenten (ä, ö, ü und ß) erkannt werden, muß man `locale: de_DE` hier einstellen. Dann werden diese Buchstaben erkannt, und zwar **egal**, ob sie in der HTML-Seite ä, ö, ü und ß, ä, ö, ü und ß oder Ä, Æ, È und Ö geschrieben sind.

`excerpt_length`

Hier kann angegeben werden, wie viele Bytes (also Zeichen) das kleine Excerpt, das beim Suchergebnis ausgegeben wird, haben darf.

`match_method`

hier wird festgelegt, ob eingegebene Wörter "und", "oder" bzw. mit booleschen Operatoren verknüpft werden. "und" ist hier default. Die anderen Werte heißen "or" und "boolean".

`use_meta_description`

Wird hier `true` eingegeben, wird bei der langen Ergebnisausgabe (mit Excerpt) der Inhalt des Metatags `description` ausgegeben und nicht einfach die oberen Zeilen des Textbods.

Die folgenden vier Attribute sind für eine flexible Suche in deutschen Webseiten extrem hilfreich bzw. fast unerlässlich. In der Original-Programmversion, die nur in Englisch existiert und die auch nur

konzipiert ist, englische Seiten zu durchsuchen, nützt das englische Endungswörterbuch und das Endungsskript, das schließlich die Endungen erzeugt, wenig. Man muß sich also im Netz deutsche Wörterbücher suchen und das entsprechende Skript (das "affix-file"). Bei der Suche in englischsprachigen Seiten sind diese Attribute nicht nötig, da dort jeweils die default-Einstellungen greifen, die das vorinstallierte Wörterbuch und das Script benutzen.

`endings_affix_file`

Eingegeben wird dann für deutsche Webseiten: `endings_affix_file: german.aff`

`endings_dictionary`

hier wird das Wörterbuch eingetragen: `endings_dictionary: german.0`

Um diese Wörter zu finden, muß zusätzlich *htfuzzy* einmal gelaufen sein. Es erzeugt die großen Endungsdatenbanken. Diese werden mit den folgenden Attributen festgelegt (bzw. deren Pfade), gewöhnlich werden sie im common-Verzeichnis abgelegt.

`endings_root2word_db`

Beispiel: `endings_root2word_db: ${common_dir}/r2wgerman.db`

`endings_word2root_db`

Beispiel: `endings_word2root_db: ${common_dir}/w2rgerman.db`

Noch ein Wort zur Datei `${common_dir}/bad_words`: Hier sind standardmäßig die (englischen) Wörter aufgelistet, die nicht indiziert werden sollen, da man nach solchen Wörtern normalerweise niemals sucht. Hier stehen z.B. the, and, for, with, not, by.... Damit auch die entsprechenden deutschen Wörter nicht indiziert werden, müssen sie von Hand hier ergänzt werden, z.B. und, an, auf, bei, zu, in, an, ab, der, die, das....

Die Ausgabemplates

Prinzipiell gibt es zwei Möglichkeiten, sich die Suchergebnisse mit den Templates anzeigen zu lassen bzw. die Ergebnisseiten zu generieren. Eine Möglichkeit ist, jeweils ein Template für den Kopf, den Body (die eigentlichen Ergebnisse) und den Fuß zu benutzen. Im Normalfall stehen diese Templates dann in den Dateien `header.html`, `long.html` und `footer.html`. Für den Mittelteil, also die eigentlichen Ergebnisse gibt es als Alternative noch die Datei `short.html`, die allerdings ihrem Namen Ehre macht, weshalb doch eher die lange Ausgabe empfohlen wird.

Die zweite Möglichkeit ist die Datei `wrapper.html`, in der alles in einer Datei abgearbeitet wird und wo die Ausgabe der Links etwas anders generiert wird. Diese Datei ist etwas weniger flexibel zu handhaben als der Weg über die drei Dateien, dafür etwas einfacher. In den folgenden Beispielen wird die Möglichkeit über die drei Dateien bevorzugt.

Wird kein Treffer gefunden oder bei Boolescher Suche eine falsche Suchsyntax eingegeben, kommen die Dateien `nomatch.html` und `syntax.html` zum Zuge.

