DMS Documentation

Administration Operational Procedures

Adding a DNS Slave to DMS

Based on Debian Wheezy

To prevent installation of recommended packages add the following to /etc/apt/apt.conf.d/00local.conf:

```
// No point in installing a lot of fat on VM servers
APT::Install-Recommends "0";
APT::Install-Suggests "0";
```

Install these packages:

```
# aptitude install bind9 racoon rsync cron-apt bind9-host \
  screen psmisc procps tree sysstat lsof open-vm-tools
```

Select raccon-tool method of raccon configuration

Sysctl IPSEC settings

To prevent network problems with running out of buffers, create the file /etc/sysctl.d/30-dms-core-net.conf with the following contents:

```
# Tune kernel for heave IPSEC DNS work.
# Up the max connection buffers
net.core.somaxconn=8192
net.core.netdev_max_backlog=8192
# Reduce TCP final timeout
net.ipv4.tcp_fin_timeout=10
# Increase size of xfrm tables
net.ipv6.xfrm6_gc_thresh=16384
net.ipv4.xfrm4_gc_thresh=16384
```

and then reload sysctls with

```
# service procps start
```

Racoon

Edit /etc/racoon/racoon-tool.conf

```
connection(%default):
        src_ip: 2001:470:c:110e::2
        admin_status: disabled

peer(2001:470:f012:2::2):

# DMS Master 1
connection(shalom-ext):
        dst_ip: 2001:470:f012:2::2
        admin_status: enabled

# DMS Master 2
peer(2001:470:f012:2::3):

connection(shalom-dr):
        dst_ip: 2001:470:f012:2::3
        admin_status: enabled
```

Note that IPSEC connections to each DMS master server have to be configured on each Slave.

Edit /etc/racoon/psk.txt to set PSK secrets. Generate with

```
# pwgen -asn 128 1
Dbj4gxNUgnW0wx8ax8de2Sa38izza9HtzNWCjxkCXFXGWxokeqOd7VgX6sRRDiFVptNQTBBMPNP
mC8iK6QCJz3bJzQT06O90QqR6RGbE1hpLKG5xPfNd7DOBLKlnQtcn
```

(You can apt-get install it). Make **SURE** each individual IPSEC connection has a unique PSK key for security. They can be generated easily, and cut/paste over terminal root session, so no big loss if they are lost.

Rsync

1. Edit /etc/default/rsync, and enable rsyncd
2. Create /etc/rsyncd.conf:

```
hosts allow = 2001:470:f012:2::2/128 2001:470:f012:2::3/128
secrets file = /etc/rsyncd.secrets

[dnsconf]
        path = /etc/bind/rsync-config
        uid=bind
        gid=bind
        comment = Slave server config area
        auth users = dnsconf
        use chroot = yes
        read only = no
```

3. Create /etc/rsyncd.secrets

```
dnsconf:etc-net24-rsync-slave-password
```

4. Do this at the shell to create target /etc/bind/rsync-config directory:

```
mkdir /etc/bind/rsync-config
chown bind:bind /etc/bind/rsync-config
```

5. and named slave directory

```
mkdir /var/cache/bind/slave
chown root:bind /var/cache/bind/slave
chmod 775 /var/cache/bind/slave
```

6. Start rsyncd

```
# service rsync start
```

7. Test connectivity from DMS Masters

```
dms-master1# telnet new-slave domain
dms-master1# telnet new-slave rsync
dms-master1# telnet new-slave 973
dms-master2# telnet new-slave domain
dms-master2# telnet new-slave rsync
dms-master2# telnet new-slave 973


zone_tool> create_slave new-slave-name ip-address
zone_tool> rsync_admin_config no_rndc
```

Bind9

Change /etc/bind/named.conf.options to the following:

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses
replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };


//========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See
https://www.isc.org/bind-keys

//========================================================================
        // dnssec-validation auto;

        // auth-nxdomain no;    # conform to RFC1035

        listen-on { localhost; };
        listen-on-v6 { any; };
        include "/etc/bind/rsync-config/options.conf";
};
```

Note that the listen directives are given in file, Debian options commented out, as they are set in the rsynced include at the bottom.

Change /etc/bind/named.conf.local to the following:

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

// rndc config
include "/etc/bind/rndc.key";
include "/etc/bind/rsync-config/rndc-remote.key";
include "/etc/bind/rsync-config/controls.conf";
// Logging configuration
include "/etc/bind/rsync-config/logging.conf";
// Secondary zones
include "/etc/bind/rsync-config/bind9.conf";
```

This file is used to include all the required bits from the /etc/bind/rsync-config directory. All this configuration can now be updated from the master server, and the slave reconfigured – but watch it when you go changing the rndc keys.

Restart bind9

```
# service bind9 restart
```

and check /var/log/syslog for any errors.

Check that on the master servers that zone_tool rsync_admin_config works - it by default will rndc the slave.

```
dms-master1# zone_tool write_rndc_conf
dms-master1# zone_tool rsync_server_admin_config

dms-master2# zone_tool write_rndc_conf
dms-master2# zone_tool rsync_server_admin_config
```

Enable Server

On the live DMS master, enable the slave, and watch that it changes state to OK. This may take 15-20 minutes

```
dms-master-live# zone_tool
zone_tool > enable_server <slave-name>
.
.
.
zone_tool > ls_pending_events
ServerSMConfigChange dms-master1                          Tue Aug 21
14:19:39 2012
ServerSMConfigChange dms-master2                          Tue Aug 21
14:19:39 2012
ServerSMConfigChange dms-slave0                           Tue Aug 21
14:19:39 2012
ServerSMConfigChange dms-slave1                           Tue Aug 21
14:19:39 2012
MasterSMHoldTimeout                                       Tue Aug 21
14:29:39 2012

.
.
.
zone_tool > show_sg vygr-one
        sg_name:              vygr-one
        config_dir:           /etc/net24/slave-config-templates
        master_address:       None
        master_alt_address:   None
        replica_sg:           False
        zone_count:           37

        Named slave configuration state:
        dms-s1-akl                    2406:1e00:1001:2::2

                OK
        dms-s1-chc                    2406:3e00:1001:2::2

                OK
zone_tool >
```

Break Fix Scenarios

- Log and Configuration Files
- Checking DMS status
- Failing Over as Master Server has Burned (or Subject to EQC Claim)
- Stuck Zone not Propagating
- MasterSM Stuck, New Zones not Being Created
- Stuck ServerSM
- Rebuilding named data from database
- Failed Master, Replica /etc not up to date
- Recovering DB from Backup

Log and Configuration Files

The following are detailed elsewhere in the documentation

/var/log/net24/net24dmd.log*	net24dmd logs
/var/log/local7.log	DMS named logs
/var/log/syslog	Basically everything
/et/net24/net24.conf	net24dmd, wsgi and zone_tool configuration file
/etc/net24	Various passwords, templates and things

See Named.conf and Zone Templating for more details.

Checking DMS status

1) Check that named, postgres, and net24dmd are running on the master

2) Using zone_tool show_dms_status on master server

```
zone_tool > show_dms_status

show_master_status:

        MASTER_SERVER:     dms-akl

        NAMED master configuration state:

        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        Wed Nov  7 16:52:36 2012
        hold_stop:         Wed Nov  7 17:02:36 2012
        replica_sg_name:   vygr-replica
        state:             HOLD

show_replica_sg:
        sg_name:             vygr-replica
        config_dir:          /etc/net24/server-config-templates
        master_address:      2406:1e00:1001:1::2
        master_alt_address:  2406:3e00:1001:1::2
        replica_sg:          True
        zone_count:          37

        Replica SG named status:
        dms-chc                     2406:3e00:1001:1::2

                OK
```

```
ls_server:
dms-akl                      Wed Nov  7 16:52:46 2012               OK
        2406:1e00:1001:1::2                     None
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dms-chc                      Wed Nov  7 16:52:46 2012               OK
        2406:3e00:1001:1::2                     210.5.48.242
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dms-s1-akl                   Wed Nov  7 16:31:04 2012               RETRY
        2406:1e00:1001:2::2                     103.4.136.226
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
        retry_msg:
          Server 'dms-s1-akl': SOA query - timeout waiting for
          response, retrying
dms-s1-chc                   Wed Nov  7 16:52:46 2012               OK
        2406:3e00:1001:2::2                     210.5.48.226
        ping: 5 packets transmitted, 5 received, 0.00% packet loss

list_pending_events:
ServerSMConfigure         dms-s1-akl                  Wed Nov  7 16:57:22
2012
ServerSMCheckServer       dms-chc                     Wed Nov  7 16:53:55
2012
ServerSMCheckServer       dms-akl                     Wed Nov  7 16:55:46
2012
ServerSMCheckServer       dms-s1-chc                  Wed Nov  7 16:57:06
2012
MasterSMHoldTimeout                                   Wed Nov  7 17:02:36
2012
```

```
zone_tool >
```

- Check Master server name, that machine is actually the master
- Check master state, HOLD means named reconfigured in the last 10 minutes
- All servers shown at bottom should be in OK ior CONFIG states, staying in RETRY or BROKEN means server may not be contactable. RETRY or BROKEN states should also have a retry_msg: field giving the associated log message
- list_pending_events shows events that have to be processed
  Any events that are scheduled in the past may indicate net24dmd having serious problems

Failing Over as Master Server has Burned (or Subject to EQC Claim)

On the Replica:

```
dms-chc: -root- [~]
# dms_promote_replica
+ perl -pe s/^#(\s*local7.* :ompgsql:\S+,dms,rsyslog,.*$)/\1/ -i
/etc/rsyslog.d/pgsql.conf
+ set +x
[ ok ] Stopping enhanced syslogd: rsyslogd.
[ ok ] Starting enhanced syslogd: rsyslogd.
+ perl -pe s/^NET24DMD_ENABLE=.*$/NET24DMD_ENABLE=true/ -i
/etc/default/net24dmd
+ perl -pe s/^OPTIONS=.*$/OPTIONS="-u bind"/ -i /etc/default/bind9
+ set +x
[....] Stopping domain name service...: bind9waiting for pid 8744 to die
. ok
[ ok ] Starting domain name service...: bind9.
[ ok ] Starting net24dmd: net24dmd.
+ zone_tool write_rndc_conf
+ zone_tool reconfig_all
+ perl -pe s/^#+(.*zone_tool vacuum_all)$/\1/ -i /etc/cron.d/dms-core
+ do_dms_wsgi
+ return 0
+ perl -pe s/^(\s*exit\s+0.*$)/#\1/ -i /etc/default/apache2
+ set +x
[ ok ] Starting web server: apache2.

dms-chc: -root- [~]
#
```

Wait till servers started, and then use zone_tool show_dms_status to check that everything becomes OK. This may take 15 minutes. The section about ls_pending_events will give scheduled times for servers to become configured.

```
dms-chc: -root- [~]
# zone_tool show_dms_status

show_master_status:
```

```
        MASTER_SERVER:     dms-chc

        NAMED master configuration state:

        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        Fri Nov  9 08:30:49 2012
        hold_stop:         Fri Nov  9 08:40:49 2012
        replica_sg_name:   vygr-replica
        state:             HOLD

show_replica_sg:
        sg_name:             vygr-replica
        config_dir:          /etc/net24/server-config-templates
        master_address:      2406:1e00:1001:1::2
        master_alt_address:  2406:3e00:1001:1::2
        replica_sg:          True
        zone_count:          37

        Replica SG named status:
        dms-akl                      2406:1e00:1001:1::2

                RETRY

ls_server:
dms-akl                      Fri Nov  9 08:23:08 2012               RETRY
        2406:1e00:1001:1::2                     None
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
        retry_msg:
          Server 'dms-akl': SOA query - timeout waiting for response,
          retrying
dms-chc                      Fri Nov  9 08:30:58 2012               OK
        2406:3e00:1001:1::2                     210.5.48.242
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dms-s1-akl                   Fri Nov  9 08:30:58 2012               OK
        2406:1e00:1001:2::2                     103.4.136.226
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dms-s1-chc                   Fri Nov  9 08:30:58 2012               OK
        2406:3e00:1001:2::2                     210.5.48.226
        ping: 5 packets transmitted, 5 received, 0.00% packet loss

list_pending_events:
ServerSMCheckServer       dms-chc                    Fri Nov  9 08:39:53
2012
MasterSMHoldTimeout                                  Fri Nov  9 08:40:49
2012
ServerSMCheckServer       dms-s1-chc                 Fri Nov  9 08:40:08
2012
ServerSMCheckServer       dms-s1-akl                 Fri Nov  9 08:36:01
2012
ServerSMConfigure         dms-akl                    Fri Nov  9 08:50:17
2012
```

```
dms-chc: -root- [~]
#
```

A new replica will need to be installed as per DMS Master Server Install

Stuck Zone not Propagating

```
zone_tool > show_zonesm wham-blam.org
        name:            wham-blam.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 10:51:14 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_at:       None
        locked_by:       None
        mtime:           Thu Aug 23 10:51:14 2012
        nsec3:           True
        reference:       nutty-nutty@ANATHOTH-NET
        sg_name:         anathoth-internal
        soa_serial:      2012091400
        state:           UNCONFIG
        use_apex_ns:     True
        zi_candidate_id: 102880
        zi_id:           102880
        zone_id:         101448
        zone_type:       DynDNSZoneSM

        zi_id:           102880
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Fri Sep 14 10:55:59 2012
        mtime:           Fri Sep 14 11:12:10 2012
        ptime:           Fri Sep 14 11:12:10 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.internal.anathoth.net.
        soa_refresh:     24h
        soa_retry:       900
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012091400
        soa_ttl:         None
        zone_id:         101448
        zone_ttl:        24h
```

Maybe as above. Can be caused by:

- Failed events (manually failed or otherwise, Events queue deleted in DB, permissions problems as follows)

- Permissions problems on the master server on the /var/lib/bind/dynamic directory - should be

```
ls -ld /var/lib/bind/dynamic/
drwxrwsr-x 2 bind net24dmd 487424 Nov  9 08:47 /var/lib/bind/dynamic/
```

```
zone_tool > show_zonesm wham-blam.org
        name:            wham-blam.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 10:51:14 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_at:       None
        locked_by:       None
        mtime:           Thu Aug 23 10:51:14 2012
        nsec3:           True
        reference:       nutty-nutty@ANATHOTH-NET
        sg_name:         anathoth-internal
        soa_serial:      2012091400
        state:           RESET
        use_apex_ns:     True
        zi_candidate_id: 102880
        zi_id:           102880
        zone_id:         101448
        zone_type:       DynDNSZoneSM

        zi_id:           102880
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Fri Sep 14 10:55:59 2012
        mtime:           Fri Sep 14 11:12:10 2012
        ptime:           Fri Sep 14 11:12:10 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.internal.anathoth.net.
        soa_refresh:     24h
        soa_retry:       900
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012091400
        soa_ttl:         None
        zone_id:         101448
        zone_ttl:        24h
```

and then use show_zonesm to check that zone state goes to PUBLISHED within 15 minutes. ls_pending_events

may also be useful, as it will show the events to do with the zone being published.

```
show_zonesm wham-blam.org
        name:            wham-blam.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 10:51:14 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_at:       None
        locked_by:       None
        mtime:           Thu Aug 23 10:51:14 2012
        nsec3:           True
        reference:       nutty-nutty@ANATHOTH-NET
        sg_name:         anathoth-internal
        soa_serial:      2012091400
        state:           RESET
        use_apex_ns:     True
        zi_candidate_id: 102880
        zi_id:           102880
        zone_id:         101448
        zone_type:       DynDNSZoneSM

        zi_id:           102880
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Fri Sep 14 10:55:59 2012
        mtime:           Fri Sep 14 11:12:10 2012
        ptime:           Fri Sep 14 11:12:10 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.internal.anathoth.net.
        soa_refresh:     24h
        soa_retry:       900
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012091400
        soa_ttl:         None
        zone_id:         101448
        zone_ttl:        24h
zone_tool > ls_pending_events
ServerSMCheckServer       shalom                    Fri Nov  9 08:50:35
2012
ServerSMCheckServer       shalom-ext                Fri Nov  9 08:50:40
2012
ServerSMCheckServer       shalom-dr                 Fri Nov  9 08:50:46
2012
ServerSMCheckServer       dns-slave1                Fri Nov  9 08:50:53
2012
ServerSMConfigure         en-gedi-auth              Fri Nov  9 08:55:31
2012
```

```
ZoneSMConfig              wham-blam.org.              Fri Nov  9 08:47:07
2012
MasterSMHoldTimeout                                   Fri Nov  9 08:56:52
2012
ServerSMCheckServer       dns-slave0                  Fri Nov  9 08:54:29
2012
zone_tool > show_zonesm wham-blam.org
        name:            wham-blam.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 10:51:14 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_at:       None
        locked_by:       None
        mtime:           Thu Aug 23 10:51:14 2012
        nsec3:           True
        reference:       nutty-nutty@ANATHOTH-NET
        sg_name:         anathoth-internal
        soa_serial:      2012091400
        state:           UNCONFIG
        use_apex_ns:     True
        zi_candidate_id: 102880
        zi_id:           102880
        zone_id:         101448
        zone_type:       DynDNSZoneSM

        zi_id:           102880
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Fri Sep 14 10:55:59 2012
        mtime:           Fri Sep 14 11:12:10 2012
        ptime:           Fri Sep 14 11:12:10 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.internal.anathoth.net.
        soa_refresh:     24h
        soa_retry:       900
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012091400
        soa_ttl:         None
        zone_id:         101448
        zone_ttl:        24h
zone_tool > ls_pending_events
ServerSMCheckServer       shalom                      Fri Nov  9 08:50:35
2012
ServerSMCheckServer       shalom-ext                  Fri Nov  9 08:50:40
2012
ServerSMCheckServer       shalom-dr                   Fri Nov  9 08:50:46
2012
ServerSMCheckServer       dns-slave1                  Fri Nov  9 08:50:53
```

```
2012
ServerSMConfigure         en-gedi-auth                Fri Nov  9 08:55:31
2012
MasterSMHoldTimeout                                   Fri Nov  9 08:56:52
2012
ServerSMCheckServer       dns-slave0                  Fri Nov  9 08:54:29
2012
ZoneSMReconfigUpdate      wham-blam.org.              Fri Nov  9 08:57:10
2012
zone_tool > ls_pending_events
ServerSMCheckServer       shalom-ext                  Fri Nov  9 09:00:25
2012
ServerSMCheckServer       shalom-dr                   Fri Nov  9 09:00:44
2012
ServerSMCheckServer       dns-slave0                  Fri Nov  9 09:01:25
2012
ServerSMCheckServer       dns-slave1                  Fri Nov  9 09:02:11
2012
ServerSMConfigure         en-gedi-auth                Fri Nov  9 09:06:15
2012
MasterSMHoldTimeout                                   Fri Nov  9 09:06:57
2012
ServerSMCheckServer       shalom                      Fri Nov  9 09:05:11
2012
zone_tool > show_zonesm wham-blam.org
        name:            wham-blam.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 10:51:14 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_at:       None
        locked_by:       None
        mtime:           Thu Aug 23 10:51:14 2012
        nsec3:           True
        reference:       nutty-nutty@ANATHOTH-NET
        sg_name:         anathoth-internal
        soa_serial:      2012091400
        state:           PUBLISHED
        use_apex_ns:     True
        zi_candidate_id: 102880
        zi_id:           102880
        zone_id:         101448
        zone_type:       DynDNSZoneSM

        zi_id:           102880
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Fri Sep 14 10:55:59 2012
        mtime:           Fri Nov  9 08:57:13 2012
        ptime:           Fri Nov  9 08:57:13 2012
```

```
soa_expire:      7d
soa_minimum:     600
soa_mname:       ns1.internal.anathoth.net.
soa_refresh:     24h
soa_retry:       900
soa_rname:       matthewgrant5.gmail.com.
soa_serial:      2012091400
soa_ttl:         None
zone_id:         101448
```

```
        zone_ttl:        24h
zone_tool >
```

MasterSM Stuck, New Zones not Being Created

Can be caused by:

- Failed MasterSMHoldTimeout events (manually failed or otherwise, Events queue deleted in DB etc)
- Permissions problems on the master server on the /etc/bind/master-config directory - Should be '2755 net24dmd:bind'

```
shalom-ext: -grantma- [~]
$ ls -ld /etc/bind/master-config
drwxr-sr-x 2 net24dmd bind 4096 Nov  9 08:56 /etc/bind/master-config
```

This shows up in zone_tool show_dms_status:

```
zone_tool > show_dms_status

show_master_status:

        MASTER_SERVER:     dms-akl

        NAMED master configuration state:

        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        Wed Nov  7 16:52:36 2012
        hold_stop:         Wed Nov  7 17:02:36 2012
        replica_sg_name:   vygr-replica
        state:             HOLD

show_replica_sg:
        sg_name:             vygr-replica
        config_dir:          /etc/net24/server-config-templates
        master_address:      2406:1e00:1001:1::2
        master_alt_address:  2406:3e00:1001:1::2
        replica_sg:          True
        zone_count:          37

        Replica SG named status:
        dms-chc                      2406:3e00:1001:1::2

                OK

ls_server:
dms-akl                     Wed Nov  7 16:52:46 2012                OK
        2406:1e00:1001:1::2                     None
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
```

```
dms-chc                      Wed Nov  7 16:52:46 2012                OK
        2406:3e00:1001:1::2                     210.5.48.242
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dms-s1-akl                   Wed Nov  7 16:31:04 2012                RETRY
        2406:1e00:1001:2::2                     103.4.136.226
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
        retry_msg:
          Server 'dms-s1-akl': SOA query - timeout waiting for
          response, retrying
dms-s1-chc                   Wed Nov  7 16:52:46 2012                OK
        2406:3e00:1001:2::2                     210.5.48.226
        ping: 5 packets transmitted, 5 received, 0.00% packet loss

list_pending_events:
ServerSMConfigure         dms-s1-akl                  Wed Nov  7 16:57:22
2012
ServerSMCheckServer       dms-chc                     Wed Nov  7 16:53:55
2012
ServerSMCheckServer       dms-akl                     Wed Nov  7 16:55:46
2012
ServerSMCheckServer       dms-s1-chc                  Wed Nov  7 16:57:06
2012


zone_tool > exit

dms-akl: -root- [~]
```

```
# date
Wed Nov  7 16:54:42 NZDT 2012
```

Key things to look for:

- master status section shows hold_start and hold_stop being in the past
- there is no MasterSMHoldTimeout event

> The MasterSM state machine forward posts the MasterSMHoldTimeout event when entering the HOLD state. If it does not get created or disappears or fails due to unforeseen events with outages etc, the MasterSM will end up stuck as above.

