

UpdateDD Version 2.4
Dokumentation

Copyright (c) 2004 Philipp Benner.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

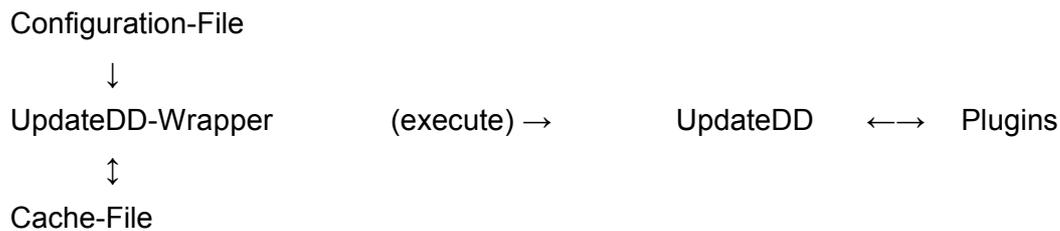
1 Inhaltsverzeichnis

| | |
|---------------------------------|---|
| 1 Beschreibung..... | 1 |
| 2 Verwendung..... | 1 |
| 3 UpdateDD-Wrapper..... | 2 |
| 3.1 Configuration File..... | 3 |
| 4 Scripting..... | 4 |
| 5 Einbindung in das System..... | 4 |
| 6 Bugreports..... | 5 |
| 7 Bezugsquellen..... | 6 |

1 Beschreibung

UpdateDD ist ein Programm für Unix ähnliche Betriebssysteme um Dynamic DNS Eintrag zu aktualisieren. Das Programm unterstützt mehrere Dienste, welche mit dem Befehl ``updatedd -L`` aufgelistet werden können. Dementsprechend ist das Programm so gegliedert, dass Unterstützung für neue Dienste leicht hinzugefügt werden kann. Für jeden Dienst gibt es ein Plugin, welches von UpdateDD aufgerufen wird. Diese Plugins führen die Aktualisierung durch und geben den Status an UpdateDD zurück.

Das Paket enthält außerdem ein Perl-Programm „UpdateDD-Wrapper“, welches Unterstützung für IP-Address-Caching und Konfigurationsdateien bereit stellt.



2 Verwendung

Bei UpdateDD gibt es zwei Arten von Argumenten. Großgeschriebene, welche für UpdateDD selbst sind und kleingeschriebene, welche den Plugins übergeben werden. Eine Liste mit Verfügbaren Argumenten und Optionen bekommt man mit dem Befehl ``updatedd --help`` bzw. ``updatedd <service> -- --help``.

```
cd81:~$ updatedd --help
```

```
Usage: updatedd [OPTION]... SERVICE -- ...
```

Options:

```
-L      list installed plugins (services) and exit
-Y      use syslog
--help  print help and exit
--version print version information and exit
```

```
cd81:~$ updatedd dyndns -- --help
```

```
Usage: updatedd [...] dyndns -- [OPTION]... [USERNAME:PASSWORD] HOSTNAME
```

For security reasons use the environment variable LOGIN instead of passing the login information directly.

Options:

```
-4      --ipv4 <address>    ip address version 4
-b      --with-backmx       enable backmx
-m      --mx <host>        hostname of your mail exchange
-o      --offline           set host to offline mode
-s      --system <system>  specify the system
                          (dyndns [default], statdns, custom)
-w      --wildcard <on|off> switch wildcard on or off
--help  print help and exit
--version display version information and exit
```

Jedes Plugin kann auf dieselbe Weise aufgerufen werden. Außerdem gibt es bei jedem Plugin die Option '-4' bzw. '--ipv4' mit der eine IPv4 Adresse übergeben werden kann, mit welcher der DynDNS Account aktualisiert wird. Wenn der Dienst IPv6 Adressen unterstützt, dann kann diese mit der Option '-6' bzw. '--ipv6' übergeben werden. Die restlichen Optionen sind vom jeweiligen Dienst abhängig und können variieren.

Folgende Beispiele funktionieren demnach immer:

```
updatedd <service> USER:PASS HOSTNAME
updatedd <service> -- -4 IPADDR USER:PASS HOSTNAME
```

```
updatedd -Y <service> USER:PASS HOSTNAME
updatedd -Y <service> -- -4 IPADDR USER:PASS HOSTNAME
```

Wenn die IP Adresse nicht übergeben wird, dann versucht der jeweilige Dienst die IP Adresse automatisch herauszufinden.

Die Option '-Y' veranlasst UpdateDD sämtliche Status Informationen in die Syslog zu schreiben, welche sich normalerweise im Verzeichnis '/var/log' befindet.