Auch hier sind die angebotenen Beispieldateien sehr brauchbar. Da sie in HTML-Format sind, sind sie auch sehr einfach zu editieren (natürlich nur, falls man HTML versteht ;-). Hier sind auch problemlos JavaScript und Cascading Style Sheets integrierbar. PHP kann leider nicht verwendet werden, weshalb man bei rein auf PHP basierten Webseiten hier bei den Suchergebnissen auf die Grenzen der Seitendynamik stößt, indem man eine statische Ausgabe anbieten muß. Etwas trickreich sind die Variablen, die hier verwendet werden können. Sie sind in dem Programmcode von *htsearch* definiert und können daher natürlich nicht verändert werden (es sei denn, man kann in C++ programmieren und schreibt sich *htsearch* um).

Die wichtigsten Variablen für die Ausgabemplates sind hier kurz aufgeführt. Eine Liste aller Templatevariablen finden Sie [hier](#).

- `$(URL)` : hier wird der Link gespeichert.
- `$(EXCERPT)` : der kurze Text, der zu jedem Treffer angezeigt wird. Inhalt und Länge wird im Konfigurationsfile festgelegt.
- `$(WORDS)` : darin werden die eingegebenen Suchworte gespeichert, auf der Ergebnisseite in der URL als `words=...` erkennbar. Wird in `header.html` verwendet, also dem Kopf der Ergebnisseite (und auch in `nomatch.html`).
- `$(LOGICAL_WORDS)` : das sind alle Wörter, nach denen die Maschine in den fuzzy-Datenbanken sucht (also ergänzte Substrings und die Suchwörter mit allen möglichen Endungen. Will man diese alle auf der Ergebnisseite anzeigen, muß im header-Template (`header.html`) `$(LOGICAL_WORDS)` anstatt `$(WORDS)` stehen. Zum anschauen kann man einen Begriff in die Quick Search-Suchmaske auf der `ht://Dig`-Homepage eingeben, dann erscheinen die "logical words" auf der Ergebnisseite.
- `$(MATCHES)` : die Trefferzahl

So sieht die Datei `long.html`, mit der die Suchergebnisse generiert werden (ohne Kopf und Fuß), aus. Es handelt sich schlicht um eine Definitions-Liste, die mit den verschiedenen Variablen gefüllt wird.

```
<dl><dt><strong><a href="$(URL)">$(TITLE)</a></strong>$(STARSLEFT)
</dt><dd>$(EXCERPT) <br>
<i><a href="$(URL)">$(URL)</a></i>
<font size="-1">$(MODIFIED), $(SIZE) Bytes</font>
<br>
</dd></dl>
```

Start und Automatismus

Nun muß nur noch dafür gesorgt werden, daß der Spider und der Indexer regelmäßig laufen, damit die Datenbanken immer auf dem neuesten Stand sind. Beim ersten Mal sollte man wie erwähnt *htfuzzy* ebenfalls einmal laufen lassen, um die Endungsdatenbanken zu erzeugen. Ansonsten sieht der Start folgendermaßen aus: Wenn die Default-Konfigurationsdatei, also `/opt/www/htdig/conf/htdig.conf` benutzt wird, muß man lediglich als `root` in das Verzeichnis `/opt/www/htdig/bin` wechseln und die beiden Programme starten. Will man dabei noch die alten Datenbanken jeweils überschreiben anstatt neue Ergebnisse einfach anzuhängen, muß man *htdig* zusätzlich mit der Option `-i` benutzen. Der Start sieht dann folgendermaßen aus:

```
./htdig -i
./htmerge
```

Wird als Konfigurationsdatei z.B. `/opt/www/htdig/conf/myfiles.conf` verwendet, wechselt man ebenfalls nach `/opt/www/htdig/bin` und startet mit:

```
./htdig -c ../conf/myfiles.conf -i
./htmerge -c ../conf/myfiles.conf
```

Zur Erstellung der Datenbanken durch *htfuzzy* gilt folgender Aufruf:

```
./htfuzzy endings -c ../conf/myfiles.conf #generiert die Endungsdatenbanken
./htfuzzy synonyms -c ../conf/myfiles.conf #generiert die Synonym-Datenbanken
```

Weitere mögliche Optionen für *htfuzzy* sind `metaphone` und `soundex`, wodurch jeweils weitere Datenbanken generiert werden. Mehr dazu [s.o.](#)