The fix is to do 'zone_tool reset_master'. This will reset the MasterSM state machine.

Stuck ServerSM

Just like the Master state machine getting stuck because of a missing MasterSMHoldTimeout event, Server SMs can end up being stuck in the CONFIG, RETRY or BROKEN states due to missing events. There will be missing 'ServerSMConfigure' events for the server in the ls_pending_events output.

```
zone_tool > show_dms_status
show_master_status:
        MASTER_SERVER:     shalom-ext
        NAMED master configuration state:
        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        None
        hold_stop:         None
        replica_sg_name:   anathoth-replica
        state:             READY
show_replica_sg:
        sg_name:             anathoth-replica
        config_dir:          /etc/bind/anathoth-master
        master_address:      2001:470:f012:2::2
        master_alt_address:  2001:470:f012:2::3
        replica_sg:          True
        zone_count:          14
        Replica SG named status:
        shalom-dr                    2001:470:f012:2::3
               OK
ls_server:
dns-slave0                   Fri Nov  9 09:56:48 2012                OK
        2001:470:c:110e::2                      111.65.238.10
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dns-slave1                   Fri Nov  9 09:56:38 2012                OK
        2001:470:66:23::2                       111.65.238.11
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
en-gedi-auth                 Thu Nov  8 18:01:07 2012                RETRY
        fd14:828:ba69:6:5054:ff:fe39:54f9       172.31.12.2
        ping: 5 packets transmitted, 0 received, 100.00% packet loss
        retry_msg:
          Server 'en-gedi-auth': failed to rsync include files,
          Command ['rsync', '--quiet', '-av', '--password-file',
          '/etc/net24/rsync-dnsconf-password', '/var/lib/net24/dms-sg
          /anathoth-internal/',
          'dnsconf@[fd14:828:ba69:6:5054:ff:fe39:54f9]::dnsconf/']'
          returned non-zero exit status 10, rsync: failed to connect
          to fd14:828:ba69:6:5054:ff:fe39:54f9
          (fd14:828:ba69:6:5054:ff:fe39:54f9): Connection timed out
          (110), rsync error: error in socket IO (code 10) at
          clientserver.c(122) [sender=3.0.9]
shalom                       Fri Nov  9 09:56:19 2012                OK
        fd14:828:ba69:1:21c:f0ff:fefa:f3c0      192.168.110.1
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom-dr                    Fri Nov  9 09:56:56 2012                OK
        2001:470:f012:2::3                      172.31.10.4
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom-ext                   Fri Nov  9 09:58:21 2012                OK
        2001:470:f012:2::2                      172.31.10.2
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
list_pending_events:
ServerSMCheckServer       shalom                      Fri Nov  9 10:01:43 2012
ServerSMCheckServer       dns-slave1                  Fri Nov  9 10:01:55 2012
ServerSMCheckServer       dns-slave0                  Fri Nov  9 10:03:17 2012
ServerSMCheckServer       shalom-dr                   Fri Nov  9 10:05:25 2012
ServerSMCheckServer       shalom-ext                  Fri Nov  9 10:04:49 2012
zone_tool >
```

Above, the ls_server section of show_dms_status displays the reason for going to RETRY or BROKEN in the displayed retry_msg field.

The fix, reset_server the server, and use ls_pending events to check ServerSMConfigure is created

```
zone_tool > reset_server en-gedi-auth
zone_tool > ls_pending_events
ServerSMCheckServer       shalom                    Fri Nov  9 12:11:17 2012
ServerSMCheckServer       shalom-ext                Fri Nov  9 12:11:47 2012
ServerSMCheckServer       en-gedi-auth              Fri Nov  9 12:14:57 2012
ServerSMCheckServer       dns-slave0                Fri Nov  9 12:18:02 2012
ServerSMCheckServer       shalom-dr                 Fri Nov  9 12:15:09 2012
ServerSMCheckServer       dns-slave1                Fri Nov  9 12:19:08 2012
ServerSMConfigure         en-gedi-auth              Fri Nov  9 12:10:39 2012
zone_tool >
```

Wait until the scheduled time posted for ServerSMConfigure, and then do a zone_tool show_dms_status to make sure everything is going.

```
zone_tool > show_dms_status
show_master_status:
        MASTER_SERVER:     shalom-ext
        NAMED master configuration state:
        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        None
        hold_stop:         None
        replica_sg_name:   anathoth-replica
        state:             READY
show_replica_sg:
        sg_name:             anathoth-replica
        config_dir:          /etc/bind/anathoth-master
        master_address:      2001:470:f012:2::2
        master_alt_address:  2001:470:f012:2::3
        replica_sg:          True
        zone_count:          14
        Replica SG named status:
        shalom-dr                    2001:470:f012:2::3
                OK
ls_server:
dns-slave0                   Fri Nov  9 12:08:29 2012                OK
        2001:470:c:110e::2                      111.65.238.10
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dns-slave1                   Fri Nov  9 12:10:19 2012                OK
        2001:470:66:23::2                       111.65.238.11
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
en-gedi-auth                 Fri Nov  9 12:10:43 2012                OK
        fd14:828:ba69:6:5054:ff:fe39:54f9       172.31.12.2
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom                       Fri Nov  9 12:11:19 2012                OK
        fd14:828:ba69:1:21c:f0ff:fefa:f3c0      192.168.110.1
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom-dr                    Fri Nov  9 12:09:44 2012                OK
        2001:470:f012:2::3                      172.31.10.4
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom-ext                   Fri Nov  9 12:11:47 2012                OK
        2001:470:f012:2::2                      172.31.10.2
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
list_pending_events:
ServerSMCheckServer       en-gedi-auth                Fri Nov  9 12:14:57 2012
ServerSMCheckServer       dns-slave0                  Fri Nov  9 12:18:02 2012
ServerSMCheckServer       shalom-dr                   Fri Nov  9 12:15:09 2012
ServerSMCheckServer       dns-slave1                  Fri Nov  9 12:19:08 2012
ServerSMCheckServer       shalom                      Fri Nov  9 12:17:44 2012
ServerSMCheckServer       shalom-ext                  Fri Nov  9 12:17:31 2012
zone_tool >
```

Rebuilding named data from database

The named dynamic data in /var/lib/bind/dynamic is corrupt, or missing

1. Stop named and net24dmd

```
root@dms3-master:~# service bind9 stop
[....] Stopping domain name service...: bind9waiting for pid 15462 to die
. ok
root@dms3-master:~# service net24dmd stop
[ ok ] Stopping net24dmd: net24dmd.
```

2. Check /etc/bind/master_config and /var/lib/bind/dynamic permissions.
   /etc/bind/master-config, should be 2755 net24dmd:bind :

```
root@dms3-master:~# ls -ld /etc/bind/master-config/
drwxr-sr-x 2 net24dmd bind 4096 Nov  9 12:39 /etc/bind/master-config/
root@dms3-master:~#
```

   /var/lib/bind/dynamic, should be 2775 bind:net24dmd :

```
root@dms3-master:~# ls -ld /var/lib/bind/dynamic
drwxrwsr-x 2 bind net24dmd 1683456 Nov  9 12:39 /var/lib/bind/dynamic
root@dms3-master:~#
```

3. Clear any files from /var/lib/bind/dynamic if needed:

```
root@dms3-master:~# rm -rf /var/lib/bind/dynamic/*
root@dms3-master:~#
```

4. Run the restore process which recreates /etc/bind/master-config/ contents, and recreates contents of /var/lib/bind/dynamic. This may take some time. 40000 zones takes 20 - 30 minutes.

```
root@dms3-master:~# zone_tool restore_named_db
***   WARNING - doing this destroys DNSSEC RRSIG data. It is a last
      resort in DR recovery.
***   Do really you wish to do this?
 --y/[N]> y
```

5. Start named and net24dmd

```
root@dms3-master:~# service net24dmd start
[ ok ] Starting net24dmd: net24dmd.
root@dms3-master:~# service bind9 start
[ ok ] Starting domain name service...: bind9.
root@dms3-master:~#
```

Failed Master, Replica /etc not up to date

The master and DR replica have the etckeeper git archive mirrored every 4 hours to the alternate server. See etckeeper and /etc on Replica and Master Servers

Recovering DB from Backup

/etc/cron.d/dms-core does daily FULL pg_dumpall to /var/backups/postresql-9.1-dms.sql.gz, on replica and master, which are rotated for 7 days.

To recover:

```
# cd /var/backups
# gunzip -c postregresql-9.1-dms.sql.gz | psql -U pgsql
```

There will be lots of SQL output. The dumpall also contains DB user passwords, and ACL/permissions information, along with DB details for the whole postgresql 'dms' cluster.

Regenerating ds/ DS material directory from Private Keys

Use the dns-recreateds command to recreate a domains DNSSEC DS material. The /var/lib/bind/keys directory is rsynced to the DR replica by the master server net24dmd daemon. Use a '*' argument to regenerate all DS material.

```
shalom-ext: -root- [/var/lib/bind/keys]
# dns-recreateds anathoth.net
+ dnssec-dsfromkey -2 /var/lib/bind/keys/Kanathoth.net.+007+57318.key
+ set +x
shalom-ext: -root- [/var/lib/bind/keys]
#
```

IPSEC not going

These examples are between DNS slave server dns-slave1 and master shalom-ext.

Diagnosis

Ping6 server from master and vice-versa to check unencrypted network level. (Transport mode encryption does not encrypt ICMPv6). Use the zone_tool ls_server -v command to get the DMS configured IPv6 addresses of both servers.

```
shalom-ext: -grantma- [~/dms]
$ zone_tool ls_server -v dns-slave1
dns-slave1 Mon Nov 12 13:57:20 2012 OK
 2001:470:66:23::2 111.65.238.11

shalom-ext: -grantma- [~/dms]
$ zone_tool ls_server -v shalom-ext
shalom-ext                  Mon Nov 12 13:59:29 2012               OK
        2001:470:f012:2::2                      172.31.10.2
shalom-ext: -grantma- [~/dms]
$ ping6 2001:470:66:23::2
PING 2001:470:66:23::2(2001:470:66:23::2) 56 data bytes
64 bytes from 2001:470:66:23::2: icmp_seq=1 ttl=58 time=312 ms
64 bytes from 2001:470:66:23::2: icmp_seq=2 ttl=58 time=310 ms
64 bytes from 2001:470:66:23::2: icmp_seq=3 ttl=58 time=310 ms
^C
--- 2001:470:66:23::2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 310.646/311.293/312.518/0.866 ms
shalom-ext: -grantma- [~/dms]
$
```

Telnet domain TCP ports both ways, and rsync out to slave server from master. This checks that IPSEC encryption is running.

From shalom-ext:

```
shalom-ext: -grantma- [~/dms]
$ telnet 2001:470:66:23::2 53
Trying 2001:470:66:23::2...
Connected to 2001:470:66:23::2.
Escape character is '^]'.
^]c
telnet> c
Connection closed.
shalom-ext: -grantma- [~/dms]
$ telnet 2001:470:66:23::2 rsync
Trying 2001:470:66:23::2...
Connected to 2001:470:66:23::2.
Escape character is '^]'.
@RSYNCD: 30.0
^]c
telnet> c
Connection closed.
shalom-ext: -grantma- [~/dms]
$
```

From dns-slave1:

```
grantma@dns-slave1:~$ telnet 2001:470:f012:2::2 53
Trying 2001:470:f012:2::2...
Connected to 2001:470:f012:2::2.
Escape character is '^]'.
^]c
telnet> c
Connection closed.
grantma@dns-slave1:~$
```

If the DNS server is a DR replica, telnet the rsync port the other way also.

Recovery

If things are not working restart the IPSEC connection at both ends:

shalom-ext master:

```
shalom-ext: -root- [/home/grantma/dms]
# racoon-tool vlist
shalom-dr
dns-slave1
%anonymous
shalom-ext
shalom
dns-slave0
en-gedi-auth
shalom-ext: -root- [/home/grantma/dms]
# racoon-tool vreload dns-slave1
Reloading VPN dns-slave1...The result of line 2: No entry.
The result of line 5: No entry.
done.
shalom-ext: -root- [/home/grantma/dms]
#
```

dns-slave1:

```
root@dns-slave1:/home/grantma# racoon-tool vlist
shalom-dr
%anonymous
shalom-ext
root@dns-slave1:/home/grantma# racoon-tool vreload shalom-ext
Reloading VPN shalom-ext...The result of line 2: No entry.
The result of line 5: No entry.
done.
root@dns-slave1:/home/grantma#
```

Wait 10 minutes for IPSEC replay timing to expire. Then retry the telnet steps above.

If IPSEC still will not work:

Use racoon-tool restart on both ends:

shalom-ext:

```
shalom-ext: -root- [/home/grantma/dms]
# racoon-tool restart
Stopping IKE (ISAKMP/Oakley) server: racoon.
Flushing SAD and SPD...
SAD and SPD flushed.
Unloading IPSEC/crypto modules...
IPSEC/crypto modules unloaded.
Loading IPSEC/crypto modules...
IPSEC/crypto modules loaded.
Flushing SAD and SPD...
SAD and SPD flushed.
Loading SAD and SPD...
SAD and SPD loaded.
Configuring racoon...done.
Starting IKE (ISAKMP/Oakley) server: racoon.
shalom-ext: -root- [/home/grantma/dms]
#
```

dns-slave1:

```
root@dns-slave1:/home/grantma# racoon-tool restart
Stopping IKE (ISAKMP/Oakley) server: racoon.
Flushing SAD and SPD...
SAD and SPD flushed.
Unloading IPSEC/crypto modules...
IPSEC/crypto modules unloaded.
Loading IPSEC/crypto modules...
IPSEC/crypto modules loaded.
Flushing SAD and SPD...
SAD and SPD flushed.
Loading SAD and SPD...
SAD and SPD loaded.
Configuring racoon...done.
Starting IKE (ISAKMP/Oakley) server: racoon.
root@dns-slave1:/home/grantma#
```

DMS Master Server Install

Base Operating System

Debian Wheezy

Create /etc/apt/apt.conf.d/00local.conf:

```
// No point in installing a lot of fat on VM servers
APT::Install-Recommends "0";
APT::Install-Suggests "0";
```

Create /etc/apt/sources.list.d/00local.conf

```
deb http://deb-repo.devel.net.nz/debian/ wheezy main
deb-src http://deb-repo.devel.net.nz/debian/ wheezy main
```

Install these packages:

cron-apt screen tree procps psmisc sysstat sudo lsof open-vm-tools open-vm-dkms

dms

To properly install netscript-2.4 because of cyclic boot dependencies (I will look into this when have some spare time, and log an RC Debian bug):

```
# dpkg --force --purge ifupdown
# apt-get -f install
```

Edit /etc/netscript/network.conf to configure static addressing. Look for IF_AUTO, set eth0_IPADDR, and further down comment out eth_start and eth_stop functions to turn off DHCP. Netscript manages iptables and ip6tables via iptables-save/iptables-restore, and keeps a cyclic history which you can change back to if your filter changes go wrong vi netscript ipfilter/ip6filter save/usebackup.

Then:

```
# aptitude update
# aptitude upgrade
```

To fix shell prompt for larger terminals on master server makes typing in long zone_tool commands at shell a lot clearer:

```
# tar -C / -xzf shell.tar.gz
```

shell.tar.gz

Replaces /etc/skel shell and /root dot files with single line feed to force use of file in /etc

Then edit /etc/environment.sh to turn off various things like umask 00002 for user id < 1000

Then follow Net24 DMS Debian Install Documentation

Archive and Work in Progress

> This section contains documents which are no longer current and are not to be used, but are made available for reference purposes only. For example build guides which have been superseded by automated deployment and management systems.

Frequent Procedures (Howtos for help desk and basic administration)

Various frequent procedures are listed here. They are typical of the daya to day management of zones with DMS.

Creating and copying Zones

Zones are created using the create_zone command:

```
zone_tool > create_zone test1.com
zone_tool > show_zone test1.com
$TTL 1h
$ORIGIN test1.com.

;
; Zone:       test1.com.
; Reference:  anathoth
; change_by:  grantma@shalom-ext.internal.anathoth.net/Admin
; zi_id:      103187
; zi_ctime:   Wed Oct 17 13:19:15 2012
; zi_mtime:   Wed Oct 17 13:19:15 2012
;


;| Apex resource records for test1.com.
;!REF:anathoth
@                       IN      SOA             ( ns1.anathoth.net. ;Master
NS
                                                matthewgrant5.gmail.com.
;RP email
                                                2012101700   ;Serial
yyyymmddnn
                                                86400        ;Refresh
                                                900          ;Retry
                                                604800       ;Expire
                                                3600
;Minimum/Ncache
                                                )
                        IN      NS              ns3.anathoth.net.
                        IN      NS              ns2.anathoth.net.
                        IN      NS              ns1.anathoth.net.


zone_tool > create_zone test1.com
***   Zone 'test1.com.' already exists.
zone_tool >
```

When a zone is just created, only the Apex records are filled in thus achieving the result of just technically parking the domain if it is then registered in ENOM or the NZRS.

They can also be created from any given ZI by using the copy_zone command:

```
zone_tool > help copy_zone
        Copy a zone:

        copy_zone [-g <ssg-name>] [-i] [ -r reference] [-z zi_id]
                          <src-domain-name> <domain-name> [zone-option] ...
        where   -g <ssg-name>: specify an SSG name other than default_ssg
                -i:            set inc_updates flag on the new zone
                -r reference:  set reference
                -z zi_id:      set zi_id used for copy source
                zone-option:   use_apex_ns|auto_dnssec|edit_lock|nsec3
                                |inc_updates
                                        up to 5 times

zone_tool > copy_zone test1.com bad-thing.org
zone_tool > show_zone bad-thing.org
$TTL 24h
$ORIGIN bad-thing.org.

;
; Zone:       bad-thing.org.
; Reference:  anathoth
; change_by:  grantma@shalom-ext.internal.anathoth.net/Admin
; zi_id:      102602
; zi_ctime:   Thu Aug 23 14:54:07 2012
; zi_mtime:   Thu Aug 23 14:54:07 2012
;


;| Apex resource records for bad-thing.org.
;!REF:anathoth
@                       IN      SOA             ( ns1.anathoth.net. ;Master
NS
                                                matthewgrant5.gmail.com.
;RP email
                                                2012082300   ;Serial
yyyymmddnn
                                                600          ;Refresh
                                                600          ;Retry
                                                604800       ;Expire
                                                600
;Minimum/Ncache
                                                )
                        IN      NS              ns3.anathoth.net.
                        IN      NS              ns2.anathoth.net.
                        IN      NS              ns1.anathoth.net.
```

ZIs can also be copied from one zone to another by using the copy_zi command. This command will not result in

the copied ZI being published unless the zone is refreshed to use it.