Aus Sicherheitsgründen sollte vermieden werden, Passwörter als Argumente an ein Programm zu übergeben. Da ein Aktualisierungsversuch aufgrund einer langsamen Verbindung zum Internet sehr lange dauern kann, sind andere Benutzer in der Lage die Login Informationen mit 'ps' zu sehen. Deshalb können die Login Informationen auch in der Environment Variable \$LOGIN gespeichert werden. Die Plugins lesen diese anschließend aus der Variablen aus.

```
export LOGIN=user:pass
updatedd <service> HOSTNAME
updatedd <service> -- -4 IPADDR HOSTNAME
```

3 UpdateDD-Wrapper

Das Programm UpdateDD-Wrapper wird dazu verwendet UpdateDD auszuführen. Es liest die Konfiguration aus einer Datei aus und führt anschließend UpdateDD aus. Wenn in der Konfigurationsdatei die Option 'IP-ADDR' verwendet wird, dann speichert UpdateDD-Wrapper diese in einem Cache. Auf diese Weise kann verhindert werden, dass eine Aktualisieren mit derselben IP-Adresse mehrmals durchgeführt wird, was oft zu einer Sperrung des Accounts führt.

UpdateDD-Wrapper können folgende Optionen übergeben werden:

- c <path> Eine alternative Konfigurationsdatei, normalerweise wird */etc/updatedd-wrapper.conf* bzw. *./usr/local/etc/updatedd-wrapper.conf* verwendet.
- d Debug Informationen werden angezeigt.
- f Veranlasst UpdateDD-Wrapper die gespeicherten IP-Adressen im Cache nicht zu beachten.
- i <sec> Bei fehlgeschlagenen Aktualisierungsversuchen kann mit dieser Option die Zeit geändert werden, die das Programm wartet bis ein erneuter Versuch durchgeführt wird.
- r <num> Mit dieser Option kann festgelegt werden, wie oft eine Aktualisierung versucht werden darf.
- s Bei dieser Option wird UpdateDD nicht ausgeführt. Es werden lediglich die Befehle ausgegeben, mit denen UpdateDD ausgeführt werden kann. Diese Option ist hilfreich um zu überprüfen ob die Konfigurationsdatei korrekt ausgeführt wird.
- t Hiermit kann eine andere Cache Datei angegeben werden.

- y Die Ausgabe von UpdateDD wird in die syslog geschrieben. Einstellungen in der Konfigurationsdatei werden übergangen.

3.1 Configuration File

Die Konfigurationsdatei von UpdateDD-Wrapper befindet sich normalerweise im Verzeichnis */etc* bzw. in */usr/local/etc*.

Ihr Syntax ist ähnlich der von der Programmiersprache C, mit einer Ausnahme: Kommentare beginnen mit einer Raute.

Die Datei beinhaltet zwei Arten von Blöcken. Die erste wird verwendet um dem Programm Login-Informationen zu übergeben und diese mit Alias Namen zu bezeichnen.

Beispiel:

```
login {
    dyndns_1    = "user:pass";
    regfish_1   = "TOKEN=token";
    regfish_2   = "STD=user:pass";
    tzo_1       = "user:pass";
};
```

Dem Alias 'dyndns_1' wird bei diesem Beispiel der Login 'user:pass' zugewiesen. Hier muss beachtet werden, dass für die Login Informationen Anführungszeichen verwendet werden sollten und dass jede Zeile mit einem Semikolon abgeschlossen werden muss. Die Syntax der Login Informationen hängt vom jeweiligen Plugin ab. Mit dem Befehl `updatedd <service> -- --help` können Sie diese nachschlagen.

Die zweite Art von Blöcken gibt an, um welchen Dienst es sich handelt, welche Hostnamen aktualisiert werden sollen und mit welchen Optionen die Aktualisierung durchgeführt werden soll.

Beispiel:

```
regfish(active) {
    login        = regfish_1;
    hostnames    = "host1.org host2.org";
    ip-addr      = ipv4:`/usr/local/share/updatedd/ipserv.pl regfish`;
    use-syslog   = yes;
    options      = "--ismx";
#    force;
}
```

Bei diesem Beispiel wird das Plugin mit dem Namen 'regfish' ausgewählt ist, welches für den Dienst „Regfish.com“ zuständig ist. Nach dem Namen des Plugins steht in Klammern entweder 'active' oder 'disabled'. Dies gibt an, ob die folgende Block von UpdateDD-Wrapper abgearbeitet oder ignoriert werden soll.

Jeder dieser Blöcke muss die folgenden Argumente beinhalten:

- login Übergibt einen Alias Namen für einen Login.
- hostnames Hier werden die Hostnamen angegeben (jeweils mit space separiert).

Die folgenden Optionen können zusätzlich angegeben werden:

- ip-addr Hier wird ein Befehl angegeben, welcher die IP Adresse in den Standard

Output schreibt. Normalerweise handelt es sich bei dem Befehl um ein Script, welches die Adresse ermittelt. Von UpdateDD werden einige Scripte mitgeliefert, welche sich in `/usr/[local/]share/updatedd` befinden. Vor dem Befehl wird angegeben, um welche Art von IP Adresse es sich handelt. Diese Option sollte immer verwendet werden, da sonst die IP Adresse nicht im Cache gespeichert werden kann und unnötige Aktualisierungen durchgeführt werden.

| | |
|------------|--|
| use-syslog | Die Ausgabe von UpdateDD wird in die syslog geschrieben. Per default ist diese Option auf 'yes'. |
| options | Hier können extra Optionen an das Plugin übergeben werden. |
| force | IP-Adressen im Cache werden ignoriert. |

4 Scripting

Mit den folgenden Informationen können sie leicht effektive Scripte selbst schreiben:

UpdateDD gibt verschiedene Werte zurück, damit es möglich ist Schleifen in Scripte einzubauen.

