

Piccola guida su come offrire i servizi dhcp e named coordinandoli tra loro. In sostanza, dhcp oltre a rilasciare un indirizzo ip valido ai client che ne fanno richiesta, provvedera' ad aggiornare i files di zona di bind. Sara' quindi possibile effettuare il classico comando " ping client " e controllare se tale client e' acceso oppure spento. La presente guida non e' esaustiva, anzi e' solo un accenno a quanto occorre sapere effettivamente. La distribuzione di riferimento e' slackware 12. Il server dhcp e bind9 risultano gia' installati di default. Cominciamo con la configurazione di dhcpd. Il server dhcpd adegua il suo comportamento in base alle direttive del file di configurazione

`/etc/dhcpd.conf`

Il file dhcpd.conf e' un file di testo piano (ascii). Le linee vuote vengono ignorate, le linee che cominciano col carattere cancelletto " # " sono considerate commenti e non modificano il comportamento del servizio. Ogni riga di configurazione deve terminare col punto e virgola " ; " . Il file si compone di una sezione globale e di almeno una sezione locale. All'interno della sezione locale e' possibile ridefinire le direttive indicate nella sezione globale. Senza indicazione alcuna ecco che le direttive presenti nella sezione globale si applicano anche alle sezioni locali. All'interno del file sono presenti

a) parametri

b) dichiarazioni

I parametri indicano al servizio "come" fare qualcosa, mentre le dichiarazioni indicano al servizio la tipologia della rete (o delle reti). Analizziamo ora un elenco (non completo) delle direttive possibili all'interno del file `/etc/dhcpd.conf`

`authoritative;` Specifica che il servizio erogato e' "autorevole", e che puo' ricostituire i nodi configurati in modo errato.

`not-authoritative;` l'esatto opposto di cui sopra. Questa e' la direttiva predefinita. In pratica se non indichiamo nulla, il server dhcp assume la direttiva " not-authoritative "

`ddns-update-style interim;` Tale direttiva puo' assumere due valori: " internim " oppure " ad hoc " . Serve quando abbiamo anche un servizio dns da aggiornare in maniera automatica con gli indirizzi ip assegnati dal dhcp.

`default-lease-time 86400;` Tempo di affitto di un indirizzo ip da parte di un elaboratore client. Tale tempo e' espresso in secondi .

`max-lease-time 86400;` Tempo massimo di affitto di un indirizzo ip da parte di un elaboratore.

`option subnet-mask 255.255.255.0;` indica la corretta maschera di rete da rilasciare ai client

`option domain-name " nome ";` indica il nome di dominio da assegnare agli elaboratori client

`option domain-name-servers 10.0.0.1;` indica il corretto dns da indicare agli elaboratori client

option routers 10.0.0.1;            indica il corretto router da indicare  
agli elaboratori client

range 10.0.0.20 10.0.0.50; definisce l'intervallo di indirizzi ip da  
rilasciare.

host            definisce un indirizzo ip da assegnare staticamente ,  
riservandolo ad un mac adress specificato.

queste sono le direttive piu' immediate da utilizzare nella configurazione del  
server dhcp. Per ottenere un elenco completo delle direttive possibili  
inseribili nel file dhcpd.conf occorre consultare la pagina man dhcp-options .  
Analizziamo ora un esempio pratico. vogliamo creare un piccolo server dhcp  
che eroghi 20 indirizzi ip alla nostra lan. un ipotetico file di  
configurazione potrebbe essere il seguente:

authoritative;            # istruiamo il nostro server  
                          # affinche' sia autorevole

default-lease-time 28800; # impostiamo un tempo standard  
                          # di affitto dell'indirizzo

max-lease-time 28800;        # il tempo di affitto dell'indirizzo  
                          # qui indicato, non puo' essere  
                          # oltrepassato

ddns-update-style interim; # istruiamo il server dhcp affinche'  
                          # aggiorni anche un server dns se  
                          # presente

option subnet-mask 255.0.0.0; # indichiamo la corretta subnet mask  
                          # da rilasciare ai client.

option domain-name dominio; # indichiamo il corretto valore di  
                          # dominio da rilasciare ai client

option domain-name-servers 10.0.0.7; # indichiamo l'indirizzo dei server  
                          # dns da consultare

option routers 10.0.0.7;    # forniamo l'indirizzo del router

option netbios-name-servers 10.0.0.7; # indichiamo l'indirizzo del server  
                          # netbios. [in questo caso occorre  
                          # configurare anche samba affinche'  
                          # rilasci le informazioni netbios  
                          # richieste dai client. lo vedremo  
                          # in seguito]

option netbios-node-type 8; # abilitiamo anche il netbios over  
                          # tcp/ip