Deleting and Undeleting Zones

Deleting a Zone

The command for deleting a zone is delete_zone.

```
zone_tool > ls bad-thing.org
bad-thing.org.
zone_tool > delete_zone bad-thing.org.
***   Zone 'bad-thing.net.' not present.
zone_tool > delete_zone bad-thing.org.
zone_tool > ls bad-thing.org.
***   Zones: bad-thing.org. - not present.
```

Undeleting a Zone

The ls_deleted command can be used in conjunction with the undelete_zone command. The undelete_zone command only takes a zone_id argument, as there are likely to be multiple deleted zones with the same name. The show_zone_byid command can be used to display the deleted zone.

```
zone_tool > ls_deleted bad-thing.*
bad-thing.org.                  101449       anathoth
zone_tool > show_zone_byid 101449
$TTL 24h
$ORIGIN bad-thing.org.

;
; Zone:       bad-thing.org.
; Reference:  anathoth
; change_by:  grantma@shalom-ext.internal.anathoth.net/Admin
; zi_id:      102602
; zi_ctime:   Thu Aug 23 14:54:07 2012
; zi_mtime:   Wed Aug 29 17:10:15 2012
; zi_ptime:   Wed Aug 29 17:10:15 2012
;


;| Apex resource records for bad-thing.org.
;!REF:anathoth
@                       IN      SOA             ( ns1.anathoth.net. ;Master
NS
                                                matthewgrant5.gmail.com.
;RP email
                                                2012082300   ;Serial
yyyymmddnn
                                                600          ;Refresh
                                                600          ;Retry
                                                604800       ;Expire
                                                600
;Minimum/Ncache
                                                )
                        IN      NS              ns3.anathoth.net.
                        IN      NS              ns2.anathoth.net.
                        IN      NS              ns1.anathoth.net.


zone_tool > undelete_zone 101449
zone_tool > ls bad-thing.*
bad-thing.org.
zone_tool >
```

Deleted zones will have their ZIs pared down to what was the published ZI after 90 days by the vacuum_all command, which is croned to run daily.

Editing a Zone

Use the edit_zone <domain-name> [zone-instance] command. If you are using the default vim-nox editor, it will drop you into a syntax highlighted editing session.

In /usr/share/vim/vimcurrent/debian.vim vim has been set up for:

```
set nocompatible " Use Vim defaults instead of 100% vi compatibility
set backspace=indent,eol,start " more powerful backspacing
```

which means Insert mode behaves like a normal editor. Arrow keys do not finish insert mode session. Backspace and delete delete across line ends with a logical sense as to directionality when in insert mode etc. (Whew! standard vi - !@#$%@$%&$%^*@#$%^ - can't find **spanner** to resolve **insertion** into **works** trajectory)

At a minumum you still have to know about :w to save, and :q to quit and save. 'ESC' is also useful to cancel something if you think you have pressed something wrong, and to exit insert mode back to visual command mode. Pressing 'u' in visual mode will undo the last change, with multiple undo for recent change history.

Vim keys	Action
ESC	**cancel current thing, exit Insert mode. Dive for this key if you want to back out of what ever you are not sure you have just started (in visual mode). Press multiple times just to reassure yourself operation is cancelled, even though once is all you need to do 95% of the time. This should 'unstick' any vi. REMEMBER THIS! (vi safety rule number 1!)**
i	Go to insert mode from visual
:w	In visual mode, save file
:e!	revert all changes until last save
:q	quit
:q!	forced quit if you have changed something
:wq	save file and quit vi
/<regexp>	search forwards
?<regexp>	search backwards
n	search again in search direction
N	search again in reverse search direction
dd	delete current line
gg	go to start of file
G	go to end of file

d$	delete from cursor to end of line
V	select current line and then use arrows to select block
v	select from cursor posN and then use arrows to select block
d	delete, then press locational key of where to delete to (^,$,G,gg)
^	beginning of line
$	end of line
:s/<regexp>/<replacement>/gc	search and replace with confirmation. Use with v or V selection to apply to block. g suffix means replace multiple times on one line, rather than first occurrence, c means confirm
:%	Apply following command across whole file. ':%s/<regexp>/<replacement>/' very useful
p	paste last deletion
y	copy 'yy' copy current line, y$ y^ as you would expect.
2yy	copy current line and one following
2dd	well, work this one out...
2p	paste twice (paste one line 2 times etc)

```
zone_tool > edit_zone 192.168.110/24

$TTL 24h
$ORIGIN 110.168.192.in-addr.arpa.

;
; Zone:       110.168.192.in-addr.arpa.
; Reference:  anathoth
; change_by:  grantma@shalom-ext.internal.anathoth.net/Admin
; zi_id:      102584
; zi_ctime:   Sun Aug 19 20:10:16 2012
; zi_mtime:   Sun Aug 19 20:10:16 2012
; zi_ptime:   Sun Aug 19 20:10:16 2012
;


;| Apex resource records for 110.168.192.in-addr.arpa.
;!REF:anathoth
@                       IN      SOA             (
ns1.internal.anathoth.net. ;Master NS
                                                matthewgrant5.gmail.com.
;RP email
```

```
                                                2012081900   ;Serial
yyyymmddnn
                                                600          ;Refresh
                                                600          ;Retry
                                                604800       ;Expire
                                                600
;Minimum/Ncache
                                                )
                        IN      NS              ns2.internal.anathoth.net.
                        IN      NS              ns1.internal.anathoth.net.


;!LOCKPTR
1                       IN      PTR
shalom.internal.anathoth.net.
;!REF:anathoth
149                     IN      PTR
something-here.failover.internal.anathoth.net.
;!REF:anathoth
16                      IN      PTR
openwrt.internal.anathoth.net.
;!LOCKPTR REF:anathoth
2                       IN      PTR
shalom-auth.internal.anathoth.net.
;!LOCKPTR REF:anathoth
20                      IN      PTR
phone-800.internal.anathoth.net.
230                     IN      PTR             ballywack.anathoth.net.
;!LOCKPTR REF:anathoth
254                     IN      PTR
shalom-fw.internal.anathoth.net.
;!REF:anathoth
3                       IN      PTR
sid-dev.internal.anathoth.net.
;!REF:anathoth
4                       IN      PTR             joy.internal.anathoth.net.
;!REF:anathoth
5                       IN      PTR
sid-test.internal.anathoth.net.
;!REF:anathoth
69                      IN      PTR
phone-802.internal.anathoth.net.
;!REF:anathoth
96                      IN      PTR
openwrt.internal.anathoth.net.

***   Do you wish to Abort, Change, Diff, or Update the zone
'110.168.192.in-addr.arpa.'?
--[U]/a/c/d> d
@@ -47,7 +47,7 @@
 5                       IN      PTR
sid-test.internal.anathoth.net.
 ;!REF:anathoth
```

```
 69                      IN      PTR
phone-802.internal.anathoth.net.
-;!LOCKPTR REF:anathoth
+;!REF:anathoth
 96                      IN      PTR
openwrt.internal.anathoth.net.

--[U]/a/c/d>
```

```
zone_tool >
```

DMS Zone File Format

The DMS zone file format builds on the format described in RFCs 1034 and 1035 by the use of 2 character comment tags. In the example above note the Apex RR group started by the ;| RR croup comment, with the block finished by a blank line. Individual RR record comments start with ;# on the line just before the record. Bioth types of comment can be multi line. An new RR Group can be started by giving a comment starting with ';|', with the RR Group comment naming the RR Group. RR Groups tend to be sorted alphabetically, except that the Apex group containing the SOA and NS records is at the top of the zone file, with the unlabelled default RR Group last of all. RR flag comments also exist, mostly to control auto reverse PTR functionality, and to disable any individual RR.

DMS comment	Description
;|	RR Group comment
;#	Individual RR comment
;!	RR flag comment
;!LOCKPTR	Lock the PTR record preventing any auto update.
;!REF:0000@1STDOMAINS-NZ	PTR RR reference. Any changes coming from a zone 'owned' by the given reference are allowed to change the record. The ;!REF on the SOA declares the ownership of the zone.
;!FORCEREV	One shot force reverse update of PTR from A or AAAA record unless it is locked.
;!TRACKREV	Track reverse update of PTR from A or AAAA unless it is locked.
;!DISABLE	Disable the RR and remove it from published zone.
;!RROP: ADD, DELETE, UPDATE_RRTYPE	zone_tool update_rrs incremental update operation. See zone_tool help update_rrs for all the details. 'Wildcard' arguments can be given to DELETE operation.

Note that multiple ';!' RR flags are all given on one line before the RR.

Auto-reverse PTR record management

The DMS system can do this, and it checks every A and AAAA record on ZI submission to do auto reverse if it is configured for the reverse zones the DMS system holds. The reference of the source zone has to match the reference of the reverse zone, or the reference on a PTR record to effect a change or the source of the update has to be a user interface with the 'Admin' sectag. Given the former conditions, if a PTR record does not exist, one is created. An existing PTR record is only updated if the FORCEREV RR flag is given, and the RR is not LOCKPTR. BTW, the inc_updates flag MUST be set on a reverse zone for auto updating to operate on it.

The update mechanism uses a network database table to choose the most specific (by CIDR netmask) existing

reverse zone to apply the update to. This is also the smarts behind the CIDR network block/IP address -> reverse zone domain resolution in zone_tool.

Edit Locking

Zones may have edit_lock flag set, which means timed edit locking is enforced on the zone. The lock has an activity time out, and edit_zone will give a lock failure with the locked_by string for the zone if it is locked. The lock can be cleared with cancel_edit_zone or clear_edit_lock, which will ask for the zone name and the lock token that is returned with the lock failure error message.

```
shalom-ext: -grantma- [~]
$ zone_tool edit_zone anathoth.net
***   Event ZoneSMEdit(885379) failed - ZoneEditLocked: Zone
      'anathoth.net.' is locked with token '885378', held by 'grantma
      @shalom-ext.internal.anathoth.net/Admin'.

shalom-ext: -grantma- [~]
$ zone_tool
Welcome to the zone_tool program.  Type help or ? to list commands.

zone_tool > clear_edit_lock anathoth.net 885378
zone_tool > edit_zone anathoth.net
***   File '/tmp/zone_tool-mtjo4s.zone'
    unchanged after editing - exiting.
zone_tool >
```

Enabling and Disabling Zones

Enabling and Disabling a Zone

This completely removes the zone from the DNS servers, while still holding it in the database. The show_zonesm <domain-name> command is used to display the zone state, though you could also use ls -v <domain-name> The zone_tool commands are enable_zone and disable_zone.

ls_pending_events can be used to display what is waiting in the DMS event queue. Note the 10 minute delay between updating the named.conf files enforced by the DMS ConfiSM state machine.

For example have a look at the following screen capture:

```
zone_tool > disable_zone bad-thing.org
zone_tool > show_zonesm bad-thing.org
        name:            bad-thing.org.
        alt_sg_name:     None
        auto_dnssec:     False
```

```
        ctime:           Thu Aug 23 14:54:07 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_by:       None
        mtime:           Thu Aug 23 15:07:07 2012
        nsec3:           True
        reference:       anathoth
        soa_serial:      2012082300
        sg_name:         anathoth-external
        state:           DISABLED
        use_apex_ns:     True
        zi_candidate_id: 102602
        zi_id:           102602
        zone_id:         101449
        zone_type:       DynDNSZoneSM

        zi_id:           102602
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Thu Aug 23 14:54:07 2012
        mtime:           Thu Aug 23 14:54:26 2012
        ptime:           Thu Aug 23 14:54:26 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.anathoth.net.
        soa_refresh:     600
        soa_retry:       600
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012082300
        soa_ttl:         None
        zone_id:         101449
        zone_ttl:        24h
zone_tool >

shalom-ext: -grantma- [~/dms-2011]
$ dig -t AXFR bad-thing.org @::1

; <<>> DiG 9.8.1-P1 <<>> -t AXFR bad-thing.org @::1
;; global options: +cmd
; Transfer failed.

zone_tool > enable_zone bad-thing.org
zone_tool > show_zonesm bad-thing.org
        name:            bad-thing.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 14:54:07 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
```

```
        lock_state:      EDIT_UNLOCK
        locked_by:       None
        mtime:           Thu Aug 23 15:08:58 2012
        nsec3:           True
        reference:       anathoth
        soa_serial:      2012082300
        sg_name:         anathoth-external
        state:           UNCONFIG
        use_apex_ns:     True
        zi_candidate_id: 102602
        zi_id:           102602
        zone_id:         101449
        zone_type:       DynDNSZoneSM

        zi_id:           102602
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Thu Aug 23 14:54:07 2012
        mtime:           Thu Aug 23 14:54:26 2012
        ptime:           Thu Aug 23 14:54:26 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.anathoth.net.
        soa_refresh:     600
        soa_retry:       600
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012082300
        soa_ttl:         None
        zone_id:         101449
        zone_ttl:        24h
.
.
.

zone_tool > ls_pending_events
ConfigSMHoldTimeout                          Thu Aug 23 15:17:09
2012
ZoneSMReconfigUpdate     bad-thing.org.      Thu Aug 23 15:17:27
2012
zone_tool > ls -v bad-thing.org
bad-thing.org.                   2012082300   UNCONFIG       anathoth
zone_tool >
.
.
.
shalom-ext: -grantma- [~/dms-2011]
$ dig -t AXFR bad-thing.org @::1
; <<>> DiG 9.8.1-P1 <<>> -t AXFR bad-thing.org @::1
;; global options: +cmd
bad-thing.org.  86400 IN SOA ns1.anathoth.net. matthewgrant5.gmail.com.
2012082300 600 600 604800 600
bad-thing.org.  86400 IN NS ns1.anathoth.net.
bad-thing.org.  86400 IN NS ns2.anathoth.net.
bad-thing.org.  86400 IN NS ns3.anathoth.net.
```

```
bad-thing.org.  86400 IN SOA ns1.anathoth.net. matthewgrant5.gmail.com.
2012082300 600 600 604800 600
;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Thu Aug 23 15:18:56 2012
;; XFR size: 5 records (messages 1, bytes 192)
```

```
zone_tool > ls -v bad-thing.org
bad-thing.org.                  2012082300   PUBLISHED      anathoth
zone_tool >
```

Refreshing and Resetting a Zone

Refreshing a Zone

This causes a refresh of the zone against the master DMS server. If there are any diifferences, they are resolved.

```
zone_tool > refresh_zone bad-thing.org
zone_tool > ls_zi bad-thing.org
        *102602               2012082300    Thu Aug 23 14:54:07 2012
zone_tool > show_zonesm bad-thing.org
        name:            bad-thing.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 14:54:07 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_by:       None
        mtime:           Thu Aug 30 09:11:45 2012
        nsec3:           True
        reference:       anathoth
        soa_serial:      2012082300
        sg_name:         anathoth-external
        state:           PUBLISHED
        use_apex_ns:     True
        zi_candidate_id: 102602
        zi_id:           102602
        zone_id:         101449
        zone_type:       DynDNSZoneSM

        zi_id:           102602
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Thu Aug 23 14:54:07 2012
        mtime:           Thu Aug 30 09:25:44 2012
        ptime:           Thu Aug 30 09:25:44 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.anathoth.net.
        soa_refresh:     600
        soa_retry:       600
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012082300
        soa_ttl:         None
        zone_id:         101449
        zone_ttl:        24h
zone_tool >
```

Resetting a Zone

This withdraws the zone completely from the DNS servers, and reconfigures it through out the DNS servers. During the 15 minutes that this takes, the Zone will NOT be served. The main use of this instruction is if a zones state machine is 'stuck' and not PUBLISHED. A yes/no confirmation is asked before doing it. **BE CAREFUL**!

```
zone_tool > reset_zonesm bad-thing.org
***   WARNING - doing this destroys DNSSEC RRSIG data.
***   Do really you wish to do this?
 --y/[N]> y
zone_tool > show_zonesm bad-thing.org
        name:            bad-thing.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 14:54:07 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_by:       None
        mtime:           Thu Aug 30 09:11:45 2012
        nsec3:           True
        reference:       anathoth
        soa_serial:      2012082300
        sg_name:         anathoth-external
        state:           RESET
        use_apex_ns:     True
        zi_candidate_id: 102602
        zi_id:           102602
        zone_id:         101449
        zone_type:       DynDNSZoneSM

        zi_id:           102602
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Thu Aug 23 14:54:07 2012
        mtime:           Thu Aug 30 09:25:44 2012
        ptime:           Thu Aug 30 09:25:44 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.anathoth.net.
        soa_refresh:     600
        soa_retry:       600
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012082300
        soa_ttl:         None
        zone_id:         101449
        zone_ttl:        24h
zone_tool >
```

The DMS zone_tool Session etc

For help desk, ssh to dms-server.failover.vygr.net with your DMS system login name. For help desk accounts, you will be dropped into a restricted zone_tool shell, which should have all the commands you need to do day to day

zone management.

The default editor in the shell is vim with zone file syntax highlighting. Invalid syntax will usually be will be highlighted in red as soon as you type it. Vim is set up to allow normal cursor navigation with arrow keys in a friendly 'Insert' mode, and other niceties, as detailed in Editing a Zone. Nano is available on request, but it wont be so helpful when editing.

To exit the shell, use Ctrl-D, exit or quit as you would with a normal *nix terminal session.

> Operations that cause an amount of down time, or may result in irreversible or really large changes in zone_tool have a confirmation question before proceeding. Be careful.