Zum Schluß sollte man der Einfachheit halber ein Shellscript schreiben, daß den Start übernimmt und dieses dann sooft es erforderlich wird per Cronjob starten. Heißt das Script `digger.sh`, steht in `/usr/local/httpd/cgi-bin` und soll einmal nachts um 4.30 Uhr laufen, starte man den Crontabeditor als `root` mit `crontab -e`, tippe `i` für den Insert-Modus des `vi`, der gestartet ist, und

gebe folgende Zeile ein:

```
30 4 * * * /usr/local/httpd/cgi-bin/digger.sh
```

Dann Escape und ZZ und der Cronjob ist gespeichert. Ein (zu kompliziertes) Beispielstartscript befindet sich nach der Installation ebenfalls in /opt/www/htid/bin und heißt `rundig.sh`

Trickreiches zum Abschluß

Dokumente in anderen Formaten

Ht://Dig bietet die Möglichkeit, außer HTML- und TXT-Dateien auch andere Dokumente zu untersuchen, z.B. Word- oder .pdf-Dateien. Man benötigt dazu nur die jeweils geeigneten Parserprogramme. Mit dem Acrobat Reader ist es z.B. direkt möglich, .pdf-Dateien zu durchsuchen. Dazu muß man lediglich den Parser mit folgender Zeile im Konfigurationsfile aktivieren:

```
pdf_parser: PfadzumParser/acroread -toPostScript
```

In PHP eingebettete Dokumente

Ist der gesamte Webauftritt in PHP realisiert, werden die Dokumente möglicherweise nicht mittels ihres harten Links auf der Seite gezeigt, sondern über ein PHP-Script aufgerufen, um z.B. bestimmte Benutzereinstellungen mitzugeben oder um die richtigen Templates um das Dokument zu bauen. Da die Ausgaben der Suchmaschine kein PHP zulassen, und die Suchergebnisse stets harte Links liefern, gibt es hierfür einen kleinen Trick in Form des Attributes `url_part_aliases`. Damit kann erreicht werden, daß der Ausgabelink anders lautet als der eigentlich gefundene. Zur Realisierung sind hier zwei Konfigurationsdateien nötig, eine "FROM"-Datei mit einem "FROM"-String und eine "TO"-Datei mit einem "TO"-String. Als Beispiel kann man die Suchfunktion des Portals anführen. Hier wird jeder gefundene Artikel durch das Script `content.php` aufgerufen, mit dem dann die Userkonfiguration übergeben wird. *Htsearch* ersetzt nun Teile des harten Links auf den jeweiligen Artikel, den es gefunden hat, durch einen String, der das Script `content.php` enthält und zusätzlich eine Standard-Userkonfiguration. Die genaue Syntax, die dabei zu verwenden ist, finden Sie [hier](#).

Links

Leider ist die Homepage von ht://Dig mit Frameset realisiert, so daß die meisten der folgenden Links nur zu den jeweiligen Frames führen ohne die nötigen Navigationsleisten links. Ansonsten wählt man eben die Hauptseite und hat sich sehr schnell einen Überblick über die gute Navigation verschafft. Die wichtige Seite mit den Attributen für die Konfigurationsdatei ist über die Navigation so zu erreichen: Hauptseite aufrufen (<http://htdig.org>) --> Configuration file --> Alphabetical oder By Program

Homepage: <http://htdig.org>

Mirror: <http://htdig.sourceforge.net>

Attribute: <http://htdig.org/attrs.html>

Features und Voraussetzungen: <http://htdig.org/require.html>

FAQ: <http://htdig.org/FAQ.html>

Beispieldateien von ht://Dig: <http://htdig.org/config.html>

Programm *htdig*: <http://htdig.org/htdig.html>

Programm *htmerge*: <http://htdig.org/htmerge.html>

Programm *htfuzzy*: <http://htdig.org/htfuzzy.html>

Programm *htnotify*: <http://htdig.org/htnotify.html>

Programm *htsearch*: <http://htdig.org/htsearch.html>

Metatags: <http://htdig.org/meta.html>

Templates und Variablen: http://htdig.org/hts_templates.html

PHP und ht://Dig: http://www.devshed.com/Server_Side/PHP/search/ - Sehr interessante (und komplexe) Anleitung, wie man ht://dig mit dynamischen PHP-Seiten kombinieren kann

Zum downloaden: das deutsche [Endungswörterbuch](#) und das dazugehörige [Skript](#). 

Linux auf dem Server 18.05.2001