# subnet di prova            # commento: i commenti devono iniziare  
                          # obbligatoriamente con il carattere #

ubnet 10.0.0.0 netmask 255.0.0.0 {        # indichiamo la subnet ove opereremo

    deny client-updates;        # rifiutiamo le informazioni di  
                          # aggiornamento offerte dai client

    max-lease-time 28800;        # indichiamo i tempi di affitto standard

```

default-lease-time 28800;    # e massimo

authoritative;              # il nostro server e' autorevole per la
                             # subnet in oggetto

ddns-updates off;          # impediamo l'aggiornamento del dns

range 10.0.0.20 10.0.0.40; # indichiamo l'intervallo di indirizzi
                             # da erogare

}                            # chiudiamo la sottosezione. Le direttive
                             # riferite ad una data sottosezione devono
                             # essere comprese tra parentesi graffe.

```

Ipotizziamo ora di volere fare in modo che un determinato elaboratore ottenga sempre il medesimo indirizzo ip, per un motivo qualsiasi. Un buon sistema potrebbe essere aggiungere le righe seguenti nel file di configurazione /etc/dhcpd.conf

```

# il server                  # indichiamo un commento.

host p41500 {               # indichiamo il nome host ed apriamo
                             # la parentesi graffa per iniziare le
                             # dichiarazioni

    deny client-updates;    # nega espressamente ai client la
                             # possibilita' di aggiornare il server
                             # dns

    deny unknown-clients;  # non rilasciamo indirizzi ip a host
                             # sconosciuti (vedere la pagina man 5
                             # dhcpd.conf)

    ddns-updates off;      # impediamo l'aggiornamento del server
                             # dns se presente

    hardware ethernet 00:13:8f:c2:bc:eb; # stabiliamo a quale indirizzo
                                             # fisico rilasciare l'ip

    fixed-address 10.0.0.88; # indichiamo quale indirizzo ip rilasciare

}

```

Ecco un esempio di configurazione possibile per fare in modo che determinati client ottengano sempre il medesimo indirizzo:

```

# Configurazione Gruppi Host # indichiamo un commento

group {                    # indichiamo un gruppo
    option routers 192.168.0.2; # Ridefinizione di alcuni parametri
    option subnet-mask 255.255.255.0; # globali in modo
    option domain-name-servers 195.130.224.18; # specifico per il gruppo di host
                                             # elencati all'interno del gruppo
                                             # group

    host plutone {        # Identificazione dell'host da configurare
        option host-name "Plutone.Zeta.net"; # Assegnamento del nome Host
        hardware ethernet 00:A0:78:8E:9E:AA; # Identificazione tramite Mac Address della
                                             # scheda di rete

        fixed-address 192.168.0.51; # Assegnamento di un IP fisso
    }
}

```

```

}

host saturno {
option host-name "Saturno.Zeta.net";
hardware ethernet 00:48:54:6E:B0:4D;
fixed-address 192.168.0.52;
}
}

```

esempio tratto da : <http://openskill.info/infobox.php?ID=749>

Ecco ora un esempio molto semplice. In questo caso il server dhcp viene istruito per rilasciare ai client 100 indirizzi ip , come tetto massimo. Nessun altro compito viene richiesto al server dhcp

```

# dhcpd.conf
#
# Configuration file for ISC dhcpd (see 'man dhcpd.conf')
#
#
ddns-update-style ad-hoc;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.50 192.168.1.150;
default-lease-time 604800; # una settimana
max-lease-time 2592000; # 30 giorni
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
option domain-name-servers 213.156.54.80;
option domain-name "fastwebnet.it";
}

```

esempio tratto da :

[http://www.slacky.eu/wikislack/index.php?title=Configurare\\_un\\_DHCP\\_server](http://www.slacky.eu/wikislack/index.php?title=Configurare_un_DHCP_server)

Per maggiori dettagli sul server dhcp occorre consultare le seguenti pagine man

```

man dhcpd
man dhcpd-options
man dhcpd.conf

```

Su slackware 12.0 non esiste uno scrip di avvio per fare partire il server dhcp. Occorre forzare a mano la partenza del servizio, oppure inserire le direttive all'interno del file /etc/rc.d/rc.local, oppure (ed e' la soluzione ottimale) scrivere uno script di avvio e posizionare tale script nella directory /etc/rc.d . Sara' necessario inoltre modificare gli altri script di avvio, mantenendo la coerenza della distribuzione.