Viewing Zones (and a lot more about them)

Listing Zones

You can use the ls command for this. It can take multiple wild cards, '?' and '*'. Other things that are useful are the customer reference. These take the form account_id@1STDOMAINS-NZ and account_id@NET24-NZ

Examples:

Plain ls - Returns everything

```
zone_tool > ls
110.168.192.in-addr.arpa.
2.1.0.f.0.7.4.0.1.0.0.2.ip6.arpa.
31.172.in-addr.arpa.
9.6.a.b.8.2.8.0.4.1.d.f.ip6.arpa.
anathoth.net.
anathoth.org.
blam.com.
blamo.net.
failover.internal.anathoth.net.
internal.anathoth.net.
loo.org.
test1.com.
test2.com.
wilma.org.
```

Wildcard ls

```
zone_tool > ls anathoth*
anathoth.net.
anathoth.org.
```

ls with reference using -r switch

```
zone_tool > ls -r 0000@1STDOMAINS-NZ
110.168.192.in-addr.arpa.
2.1.0.f.0.7.4.0.1.0.0.2.ip6.arpa.
31.172.in-addr.arpa.
9.6.a.b.8.2.8.0.4.1.d.f.ip6.arpa.
blam.com.
blamo.net.
failover.internal.anathoth.net.
internal.anathoth.net.
loo.org.
test1.com.
test2.com.
wilma.org.
```

Verbose ls with reference

```
zone_tool > ls -v -r 0000@1STDOMAINS-NZ
110.168.192.in-addr.arpa.        2012081900   PUBLISHED      anathoth
2.1.0.f.0.7.4.0.1.0.0.2.ip6.arpa. 2012052300   PUBLISHED      anathoth
31.172.in-addr.arpa.             2012071301   PUBLISHED      anathoth
9.6.a.b.8.2.8.0.4.1.d.f.ip6.arpa. 2012081900   PUBLISHED      anathoth
blam.com.                        2012081600   PUBLISHED      anathoth
blamo.net.                       2012080902   PUBLISHED      anathoth
failover.internal.anathoth.net.  2012081601   PUBLISHED      anathoth
internal.anathoth.net.           2012081900   PUBLISHED      anathoth
loo.org.                         2012081602   PUBLISHED      anathoth
test1.com.                       2012081601   PUBLISHED      anathoth
test2.com.                       2012081602   PUBLISHED      anathoth
wilma.org.                       2012081602   PUBLISHED      anathoth
zone_tool >
```

Listing Deleted Zones

Use the ls_deleted command. It can use wild cards and reference as per the ls command. The second column displayed is the zone_id, which you use to undelete a zone. Raison d'etre: With 1st domains, knowing how people

use computers when they 'know'/think something goes a bit loopy, they will spring for deleting a zone, and recreating it, most likely multiple times. Thus there are likely to be multiple deleted zones for the same domain name, hence the use of zone_id for undelete.

```
zone_tool > help ls_deleted

        List deleted zones/domains (+ wildcards):

        ls_deleted [-v] [-r reference] [-g sg_name] [domain-name]
                        [domain-name] ...

        where:  domain-name     domain name with * or ? wildcards as needed
                reference       reference
                sg_name         server group name
                -v              verbose output

zone_tool > ls_deleted
blamo.wham.                      101374       anathoth
blamo.wham.                      101375       anathoth
toady.anathoth.net.              101407       anathoth
zone_tool >
```

Showing a Zone

Use the show_zone command. By default just displays the published Zone Instance (ZI)

```
zone_tool > show_zone anathoth.net
$TTL 24h
$ORIGIN anathoth.net.

;
; Zone:       anathoth.net.
; change_by:  hd-test@shalom-ext.internal.anathoth.net/Admin
; zi_id:      102592
; zi_ctime:   Mon Aug 20 11:07:49 2012
; zi_mtime:   Mon Aug 20 11:12:07 2012
; zi_ptime:   Mon Aug 20 11:12:07 2012
;


;|
;| Apex resource records for anathoth.net.
;|
@                       IN      SOA             ( ns1          ;Master NS
                                                matthewgrant5.gmail.com.
;RP email
                                                2012082000   ;Serial
yyyymmddnn
                                                600          ;Refresh
```

```
                                                600          ;Retry
                                                604800       ;Expire
                                                600
;Minimum/Ncache
                                                )
                        IN      NS              ns3
                        IN      NS              ns2
                        IN      NS              ns1


;| Hosts
shalom-dr               IN      AAAA            2001:470:f012:2::3
                        IN      SSHFP           1 1
07bfdd14b4be97dbe282573eecd5bc6b062a92b1
shalom-ext              IN      AAAA            2001:470:f012:2::2
                        IN      SSHFP           1 1
073b3198599c59a3c2a9db8c209a2097ea46aa09
shalom-fw               IN      AAAA            2001:470:c:2e6::2
shalom-svc              IN      AAAA            2001:470:f012:2::1


;| Internal zone lacing
internal                IN      DS              18174 7 2
c42492db9def5ca9403d26f175247dfe86d913da4bedfc7d629f5e57d6669feb
                        IN      NS              ns1.internal
                        IN      NS              ns2.internal
ns1.internal            IN      AAAA
fd14:828:ba69:1:21c:f0ff:fefa:f3c0
ns2.internal            IN      AAAA            fd14:828:ba69:2::2


;| Name server records
ns1                     IN      A               203.79.116.183
                        IN      AAAA            2001:470:f012:2::2
ns2                     IN      A               111.65.238.10
                        IN      AAAA            2001:470:c:110e::2
ns3                     IN      A               111.65.238.11
                        IN      AAAA            2001:470:66:23::2


;| Web site Urls
@                       IN      A               203.79.116.183
                        IN      AAAA            2001:470:f012:2::2
                        IN      TXT             "Some hash"
www                     IN      CNAME           @
```

```
zone_tool >
```

Use ls_zi <domain-name> to display all the ZIs in the DB for a zone.

```
ls_zi anathoth.net
         102012                2012042702    Mon Feb 27 10:06:28 2012
         102100                2012050800    Tue May  8 14:19:17 2012
         102104                2012050801    Tue May  8 14:22:25 2012
         102106                2012050802    Tue May  8 14:29:02 2012
         102108                2012050803    Tue May  8 14:34:17 2012
         102133                2012050900    Wed May  9 09:23:04 2012
         102136                2012050901    Wed May  9 09:24:14 2012
         102152                2012050902    Wed May  9 12:55:11 2012
         102155                2012050903    Wed May  9 12:56:27 2012
         102156                2012050904    Wed May  9 12:56:46 2012
         102159                2012051000    Thu May 10 10:07:52 2012
         102162                2012051012    Thu May 10 10:09:04 2012
         102164                2012051013    Thu May 10 13:31:06 2012
         102167                2012051013    Thu May 10 16:13:56 2012
         102171                2012051014    Thu May 10 16:45:33 2012
         102187                2012052100    Mon May 21 11:43:57 2012
         102189                2012052300    Wed May 23 11:47:01 2012
         102199                2012052400    Thu May 24 15:23:05 2012
         102201                2012052401    Thu May 24 15:24:18 2012
         102261                2012072500    Tue Jul  3 12:05:29 2012
         102468                2012072600    Thu Jul 26 12:13:53 2012
         102585                2012082000    Mon Aug 20 10:26:27 2012
         102588                2012082000    Mon Aug 20 10:27:36 2012
         102589                2012082000    Mon Aug 20 10:41:26 2012
        *102592                2012082000    Mon Aug 20 11:07:49 2012
```

The published ZI is asterisked.

show_zone can also take a ZI as the second argument

```
zone_tool > show_zone anathoth.net 102585
$TTL 24h
$ORIGIN anathoth.net.

;
; Zone:       anathoth.net.
; change_by:  grantma@shalom-ext.internal.anathoth.net/Admin
; zi_id:      102585
; zi_ctime:   Mon Aug 20 10:26:27 2012
; zi_mtime:   Mon Aug 20 10:26:28 2012
; zi_ptime:   Mon Aug 20 10:26:28 2012
;


;|
;| Apex resource records for anathoth.net.
;|
@                       IN      SOA             ( ns1           ;Master NS
                                                matthewgrant5.gmail.com.
;RP email
                                                2012082000   ;Serial
yyyymmddnn
                                                600          ;Refresh
                                                600          ;Retry
                                                604800       ;Expire
                                                600
;Minimum/Ncache
                                                )
                        IN      NS              ns3
                        IN      NS              ns2
                        IN      NS              ns1
.
.
.
```

Power Tricks

zi-id

Anywhere a ZI id can be entered, you can use the '^---' and '^++' notation. '^' is the published ZI, '^-' the ZI previous to the published ZI, '^+2' the ZI 2 ahead of the current published ZI, '@2d' the ZI that was published 2 days ago, '1/4' the ZI that was published on the 1st of April, 2/3/1010 the ZI published as of the 2nd March 1010. The zi_id is also used with the diff_zone and diff_zones commands.

domain-name

In the case of reverse zones, the domain name can be the exact network block in CIDR notation when creating a zone, deleting a zone, enabling/disabling/setting a zone. An IP number can be given with show_zone, edit_zone, and lszi, and the corresponding closest reverse zone will be shown/edited. This is for ease of use when working with IP addresses and network diagnosis. The IP number can be pasted into the terminal.

Differencing ZIs and Zones

Differences between the ZIs in a zone can be taken by using the diff_zone_zi command. The first zi_id parameter is the former ZI, and the 2nd the latter ZI. By default the 2nd ZI is the currently published ZI.

All diff output is in unified format, and if the system is set up properly, difference lines are colorized in the zone_tool pager.

Dates can also take a 4 digit year, ISO date format, whith hh:mm after a comma. (ie 3/5/2012,13:45) If a time is not given with a date, it is taken as being at midnight on the date, the start of the day, 00:00. This is in line with the international date time standards used for time zones.

Times in hh:mm can also be used as a zi_id.

> Zone SOA serial numbers for a ZI 'float'. They are updated if a ZI for a zone is republished, of if an update is made to the zone apex records, of if the ZI for the zone is refreshed resulting in it publication. The SOA serial for a ZI is worked out via an RFC compliant 'bargaining' process with named when named is updated with the ZI via dynamic differencing from net24dmd. A current serial number of 'YYYYMMDDnn' format is the first 'offer' if the named zone SOA serial is before the current day.
>
> The best thing when looking for a SOA serial number for a zone is to give it as a zi_id date.

Differencing between ZI at 1/5 (1st May) of current year and published for zone anathoth.net.

```
zone_tool > diff_zone_zi anathoth.net 1/5
@@ -6,7 +6,7 @@
 ;|
 @                       IN      SOA             ( ns1           ;Master NS
                                                 matthewgrant5.gmail.com.
;RP email
-                                                2012042702   ;Serial
yyyymmddnn
+                                                2012082000   ;Serial
yyyymmddnn
                                                 600          ;Refresh
                                                 600          ;Retry
                                                 604800       ;Expire
@@ -18,7 +18,10 @@


 ;| Hosts
+shalom-dr               IN      AAAA            2001:470:f012:2::3
+                        IN      SSHFP           1 1
07bfdd14b4be97dbe282573eecd5bc6b062a92b1
 shalom-ext             IN      AAAA            2001:470:f012:2::2
+                        IN      SSHFP           1 1
073b3198599c59a3c2a9db8c209a2097ea46aa09
 shalom-fw              IN      AAAA            2001:470:c:2e6::2
 shalom-svc             IN      AAAA            2001:470:f012:2::1

@@ -43,6 +46,7 @@
 ;| Web site Urls
 @                       IN      A               203.79.116.183
                         IN      AAAA            2001:470:f012:2::2
+                        IN      TXT             "Some hash"
 www                     IN      CNAME           @


zone_tool >
```

Differencing between ZI 65 days ago and published for zone anathoth.net Note the 2 days ago, no difference, produces no output. Other time specifiers are s for seconds, m for minutes, h for hours. Months is not available as Python standard lib datetime.timedelta class does not support it (months varying in length?).

```
zone_tool > diff_zone_zi anathoth.net @2d ^
zone_tool > diff_zone_zi anathoth.net @25d ^
@@ -6,7 +6,7 @@
 ;|
 @                       IN      SOA             ( ns1           ;Master NS
                                                 matthewgrant5.gmail.com.
;RP email
-                                                2012072600   ;Serial
yyyymmddnn
+                                                2012082000   ;Serial
yyyymmddnn
                                                 600          ;Refresh
                                                 600          ;Retry
                                                 604800       ;Expire
zone_tool >
```

Differencing between anathoth.net on 2/4/2012,14:04 and the ZI 4 previous to the current published one (could also be given as '^----'):

```
diff_zone_zi anathoth.net 3/4/2012,14:04 ^-4
@@ -6,7 +6,7 @@
 ;|
 @                        IN      SOA             ( ns1           ;Master NS
                                                  matthewgrant5.gmail.com.
;RP email
-                                                 2012042702   ;Serial
yyyymmddnn
+                                                 2012072600   ;Serial
yyyymmddnn
                                                  600          ;Refresh
                                                  600          ;Retry
                                                  604800       ;Expire
@@ -18,7 +18,10 @@


 ;| Hosts
+shalom-dr               IN      AAAA            2001:470:f012:2::3
+                        IN      SSHFP           1 1
07bfdd14b4be97dbe282573eecd5bc6b062a92b1
 shalom-ext              IN      AAAA            2001:470:f012:2::2
+                        IN      SSHFP           1 1
073b3198599c59a3c2a9db8c209a2097ea46aa09
 shalom-fw               IN      AAAA            2001:470:c:2e6::2
 shalom-svc              IN      AAAA            2001:470:f012:2::1

@@ -43,6 +46,7 @@
 ;| Web site Urls
 @                        IN      A               203.79.116.183
                          IN      AAAA            2001:470:f012:2::2
+                        IN      TXT             "Some hash"
 www                      IN      CNAME           @


zone_tool >
```

The zi_id date format arguments can be used with show_zone and edit_zone instead of a straight zi_id. So you can workflow using command line history. edit_zone will take the specified ZI ID as the source to change, and make it the published ZI on completion (you can also abort, and also diff your edit before updating).

Differencing Zones

The diff_zones command can be used to show the difference between 2 zones. This is useful if the latter zone was created from the other . The zi_id arguments are given in the order of the zone names.

To show it works:

```
zone_tool > diff_zones anathoth.net anathoth.net ^-- ^
@@ -3,11 +3,11 @@

 ;
 ; Zone:       anathoth.net.
-; change_by:  grantma@shalom-ext.internal.anathoth.net/Admin
-; zi_id:      102588
-; zi_ctime:   Mon Aug 20 10:27:36 2012
-; zi_mtime:   Mon Aug 20 10:27:38 2012
-; zi_ptime:   Mon Aug 20 10:27:38 2012
+; change_by:  hd-test@shalom-ext.internal.anathoth.net/Admin
+; zi_id:      102592
+; zi_ctime:   Mon Aug 20 11:07:49 2012
+; zi_mtime:   Mon Aug 20 11:12:07 2012
+; zi_ptime:   Mon Aug 20 11:12:07 2012
 ;
```

And of course:

```
zone_tool > ls_zi anathoth.net
         102012               2012042702    Mon Feb 27 10:06:28 2012
         102100               2012050800    Tue May  8 14:19:17 2012
         102104               2012050801    Tue May  8 14:22:25 2012
         102106               2012050802    Tue May  8 14:29:02 2012
         102108               2012050803    Tue May  8 14:34:17 2012
         102133               2012050900    Wed May  9 09:23:04 2012
         102136               2012050901    Wed May  9 09:24:14 2012
         102152               2012050902    Wed May  9 12:55:11 2012
         102155               2012050903    Wed May  9 12:56:27 2012
         102156               2012050904    Wed May  9 12:56:46 2012
         102159               2012051000    Thu May 10 10:07:52 2012
         102162               2012051012    Thu May 10 10:09:04 2012
         102164               2012051013    Thu May 10 13:31:06 2012
         102167               2012051013    Thu May 10 16:13:56 2012
         102171               2012051014    Thu May 10 16:45:33 2012
         102187               2012052100    Mon May 21 11:43:57 2012
         102189               2012052300    Wed May 23 11:47:01 2012
         102199               2012052400    Thu May 24 15:23:05 2012
         102201               2012052401    Thu May 24 15:24:18 2012
         102261               2012072500    Tue Jul  3 12:05:29 2012
         102468               2012072600    Thu Jul 26 12:13:53 2012
         102585               2012082000    Mon Aug 20 10:26:27 2012
         102588               2012082000    Mon Aug 20 10:27:36 2012
         102589               2012082000    Mon Aug 20 10:41:26 2012
        *102592               2012082000    Mon Aug 20 11:07:49 2012
zone_tool > copy_zone -z 102592 anathoth.net wham-blam.org
zone_tool > edit_zone wham-blam.org
***   Do you wish to Abort, Change, Diff, or Update the zone
```

```
'wham-blam.org.'?
--[U]/a/c/d>
zone_tool > diff_zones anathoth.net wham-blam.org ^-- ^
@@ -1,33 +1,35 @@
 $TTL 24h
-$ORIGIN anathoth.net.
+$ORIGIN wham-blam.org.

 ;
-; Zone:       anathoth.net.
+; Zone:       wham-blam.org.
+; Reference:  anathoth
 ; change_by:  grantma@shalom-ext.internal.anathoth.net/Admin
-; zi_id:      102588
-; zi_ctime:   Mon Aug 20 10:27:36 2012
-; zi_mtime:   Mon Aug 20 10:27:38 2012
-; zi_ptime:   Mon Aug 20 10:27:38 2012
+; zi_id:      102598
+; zi_ctime:   Thu Aug 23 10:52:16 2012
+; zi_mtime:   Thu Aug 23 10:52:18 2012
+; zi_ptime:   Thu Aug 23 10:52:18 2012
 ;


-;|
-;| Apex resource records for anathoth.net.
-;|
-@                       IN      SOA             ( ns1           ;Master NS
+;| Apex resource records for wham-blam.org.
+;!REF:anathoth
+@                       IN      SOA             ( ns1.anathoth.net.
;Master NS
                                                  matthewgrant5.gmail.com.
;RP email
-                                                 2012082000   ;Serial
yyyymmddnn
+                                                 2012082301   ;Serial
yyyymmddnn
                                                  600          ;Refresh
                                                  600          ;Retry
                                                  604800       ;Expire
                                                  600
;Minimum/Ncache
                                                  )
-                       IN      NS              ns3
-                       IN      NS              ns2
-                       IN      NS              ns1
+                       IN      NS              ns3.anathoth.net.
+                       IN      NS              ns2.anathoth.net.
+                       IN      NS              ns1.anathoth.net.


 ;| Hosts
```

```
+bingo                    IN      AAAA            ::1
+                         IN      TXT             "Samson was here"
 shalom-dr                IN      AAAA            2001:470:f012:2::3
                          IN      SSHFP           1 1
07bfdd14b4be97dbe282573eecd5bc6b062a92b1
 shalom-ext               IN      AAAA            2001:470:f012:2::2
```

```
zone_tool >
```

Programming Information

WSGI JSON RPC Protocol

Net24dmsd Communications Protocol

> *This document will be updated as the DMS protocol is implemented.*

The server to be tightly coded to a standard so it behaves reasonably. Clients won't have to be so fussy, but should not request anything they are not coded to deal with! Comprehensive error processing by the client is encouraged.

The protocol will be JSON-RPC over HTTP 1.1+. This will enable the processing of multiple requests over the same TCP connection. TCP connections to the server will be cacheable, and can be held open up to a limit set on the server. Multiple POSTs over the connection are allowed, and multiple RPC requests can be submitted within a POST request, with the id: set to a UUID string generated as per RFC 4122.

JSON/RPC 2.0 specification will be used. JSON RFC 4627 will be used as the data format.

JSON-RPC over HTTP will be used to access the server, with the limitation being that HTTP POST shall be used, not GET with its encoded URL.... (blech!). Batch mode requests will also be implemented.

Authentication will be via HTTP Basic authentication, with the deployed implementation using HTTPS for integrity. Privileged access stratification will be achieved by accessing different Python WSGI scripts at different URLs. Initially 2 different levels of access will be provided:

1. Customer for 1st domains and Net24 customer front ends,
2. HelpDesk for normal administrative work on the DNS.

Comprehensive administrative functionality will be available via the zone-tool command line UI on the Master DNS server.

Contents

Error Information

Errors shall be python exceptions translated to JSON-RPC errors. The 'data' section will contain relevant exception attributes, along with an error message. There will different classes of error, dependent on the operation being performed.

Errors to do with Zone Instance submission will return RR Group and RRS index information into the ZI structure sent in the request.

Please not that zones outside the client role are treated as if they do not exist unless otherwise noted.

Please see The DMS Errors page for a full listing.

Editing Cycle

Please note that an edit cycle starts with the 'edit_zone' call below, and is finished with an 'update_zone' call. When edit locking is enabled for the zone (typically only helpdesk, admin, and special customers) the 'tickle_editlock' (keep a locked editing session live, called on receiving any data from web browser) and 'cancel_edit_zone' (to cancel edit session) calls should be used.

Incremental Updates

The 'update_rrs' call is to be used for incremental updates. The update_type is a unique ID identifying the operation type, of which only one per zone can be queued at a time. Each update call eventually generates a new ZI incorporating the changes after the call returns. When the call is made, a forward-looking check is made with the current (or candidate) ZI to make sure the changes to be made are consistent.

This mechanism is only for the simple consistent changes required for adding/removing a Web site to a domain, adding/removing mail MX records for adding Web hosting or Mail to a domain.

Note: The error checking is forward looking and would probably fail to produce a published zone for complex change sets. It is NOT for making general editing changes such as these to the zone. Use the Editing Cycle above for user UI editing sessions, not this.

Request calls

list_zone(<names>, [reference], [include_deleted], [toggle_deleted], [include_disabled])

<names> array of wildcard-names
[reference] customer ID or other ID meta data
[include_deleted] boolean true/false whether to include deleted domains in listing
[toggle_deleted] boolean true/false list only deleted domains
[include_deleted] boolean true/false include disabled domains, defaults to true

To list domains. Many wild carded domains can be specified. Response will either be the list of domain names, or an empty list as domains cannot be found. Customer facing DMIs will be set up so that a *ZoneSearchPatternError* exce ption will be thrown if list_zone is called with no *names*, or *names* set to '*', without *reference* being given.

list_zi(<name>)

<name> domain to list

List all zis for a domain. Returns just the base zone_sm object, and the list of zis 'all_zis'. The published zi is the 'zi' in the zone_sm object, and its full structure is returned, Each zi is accompanied by its ctime and mtime. The output

is shown below in show_zone.

show_zone(<name>, [zi_id])

show_zone_text(<name>, [zi_id], [all_rrs])

<name> domain to show.
[zi_id] optional zone instance
[all_rrs] optional NOT showing of Apex RRs. Only for show_zone_text

Like the previous operation, except that the full zi returned can be given.

show_zone_text returns a zone file text blob, JSON encoded. Note that this means new line, tab, etc are encoded as '\n', '\t' not as a control characters.

