

[Four Calorie DNS Server Tour Page](#)

Welcome to the Four Calorie DNS Server for FreeBSD 6.0 (i386), (fourcdns - beta test version)

This program was developed as a companion to the "Four Calorie Web" server application, another of our programs that is available on the Internet.

Older versions may be referred to as "Zero Calorie DNS" You may substitute the term "four" for "zero" (and vice versa) in the documentation and license.txt files.

Please read the license file (License.txt)

This is the maiden voyage for this beta test version, so please be informed that the user accepts all risks involved with the usage of the software, or the inability to use the software.

This version is very preliminary, and should not be used on the public Internet (technically, it is capable of DNS service on an Internet or an intranet).

Basic configuration

The server may be bound to any legal address and any legal port (valid ports are 1 thru 65535). The default port is 53, and the default address is 127.0.0.1, as specified in the configuration file

`/usr/local/zerocaloriedns/zerocaloriedns.set`

which can be found in the `/usr/local/zerocaloriedns` directory. The configuration file is named "zerocaloriedns.set" in both the "zero" calorie and "four" calorie distributions. The application directory is named "`/usr/local/zerocaloriedns`" in both distributions.

Note that the configuration file is a simple text file. (You may edit it with vi or gedit if you wish.)

The configuration file may contain four lines, as follows:

```
ServerAddress
Port
DomainName
DomainAddress
```

Four Calorie DNS Server Tour Page

Only the information is supplied, without tags, and it must be in the correct order. The following example displays the contents of the default configuration file that is supplied with the installation media:

```
127.0.0.1
53
zerocaloriweb.com
127.0.0.1
```

Note that the domain name is the primary domain name, and does not include the www "server" segment of the domain name. The third segment (immediately to the left of the primary domain name) is often a mnemonic for the server type. The DNS server will be able to resolve the "www" segment automatically, but it should not be added to the configuration file.

If the domain address is too short, it will be padded with zeroes. (An entry of "2" would be translated to "2.0.0.0") The moral is that you should make sure that your configuration file is accurate. Use complete four segment Internet addresses.

The first entry (the address of the machine that the server is running on) will be used as a binding hint, but the server may bind to other addresses that it is able to find on the machine. Errors in the configuration file may cause the server to immediately exit its process when it is executed.

Changes made to the configuration file(s) are not effective until the server is stopped and restarted. Re-executing the fourcdns file will not stop the server. Therefore, the user should execute a kill command first. Assuming that there are no other unrelated fourcdns processes running, the user could restart the server by executing:

```
killall fourcdns
./fourcdns
```

The Windows version sports a nice utility for DNS Server management/monitoring. It is ZeroCDNSManager.exe, and is one of the start menu items.

The FreeBSD 6.0 X Windows GUI for DNS server management is

```
fourcdnsmgr6:
./fourcdnsmgr6
```

Four Calorie DNS Server Tour Page

A Little Background ...

This server was developed as a companion to the "Four-Calorie-Web" server program, which is an HTTP web server. A web server without a DNS server is not very useful, unless your visitors have a good memory for Internet addresses!

This is the preliminary beta version of the DNS server, and should be used only for purposes that are not important or that are tolerant of failure. (See license.txt)

There seems to be a dearth of DNS server applications for the Windows, Solaris, and FreeBSD platforms. Hopefully, this program will (when it eventually passes the hurdle of the beta-testing period) alleviate the DNS-application-scarcity problem. This particular server was designed to have only the functionality needed to power the companion "Four-Calorie-Web-Server" application.

The server resolves only the user's domain IP address, and no others. It cannot run as a general purpose public DNS server in it's current configuration. This is an artificial constraint, but it allows us to concentrate our efforts toward our specific goal (towards powering the "Four-Calorie-Web" Server).

The server currently responds only to UDP requests, although TCP backup functionality will be considered in the future. The server implements minimum functionality in other ways. For instance, the header flags recursive bit is not set (recursion is not available). Some things are "hardwired" this way to allow for maximum simplicity. A user should be able to set-up the Four Calorie DNS server in a few minutes (If there are any other DNS servers that can be set-up in two or three minutes, I am not aware of them).

The server auto-resolves all "server-type mnemonic" requests to the same one-and-only IP address that is supplied by the user in the configuration file (So long as the primary domain is correct).

In other words, `www.yourdomain.com` and `mail.yourdomain.com` and `ftp.yourdomain.com`, and `ns111.yourdomain.com` all resolve to the domain IP address. This auto-resolution would not be desirable in a large server farm. However, for an individual's toy web-site, it probably is a positive attribute. The server does not resolve lower sub-domains.

Four Calorie DNS Server Tour Page

Some resolvers do not handle CNAME entries well, so after some experimentation, it was decided to disallow CNAME functionality, and just return the primary domain IP address for the next-level domain entries. A maximum limit of 50 characters has been implemented for the entire domain.

The server currently handles standard forward (domain name) queries and reverse IP address queries, MX, NS, and SOA queries, and little else. However, for a web server to function properly, the DNS server need not service much more than standard forward and reverse queries.

The authoritative server flag is always set. The server does not currently support a zone transfer, but version 1.0.5 now responds to SOA (Start of authority), NS (Nameserver), A (Address record), MX and PTR (Reverse) requests.

