

ddclient

24 x 7 Availability

A home user's Linux computer is typically crammed full of data such as holiday photos, configuration files that solve a specific problem or word processor templates. Users also typically want to access that data wherever they are. Dynamic DNS services and the ddclient allow users without a static IP address to expose their machines to the perils of the Internet. **BY FREDERIK BIJLSMA**

Users wanting to access their SOHO PCs remotely are typically frustrated by the fact that most providers ditch the IP lease after about 24 hours. Traditional tools will allow you to provide remote access facilities for this period only. Fortunately, help is at hand in the form of a dynamic DNS service that allows you to update the mapping between **DNS hostnames** and numerical IP addresses when required.

However, to do so the service provider must be informed of any IP address changes. It would be difficult or impossible to modify the mapping manually, each time the address changes – particularly if you cannot access your computer's console at that time. ddclient by Paul Burry automatically supplies the major dynamic DNS services (that is <http://www.dyndns.org/>, <http://www.hn.org/>, <http://www.zoneedit.com/> and <http://www.easydns.com/>) with appropriate information.

Listing 1: Installing ddclient

```
tar xzf ddclient-3.6.2.tar.gz
cd ddclient-3.6.2
su (type your root password here)
cp ddclient /usr/sbin
cp sample-etc_ddclient.conf /etc/ddclient.conf
cp sample-etc_rc.d_init.d_ddclient /etc/rc.d/init.d/ddclient
exit
```

As ddclient is a Perl script, you do not even need to compile it, and this makes the installation a piece of cake; of course, this assumes that you have already installed Perl. Fortunately, most major distributions tend to install Perl by default.

Easy Installation

ddclient itself and a few sample files are available on the subscription CD or from the following website: <http://burry.ca:4141/ddclient/> – although we did have some trouble reaching the website. Users of the current Debian distribution or Red Hat Linux 8/9 can use the pre-compiled packages on the subscription CD.

```
dpkg -i ddclient_3.6.2-2_all.deb
```

will install the package on Debian, and:

```
rpm -ivh ddclient-3.6.2-4.
noarch.rh8.rpm
rpm -ivh ddclient-3.6.2-5.
noarch.rh9.rpm
```

will install it on Red Hat 8.x (first command) or 9.x (second command).

Debian even offers to configure ddclient directly after installing, using the character based interface shown in Figure 1. If you need to change the settings at a later stage, you can launch:

```
dpkg-reconfigure ddclient
```

to do so. We will be looking at the manual configuration in this article.

Red Hat will need to add the program to the list of services launched on booting the system, by typing:

```
chkconfig --add ddclient
```

Users of other distributions can refer to Listing 1 for an installation guide.

Dynamics

To put ddclient to work, you will need an account with one of the above mentioned providers of dynamic DNS services. This account is completely independent from your Internet account!

The most popular provider, DynDNS, provides access via <http://www.dyndns.org/account/create.html>, allowing you to configure up to five hostnames free of charge. After logging on to <http://www.dyndns.org/>

OUT OF THE BOX

There are thousands of tools and utilities for Linux. "Out of the box" takes a pick of the bunch and each month suggests a little program, which we feel is either absolutely indispensable or unduly ignored.

GLOSSARY

DNS hostname: The unique name of a computer in the "Domain Name Service" (DNS), for example, www.linux-magazine.com. Dynamic DNS services map the name to the numerical IP address assigned to the computer when it connects to the Internet and passes this information to the ddclient. Of course, this only applies to domains where you are responsible for the DNS entries.

Perl: The "Practical extraction and report language" is a fully-featured and popular script language. Code written in Perl does not need to be compiled, but can be executed by the Perl interpreters.

Routing table: This is a list, managed by the operating system, that provides a machine with information on other routers (computers) that provide the best paths to a target system (the Internet in this case).