Ecco un ipotetico script di avvio, il cui nome e' rc.dhcpd

```

#!/bin/sh
#
# /etc/rc.d/rc.dhcpd
# This shell script takes care of starting and stopping
# the ISC DHCPD service
#

# Put the command line options here that you want to pass to dhcpd:
DHCPD_OPTIONS="-q eth0"

[ -x /usr/sbin/dhcpd ] || exit 0

[ -f /etc/dhcpd.conf ] || exit 0

```

```

start() {
    # Start daemons.
    echo -n "Starting dhcpd: /usr/sbin/dhcpd $DHCPD_OPTIONS "
    /usr/sbin/dhcpd $DHCPD_OPTIONS
    echo
}
stop() {
    # Stop daemons.
    echo -n "Shutting down dhcpd: "
    killall -TERM dhcpd
    echo
}
status() {
    PIDS=$(pidof dhcpd)
    if [ "$PIDS" == "" ]; then
        echo "dhcpd is not running!"
    else
        echo "dhcpd is running at pid(s) ${PIDS}."
    fi
}
restart() {
    stop
    start
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        start
        ;;
    status)
        status
        ;;
    *)
        echo "Usage: $0 {start|stop|status|restart}"
        ;;
esac

exit 0
#
# end script
#

```

Esempio tratto da :

[http://www.slackware.com/~alien/rc\\_scripts/other\\_rc\\_scripts/rc.dhcpd](http://www.slackware.com/~alien/rc_scripts/other_rc_scripts/rc.dhcpd)

Passiamo ora alla configurazione di named.

Il file di configurazione principale di named

e' named.conf. Posizionato

solitamente in /etc/. E' un file di testo piano (ascii). Le righe che

iniziano col doppio slash " // " sono considerate commenti e non modificano

il comportamento del servizio. Siccome la configurazione di bind e' davvero

laboriosa, preferisco riportare diversi esempi, compreso un caso reale, e

rimandare all'abbondante (ma anche criptica) documentazione. Si consulti :

```
man named
man named.conf
http://www.lnf.infn.it/computing/doc/AppuntiLinux/a2157.html
http://openskill.info/infobox.php?ID=742
/usr/doc/bind-9.4.1 [ su slackware 12.0 ]
```

Ecco il primo esempio di configurazione di named.conf

```
// Boot file for LAND-5 name server           // indichiamo un commento

options {                                     // specifichiamo la directory
    directory "/var/named";                  // ove sono presenti i files di
};                                           // zona

controls {                                   // indichiamo a named
    inet 127.0.0.1 allow { localhost; } keys { rndc_key; }; // su quale interfaccia
};                                           // mettersi in ascolto
                                           // e quale chiave usare

key "rndc_key" {                             // questa e' la chiave
    algorithm hmac-md5;                     // da utilizzare per dialogare
    secret "c3Ryb25nIGVub3VnaCBmb3IgYSBtYW4  // con named. E' spezzata su due
        gYnV0IG1hZGUgZm9yIGEgd29tYW4K";    // righe per motivi tipografici,
};                                           // in realta' risiede su un'unica
                                           // linea

zone "." {                                   // indica il file di zona
    type hint;                               // che contiene gli indirizzi
    file "root.hints";                       // dei root server
};

zone "0.0.127.in-addr.arpa" {                // risoluzione inversa di
    type master;                             // localhost.
    file "zone/127.0.0";                     //
};

zone "land-5.com" {                          // named e' autorevole per
    type master;                             // la zona " land-5.com "
    file "zone/land-5.com";                 // ed il file di zona e'
};                                           // si chiama land-5.com

zone "177.6.206.in-addr.arpa" {              // risoluzione inversa.
    type master;                             // il file di zona per la
    file "zone/206.6.177";                 // risoluzione inversa e'
};                                           // 206.6.177
```