Sample JSON dump of output of show_zone. Note that "sectags" sub-array only shows up in Admin DMS client RPC interface

```
{   'all_zis': [   {   'ctime': 'Mon Mar  5 14:11:25 2012',
                       'mtime': 'Mon Mar  5 14:46:21 2012',
                       'ptime': 'Mon Mar  5 14:46:21 2012',
                       'zi_id': 45,
                       'zone_id': 32}],
    'alt_sg_name': null,
    'auto_dnssec': false,
    'ctime': 'Mon Mar  5 14:11:25 2012',
    'deleted_start': null,
    'edit_lock': false,
    'edit_lock_token': null,
    'inc_updates': false,
    'lock_state': 'EDIT_UNLOCK',
    'mtime': 'Mon Mar  5 14:11:25 2012',
    'name': 'anathoth.net.',
    'nsec3': false,
    'reference': 'net24',
    'sectags': [{'sectag_label': 'Admin'}],
    'soa_serial': 2012030500,
    'sg_name': 'net24-one',
    'state': 'PUBLISHED',
    'use_apex_ns': true,
    'zi': {   'ctime': 'Mon Mar  5 14:11:25 2012',
              'mtime': 'Mon Mar  5 14:46:21 2012',
              'ptime': 'Mon Mar  5 14:46:21 2012',
              'rr_groups': [   {   'comment': 'Apex resource records for
anathoth.net.',
                                   'rrs': [   {   'class': 'IN',
                                                  'disable': false,
                                                  'label': '@',
                                                  'lock_ptr': false,
                                                  'rdata':
'ns2.net24.net.nz.',
                                                  'reference': null,
```

```
                                                'rr_id': 5126,
                                                'ttl': null,
                                                'type': 'NS',
                                                'zi_id': 45},
                                            {   'class': 'IN',
                                                'disable': false,
                                                'label': '@',
                                                'lock_ptr': false,
                                                'rdata':
'ns1.net24.net.nz.',
                                                'reference': null,
                                                'rr_id': 5125,
                                                'ttl': null,
                                                'type': 'NS',
                                                'zi_id': 45},
                                            {   'class': 'IN',
                                                'disable': false,
                                                'label': '@',
                                                'lock_ptr': false,
                                                'rdata':
'ns1.net24.net.nz. soa.net24.net.nz. 2012030500 7200 7200 604800 86400',
                                                'reference': null,
                                                'rr_id': 5124,
                                                'ttl': null,
                                                'type': 'SOA',
                                                'zi_id': 45}],
                                    'tag': 'APEX_RRS'}],
                'soa_expire': '7d',
                'soa_minimum': '24h',
                'soa_mname': 'ns1.net24.net.nz.',
                'soa_refresh': '7200',
                'soa_retry': '7200',
                'soa_rname': 'soa.net24.net.nz.',
                'soa_serial': 2012030500,
                'soa_ttl': null,
                'zi_id': 45,
                'zone_id': 32,
                'zone_ttl': '24h'},
    'zi_candidate_id': 45,
    'zi_id': 45,
    'zone_id': 32,
```

```
    'zone_type': 'DynDNSZoneSM'}
```

show_zone_text(<name>, zi_id=None)

create_zone(<name> <reference> <login_id> [zi_data] [sectags] [sg_name] [edit_lock] [auto_dnssec] [nesc3] [inc_updates])

create_zone(<name> <reference> <login_id>)

copy_zone(<src_name> <name> <reference> <login_id> [zi_id] [sectags] [sg_name] [edit_lock] [auto_dnssec] [nesc3] [inc_updates]

<src_name> source domain to be copied
[zi_id] source ZI to be copied
<name> domain to be created
<reference> reference for domain being created - can be missed, but domain will be owned by default_ref, ie VOYAGERNET-NZ!
<login_id> DMI login ID. Email address, or numerical login_id
[zi_data] optional zi_data (for feeding in a template)
[sg_name] optional sg where zone is to be created Admin DMS only.
[sectags] optional list of security tags for new zone. Admin DMS only. Same array/object format as listing above.
[edit_lock] optional boolean for turning on edit_lock mode, default false
[auto_dnssec] optional boolean for turning on automatic DNSSEC, default false
[nsec3] optional boolean for enabling NSEC3 under DNSSEC, default false
[inc_updates] optional boolean for enabling incremental updates for zone, default true for basic interface, false for help desk and admin interfaces.

Returns: true

Errors are returned if a zone already exists. Optional zi_data in the format above can be feed in for a template. Please note that Apex SOA and NS records will not be taken. Basic call used by default for 1stdomains and Register Direct.

enable_zone(<name>)

<name> domain to be enabled.

Returns: true

Errors will be returned if the zone does not exist.

disable_zone(<name>)

<name> domain to be disabled.

Returns: true

Errors will be returned if the zone does not exist.

delete_zone(<name>)

<name> domain to be deleted.

Returns: true

Errors can be returned if the zone does not exist.

set_zone(<name> [edit_lock] [auto_dnssec] [nsec3] [inc_updates])

<name> domain to be created
[edit_lock] optional boolean for turning on edit_lock mode, default false
[auto_dnssec] optional boolean for turning on automatic DNSSEC, default false
[nsec3] optional boolean for enabling NSEC3 under DNSSEC, default false
[inc_updates] optional boolean for enabling incremental updates for zone, default true for basic interface, false for help desk and admin interfaces.

Returns: true

Errors are returned if a zone already exists.

undelete_zone(<zone_id>)

<zone_id> Id of deleted zone to be undeleted

Undelete a zone.. This can only be done to a deleted zone, and if there are no active zones with the same name.

Returns: true

destroy_zone(<zone_id>)

<zone_id> Id of deleted zone to be destroyed

Destroy a zone.. This can only be done to a deleted zone.

Returns: true

copy_zi(<src_name>, <name>, [zi_id])

<src_name> Source zone name
<name> destination domain name
<login_id> DMI login ID. Email address, or numerical login_id
[zi_id] ZI ID to be copied, default published ZI of source zone.

Copy a ZI from a source zone to another.

Returns: true

delete_zi(<name> <zi_id>)

<name> domain name
<zi_id> ZI ID

Delete a zi. This can only be done for a ZI that is not currently in use.

edit_zone(<name> <login_id> [zi_id])

<name> domain to be edited.
[zi_id] optional zone-instance number or Null

Returns: list (zone_zi_data, edit_lock_token).

Can be: [zi_data, edit_lock_token] if edit_lock obtained.

[zi_data, Null] if zone does not have edit locking enabled.

Errors are returned if the zone does not exist, zi_id is invalid, an edit_lock is not able to be obtained.

Returns a zone structure, with a list of all zis in database for domain, accompanied by the zi's date. This structure is the one show above for show_zone.

The zi structure contains all the SOA data. Depending on the the value of 'use_apex_ns', for 'True' the Apex NS records are supplied, and the secondary DNS server parameters of the SOA record are setable. Otherwise, the Apex NS records are not supplied as they are the global DNS secondary server settings, and the only editable SOA fields (always editable) are soa_minimum, soa_ttl, and zone_ttl. Net24dmd always generates the SOA record for a zone from the values in the zi structure, and automatically calculates the zone SOA serial number based on the algorithm used in the RFCs(RFC 2316 Sec 3.4.2.2, RFC 1982 Section 3) and conventional serial number guidelines based on the date, if it is possible.

The zone-instance parameter defaults to the published ZI, and another ZI can be given. The edit lock is an optional feature zone state machine that can be enabled from zone-tool for domains the are often edited, to prevent unpredictable updates to published zones (Ie 2 people editing isx.net.nz simultaneously, and then one having his changes wiped out by the later publish action). The edit lock is covered by an inactivity timeout which ist reset by the tickle_editlock() method.

tickle_editlock(<name>, <edit_lock_token>)

<name> domain being edited
<edit_lock_token> edit lock token to be tickled

Notification of UI activity to reset edit lock time out.

cancel_edit_zone(<name>, <edit_lock_token>)

<name> domain being edited
<edit_lock_token> edit lock token to be canceled

Cancels a locked zone editing session.

update_zone(<name>, <zi_data>, <login_id>, [edit_lock_token])

<name> domain to be updated
<zi_data> new zi structure to be published.
<login_id> DMI login_id. Email format, or numerical string.
[edit_lock_token] Must be supplied to finish an edit locked session.

Saves zi_data to database for a zone. Queues a ZoneSMEditUpdate (edit_locked zone) or ZoneSMUpdate event to publish domain with new zi.

show_sectags()

List all possible security tags. This command is only available with Admin level DMS client privilege. Sectags are created and deleted from the one_tool command line. Each WSGI back end has its privilege assigned by configuring it with a given security tag.

show_zone_sectags(<name>)

<name> domain to be queried.

List the security attached to the given zone. This command is only available with Admin level DMS client privilege.

add_zone_sectag(<name>, <sectag>):

<name> domain
<sectag> sectag to be added

Adds a sectag to a zone. Admin Level DMS client privilege only.

Returns: true

delete_zone_sectag(<name>, <sectag>):

<name> domain
<sectag> sectag to be deleted

Deletes a sectag from a zone. Admin Level DMS client privilege only.

Returns: true

replace_zone_sectags(<name>, <sectags>):

<name> domain to be operated on
<sectags> list of sectags as per above format in listing.

Completely replaces the zones current sectags with the ones specified in the list. This command is only available with Admin level DMS client privilege.

Thus you can use show_sectags() to get all possible sectags, show_zone_sectags() to fill out checkboxs in a dialogue/list, and then call replace_zone_sectags() with all checked values when user clicks <OK>/submits in Web UI.

sign_zone(<name>)

<name> domain to be operated on.

Resign a DNSSEC zone.

Returns: true

load_keys(<name>)

<name> domain to be operated on.

Load the DNSSEC keys for a zone.

Returns: true

refresh_zone(<name>)

<name> domain to be refreshed.

Refresh/update the contents of a zone from the DB into the DNS. Issues a publish event to zone.

Returns: true

reset_zone(<name>)

<name> domain to be reset

Resets the zone state machine. Useful for when net24dmd has an internal error, or when named is mis-configured for write access.

Returns: true

refresh_zone_ttl(<name> [zone_ttl])

<name> domain name of zone
[zone_ttl] named TTL string

Refresh a zones TTL, using the global default for zone creation if none given.

Returns: true

show_configsm()

Show the current status of the master named configuration state machine. Useful as it show when the next rndc config can happen.

Returns: true

create_reference(<reference>)

<reference> entity reference string

Creates an entity reference string in the DMS for use with a set of zones.

Returns: true

delete_reference(<reference>)

<reference> entity reference string

Deletes an unused entity reference string from the DMS when there are no more zones against it.

Returns: true

rename_reference(<reference> <dst_reference>)

<reference> original entity reference string
<dst_reference> new entity reference string

Rename a reference in the DMS. This should check with the user first to see if they really want to do this. I can see someone like Mike wanting to use this from DMI if the ID in the DMS zone database is wrong, if it is an account ID.

Returns: true

list_reference([reference-wildcard], [<reference-wildcard], ...)

[reference-wildcard] reference wildcard string.

Lists references. Help desk and admin level functionality.

returns list of references in JSON.

set_zone_reference(<name>, <reference>)

<name> domain to be operated on
<reference> reference to be set on domain

Change the reference on a domain. Again Admin level only functionality.

Returns: true

rr_query_db(<label> [name] [type] [rdata] [zi_id] [show_all])

<label> host name or other DNS label
[name] domain to be queried
[type] RR type
[rdata] RR rdata string
[zi_id] ZI ID
[show_all] boolean rue/false, show all records, including disabled ones.

Query the DB ala the OS libc/libresolv hostname() call. This uses a cross zone DB query looking for any records. This is Admin level only functionality.

update_rrs(<name> <update_data> <update_type> <login_id>)

<name> domain being updated
<update_data> update data for zone
<update_type> client update type
<login_id> Email format, or numerical string.

Do incremental updates on a zone. The update data is the same ZI data format as in create_zone()

Example update file from equiv zone_tool update_rrs command:

```
$ORIGIN     foo.bar.org.
        $UPDATE_TYPE SpannerReplacement_ShouldBeUUIDperClientOpType

        ;!RROP:DELETE
        ns5             IN  ANY     ""  ; All records for ns5
        ;!RROP:DELETE
        ns7             IN  A       ""  ; All A records for ns2
        ;!RROP:DELETE
        ns67            IN  A       192.168.2.3 ; Specific record

        ;!RROP:ADD
        ns99            IN  TXT     "Does not know Maxwell Smart"
        ;!RROP:ADD
        ns99            IN  AAAA       2002:fac::1

        ;!RROP:UPDATE_RRTYPE
        ns99            IN  AAAA    ::1
```

The ZI data RRs are augmented with the **update_op** property, which takes the RROP text values of ADD, DELETE, and UPDATE_RRTYPE. As seen above the DELETE update_op can use RR type ANY, and blank rdata as wildcards. UPDATE_RRTYPE replaces all records of that type for the DNS zone node concerned.

The update_type property is used to make sure that only one update_type is queued per zone for execution. Each update is a unique transaction for the zone
concerned.

Note that their are separate privilege levels for the Admin, helpdesk, and ordinary customer front ends, and these can affect the auto reverse parameters that can be used in the call, exactly the same as for update_zone/create_zone above.

Example of the JSON params object feed to the update_rrs() call:

```
{   'name': 'foo.bar.org.',
    'update_data': {   'rr_groups': [   {   'rrs': [   {   'class': 'IN',
                                                           'disable':
false,
                                                           'force_reverse':
false,
                                                           'label':
'ns5.foo.bar.org.',
                                                           'lock_ptr':
false,
                                                           'rdata': null,
                                                           'reference':
null,
                                                           'type': 'ANY',
                                                           'update_op':
'DELETE'},
                                                       {   'class': 'IN',
```

```
                                                    'disable':
false,
                                                    'force_reverse':
false,
                                                    'label':
'ns7.foo.bar.org.',
                                                    'lock_ptr':
false,
                                                    'rdata': null,
                                                    'reference':
null,
                                                    'type': 'A',
                                                    'update_op':
'DELETE'},
                                                {   'class': 'IN',
                                                    'disable':
false,
                                                    'force_reverse':
false,
                                                    'label':
'ns67.foo.bar.org.',
                                                    'lock_ptr':
false,
                                                    'rdata':
'192.168.2.3',
                                                    'reference':
null,
                                                    'type': 'A',
                                                    'update_op':
'DELETE'}]},
                                    {   'rrs': [    {   'class': 'IN',
                                                    'disable':
false,
                                                    'force_reverse':
false,
                                                    'label':
'ns99.foo.bar.org.',
                                                    'lock_ptr':
false,
                                                    'rdata': '"Does
not know Maxwell Smart"',
                                                    'reference':
null,
                                                    'type': 'TXT',
                                                    'update_op':
'ADD'},
                                                {   'class': 'IN',
                                                    'disable':
false,
                                                    'force_reverse':
false,
                                                    'label':
'ns99.foo.bar.org.',
```

```
                                                                'lock_ptr':
false,
                                                                'rdata':
'2002:fac::1',
                                                                'reference':
null,
                                                                'type': 'AAAA',
                                                                'update_op':
'ADD'}]},
                                                {   'rrs': [   {   'class': 'IN',
                                                                'disable':
false,
                                                                'force_reverse':
false,
                                                                'label':
'ns99.foo.bar.org.',
                                                                'lock_ptr':
false,
                                                                'rdata': '::1',
                                                                'reference':
null,
                                                                'type': 'AAAA',
                                                                'update_op':
'UPDATE_RRTYPE'}]}]},
    'update_type': 'SpannerReplacement_ShouldBeUUIDperClientOpType'}
```

set_zone_sg(<name>, <sg_name>)

<name> domain to be operated on.
<sg_name> sg the zone is being moved to.

Set the sg a zone is served on. Note that this call at present can only be used on disabled zones. Admin level only call.

Returns: true

set_zone_alt_sg(<name>, <sg_name>)

<name> domain to be operated on.
<sg_name> Alternate sg the zone is being served on.

Set an additional sg a zone will be served on. Note that this call at present can only be used on disabled zones. Note that the sg concerned has to be refreshed. Admin level only call.

Returns: true

list_sg()

List all sgs that are existent on the master DNS server. Admin level only call, for populating menu drop boxes when creating zones etc.

Errors are exceptions in net24/dms/exceptions.py, as listed above

DMS Errors

DMS Errors

Error Exceptions

The data section of each JSON RPC error message has a 'exception_type', 'stack_trace', 'exception_message', as well as following extra fields below. The error class hierachy is at the top of the listing. The 'exception_type' contains a name for the error such as 'BaseJsonRpcError.DMSError.CancelEditLockFailure'. The intent is to allow the error messages to be classified in the DMI client.

```
Help on module net24.dms.exceptions in net24.dms:

NAME
    net24.dms.exceptions - Exceptions module for the DMS

CLASSES
    net24.core.wsgi.jsonrpc_server.BaseJsonRpcError(builtins.Exception)
        DMSError
            BadInitialZoneName
            BinaryFileError
```