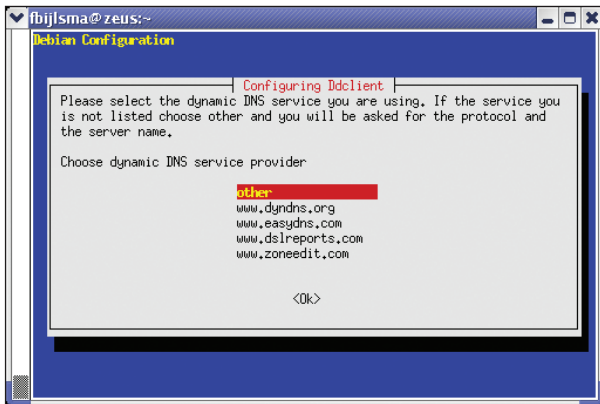


Figure 1: Debian provides a configuration tool for ddclient

`dyndns.org/account/`, you can select *Dynamic DNS / Add Host* and add your hostname in the *Hostname* line. Then select one of the domains provided by DynDNS on the right. You do not need to change any other fields. Your computer will later be accessible as *Hostname.selectedDomain*, this corresponds to `linuxusertest.dyndns.org`.

As DynDNS reserves the right to delete unused hostnames that have not updated their IP address for 35 days from the database, it is not worth securing a supply of hostnames. If you intend to map your computers to domains of your own, you can ask the dynamic DNS provider to maintain the DNS information for your domain – of course, this service is not free of charge.

Tell me who I am!

The next step is to configure *ddclient* to inform the dynamic DNS service of your current Internet IP address; the `/etc/ddclient.conf` is used for this purpose. You can transmit a static IP address, as follows:

```
use=ip, ip=127.0.0.1
```

However, this would mean reconfiguring *ddclient* each time you change your IP address, and this would be self-defeating as the address is no longer dynamic. A better approach would be to enter

```
use=if, if=ppp0
```

thus telling the tool to read the current IP address of the interface `ppp0`. Of course you could use other network interfaces instead, assuming you enter the IP address that is available on the Internet.

Another alternative is to let *ddclient* send a Web request to the dynamic DNS server, allowing the server to read the IP address. This method can be enabled by specifying `use=web`, and reflects the machine's current **routing table**, which in turn means that it makes no difference what medium you use to connect to the Internet. However, this

method will not work if you stipulate a proxy, as the proxy address would be entered into the database.

ddclient additionally reads the dial-up information for some dial-up routers (type `ddclient --help` for more information), but the Web method will typically work just as well, if not better.

Your choice of *use* method will not affect the remaining configuration steps performed in `/etc/ddclient.conf`. However, you will need to modify the values for *login* and *password* in Listing 2 to reflect your own DynDNS login and password, and enter your registered hostname in the *server* line at the end of the file. When you type `/etc/init.d/ddclient start` to launch *ddclient* in dae-

mon mode, the tool checks for IP address changes at definable intervals, ensuring that DynDNS still has your current IP. As soon as your IP address changes, *ddclient* sends the new IP to the dynamic DNS provider.

If you do not have a flat rate Internet account, the cost of continually establishing connections to the Internet may be non-trivial, and it does cause additional traffic. In this case, you might prefer to launch *ddclient* without any additional arguments each time you connect to the Internet.

Troubleshooting

A quick look at `/var/log/messages` indicates that *ddclient* has successfully transmitted our current IP address to the dynamic server specified in the *server* line:

```
May 13 14:54:58 freedom ddclient[4196]: SUCCESS:
updating myhostname.dyndns.org
: good: IP address set
to 62.245.157.217
```

If the IP address in the log entry does not correspond to the IP on your machine, you may need to experiment with the `use=` parameter. This issue is normally quite easy to resolve. ■

Listing 2: /etc/ddclient.conf

```
daemon=300 # The daemon checks for an IP change
# every 300 seconds...
syslog=yes # ... and writes a syslog entry
# (typically /var/log/messages).
# mail=root # Send email if IP address changes
# (disabled)
pid=/var/run/ddclient.pid # Store daemon process ID in
# /var/run/ddclient.pid
# use=ip, ip=127.0.0.1 # Send static IP (disabled)
# use=if, if=ppp0 # Read IP for interface (disabled)
use=web # Get IP via Web access
protocol=dyndns2 # Default protocol
server=members.dyndns.org # Default server
login=linuxusertest # Default login for DynDNS account
password=blablabl # Default password
server=members.dyndns.org, protocol=dyndns2, myhostname.dyndns.org
# Additional zoneedit entry. As this service uses different credentials
# than DynDNS, these lines contain different protocol, login and
# password values.
server=www.zoneedit.com, protocol=zoneedit1, login=zoneeditlogin,
password=zoneeditpassword, myhostname.my.domain
```