The serial number is auto-incremented, and changes only once per day, even when the server is restarted. Since zone transfers are not allowed, this may or may not be an issue. The upshot of this limitation is that a change made to a domain name or IP address may not be "seen" by other servers that are caching the output of this server for one day. However; the NS, ARECORD and PTR requests served by **this** server will be accurate "up to the minute", so long as the server was restarted after the configuration file change.

If one day serial updates is an issue, then don't use this server for your application. The expiration is fixed (hard coded) to two weeks. For the serial updates to work correctly, your machine must have a correct CMOS date (Your battery should be working).

The domain administrator's email is hard coded to de@somedomain.com, where somedomain.com is the content of the third line of the configuration file. The MX record is hard coded to mail.somedomain.com, where somedomain.com is defined in the third line of the configuration file.

Installation and De-installation Notes

Installation

The installation has been changed to use the FreeBSD ports system. Instead of extracting the tarball, simply copy the fourcdns-1.0.8.tar.gz

file to the /usr/ports/distfiles directory. Then, while the current directory is still the directory that the outside tarball was extracted into (fourcdnsTempInstall) :

```
type:
make install
```

Reboot. Upon restart, you should be able to type the name of the GUI manager in an X windows terminal (fourcdnsmgr6) to start the GUI manager. You should be able to start the DNS server manually by typing:

```
killall fourcdns
fourcdns
```

You should be able to see the manual pages by typing man fourcdns. Note that the configuration file should still reside in the: /usr/ local/ zerocaloriedns directory.

The server will display a banner unless the -silent switch is used. The banner will not indicate whether or not the server was able to bind to the configured address. You must check the server status with sockstat, top, and nslookup or dig:

i.e. dig @127.0.0.1 MX zerocaloriweb.com

In the future, a management application will make this easier.

DNS Server Monitor/Manager

The user should be able to use a DNS tool to test the server. Type "zerocaloriweb.com" into the appropriate text box in your chosen DNS tool (or use whatever domain you have included in the ZeroCalorieDNS.SET configuration file).

DNS Tools

Your DNS tool should allow you to obtain a response appropriate for the configuration file that you have supplied. NSLookup is a good tool for DNS testing, and is supplied with the wonderful MSys package. (See the shameless plug in the next section) Alternatively, if you have a Unix machine on your network, you can use the DIG utility to get a nice summary output from the DNS server.

Ethereal utility

Ethereal is a utility that is available for Windows and for various unix variants. The use of this utility is a great way to view the more arcane detail of the output of the "Four Calorie DNS Server". Note that on windows, the Ethereal utility requires the "PCAP for Windows" software to be pre-installed. The PCAP packet capture software is available as a freeware (beta version) download.

Single IP Setups

Note that many Internet web servers run in concert with a DNS server that handles only one domain. The Internet registry record points to a domain IP address for the owner's nameserver, and that nameserver may in many cases be capable of returning only the IP-address for a single domain. (The owner's domain)

This DNS server is designed to do just that -- return the domain information for a single domain. This model was chosen to create maximum simplicity, and a minimum of configuration headaches. The name server has a fixed built-in name (ns111.yourdomain.com), where yourdomain.com is the domain that the user entered into the third line of the configuration file. The last line of the configuration file is the address of the domain and nameserver. (This is in contrast to the first line of the configuration file, which is the address of the machine that the DNS server is running on). The addresses do not need to reference the same machine.

If there are conflicts with the ns111 nameserver name, then you will have to adjust the other conflicting name server to resolve the issue. If you have a big server farm, you probably are not using this DNS server anyway. (And shouldn't be, as this version is the number one cut of the beta test distribution!)

Easy DNS - HTTP setup ... all on one machine!

We stipulate that this beta version cannot be used on the public Internet, however; the following list describes what would be the relationship between the HTTP server and the DNS server in any future public deployment of the HTTP and DNS servers:

The domain name registry record at Register.com (or wherever it is maintained) contains the address of the name server (Four Calorie DNS server, in our case) for the somedomain.com domain

1. The DNS configuration file of our little DNS server contains the address of the domain (which in a one server setup, will also be the address of the HTTP server. (Four Calorie Web Server, in our case)
2. Our little web server is bound to port 80 on a machine that has been assigned the aforementioned address. (Everything is on one machine, in our case, and all three configuration file addresses are the same domain address)
3. A random user browses the Internet, and finds a link to our www.somedomain.com site. Random user clicks on the browser link. The browser uses what is known as a "resolver" to query the client's assigned DNS address, which at some point will query the root servers (or some other server) for our nameserver address as it is recorded by the registrar record.
4. The browser then knows where our nameserver is, and queries our nameserver for the resolution of www.somedomain.com. Our nameserver (Four Calorie DNS Server) returns a result to the browser (which is the same domain IP address as the web http server in our one-machine configuration.)
5. The browser, with our web server's address in hand, transfers our web pages from the web server to the browser, and displays same.

Please read the license.txt file included with the installation media. This version of the DNS server runs on FreeBSD 6.0 (i386)

No particular level of security is indicated or warranted for any portion of this software.

Monitoring the server operation:

The MSys netstat utility can detail the connection information for the `Four Calorie DNS Server`. (Here is a plug for the wonderful MSys package. It provides a unix-like terminal and environment on Windows, and it contains utilities that allow more convenient management of network related issues than Windows)

The server is in the development and testing phase, and so perfect function is not assured.

Let us know what you think!

Note that this program is still in the development and testing process
- don't use it for anything important!

Email suggestions to: rleesBSD@gmail.com, with subject line /Attn: RScheckelhoff /REF: Datazygte DNS