ConfigBatchHoldFailed
DBReadOnlyError
IncrementalUpdateNotInTrialRun
IncrementalUpdatesDisabled
InvalidDomainName
    ReverseNamesNotAccepted
InvalidHmacType
LoginIdError
    LoginIdFormatError
    LoginIdInvalidError
MultipleReferencesFound
NoPreviousLabelParseError
NoReferenceFound
NoReplicaSgFound
NoSecTagsExist
NoSgFound
NoZoneSecTagsFound
NoZonesFound
ReferenceDoesNotExist
ReferenceExists
ReferenceFormatError
ReferenceStillUsed
ReplicaSgExists
RestoreNamedDbError
    NamedConfWriteError
    NamedStillRunning
    Net24dmdStillRunning
    PidFileAccessError
    PidFileValueError
    ZoneFileWriteError
RrQueryDomainError
SOASerialError
    SOASerialArithmeticError
    SOASerialCandidateIgnored
    SOASerialOcclusionError
    SOASerialPublishedError
    SOASerialRangeError
    SOASerialTypeError
SecTagPermissionDenied
ServerError
    NoServerFound
    NoServerFoundByAddress
    ServerAddressExists
    ServerExists
    ServerNotDisabled
ServerSmFailure
SgExists
SgMultipleResults
SgNameRequired
SgStillUsed
    SgStillHasServers
    SgStillHasZones
UpdateError

```
    DynDNSUpdateError
        DynDNSCantReadKeyError
    NoSuchZoneOnServerError
UpdateTypeAlreadyQueued
UpdateTypeRequired
ZiIdSyntaxError
    ZiIdAdjStringSyntaxError
    ZiIdDdMmYyyySyntaxError
    ZiIdDdSlashMmSyntaxError
    ZiIdHhMmSyntaxError
    ZiIdIsoDateSyntaxError
    ZiIdTimeAmountSyntaxError
    ZiIdTimeUnitSyntaxError
ZiInUse
ZiParseError
    HostnameZiParseError
    TtlZiParseError
ZiTextParseError
ZoneAdminPrivilegeNeeded
ZoneBeingCreated
ZoneCfgItem
    ZoneCfgItemNotFound
    ZoneCfgItemValueError
ZoneDisabled
ZoneExists
ZoneHasNoSOARecord
ZoneHasNoZi
ZoneMultipleResults
ZoneNameUndefined
ZoneNoAltSgForSwap
ZoneNotDeleted
ZoneNotDisabled
ZoneNotDnssecEnabled
ZoneNotFound
    ZiNotFound
    ZoneNotFoundByZoneId
ZoneNotPublished
ZoneParseError
    BadNameOwnerError
    BadNameRdataError
    DirectiveParseError
        HostnameParseError
        TtlInWrongPlace
        TtlParseError
    GenerateNotSupported
    IncludeNotSupported
    InvalidUpdateOperation
    LabelNotInDomain
    Not7ValuesSOAParseError
    PrivilegeNeeded
        AdminPrivilegeNeeded
        HelpdeskPrivilegeNeeded
    RRNoTypeGiven
```

```
                RdataParseError
                RropNotSupported
                SOASerialMustBeInteger
                UnhandledClassError
                UnhandledTypeError
                UpdateTypeNotSupported
                ZoneError
                    DuplicateRecordInZone
                    ZoneAlreadyHasSOARecord
                    ZoneCNAMEExists
                    ZoneCNAMELabelExists
                    ZoneCheckIntegrityNoGlue
                    ZoneHasNoNSRecord
                    ZoneSOARecordNotAtApex
            ZoneSearchPatternError
                OnlyOneLoneWildcardValid
                ReferenceMustBeGiven
            ZoneSecTagDoesNotExist
                ZoneSecTagConfigError
            ZoneSecTagExists
            ZoneSecTagStillUsed
            ZoneSmFailure
                ActiveZoneExists
                CancelEditLockFailure
                EditLockFailure
                TickleEditLockFailure
                UpdateZoneFailure
                ZoneFilesStillExist
            ZoneTTLNotSetError

    class ActiveZoneExists(ZoneSmFailure)
     |  Another zone instance is active - this one cannot be activated.
     |
     |  JSONRPC Error:      -69
     |  JSONRPC data keys:  'name'          - domain name
     |  JSONRPC data keys:  'event_message' - event message
     |  JSONRPC data keys:  'event_results' - event results object
     |

    class AdminPrivilegeNeeded(PrivilegeNeeded)
     |  Administrative privilege is needed to set this RR field
     |
     |  JSONRPC Error:      -26
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class BadInitialZoneName(DMSError)
     |  Name of the Zone can not be determined.
```

```
      |
      |  JSONRPC Error:      -55
      |  JSONRPC data keys:  'file_name'   - file name being loaded.
      |

    class BadNameOwnerError(ZoneParseError)
      |  Owner name of an A AAAA or MX record is not a valid hostname
      |
      |  JSONRPC Error:      -13
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |                      'label_thing'       - thing given as RR label
      |

    class BadNameRdataError(ZoneParseError)
      |  Name in the rdata of a record is not a valid hostname
      |
      |  JSONRPC Error:      -14
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |                      'rdata_thing'   - bad RDATA of RR
      |                      'bad_name'      - bad hostname in RDATA
      |

    class BinaryFileError(DMSError)
      |  Trying to load a binary file.
      |
      |  JSONRPC Error:      -39
      |  JSONRPC data keys:  'file_name'   - file name
      |

    class CancelEditLockFailure(ZoneSmFailure)
      |  For a DMI, can't clear edit_lock for zone.
      |
      |  JSONRPC Error:      -33
      |  JSONRPC data keys:  'name'          - domain name
      |  JSONRPC data keys:  'event_message' - Cancel Event Message
      |  JSONRPC data keys:  'event_results' - Event results object
      |

    class ConfigBatchHoldFailed(DMSError)
      |  Configuration SM Failed to enter CONFIG_HOLD for batch zone
creation
      |
      |  JSONRPC Error:      -56
```

```
     |

    class DBReadOnlyError(DMSError)
     |  Database is in Read Only mode.
     |
     |  JSONRPC Error: - 122
     |  JSONRPC data keys: 'exc_msg'  - original exception message
     |

    class DMSError(net24.core.wsgi.jsonrpc_server.BaseJsonRpcError)
     |  Base DMS Error Exception
     |
     |  JSONRPC Error: JSONRPC_INTERNAL_ERROR
     |

    class DirectiveParseError(ZoneParseError)

    class DuplicateRecordInZone(ZoneError)
     |  Zone already has a record for this.
     |
     |  JSONRPC Error:      -20
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class DynDNSCantReadKeyError(DynDNSUpdateError)
     |  Can't read in configured TSIG for Dynamic DNS update
     |
     |  JSONRPC Error: -5
     |  JSONRPC data keys: 'name'  - None
     |

    class DynDNSUpdateError(UpdateError)
     |  Error during update of zone
     |
     |  JSONRPC Error: -4
     |  JSONRPC data keys: 'name'  - domain name
     |

    class EditLockFailure(ZoneSmFailure)
     |  For a DMI, can't obtain an edit_lock for zone.
     |
     |  JSONRPC Error:      -34
     |  JSONRPC data keys:  'name'          - domain name
     |  JSONRPC data keys:  'event_message' - Lock Event Message
     |  JSONRPC data keys:  'event_results' - Event results object
     |

    class GenerateNotSupported(ZoneParseError)
```

```
     |  Our zone parser does not support the $GENERATE statement
     |
     |  JSONRPC Error:      -54
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class HelpdeskPrivilegeNeeded(PrivilegeNeeded)
     |  Help desk privilege is needed to set this RR field
     |
     |  JSONRPC Error:      -27
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class HostnameParseError(DirectiveParseError)
     |  Hostname parse error while parsing zone file.
     |
     |  JSONRPC Error:      -51
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class HostnameZiParseError(ZiParseError)
     |  Zi related SOA mname or rname value error.
     |
     |  JSONRPC Error:      -48
     |  JSONRPC data keys:  'name'      - domain name
     |                      'zi_field'  - ZI field where error found
     |                      'value'     - value in error
     |

    class IncludeNotSupported(ZoneParseError)
     |  Our zone parser does not support the $INCLUDE statement
     |
     |  JSONRPC Error:      -50
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
```

```
      |                                                rr_groups
      |

    class IncrementalUpdateNotInTrialRun(DMSError)
      |  Error in Incremental Update mechanism.  Update mechanism not in
      |  Trial Run Mode.
      |
      |  JSONRPC Error:      JSON_RPC_INTERNAL_ERROR
      |  JSONRPC data keys:  'name'      - domain name
      |

    class IncrementalUpdatesDisabled(DMSError)
      |  Incremental Updates are disabled for this zone.
      |
      |  JSONRPC Error:      -90
      |  JSONRPC data keys:  'name'      - domain name
      |

    class InvalidDomainName(DMSError)
      |  Domain name is invalid.
      |
      |  JSONRPC Error:      -89
      |  JSONRPC data keys:  'name'      - domain name
      |

    class InvalidHmacType(DMSError)
      |  Invalid Hmac type given
      |
      |  JSONRPC Error:      -96
      |  JSONRPC data keys: 'hmac_type'      - Given hmac type
      |

    class InvalidUpdateOperation(ZoneParseError)
      |  RR type is one we don't handle.
      |
      |  JSONRPC Error:      -83
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class LabelNotInDomain(ZoneParseError)
      |  FQDN Label outside of domain
      |
      |  JSONRPC Error:      -12
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
```

```
      |                                                rr_groups
      |                     'label_thing'       - thing given as RR label
      |

    class LoginIdError(DMSError)
      |  DMS Error class to cover login_id exceptions
      |

    class LoginIdFormatError(LoginIdError)
      |  A login_id can only consist of the characters '-_a-zA-Z0-9.@',
      |  and must start with a letter or numeral.  It also must be less than
      |  512 characters long.
      |
      |  JSONRPC Error:      -124
      |  JSONRPC data keys:  'login_id'  - login_id
      |                      'error'     - error message
      |

    class LoginIdInvalidError(LoginIdError)
      |  A login_id must be given, and be less than 512 characters long.
      |
      |  JSONRPC Error:      -125
      |  JSONRPC data keys:  'error'     - error message
      |

    class MultipleReferencesFound(DMSError)
      |  Multiple references were found
      |
      |  JSONRPC Error:      -68
      |  JSONRPC data keys:  'reference'   - reference code
      |

    class NamedConfWriteError(RestoreNamedDbError)
      |  Can't write named.conf sections
      |
      |  JSONRPCError:       -113
      |  JSONRPC data keys:  'name'  - domain name
      |                      'internal_error' - error that occured
      |

    class NamedStillRunning(RestoreNamedDbError)
      |  Named is still running
      |
      |  JSONRPCError:       -108
      |  JSONRPC data keys:  'rndc_status_exit_code' - exit code from 'rndc
status'
      |

    class Net24dmdStillRunning(RestoreNamedDbError)
      |  Net24dmd is still running
      |
      |  JSONRPCError:       -109
      |  JSONRPC data keys:  net24dmd_pid - net24dmd PID
```

```
 |

class NoPreviousLabelParseError(DMSError)
 |  No Previous Label seen. - This should not be reached in code
 |
 |  JSONRPC Error:     -7
 |  JSONRPC data keys:  'name'  - domain name
 |

class NoReferenceFound(DMSError)
 |  No Reference found.
 |
 |  JSONRPC Error:      -67
 |  JSONRPC data keys:  'reference'   - reference code
 |

class NoReplicaSgFound(DMSError)
 |  For a DMI, Master SG not found
 |
 |  JSONRPC Error:      -128
 |

class NoSecTagsExist(DMSError)
 |  No zone security tags found for this domain.
 |
 |  JSONRPC Error:      -45
 |

class NoServerFound(ServerError)
 |  Server does not exist
 |
 |  JSONRPC Error:      -75
 |  JSONRPC data keys:  'server_name'      - server name
 |

class NoServerFoundByAddress(ServerError)
 |  Server does not exist
 |
 |  JSONRPC Error:      -76
 |  JSONRPC data keys:  'address'      - server address
 |

class NoSgFound(DMSError)
 |  For a DMI, requested SG not found
 |
 |  JSONRPC Error:      -59
 |  JSONRPC data keys:  'sg_name'      - SG name
 |

class NoSuchZoneOnServerError(UpdateError)
 |  No zone found in DNS server
 |
 |  JSONRPC Error: -6
```

```
     |  JSONRPC data keys:  'name'      - zone name
     |                      'server'    - server hostname/address
     |                      'port'      - server port
     |
     |  This exception only occurs internally in net24dmd, and dyndns_tool.
It is
     |  not returned at all over HTTP JSON RPC.
     |

    class NoZoneSecTagsFound(DMSError)
     |  No zone security tags found for this domain.
     |
     |  JSONRPC Error:      -44
     |  JSONRPC data keys:  'name'   - domain name
     |

    class NoZonesFound(DMSError)
     |  For a DMI, can't find the requested zones.
     |
     |  JSONRPC Error:      -32
     |  JSONRPC data keys:  'name_pattern'  - wildcard name pattern
     |

    class Not7ValuesSOAParseError(ZoneParseError)
     |  7 fields were not supplied as required by RFC 1035
     |
     |  JSONRPC Error:      -10
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |                      'num_soa_rdata_values' - number of SOA fields
given
     |

    class OnlyOneLoneWildcardValid(ZoneSearchPatternError)
     |  Only one lone '*' or '%' for zone search pattern is valid
     |
     |  JSONRPC Error:      -98
     |  JSONRPC data keys:  'search_pattern'    - Zone search pattern
     |

    class PidFileAccessError(RestoreNamedDbError)
     |  PID file format error
     |
     |  JSONRPCError:       -111
     |  JSONRPC data keys:  pid_file - PID file name
     |                      exception - Value Error Exception
     |

    class PidFileValueError(RestoreNamedDbError)
```

```
      |  PID file format error
      |
      |  JSONRPCError:       -110
      |  JSONRPC data keys:  pid_file - PID file name
      |                      exception - Value Error Exception
      |

    class PrivilegeNeeded(ZoneParseError)
      |  Privilege is needed to set this RR field
      |
      |  JSONRPC Error:      JSONRPC_INTERNAL_ERROR
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class RRNoTypeGiven(ZoneParseError)
      |  RR has no type given.
      |
      |  JSONRPC Error:      -97
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class RdataParseError(ZoneParseError)
      |  Somewhere in the rdata processing (probably within dnspython)
      |  sense could not be made of the data
      |
      |  JSONRPC Error:      -24
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |                      'rdata_thing'       - given invalid RDATA
      |

    class ReferenceDoesNotExist(DMSError)
      |  Reference does not exist.
      |
      |  JSONRPC Error:      -65
      |  JSONRPC data keys:  'reference'   - reference code
      |

    class ReferenceExists(DMSError)
```

```
 |  Trying to create a reference that already exists.
 |
 |  JSONRPC Error:      -64
 |  JSONRPC data keys:  'reference'   - reference code
 |

class ReferenceFormatError(DMSError)
 |  A reference can only consist of the characters '-_a-zA-Z0-9.@',
 |  and must start with a letter or numeral.  It also must be less than
 |  1024 characters long.
 |
 |  JSONRPC Error:      -82
 |  JSONRPC data keys:  'reference' - reference name
 |                      'error'     - error message
 |

class ReferenceMustBeGiven(ZoneSearchPatternError)
 |  When giving a zone search pattern, a reference must be given
 |
 |  JSONRPC Error:      -99
 |  JSONRPC data keys:  'search_pattern'    - Zone search pattern
 |

class ReferenceStillUsed(DMSError)
 |  Reference is still in use
 |
 |  JSONRPC Error:      -66
 |  JSONRPC data keys:  'reference'   - reference code
 |

class ReplicaSgExists(DMSError)
 |  A master SG already exists
 |
 |  JSONRPC Error:      -114
 |  JSONRPC data keys:  'sg_name'          - SG name
 |                      'replica_sg_name'   - master SG name
 |

class RestoreNamedDbError(DMSError)
 |  Subclass for Errors relating to restore_named_db DR functionality
 |

class ReverseNamesNotAccepted(InvalidDomainName)
 |  Reverse domain names are generated from CIDR network names.
 |
 |  JSONRPC Error:      -91
 |  JSONRPC data keys:  'name'      - domain name
 |

class RrQueryDomainError(DMSError)
 |  For query an RR, domain cannot start with '.'
 |
 |  JSONRPC Error:      -81
```

```
     |  JSONRPC data keys:  'name'      - domain name
     |

    class RropNotSupported(ZoneParseError)
     |  Our zone parser does not support the RROP: RR flag in edit mode
     |
     |  JSONRPC Error:      -85
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class SOASerialArithmeticError(SOASerialError)
     |  SOA Serial Arithmetic Error.  Possibly due to memory corruption.
     |
     |  JSONRPC Error: -3
     |  JSONRPC data keys: 'name'  - domain name
     |

    class SOASerialCandidateIgnored(SOASerialError)
     |  Proposed SOA Serial Candidate ignored.
     |
     |  JSONRPC Error: -118
     |  JSONRPC data keys: 'name'  - domain name
     |

    class SOASerialError(DMSError)
     |  Ancestor for all SOA Serial arithmetic errors
     |

    class SOASerialMustBeInteger(ZoneParseError)
     |  SOA serial number must be an integer value.
     |
     |  JSONRPC Error:      -11
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |                      'soa_serial_thing'  - thing given as SOA serial
no.
     |

    class SOASerialOcclusionError(SOASerialError)
     |  SOA Serial Occlusion Error.  SOA serial as recorded in database is
     |  maximum of current SOA serial value in master DNS server.
     |
     |  JSONRPC Error: -115
     |
```

```
    class SOASerialPublishedError(SOASerialError)
     |  SOA Serial Published Error.  SOA serial number update is the same
as
     |  published value in database.
     |
     |  JSONRPC Error: -116
     |

    class SOASerialRangeError(SOASerialError)
     |  SOA Serial Number is out of range must be > 0 and <= 2**32 -1.
     |
     |  JSONRPC Error: -120
     |  JSONRPC data keys: 'name'  - domain name
     |

    class SOASerialTypeError(SOASerialError)
     |  SOA Serial Number must be an integer.
     |
     |  JSONRPC Error: -121
     |  JSONRPC data keys: 'name'  - domain name
     |

    class SecTagPermissionDenied(DMSError)
     |  Operations on security tags can only be done with Admin privilege
     |
     |  JSONRPC Error:      -46
     |  JSONRPC data keys:  'sectag_label'   - security tag label
     |

    class ServerAddressExists(ServerError)
     |  Server with the given address exists
     |
     |  JSONRPC Error:      -77
     |  JSONRPC data keys:  'address'      - server address
     |

    class ServerError(DMSError)
     |  Ancestor class for server functions, saves code.
     |

    class ServerExists(ServerError)
     |  Server already exists
     |
     |  JSONRPC Error:      -74
     |  JSONRPC data keys:  'server_name'      - server name
     |

    class ServerNotDisabled(ServerError)
     |  Server must be disabled for operation to proeceed.
     |
     |  JSONRPC Error:      -78
     |  JSONRPC data keys:  'server_name'      - server name
```

```
 |

class ServerSmFailure(DMSError)
 |  Server SM Failure - synchronous execution of the Server SM
 |  was not successful.
 |
 |  JSONRPC Error: -79
 |  JSONRPC data keys:  'server_name'    - server name
 |  JSONRPC data keys:  'event_message' - Event Message
 |  JSONRPC data keys:  'event_results' - Event results object
 |

class SgExists(DMSError)
 |  For a DMI, SG already exists
 |
 |  JSONRPC Error:      -72
 |  JSONRPC data keys:  'sg_name'      - SG name
 |

class SgMultipleResults(DMSError)
 |  For a DMI, search for one requested SG found multiple entities
 |
 |  JSONRPC Error:      -58
 |  JSONRPC data keys:  'sg_name'      - SG name
 |

class SgNameRequired(DMSError)
 |  SG Name is required for this configuration parameter
 |
 |  JSONRPC Error:      -63
 |  JSONRPC data keys:  'config_key'      - config parameter key
 |

class SgStillHasServers(SgStillUsed)
 |  For a DMI, attempted deletion, SG still has servers
 |
 |  JSONRPC Error:      -95
 |  JSONRPC data keys:  'sg_name'      - SG name
 |

class SgStillHasZones(SgStillUsed)
 |  For a DMI, attempted deletion, SG still has zones
 |
 |  JSONRPC Error:      -73
 |  JSONRPC data keys:  'sg_name'      - SG name
 |

class SgStillUsed(DMSError)
 |  Container class for SG Deleteion errors
 |

class TickleEditLockFailure(ZoneSmFailure)
 |  Can't tickle the edit lock timeout event due to an incorrect
```

```
      |  edit_lock_token
      |
      |  JSONRPC Error:      -35
      |  JSONRPC data keys:  'name'          - domain name
      |  JSONRPC data keys:  'event_message' - Timeout Event Message
      |  JSONRPC data keys:  'event_results' - Event results object
      |

    class TtlInWrongPlace(DirectiveParseError)
      |  $TTL not at top of zone file.
      |
      |  JSONRPC Error:      -53
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class TtlParseError(DirectiveParseError)
      |  Hostname parse error while parsing zone file.
      |
      |  JSONRPC Error:      -52
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class TtlZiParseError(ZiParseError)
      |  Zi related ttl value error.
      |
      |  JSONRPC Error:      -49
      |  JSONRPC data keys:  'name'      - domain name
      |                      'zi_field'  - ZI field where error found
      |                      'value'     - value in error
      |

    class UnhandledClassError(ZoneParseError)
      |  Unhandled class for record - we only ever do IN
      |
      |  JSONRPC Error:      -8
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |
```

```
    class UnhandledTypeError(ZoneParseError)
     |  RR type is one we don't handle.
     |
     |  JSONRPC Error:      -9
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class UpdateError(DMSError)
     |  Error during update of zone
     |
     |  JSONRPC Error: -2
     |  JSONRPC data keys: 'name'  - domain name
     |

    class UpdateTypeAlreadyQueued(DMSError)
     |  An update of the given type is already queued for the zone
     |
     |  JSONRPC Error:      -86
     |  JSONRPC data keys:  'name'          - domain name
     |                      'update_type'   - update type
     |

    class UpdateTypeNotSupported(ZoneParseError)
     |  Our zone parser does not support the $UPDATE_TYPE statement in edit
mode
     |
     |  JSONRPC Error:      -84
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class UpdateTypeRequired(DMSError)
     |  An update_type is required parameter for an incremental update.
     |
     |  JSONRPC Error:      -87
     |  JSONRPC data keys:  'name'          - domain name
     |

    class UpdateZoneFailure(ZoneSmFailure)
     |  Can't update zone as it is locked.
     |
     |  JSONRPC Error:      -35
     |  JSONRPC data keys:  'name'          - domain name
     |  JSONRPC data keys:  'event_message' - Timeout Event Message
```

```
 |  JSONRPC data keys:  'event_results' - Event results object
 |  JSONRPC data keys:  'zi_id' -         ID of saved ZI
 |

class ZiIdAdjStringSyntaxError(ZiIdSyntaxError)
 |  ZI id adjustment sub string has invalid syntax.
 |
 |  JSONRPC Error:      -101
 |  JSONRPC data keys:  'zi_id'     - given zi_id string
 |

class ZiIdDdMmYyyySyntaxError(ZiIdSyntaxError)
 |  ZI id sub string has an invalid DD/MM/YYYY date.
 |
 |  JSONRPC Error:      -106
 |  JSONRPC data keys:  'zi_id'     - given zi_id string
 |

class ZiIdDdSlashMmSyntaxError(ZiIdSyntaxError)
 |  ZI id sub string has an invalid DD/MM date.
 |
 |  JSONRPC Error:      -105
 |  JSONRPC data keys:  'zi_id'     - given zi_id string
 |

class ZiIdHhMmSyntaxError(ZiIdSyntaxError)
 |  ZI id sub string has an invalid HH:MM time.
 |
 |  JSONRPC Error:      -104
 |  JSONRPC data keys:  'zi_id'     - given zi_id string
 |

class ZiIdIsoDateSyntaxError(ZiIdSyntaxError)
 |  ZI id sub string has an invalid YYYY-MM-DD date.
 |
 |  JSONRPC Error:      -107
 |  JSONRPC data keys:  'zi_id'     - given zi_id string
 |

class ZiIdSyntaxError(DMSError)
 |  ZI id given has invalid syntax.
 |
 |  JSONRPC Error:      -100
 |  JSONRPC data keys:  'zi_id'     - given zi_id string
 |

class ZiIdTimeAmountSyntaxError(ZiIdSyntaxError)
 |  ZI id sub string has invalid time amount.
 |
 |  JSONRPC Error:      -103
 |  JSONRPC data keys:  'zi_id'     - given zi_id string
 |
```

```
    class ZiIdTimeUnitSyntaxError(ZiIdSyntaxError)
     |  ZI id sub string has an invalid time unit specifier.
     |
     |  JSONRPC Error:      -102
     |  JSONRPC data keys:  'zi_id'     - given zi_id string
     |

    class ZiInUse(DMSError)
     |  Trying to delete a zi that is currently published.
     |
     |  JSONRPC Error:      -38
     |  JSONRPC data keys:  'name'        - domain name
     |

    class ZiNotFound(ZoneNotFound)
     |  For a DMI, can't find the requested zi.
     |
     |  JSONRPC Error:      -30
     |  JSONRPC data keys:  'name'      - domain name
     |  JSONRPC data keys:  'zi_id'     - Zone Instance ID
     |                                    (can be None/Null)
     |

    class ZiParseError(DMSError)
     |  Zi related SOA/TTL data error.
     |
     |  JSONRPC Error:      JSONRPC_INTERNAL_ERROR
     |  JSONRPC data keys:  'name'      - domain name
     |                      'zi_field'  - ZI field where error found
     |                      'value'     - value in error
     |

    class ZiTextParseError(DMSError)
     |  Parse Error.  The zone file text input as zi_text
     |  must be of a valid format
     |
     |  JSONRPC Error:      -126
     |  JSONRPC data keys:  'parse_error'        - error message
     |                      'name'               - domain name
     |                      'lineno'             - line number
     |                      'col'                - column
     |                      'marked_iinput_line' - input line with marked
error
     |

    class ZoneAdminPrivilegeNeeded(DMSError)
     |  DMI has not been assigned the privilege required to edit this zone.
     |
     |  JSONRPC Error:      -127
     |  JSONRPC data keys:  'name'          - domain name
     |

    class ZoneAlreadyHasSOARecord(ZoneError)
```

```
      |  Zone already has an SOA record.
      |
      |  JSONRPC Error:      -15
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class ZoneBeingCreated(DMSError)
      |  A zone in the creation process can not be deleted or undeleted
      |
      |  JSONRPC Error:      -62
      |  JSONRPC data keys:  'name'          - domain name
      |  JSONRPC data keys:  'event_message' - event message
      |  JSONRPC data keys:  'event_results' - event results object
      |

    class ZoneCNAMEExists(ZoneError)
      |  Zone already has a CNAME using this label.
      |
      |  JSONRPC Error:      -18
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class ZoneCNAMELabelExists(ZoneError)
      |  Zone already has a CNAME using this label.
      |
      |  JSONRPC Error:      -19
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class ZoneCfgItem(DMSError)

    class ZoneCfgItemNotFound(ZoneCfgItem)
      |  An item with the given key name can not be found in the zone_cfg
table
      |
      |  JSONRPC Error:      -61
      |  JSONRPC data keys:  'key'      - item key name
      |
```

```
    class ZoneCfgItemValueError(ZoneCfgItem)
     |  An item with the given key name can not be interpolated from its
string
     |
     |  This can happen for string -> boolean conversions
     |
     |  JSONRPC Error:      -71
     |  JSONRPC data keys:  'key'      - item key name
     |  JSONRPC data keys:  'value'    - item value
     |

    class ZoneCheckIntegrityNoGlue(ZoneError)
     |  Record in zone does not have valid in zone glue
     |
     |  JSONRPC Error:      -21
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class ZoneDisabled(DMSError)
     |  Zone disabled. Can't do operation.
     |
     |  JSONRPC Error:      -88
     |  JSONRPC data keys:  'name'      - domain name
     |

    class ZoneError(ZoneParseError)
     |  Zone related resource record error.
     |
     |  JSONRPC Error:      JSONRPC_INTERNAL_ERROR
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class ZoneExists(DMSError)
     |  Trying to create a zone that already exists
     |
     |  JSONRPC Error:      -36
     |  JSONRPC data keys:  'name'        - domain name
     |

    class ZoneFileWriteError(RestoreNamedDbError)
     |  Can't write zone file
     |
```

```
      |  JSONRPCError:       -112
      |  JSONRPC data keys:  'name'  - domain name
      |                      'internal_error' - error that occured
      |

    class ZoneFilesStillExist(ZoneSmFailure)
      |  Can't destroy/nuke a zone as its zone files still exist
      |
      |  JSONRPC Error:      -70
      |  JSONRPC data keys:  'name'          - domain name
      |  JSONRPC data keys:  'event_message' - Event Message
      |  JSONRPC data keys:  'event_results' - Event results object
      |

    class ZoneHasNoNSRecord(ZoneError)
      |  Zone has No NS records.
      |
      |  JSONRPC Error:      -23
      |  JSONRPC data keys:  'name'      - domain name
      |                      'rr_data'   - RR data from zi, Not RDATA!
      |                      'rr_groups_index'   - index into rr_groups
array.
      |                      'rrs_index'         - index of RR in rrs of
      |                                            rr_groups
      |

    class ZoneHasNoSOARecord(DMSError)
      |  Zone has No SOA record.
      |
      |  JSONRPC Error:      -22
      |  JSONRPC data keys:  'name'      - domain name
      |

    class ZoneHasNoZi(DMSError)
      |  For a Zone, no ZI has no candidate or published ZI
      |
      |  JSONRPC Error: - 92
      |  JSONRPC data keys: 'name'  - domain name
      |

    class ZoneMultipleResults(DMSError)
      |  For a DMI, search for one requested zone found multiple entities
      |
      |  JSONRPC Error:      -57
      |  JSONRPC data keys:  'name'      - domain name
      |

    class ZoneNameUndefined(DMSError)
      |  Name of the Zone can not be determined.
      |
      |  JSONRPC Error:      -47
      |  JSONRPC data keys:  'file_name'   - file name being loaded.
      |
```

```
class ZoneNoAltSgForSwap(DMSError)
 |  Zone idoes not have an alternate SG for swapping
 |
 |  JSONRPC Error:      -123
 |  JSONRPC data keys:  'name'          - domain name
 |

class ZoneNotDeleted(DMSError)
 |  Trying to destroy a zone that is active
 |
 |  JSONRPC Error:      -37
 |  JSONRPC data keys:  'name'        - domain name
 |

class ZoneNotDisabled(DMSError)
 |  Zone disabled. Can't do operation.
 |
 |  JSONRPC Error:      -94
 |  JSONRPC data keys:  'name'      - domain name
 |

class ZoneNotDnssecEnabled(DMSError)
 |  Zone is not DNSSEC enabled.
 |
 |  JSONRPC Error:      -60
 |  JSONRPC data keys:  'name'          - domain name
 |

class ZoneNotFound(DMSError)
 |  For a DMI, can't find the requested zone.
 |
 |  JSONRPC Error:      -28
 |  JSONRPC data keys:  'name'      - domain name
 |

class ZoneNotFoundByZoneId(ZoneNotFound)
 |  For a DMI, can't find the requested zone.
 |
 |  JSONRPC Error:      -29
 |  JSONRPC data keys:  'zone_id'      - Zone ID
 |

class ZoneNotPublished(DMSError)
 |  Zone Not Published.  Can't poke DNS server.
 |
 |  JSONRPC Error: -117
 |  JSONRPC data keys: 'name'  - domain name
 |

class ZoneParseError(DMSError)
 |  Parent class for zi RR errors
 |
```

```
     |  JSONRPC Error:      JSONRPC_INTERNAL_ERROR
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class ZoneSOARecordNotAtApex(ZoneError)
     |  Zone already has an SOA record.
     |
     |  JSONRPC Error:      -16
     |  JSONRPC data keys:  'name'      - domain name
     |                      'rr_data'   - RR data from zi, Not RDATA!
     |                      'rr_groups_index'   - index into rr_groups
array.
     |                      'rrs_index'         - index of RR in rrs of
     |                                            rr_groups
     |

    class ZoneSearchPatternError(DMSError)
     |  Given zone search pattern is invalid
     |

    class ZoneSecTagConfigError(ZoneSecTagDoesNotExist)
     |  Zone security tag for DMS server does not exist.
     |
     |  JSONRPC Error:      -42
     |  JSONRPC data keys:  'sectag_label'   - security tag label
     |

    class ZoneSecTagDoesNotExist(DMSError)
     |  Zone security tag does not exist.
     |
     |  JSONRPC Error:      -41
     |  JSONRPC data keys:  'sectag_label'   - security tag label
     |

    class ZoneSecTagExists(DMSError)
     |  Trying to create a security tag that already exists.
     |
     |  JSONRPC Error:      -40
     |  JSONRPC data keys:  'sectag_label'   - security tag label
     |

    class ZoneSecTagStillUsed(DMSError)
     |  Zone security tag is still in use
     |
     |  JSONRPC Error:      -43
     |  JSONRPC data keys:  'sectag_label'   - security tag label
     |
```

```
class ZoneSmFailure(DMSError)
 |  Zone SM Failure - synchronous execution of the Zone SM
 |  was not successful.
 |
 |  JSONRPC Error: -80
 |  JSONRPC data keys:  'name'          - domain name
 |  JSONRPC data keys:  'event_message' - Event Message
 |  JSONRPC data keys:  'event_results' - Event results object
 |

class ZoneTTLNotSetError(DMSError)
 |  The zone ttl needs to be set in the RR database row
 |
 |  JSONRPC Error: -1
 |  JSONRPC data keys: 'rr_id'  - Resource Record ID
```

```
        |
```