Ora un secondo esempio di named.conf

```
options {                                     // indica a named
    directory "/var/lib/named";              // le posizioni di
    dump-file "/var/log/named_dump.db";     // lavoro, inoltre
    statistics-file "/var/log/named.stats"; // indica a quali altri
    forwarders { 62.31.176.39 ; 193.38.113.3; }; // server dns appoggiarsi
    listen-on-v6 { any; };                  //
    notify no;                               //
};
```

```

};

zone "." in {                                // indica il file contenente
    type hint;                               // gli indirizzi ip
    file "root.hint";                        // dei root server
};

zone "localhost" in {                       // indica il file da utilizzare
    type master;                             // per risolvere
    file "localhost.zone";                  // localhost
};

zone "0.0.127.in-addr.arpa" in {           // risoluzione inversa
    type master;                             // di localhost
    file "127.0.0.zone";                    //
};

zone "spring.wellho.net" in {              // named risolve la zona
    type master;                             // spring.wellho.net , usando
    file "/var/lib/named/wellho.zone";      // il file di zona wellho.zone
};

include "/etc/named.conf.include";         // viene aggiunta una direttiva
                                           // contenuta nel file esterno
                                           // named.conf.include

```

Ora un terzo esempio di named.conf :

```

options {                                   // definiamo la directory
    directory "/etc/domain";               // di lavoro
};                                           //

zone "." {                                  // definiamo le gli indirizzi
    type hint;                             // ip dei root server
    file "named.root";                     // possiamo ottenere il file presso
};                                           // ftp://ftp.rs.internic.net/domain/named.root

zone "localhost" {                         //
    type master;                           // definiamo localhost
    file "localhost";                      //
};

zone "0.0.127.in-addr.arpa" {              //
    type master;                           // risluzione inversa di localhost
    file "127.0.0";                        //
};

zone "company.xy" {                        // il file "company.xy" deve risiedere in
    type master;                           // /etc/domain/ directory, ed occorre crearlo
    file "company.xy";                     // a mano prima dell'avvio di bind
};

zone "42.168.192.in-addr.arpa" {          // il file "192.168.42" should risiedere
    type master;                           // in /etc/domain/ directory, ed occorre crearlo
    file "192.168.42";                     // a mano prima dell'avvio di bind
};

```

Ora un quarto esempio di named.conf. Si tratta di un caso reale. In questo caso il server dhcp oltre a rilasciare l'indirizzo ip a chi ne fa' richiesta, indichera' a named di eseguire l'aggiornamento del file di zona. insomma, si concretizza

l'accoppiata dhcp+bind

```
options {                                // definiamo le opzioni
    directory "/var/named/caching-example/"; // di funzionamento di
    pid-file "/var/run/named/named.pid";     // bind
    listen-on {127.0.0.1; 10.0.0.1;};      //
    forwarders { 62.94.0.1; 208.67.220.220; }; //
};

include      "/etc/rndc.key";             // inseriamo una direttiva
                                                // presente in un file esterno.
                                                // in qst caso si tratta della
                                                // chiave con la quale dhcpd
                                                // e bind dialogano.

zone "." {                                //
    type hint;                            // definiamo i corretti root server
    file "named.ca";                      //
};

zone "0.0.127.in-addr.arpa" {             // definiamo la risoluzione
    type master;                          // inversa di localhost
    file "named.local";                   //
};

zone "veronese" {                         //
    type master;                          // definiamo la zona per la quale
    file "veronese";                      // vogliamo rendere autorevole
    allow-query { any; };                 // il servizio
    allow-update { key rndc-key; };       //
};

zone "0.0.10.in-addr.arpa" {              // definiamo la risoluzione inversa
    type master;                          // per la zona ove bind
    file "risoluzione-inversa";           // e' autorevole
    allow-query { any; };                 //
    allow-update { key rndc-key; };       //
};
```

La creazione del file named.conf non e' sufficiente per eseguire il servizio. Occorre almeno un file di zona. All'interno di tale file devono essere presenti i riferimenti corretti di indirizzo\_ip e nome\_host per i quali si vuole ottenere la risoluzione. Il file assume grossomodo questo aspetto

```
computer_1  indirizzo_1
computer_2  indirizzo_2
computer_3  indirizzo_3
computer_4  indirizzo_4
```

Nei files di zona, le linee che iniziano con un punto e virgola " ; " indicano un commento. Ecco un elenco incompleto dei record DNS maggiormente utilizzati:

SOA	inizio di una zona di autorità
NS	un name server autoritativo
A	un indirizzo host
CNAME	il nome canonico per un alias
MX	mail exchanger
PTR	un puntatore a nome di dominio (usato nel DNS inverso)