Status Codes:

- 0 => Die Aktualisierung war erfolgreich.
- 1 => Warnung: Es gab einen Verbindungsfehler oder es ist nicht genügend Speicher vorhanden. Ein erneuter Versuch darf durchgeführt werden.
- 2 => Fehler: Die Aktualisierung konnte nicht durchgeführt werden. Z.B. ist der Benutzer oder das Passwort falsch, oder der Hostname existiert nicht. Ein erneuter Versuch sollte nicht durchgeführt werden, weil sonst der Account gesperrt werden könnte.
- 3 => Fehlerhafter Aufruf von UpdateDD.

5 Einbindung in das System

Grundsätzlich gibt es zwei Arten wie UpdateDD in das System eingebunden werden kann, damit es die Aktualisierungen automatisch durchführt. Wenn der rechner direkt mit dem Internet verbunden ist, dann kann pppd benutzt werden. Dazu muss nur ein kleines Script mit folgendem Inhalt im Verzeichnis `'/etc/ppp/ip-up.d'` angelegt werden:

```
#!/bin/sh
sleep 30s # warte bis die Verbindung steht
updatedd-wrapper -y [weitere Optionen]
```

Wenn PPPD verwendet wird, ist das Cachen der IP-Adressen meist unnötig, da sich bei jeder Verbindung zum Internet die IP-Adresse ändert. Wenn dies der Fall ist können System Ressourcen gespart werden, indem die Option ``ip-addr'` in der Konfigurationsdatei von UpdateDD-Wrapper nicht verwendet wird, oder indem UpdateDD direkt ausgeführt wird:

```
#!/usr/bin/perl -w
#!/usr/local/bin/perl -w

use strict;
```

```
##### Configuration #####
```

```
my $service      = "dyndns";
my $login        = "test:test";
my $hostnames    = "test.dyndns.org test2.dyndns.org";
my $ip_addr      = $ARGV[0];
my $retries      = 5;
```

```
#####
```

```
$ENV{LOGIN} = $login;
my $ret;
foreach my $hostname (split /\s,/, $hostnames) {
    my $n = $retries;
    while($n != 0) {
        $ret = system("updatedd", "-Y", $service, "--", "-4", $ip_addr, $hostname);
        if($ret != 1) {
            $n = 0;
        } elsif($ret == 1) {
            $n--;
        }
    }
}
exit($ret);
```

Mit den Informationen aus Kapitel 4 Scripting können Sie sich auch eigene Scripte schreiben. Im Verzeichnis `/usr/[local/]share/updatedd` sind zusätzliche Beispiele zu finden.

Wenn auf dem System kein ppp Daemon installiert ist, dann kann cron verwendet werden um UpdateDD in einem Zeitintervall auszuführen. Dazu muss in der Datei `/etc/crontab` der folgende Eintrag hinzugefügt werden.

```
*/10 * * * * root updatedd-wrapper -y &
```

Bei diesem Beispiel wird UpdateDD-Wrapper alle 10 Minuten aufgerufen. Aus Sicherheitsgründen wäre es gut wenn Sie einen anderen Benutzer als root für diesen Aufruf verwenden würden.

6 Bugreports

Wenn Ihnen ein Fehler bei UpdateDD auffällt, dann können sie einen Bugreport an updatedd@philipp-benner.de schreiben. Allerdings reicht es nicht aus zu schreiben, dass etwas nicht oder fehlerhaft funktioniert. Es werden zusätzliche Informationen benötigt um den Fehler zu korrigieren:

- Der Befehl, mit dem UpdateDD bzw. UpdateDD-Wrapper aufgerufen wurde und bei dem der Fehler auftritt.
- UpdateDD sollte mit `./configure --enable-debug` kompiliert werden. Es wird die vollständige Ausgabe im Terminal benötigt. Außerdem sollte sicher gestellt werden, dass die Ausgabe von UpdateDD nicht in die Syslog geschrieben wird.
- Handelt es sich um einen Fehler im Programm UpdateDD-Wrapper wird die Ausgabe von `updatedd-wrapper -ds` und die Konfigurationsdatei benötigt.

7 Bezugsquellen

UpdateDD kann unter den folgenden Adressen heruntergeladen werden:

<http://updatedd.philipp-benner.de>

<http://savannah.nongnu.org/download/updatedd/>

CVS Repository

```
export CVS_RSH="ssh"
```

```
cvs -z3 -d:ext:anoncvs@savannah.nongnu.org:/cvsroot/updatedd co updatedd
```