System Documentation

Auto Reverse PTRs

With the advent of IPv6, reverse zone management is recommended to only contain real machine names, instead of macro generated addresses. The IPv6 reverse format for ip6.arpa. is sub-domain per nibble, which is real squirrel brain food, and will literally drive you dotty.

When provisioning a new network block, the first thing to do is to set up a reverse zone, with inc_updates enabled. The auto PTR update operations are considered part of the incremental updates mechanism.

```
zone_tool > create_zone 2001:db8:1:2:3::/52 inc_updates
zone_tool . create_zone -r nuts@SQUIRREL-NZ my-new.zone.org
zone_tool > edit_zone my-new.zone.org
.
.
.
@   IN    AAAA 2001:db8:1:2:3::1
.
.
.
zone_tool > show_zone 2001:db8:1:2:3::1
.
.
.
;! REF:nuts@SQUIRREL-NZ
0.0.0.0.0.0.0.0.0.0.0.1  IN PTR my-new.zone.org.
.
.
.
zone_tool >
```

All zones instances saved have the auto PTR data collected from AAAA records, and upon request from IPv4 A records . A privilege evaluation based on the forward zones reference string, the reference string against any existing reverse and reverse zone and sectag is then performed to determine if the reverse zone is allowed to be updated.

Only the 'Admin' sectag can give auto reverse like tha above automatically. Customer DMIs dont' have that privilege . Customer DMIs can only update a PTR record if the reference for the customer domain and the PTR record match, or if the domain reference matches the reverse zone's reference.

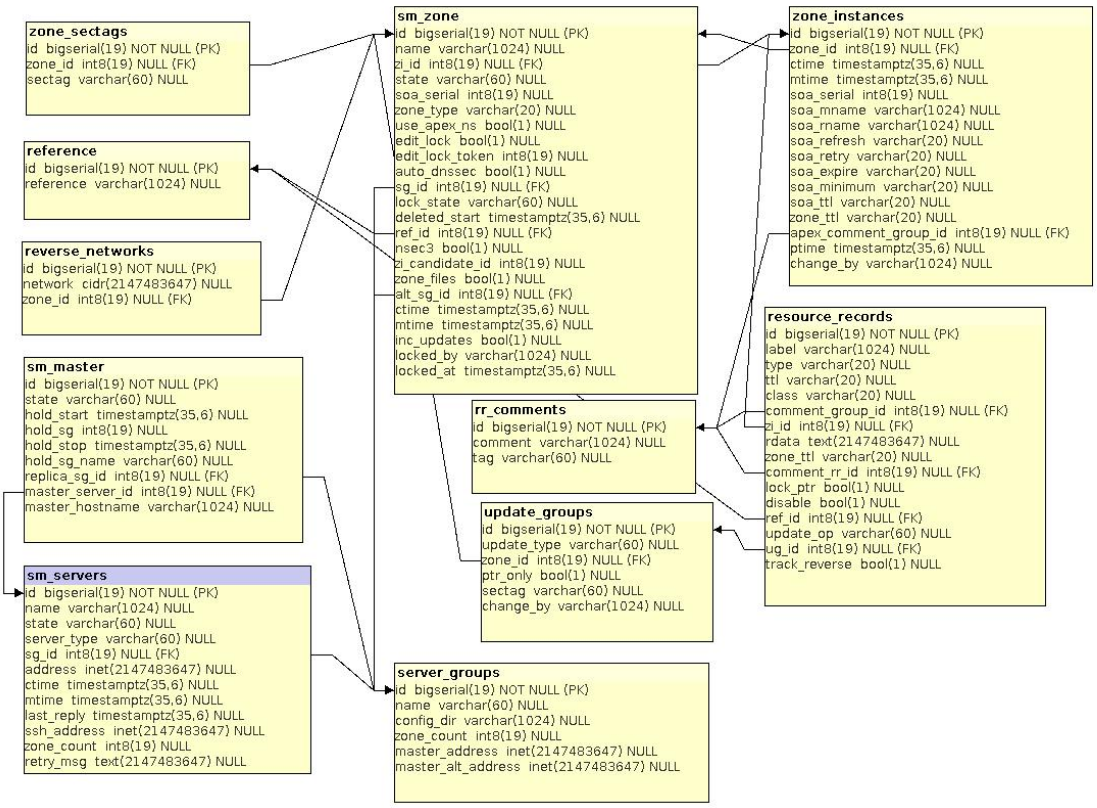
The reverse zone tables are looked up in CIDR fashion when it comes to auto-reverse PTR operations. Thus we can delegate a more specific reverse block than our IPv6 /32 to a customers XDSL connection, and they can manage the reverse space themselves from their forward domain. This CIDR functionality is also used from zone_tool to save on brain mashing. You can give an IP address to show/edit a reverse zone, and an exact network block for zone creation and zone config setting, and zone deletion.

If a reverse record does not exist yet, and the privilege checks pass, the PTR record is created from the first AAAA record seen. The FORCEREV RR flag can be used for create a PTR for an IPv4 A record. Also, a FORCREV RR flag may be used to force an update to a new value, and a TRACKREV RR flag to make the PTR record to update on each forward zone save.

The RR flags used on PTR records are REF: to delegate management operations, and LOCKPTR to prevent a reverse record from ever being altered by the auto reverse mechanism. This can be set on our main mail server and DNS server PTR records.

DB Schema - Zone Diagram

Zone Table structure.jpg



DMS Central config

The DMS central configuration stored in the DMS database is displayed with the show_config command. It contains various 'default_' settings used during zone creation, DMS vacuuming, and apex record creation and refreshment.

```
zone_tool > show_config
        auto_dnssec:       false
        default_ref:       anathoth
        default_ssg:       anathoth-external
        default_stype:     bind9
        edit_lock:         true
        event_max_age:     120.0
        inc_updates:       false
        nsec3:             true
        soa_expire:        7d
        soa_minimum:       10m
        soa_mname:         ns1.anathoth.net. (anathoth-external)
        soa_mname:         ns1.internal.anathoth.net. (anathoth-internal)
        soa_refresh:       600
        soa_retry:         600
        soa_rname:         matthewgrant5.gmail.com.
        syslog_max_age:    120.0
        use_apex_ns:       true
        zi_max_age:        90.0
        zi_max_num:        25
        zone_del_age:      0.0
        zone_del_pare_age: 90.0
        zone_ttl:          24h
zone_tool >
```

The entries shown above can be set using the set_config command:

```
zone_tool > help set_config
        Set DB Configuration settings:

        set_config [-g sg_name] [sg_name] <key> <value>
        sg_name         sg_name for soa_mname
        Key can be one of:
        default_sg       Default Server Group
        default_ref      Default reference for created zones
        auto_dnssec      Boolean defaults used during initial zone creation
        edit_lock
        inc_updates
        nsec3
        use_apex_ns
        soa_mname        Used during initial zone creation
        soa_rname
        soa_refresh
        soa_retry
        soa_expire
        soa_minimum
        soa_ttl
        zone_ttl
        event_max_age    Defaults used when vacuuming deleted zones,
        syslog_max_age   events, syslog messages and old zis.
        zi_max_num
        zi_max_age
        zone_del_age         0 turns off deleted zone aging via vacuum_*
        zone_del_pare_age    0 turns off zone zi paring to 1 via vacuum_*

zone_tool > set_config auto_dnssec false
zone_tool >
```

DNSSEC and DMS

The DMS backend supports DNSSEC operations. If the keys are created for a domain on the disk via the dns-createzonekeys command as root, then a DNSSEC enabled domain can either be created, or if the zone exists it can be DNSSEC enabled. The keys created by the shell script are NSEC3 and NSEC capable, of algorithm NSEC3RSASHA1. The keys created have no expiration date. The DS material is in the /var/lib/bind/ds directory, with the keys in /var/lib/bind/keys directory.

```
shalom-ext: -root- [/var/lib/bind/keys]
# dns-createzonekeys omg.blah.net
+ rm -f '/var/lib/bind/keys/Komg.blah.net.*'
+ rm -f /var/lib/bind/ds/omg.blah.net
+ dnssec-keygen -3 -r /dev/random -f KSK -K /var/lib/bind/keys omg.blah.net
Generating key pair.................................+++
.....................+++
```

```
Komg.blah.net.+007+56550
+ dnssec-dsfromkey -2 /var/lib/bind/keys/Komg.blah.net.+007+56550.key
+ dnssec-keygen -3 -r /dev/random -K /var/lib/bind/keys omg.blah.net
Generating key pair.......................++++++
.........................++++++
Komg.blah.net.+007+30722
+ set +x

shalom-ext: -root- [/var/lib/bind/keys]
# zone_tool
Welcome to the zone_tool program.  Type help or ? to list commands.

zone_tool > create_zone omg.blah.net auto_dnssec edit_lock
.
.
.
zone_tool > show_zonesm omg.blah.net
        name:            omg.blah.net.
        alt_ssg_name:    None
        auto_dnssec:     True
        ctime:           Fri Sep  7 15:32:18 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_by:       None
        mtime:           Fri Sep  7 15:32:18 2012
        nsec3:           True
        reference:       anathoth
        soa_serial:      2012090705
        ssg_name:        anathoth-external
        state:           PUBLISHED
        use_apex_ns:     True
        zi_candidate_id: 102843
        zi_id:           102843
        zone_id:         101523
        zone_type:       DynDNSZoneSM

        zi_id:           102843
        change_by:       root@shalom-ext.internal.anathoth.net/Admin
        ctime:           Fri Sep  7 15:32:18 2012
        mtime:           Fri Sep  7 15:32:40 2012
        ptime:           Fri Sep  7 15:32:40 2012
        soa_expire:      7d
        soa_minimum:     10m
        soa_mname:       ns1.anathoth.net.
        soa_refresh:     600
        soa_retry:       600
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012090705
        soa_ttl:         None
        zone_id:         101523
```

```
        zone_ttl:        24h
zone_tool >
zone_tool > show_zone omg.blah.net
$TTL 24h
$ORIGIN omg.blah.net.

;
; Zone:       omg.blah.net.
; Reference:  anathoth
; change_by:  root@shalom-ext.internal.anathoth.net/Admin
; zi_id:      102843
; zi_ctime:   Fri Sep  7 15:32:18 2012
; zi_mtime:   Fri Sep  7 15:32:40 2012
; zi_ptime:   Fri Sep  7 15:32:40 2012
;


;| Apex resource records for omg.blah.net.
;!REF:anathoth
@                       IN      SOA             ( ns1.anathoth.net. ;Master
NS
                                                matthewgrant5.gmail.com.
;RP email
                                                2012090705   ;Serial
yyyymmddnn
                                                600          ;Refresh
                                                600          ;Retry
                                                604800       ;Expire
                                                600
;Minimum/Ncache
                                                )
                        IN      NS              ns3.anathoth.net.
                        IN      NS              ns2.anathoth.net.
                        IN      NS              ns1.anathoth.net.

zone_tool >

shalom-ext: -root- [/var/lib/bind/ds]
# cat omg.blah.net
omg.blah.net. IN DS 56550 7 2
2B2BFD4C06AF0B2CC3CFC6995555B05A1562A62D4A73C59148AFE582 CACEAE6F
```

NSEC3 non-walkable NX Domain processing can be selected for the zone by using the zone_tool set_zone command.

```
zone_tool > set_zone omg.blah.net nsec3
zone_tool >
```

which takes the same flags as the create_zone command.