Rispettando lo schema di prima, forniamo alcuni esempi di file di zona, rimandando

alla copiosa documentazione per approfondimenti.

```
$ORIGIN .
$TTL 172800 ; 2 days
esempio.com IN SOA ns1.esempio.com. root.ns1.esempio.com. (
    2003021512 ; serial
    28800 ; refresh (8 hours)
    7200 ; retry (2 hours)
    604800 ; expire (1 week)
    86400 ; minimum (1 day)
)
NS ns1.esempio.com.
$ORIGIN esempio.com.
$TTL 3600 ; 1 hour
dhcpclient A 192.168.0.98
    TXT "311e98e7bea987126c3037d1b1f4c7b99d"
$TTL 172800 ; 2 days
ns1 A 192.168.0.1
mac A 192.168.0.2
linux A 192.168.0.3
windows A 192.168.0.4
webftp A 192.168.0.5
mail A 192.168.0.6
```

;Aliases

```
www.esempio.com. CNAME webftp.esempio.com.
ftp.esempio.com. CNAME webftp.esempio.com.
win.esempio.com. CNAME windows.esempio.com.
```

;Definisco il mail server per il dominio

```
esempio.com. MX 10 mail.esempio.com.
esempio.com. MX 20 mail.nostroisp.com.
```

esempio da :

<http://openskill.info/infobox.php?ID=763>

Un'altro esempio di file di zona:

```
$TTL 3600 ; 1 hour
example.org. IN SOA ns1.example.org. admin.example.org. (
    2006051501 ; Serial
    10800 ; Refresh
    3600 ; Retry
    604800 ; Expire
    86400 ; Minimum TTL
)
```

; DNS Servers

```
IN NS ns1.example.org.
IN NS ns2.example.org.
```

; MX Records

```
IN MX 10 mx.example.org.
IN MX 20 mail.example.org.
```

```
IN A 192.168.1.1
```

; Machine Names

```
localhost IN A 127.0.0.1
ns1 IN A 192.168.1.2
ns2 IN A 192.168.1.3
```

```
mx      IN  A    192.168.1.4
mail    IN  A    192.168.1.5
```

; Aliases

```
www     IN  CNAME @
```

esempio da :

[http://www.freebsd.org/doc/it\\_IT.ISO8859-15/books/handbook/network-dns.html](http://www.freebsd.org/doc/it_IT.ISO8859-15/books/handbook/network-dns.html)

Ora un file di zona realmente esistente. In realta' il file di zona presentato appartiene ad un server dns che accetta aggiornamenti tramite dhcp. Insomma, il server dhcp rilascia l'indirizzo ip ed aggiorna il relativo file di zona.

```
$ORIGIN .
```

```
$TTL 86400 ; 1 day
```

```
veronese      IN  SOA    portatile.veronese. root.portatile.veronese. (
                2004102522 ; serial
                28800    ; refresh (8 hours)
                7200    ; retry (2 hours)
                604800   ; expire (1 week)
                86400   ; minimum (1 day)
            )
                NS    portatile.veronese.
```

```
$ORIGIN veronese.
```

```
portatile     A    10.0.0.1
```

```
$TTL 14400 ; 4 hours
```

```
testa        A    10.0.0.120
```

```
              TXT   "31cf8f5ac67d8989413617e5cedc53e422"
```

Ora un paio di esempi di file di zona abilitati alla risoluzione inversa. Il primo esempio e' reale, cioe' si riferisce ad un server effettivamente operativo.

```
$ORIGIN .
```

```
$TTL 86400 ; 1 day
```

```
0.0.10.in-addr.arpa IN SOA    portatile.veronese. root.portatile.veronese. (
                2004095925 ; serial
                28800    ; refresh (8 hours)
                7200    ; retry (2 hours)
                604800   ; expire (1 week)
                86400   ; minimum (1 day)
            )
                NS    portatile.veronese.
```

```
$ORIGIN 0.0.10.in-addr.arpa.
```

```
1          PTR    portatile.veronese.
```

```
$TTL
```

```
14400; 4 hours
```

```
120       PTR    testa.veronese.
```

Ora un altro esempio, questa volta tratto da:

<http://openskill.info/infobox.php?ID=764>

```
$TTL 2D
```

```
0.168.192.in-addr.arpa.      IN  SOA    ns1.esempio.com. root.ns1.esempio.com. (
                2003021502 ;serial
                28800    ;refresh
                7200    ;retry
                604800   ;expire
                28800 ) ;minimum
IN  NS    ns1.esempio.com.
```

```
1          IN  PTR    ns1.esempio.com.
```

```
2          IN  PTR    mac.esempio.com.
```

```
3          IN  PTR    linux.esempio.com.
```

```
4      IN  PTR  windows.esempio.com.
5      IN  PTR  webftp.esempio.com.
6      IN  PTR  mail.esempio.com.
```

Ora che siamo in gradi di scrivere i corretti file di configurazione per named, occorre controllare che tali files siano esatti nella loro costruzione. Per ottenere questo controllo possiamo avvalerci dei comandi named-checkconf e named-checkzone . La relativa pagina man restituisce la sintassi per il loro utilizzo.

