

## Multiple\_DDNS\_Accounts

DD-WRT v23 SP2 and later use [inadyn](#) as its embedded dyndns client solution.

By default, the DD-WRT GUI allows the specification of one DDNS provider e.g. dyndns. At router startup, DD-WRT launches inadyn in daemon mode, and from this point onwards, inadyn manages updates to the DDNS provider. inadyn retrieves the DDNS provider parameters (username, password, provider name etc.) from DD-WRT via a configuration file (/tmp/ddns/inadyn.conf), that DD-WRT creates from information entered by the user on the DDNS UI screen.

However, the base DDNS implementation in DD-WRT updates one DDNS provider only. Some users may have multiple DDNS providers, and associate multiple URLs with their IP address. This wiki shows how to maintain these multiple DDNS accounts by using the embedded inadyn in DD-WRT.

Note that you may choose to create an account on [DNS-O-Matic](#) which will perform multiple Dynamic DNS updates for you automatically.

## Contents

- [1 DD-WRT Integration of Inadyn](#)
- [2 Requirements](#)
- [3 Configuration](#)
- [4 Configuration \(the old way\)](#)
  - ◆ [4.1 Create a directory on the jffs partition](#)
  - ◆ [4.2 Create a link to inadyn](#)
  - ◆ [4.3 Find the typical command line from your DDNS provider that works](#)
  - ◆ [4.4 Create the new configuration file for this new DDNS provider](#)
  - ◆ [4.5 Make start inadyn with the new config file at router startup](#)
  - ◆ [4.6 Tests](#)
- [5 External Links](#)
- [6 See also](#)

## DD-WRT Integration of Inadyn

inadyn is a non-persistent program i.e. it does not maintain any state from one session to another. This creates a problem for DD-WRT in that if inadyn is restarted for some reason (e.g. a router reboot), a couple of key pieces of information are lost:

- The WAN IP address the last time inadyn was run.

inadyn decides whether to do an update by comparing the last WAN IP address it saw with the current router WAN IP address. Thus when it starts for the first time, if it has no memory of the last IP address, it will always issue an update. Without some provision of supplying inadyn with this last WAN IP information, inadyn will always issue an update on startup, which has the potential to cause problems with DDNS providers like dyndns, that suspend their DDNS service to users who do frequent IP updates.

- The amount of time left before inadyn issues a compulsory update

## Multiple\_DDNS\_Accounts

inadyn can be configured to always update a DDNS provider at a fixed interval. By default, DD-WRT sets this to 28 days. If for some reason inadyn is restarted part way into this schedule, it needs to know when the next compulsory/forced update time is e.g. if it was killed on the 27th day, if it has no information on how far along it was before, it will start counting from 0, and not do an update for another 28 days, thereby incorrectly stretching the compulsory period from 28 days to 27+28=45 days.

The new inadyn integration with DD-WRT upgrades both DD-WRT and inadyn to maintain these two pieces of persistent information in nvram. On router startup, these nvram values are written into files in /tmp/ddns for inadyn to retrieve, and from then onwards are updated by inadyn as it performs its updates.

When we extend DD-WRT as outlined below, we must also provide some means by which inadyn can maintain these two pieces of information in a persistent fashion. For the purposes of this section, we will assume that we will achieve persistence by using the JFFS2 filesystem.

## Requirements

- V23SP2 or V24 build from 18 August 2006 or later
- Some JFFS2 space to hold 4 small text files (config file, log file, cached IP, cached forced-update time) per additional DDNS provider

## Configuration

If you fit the requirements you can update multiple DDNS domains by adding the -a parameter to the hosts list in the web interface.

So it would look something like this:

```
Hostname: account1.dyndns.org -a account2.game-host.org
```

[Image:Dns 207.png](#)

## Configuration (the old way)

### Create a directory on the jffs partition

This will store the details of the new DDNS provider. For the sake of this Wiki, we will assume this is /jffs/ddns2

```
mkdir /jffs/ddns2
```

## Create a link to inadynd

This will allow us to execute the inadynd.conf file by inadynd in the new created directory.

```
ln -s /usr/sbin/inadynd /jffs/ddns2/inadynd
```

(Optional) Copy the current cached IP value from /tmp/ddns/inadynd\_ip.cache to this directory.

## Find the typical command line from your DDNS provider that works

Determine the command line to be used for this DDNS service provider (to be placed in a config file specific to this provider). This section of the Wiki will not focus on the different DDNS providers, and how to setup the various inadynd command line options for each. Other sections of the wiki do this e.g. [DDNS - How to setup Custom DDNS settings using embedded inadynd - HOWTO](#). In case of difficulty, DD-WRT itself can be used as a guide e.g.:

- Set up this provider through the DD-WRT UI and get it working.
- Then ssh or telnet to the router and navigate to /tmp/ddns.

```
cd /tmp/ddns
```

- Copy the contents of the inadynd.conf file. This file will have one line of information containing all the parameters used to invoke inadynd for this service provider.

```
cp /tmp/ddns/inadynd.conf /tmp/ddns/inadynd.conf.backup  
cp /tmp/ddns/inadynd.conf /jffs/ddns2/inadynd.conf
```

- Modify the line to delete a parameter "--exec ddns\_successful". This will most likely be at the very end of the line.

```
vi /jffs/ddns2/inadynd.conf
```

Exit vi saving the changes with <esc> : wq

## Create the new configuration file for this new DDNS provider

Now we will modify the file we just closed to adjust the paths and other parameters appropriately.

```
vi /jffs/ddns2/inadynd.conf
```

- Go to the end of the line and change `--log_file /tmp/ddns/ddns.log` to `--log_file /jffs/ddns2/ddns.log`

E.g. the command line I use for no\_ip is: `--background --dyndns_system default@no-ip.com -u my_noip_username -p my_noip_password -a my_noip_url --update_period_sec 360 --forced_update_period 2419200 --log_file /jffs/ddns2/ddns.log`

## Multiple\_DDNS\_Accounts

- As an added bonus (and as different from the built in DD-WRT DDNS) you can tailor the parameter values to your satisfaction e.g. if you want to specify a faster check interval of 1 minute, change the 360 seconds to your preferred value of 60 etc.

## Make start inadyn with the new config file at router startup

Now we need to start inadyn with this config file at router startup. Here is an example script to do so. Note the use of the cache\_dir parameter to tell inadyn to use our newly created directory to store its persistent cache values. In the DD-WRT UI, this script can be placed in (or appended to) Administration->Commands->Command Shell. (Press "Save Startup" when done).

```
#Delay to ensure WAN is up and running and router has correct time
sleep 5
#Now startup the new inadyn ...
/jffs/ddns2/inadyn --input_file /jffs/ddns2/inadyn.conf --cache_dir /jffs/ddns2
```

## Tests

- Test your added domain from Windows or Linux with *nslookup*. At the prompt, enter the domain name and press enter. You should see your WAN IP address there if all went OK.

```
> user@myhost:~$ nslookup
> myserver.com
Server:          192.168.1.1
Address:         192.168.1.1#53

Non-authoritative answer:
Name:   myserver.com
Address: 7x.x40.249.x2
>
```

Exit nslookup with ctrl-c.

- In your browser, add the subdomain and verify it connects and opens the default web page of your host.

*Note: to add more than one account, simply repeat the above step by step directions, but using a new/different directory for each additional DDNS provider e.g. /jffs/ddns3, /jffs/ddns4*

## External Links

[INADYN scripts for multiple DDNS](#)

## See also

- [Dynamic DNS](#)

See also