Occorre ora fare dialogare tra loro i due servizi, il server dhcp ed il server dei nomi, assumendo che operino sul medesimo elaboratore.

Tale scopo viene ottenuto tramite l'utility rndc e tramite l'altra utility rndc-confgen.

Come primo passaggio occorre creare la chiave crittografica da condividere tra i due server (dhcp e bind). Otteniamo tale risultato invocando rndc-confgen anche senza argomenti (dobbiamo possedere i privilegi dell'utente root). Ecco un output possibile:

```
root@portatile:/home/claudio# rndc-confgen      [invio]

# Start of rndc.conf
key "rndc-key" {
    algorithm hmac-md5;
    secret "ffV5+eVLg4/FyThCzuRhmg==";
};

options {
    default-key "rndc-key";
    default-server 127.0.0.1;
    default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndc-key" {
#     algorithm hmac-md5;
#     secret "ffV5+eVLg4/FyThCzuRhmg==";
# };
#
# controls {
#     inet 127.0.0.1 port 953
#     allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# End of named.conf
```

ci viene fornita una chiave, e ci vengono anche proposte alcune indicazioni sul corretto utilizzo. In realta' risulta piu' efficace disattendere queste indicazioni e inserire la chiave ottenuta in un file. tale file verra' poi richiamato durante la configurazione di dhcpd (nel file /etc/dhcpd.conf) e durante la configurazione di named (esattamente nel file /etc/named.conf), tramite la direttiva " include ". L'utility rndc invece ci consente di impartire ordini a named anche durante l'esecuzione dello stesso. Insomma, e un sistema di gestione del servizio. Invocando il comando senza argomenti otteniamo un elenco abbastanza chiaro del possibile utilizzo:

```
root@portatile:/home/claudio# rndc      [invio]
```

```
Usage: rndc [-c config] [-s server] [-p port]
          [-k key-file ] [-y key] [-V] command
```

command is one of the following:

reload      Reload configuration file and zones.  
 reload zone [class [view]]  
             Reload a single zone.  
 refresh zone [class [view]]  
             Schedule immediate maintenance for a zone.  
 retransfer zone [class [view]]  
             Retransfer a single zone without checking serial number.  
 freeze      Suspend updates to all dynamic zones.  
 freeze zone [class [view]]  
             Suspend updates to a dynamic zone.  
 thaw        Enable updates to all dynamic zones and reload them.  
 thaw zone [class [view]]  
             Enable updates to a frozen dynamic zone and reload it.  
 notify zone [class [view]]  
             Resend NOTIFY messages for the zone.  
 reconfig    Reload configuration file and new zones only.  
 stats       Write server statistics to the statistics file.  
 querylog    Toggle query logging.  
 dumpdb [-all|-cache|-zones] [view ...]  
             Dump cache(s) to the dump file (named `dump.db`).  
 stop        Save pending updates to master files and stop the server.  
 stop -p     Save pending updates to master files and stop the server  
             reporting process id.  
 halt        Stop the server without saving pending updates.  
 halt -p     Stop the server without saving pending updates reporting  
             process id.  
 trace       Increment debugging level by one.  
 trace level Change the debugging level.  
 notrace     Set debugging level to 0.  
 flush       Flushes all of the server's caches.  
 flush [view] Flushes the server's cache for a view.  
 flushname name [view]  
             Flush the given name from the server's cache(s)  
 status      Display status of the server.  
 recursing   Dump the queries that are currently recursing (named `recursing`)  
 validation newstate [view]  
             Enable / disable DNSSEC validation.  
 \*restart    Restart the server.

\* == not yet implemented

Version: 9.4.1

autore: Veronese Claudio  
 collaboratore: Matteucci Gianluca  
 collaboratore: Pistamiglio Angelo

Si consente al lug di Alessandria (allug) l'utilizzo del presente  
 tutorial, anche parziale, anche rivisto e riadattato, senza limitazione  
 alcuna.