> For NSEC3, the secondary servers mast be capable of calculating NX responses to queries. Bind 9.6.3 ESRV and up are compatible with this requirement.

If the DS material for a domain needs to be recreated, use the dns-recreateds command.

```
shalom-ext: -root- [/var/lib/bind/keys]
# dns-recreateds anathoth.net
+ dnssec-dsfromkey -2 /var/lib/bind/keys/Kanathoth.net.+007+57318.key
+ set +x
```

etckeeper and /etc on Replica and Master Servers

etckeeper and ssh

etckeeper is a tool to keep the contents of /etc in a git VCS. When combinded with ssh and the appropriate git remote setup with cron, it allows the /etc of the other machine in the master/replica DR pair to be kept on its alternate, and vice-versa. This protects aginst the /etc on the master being updated, the replica being missed, and then finding that things aren't working on the replica when the master dies, with no record of the updates needed to machine configuration. For information on etckeeper usage, see /usr/share/doc/etckeeper/README.gz Example for diffing/checking out /etc/racoon/racoon-tool.conf from other machine:

```
dms3-master:/etc# git diff dms4-dr/master racoon/racoon-tool.conf
dms3-master:/etc# git checkout dms4-dr/master racoon/racoon-tool.conf
dms3-master:/etc# git checkout HEAD racoon/racoon-tool.conf
```

> Be careful with the git checkout operation as missing the trailing path argument will cause /etc to be changed to that of the other machine.

You can diff parts of etc against the other machine:

```
dms-master-chc: -root- [/etc]
# git diff dms-master-akl/master bind

diff --git a/bind/rndc.conf b/bind/rndc.conf
index 0b0d600..22f85c8 100644
--- a/bind/rndc.conf
+++ b/bind/rndc.conf
@@ -8,20 +8,14 @@ options {
 };


-server dms-master-chc {
-       addresses { 2406:3e00:1001:1::2 port 953; };
-       key "remote-key";
-};
-
-
 server dms-master-akl {
        addresses { 2406:1e00:1001:1::2 port 953; };
        key "remote-key";
 };


-server dms-ns2-chc {
-       addresses { 2406:3e00:1001:2::2 port 953; };
+server dms-master-chc {
+       addresses { 2406:3e00:1001:1::2 port 953; };
        key "remote-key";
 };

@@ -32,3 +26,9 @@ server dms-ns1-akl {
 };


+server dms-ns2-chc {
+       addresses { 2406:3e00:1001:2::2 port 953; };
+       key "remote-key";
+};
+
+
```

as well as checking out a directory/file so that it is same as on the other machine

```
dms-master-chc: -root- [/etc]
# git checkout dms-master-akl/master bind/named.conf.options
```

Use 'etckeeper commit ' to commit to the repository, and 'git fetch' on the other machine.

```
dms-master-chc: -root- [/etc]
# etckeeper commit
.
.
.
dms-master-akl: -root- [/etc]
# git fetch dms-master-chc
```

Note that in the /etc git repository, the revision trees for both of the machines are not connected together history wise, but they are very similar due to being installed from the same Linux distribution and package lists. So you have 2 completely separate trees in each git repository.

Event Queue Inspection

The zone_tool event queue inspection commands are:

show_event <event-id>	Given an event ID, show the contents of the event
ls_pending_events [-v]	List all pending events
ls_failed_events [-v] [n]	List last n failed events, by default 25
ls_events [-v] [n]	List n events in queue in reverse order, by default 25
fail_event <event-id>	Manually an event

The -v switch is for verbose output.

Listing Pending Events

This is probably the most used. By default it returns all the pending events. This is useful when you want to check when the MasterSM is going to time out of HOLD, a server is to be reconfigured or a Zone is to be reconfigured in named.conf. The example below displays the usual ServerSMCheckServer events, and then the result in the event queue of zone_tool reconfig_all.

```
zone_tool > ls_pending_events
ServerSMCheckServer       shalom-dr                    Tue Nov 13 14:39:07 2012
ServerSMCheckServer       dns-slave0                   Tue Nov 13 14:44:15 2012
ServerSMCheckServer       dns-slave1                   Tue Nov 13 14:46:13 2012
ServerSMCheckServer       en-gedi-auth                 Tue Nov 13 14:43:04 2012
ServerSMCheckServer       shalom-ext                   Tue Nov 13 14:43:19 2012
zone_tool > reconfig_all
zone_tool > ls_pending_events -v
ServerSMCheckServer  896877                      NEW
  dns-slave0
  Tue Nov 13 14:36:02 2012  Tue Nov 13 14:44:15 2012  --
ServerSMCheckServer  896878                      NEW
  dns-slave1
  Tue Nov 13 14:36:20 2012  Tue Nov 13 14:46:13 2012  --
ServerSMCheckServer  896879                      NEW
  en-gedi-auth
  Tue Nov 13 14:37:00 2012  Tue Nov 13 14:43:04 2012  --
ServerSMCheckServer  896880                      NEW
  shalom-ext
  Tue Nov 13 14:37:49 2012  Tue Nov 13 14:43:19 2012  --
ServerSMCheckServer  896881                      NEW
  shalom-dr
  Tue Nov 13 14:39:09 2012  Tue Nov 13 14:45:18 2012  --
MasterSMAllReconfig  896882                      NEW

  Tue Nov 13 14:39:15 2012  Tue Nov 13 14:39:15 2012  --
zone_tool > ls_pending_events
ServerSMCheckServer       dns-slave0                   Tue Nov 13 14:44:15 2012
ServerSMCheckServer       dns-slave1                   Tue Nov 13 14:46:13 2012
ServerSMCheckServer       en-gedi-auth                 Tue Nov 13 14:43:04 2012
ServerSMCheckServer       shalom-ext                   Tue Nov 13 14:43:19 2012
ServerSMCheckServer       shalom-dr                    Tue Nov 13 14:45:18 2012
ServerSMConfigChange      shalom                       Tue Nov 13 14:39:23 2012
ServerSMConfigChange      en-gedi-auth                 Tue Nov 13 14:39:23 2012
ServerSMConfigChange      shalom-ext                   Tue Nov 13 14:39:23 2012
ServerSMConfigChange      shalom-dr                    Tue Nov 13 14:39:23 2012
ServerSMConfigChange      dns-slave0                   Tue Nov 13 14:39:23 2012
ServerSMConfigChange      dns-slave1                   Tue Nov 13 14:39:23 2012
MasterSMHoldTimeout                                    Tue Nov 13 14:49:18 2012
zone_tool >
```

Listing failed events

Here is an example of how to list failed events

Importing, Nuking, and Destroying Zones

Zone_tool has a a number of commands to enable the import of single zones, single ZIs and multiple zones from a single directory(load_zone, load_zone_zi, load_zones), as well as the nuke_zones command to remove incorrect imports from the DMS DB.

The zone_tool destroy_zone command can also be used to erase a single zone by its zone_id once it is deleted,

and its named zone files flushed by the net24dmd daemon. The vacuum_zone/vacuum_all commands will immediately DELETE the zones from the DB once the named zone files and configuration are cleaned up on the DMS master server.

> When importing zones, take care to specify the correct SECTAG inc_updates, and SG group for the zones.

The -f flag can be given for use from scripts with these zone_tool commands. The load_zones command will continue to import other zones if a parse error occurs for some of the zones. Error messages for each incorrect zone file will be printed to stdout, including line number, for correction and subsequent import. Zone_tool load_zones expects that the file name for the zone is the actual exact domain name. Thus:

```
grantma@dms3-master:~/net24-test-zones/registerdirect$ zone_tool load_zones
-f ****   Zone file '30seconds.co.nz': zone '30seconds.co.nz.' already
      exists.
***   Zone file 'abelsoftware.com': zone 'abelsoftware.com.' already
      exists.
***   Zone file 'aegmeansbusiness.co.nz': zone
      'aegmeansbusiness.co.nz.' already exists.
***   Zone file 'aipshop.co.nz': zone 'aipshop.co.nz.' already exists.
***   Zone file 'altaine.com': zone 'altaine.com.' already exists.
***   Zone file 'alternative-essentials.co.nz': zone 'alternative-
      essentials.co.nz.' already exists.
***   Zone file 'anameg.co.nz': zone 'anameg.co.nz.' already exists.
^Cgrantma@dms3-master:~/net24-test-zones/registerdirect$
grantma@dms3-master:~/net24-test-zones/registerdirect$ ^C
grantma@dms3-master:~/net24-test-zones/registerdirect$ zone_tool nuke_zones
-f *
```

The zone import code uses the same file parsing code that is used from the zone_tool edit_zone command. The net24.dms.zone_parser module is a complete enough implementation of the RFC 1035 and RFC 1034 zone file formats. All the RFC $ directives are parsed, and unsupported directives rejected with clear error messages.

The JSON RPC mechanism also supports zone file format import operations, along with zone file based editing operations. This is so that the plesk zone file editor can be easily integrated with the system.

Master and DR Replica Setup

The master servers for the DMS are primarily a pair of servers, with the master of the pair operating named as the DNS master server for all zones, running net24dmd, and running the master PostgresQL DMS database. The replica of the pair operates a BIND DNS slave server of all the zones, and PostgresQL in hot standby mode.

Upon failure of the DMS master, the replica's named can be restarted as the master DNS server, PostgresQL promoted to full master, and net24dmd started.

The above fail-over is achieved by manually running the dms_promote_replica script on the replica server. The master server can be taken off line using the dms_master_down script (the dms_master_up script reverses this operation). After the master has been taken down, it can be restarted as the master replica by using the dms_start_as_replica script.

The settings for the dms_ scripts are in the /etc/net24/dr-settings.sh file. The full list of the DMS fail over scripts is:

DMS DR script	Function
dms_master_down	Manually take master down
dms_master_up	Manually bring master up from above operation
dms_promote_replica	Promote replica server to master
dms_start_as_replica	Restart/start a machine as a replica
dms_pg_basebackup	Component operation - create PG host standby replica data base
dms_write_recovery_conf	Component operation - create PG recovery.conf file
dms_update_wsgi_dns	Component operation - update DMS failover CNAME record

```
shalom-ext: -root- [/home/grantma]
# dms_master_down
Stopping interface: dummy0.
+ do_dms_drif
+ [ -n dummy0 ]
+ return 0
+ perl -pe s/^(IF_AUTO.*)\s+dummy0(.*$)/\1\2/g -i
/etc/netscript/network.conf
+ perl -pe s/^(IF_AUTO=")dummy0\s+(.*$)/\1\2/g -i
/etc/netscript/network.conf
+ do_dms_wsgi
+ return 1
+ perl -pe s/^([^#].*zone_tool vacuum_all)$/#\1/ -i /etc/cron.d/dms-core
+ set +x
[ ok ] Stopping net24dmd: net24dmd.
[ ok ] Stopping domain name service...: bind9.
+ perl -pe s/^(local7.* :ompgsql:\S+,dms,rsyslog,.*$)/#\1/ -i
/etc/rsyslog.d/pgsql.conf
+ set +x
[ ok ] Stopping enhanced syslogd: rsyslogd.
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Stopping PostgreSQL 9.1 database server: dms.
+ perl -pe s/^NET24DMD_ENABLE=.*$/NET24DMD_ENABLE=false/ -i
/etc/default/net24dmd
+ perl -pe s/^OPTIONS=.*$/OPTIONS="-u bind -c
\/etc\/bind\/named-dr-slave.conf"/ -i /etc/default/bind9

shalom-ext: -root- [/home/grantma]
.
.
.
root@shalom-dr:/home/grantma# dms_promote_replica
+ perl -pe s/^#(\s*local7.* :ompgsql:\S+,dms,rsyslog,.*$)/\1/ -i
/etc/rsyslog.d/pgsql.conf
+ set +x
[ ok ] Stopping enhanced syslogd: rsyslogd.
```

```
[ ok ] Starting enhanced syslogd: rsyslogd.
+ perl -pe s/^NET24DMD_ENABLE=.*$/NET24DMD_ENABLE=true/ -i
/etc/default/net24dmd
+ perl -pe s/^OPTIONS=.*$/OPTIONS="-u bind"/ -i /etc/default/bind9
+ set +x
[....] Stopping domain name service...: bind9waiting for pid 27511 to die
. ok
[ ok ] Starting domain name service...: bind9.
[....] Starting net24dmd: net24dmd***   DB in Read Only mode -
(InternalError) cannot execute INSERT in
      a read-only transaction  'INSERT INTO update_groups (update
 failed!
+ zone_tool write_rndc_conf
+ zone_tool reconfig_all
+ perl -pe s/^#+(.*zone_tool vacuum_all)$/\1/ -i /etc/cron.d/dms-core
+ do_dms_wsgi
+ return 1
+ set +x
+ perl -pe s/^(IF_AUTO=.*)"$/\1 dummy0"/g -i /etc/netscript/network.conf
+ set +x
Configuring interface: dummy0.
root@shalom-dr:/home/grantma#
.
.
.
shalom-ext: -root- [/home/grantma]
# dms_start_as_replica
dms_start_as_replica: Will replicate from 'shalom-dr.anathoth.net'
Operation will destroy all data Proceed? (y/N)y
dms_start_as_replica: replicating from 'shalom-dr.anathoth.net'
341707/341707 kB (100%), 1/1 tablespace
[ ok ] Stopping net24dmd: net24dmd.
+ do_dms_drif
+ [ -n dummy0 ]
+ return 0
+ perl -pe s/^(IF_AUTO.*)\s+dummy0(.*$)/\1\2/g -i
/etc/netscript/network.conf
+ perl -pe s/^(IF_AUTO=")dummy0\s+(.*$)/\1\2/g -i
/etc/netscript/network.conf
+ do_dms_wsgi
+ return 1
+ perl -pe s/^(local7.* :ompgsql:\S+,dms,rsyslog,.*$)/#\1/ -i
/etc/rsyslog.d/pgsql.conf
+ perl -pe s/^([^#].*zone_tool vacuum_all)$/#\1/ -i /etc/cron.d/dms-core
+ perl -pe s/^NET24DMD_ENABLE=.*$/NET24DMD_ENABLE=false/ -i
/etc/default/net24dmd
+ perl -pe s/^OPTIONS=.*$/OPTIONS="-u bind -c
\/etc\/bind\/named-dr-slave.conf"/ -i /etc/default/bind9
+ set +x
[ ok ] Starting PostgreSQL 9.1 database server: dms.
[ ok ] Starting domain name service...: bind9.
```

```
shalom-ext: -root- [/home/grantma]
#
```

At any time, the status of the cluster can be displayed on any of the Master servers (running and replicas) using the zone_tool show_dms_status command.

```
zone_tool > show_dms_status

show_master_status:

        MASTER_SERVER:     shalom-ext

        NAMED master configuration state:

        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        None
        hold_stop:         None
        replica_sg_name:   anathoth-replica
        state:             READY

show_replica_sg:
        sg_name:             anathoth-replica
        config_dir:          /etc/bind/anathoth-master
        master_address:      2001:470:f012:2::2
        master_alt_address:  2001:470:f012:2::3
        replica_sg:          True
        zone_count:          14

        Replica SG named status:
        shalom-dr                    2001:470:f012:2::3

                OK

ls_server:
dns-slave0                   Thu Nov  8 12:04:25 2012                OK
        2001:470:c:110e::2                       111.65.238.10
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dns-slave1                   Thu Nov  8 12:01:58 2012                OK
        2001:470:66:23::2                        111.65.238.11
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
en-gedi-auth                 Thu Nov  8 12:08:05 2012                OK
        fd14:828:ba69:6:5054:ff:fe39:54f9        172.31.12.2
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom                       Thu Nov  8 12:04:12 2012                OK
        fd14:828:ba69:1:21c:f0ff:fefa:f3c0       192.168.110.1
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom-dr                    Thu Nov  8 12:04:46 2012                OK
        2001:470:f012:2::3                       172.31.10.4
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom-ext                   Thu Nov  8 12:04:12 2012                OK
```

```
        2001:470:f012:2::2                      172.31.10.2
        ping: 5 packets transmitted, 5 received, 0.00% packet loss

list_pending_events:
ServerSMCheckServer       dns-slave1                  Thu Nov  8 12:11:25
2012
ServerSMCheckServer       shalom-ext                  Thu Nov  8 12:12:15
2012
ServerSMCheckServer       shalom                      Thu Nov  8 12:13:47
2012
ServerSMCheckServer       dns-slave0                  Thu Nov  8 12:14:23
2012
ServerSMCheckServer       shalom-dr                   Thu Nov  8 12:14:19
2012
ServerSMCheckServer       en-gedi-auth                Thu Nov  8 12:15:58
2012
```

```
zone_tool >
```

Master server SM, and Reconfiguration

At its core, DMS has a master server state machine (MasterSM) which is used to drive the ServerSM state machines via each SG. The master status can be shown via show_master_status:

```
zone_tool > show_master_status

        MASTER_SERVER:     shalom-ext

        NAMED master configuration state:

        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        None
        hold_stop:         None
        replica_sg_name:   anathoth-replica
        state:             READY
zone_tool >
```

It drives the 10 minute rndc cycle for the DMS master DNS server and all slaves. It has 2 states, HOLD and READY.

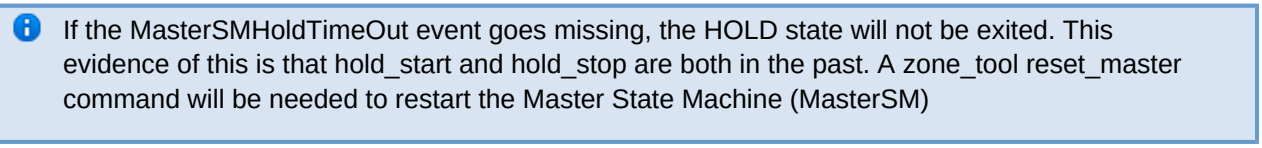
```
zone_tool > show_master_status

        MASTER_SERVER:     shalom-ext

        NAMED master configuration state:

        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        Thu Nov  8 12:58:37 2012
        hold_stop:         Thu Nov  8 13:08:37 2012
        replica_sg_name:   anathoth-replica
        state:             HOLD
zone_tool > ls_pending_events
ServerSMCheckServer       dns-slave0                Thu Nov  8 12:59:17
2012
ServerSMCheckServer       shalom                    Thu Nov  8 13:00:57
2012
ServerSMCheckServer       en-gedi-auth              Thu Nov  8 13:01:26
2012
ServerSMCheckServer       shalom-dr                 Thu Nov  8 13:02:29
2012
ServerSMCheckServer       dns-slave1                Thu Nov  8 13:02:06
2012
ServerSMCheckServer       shalom-ext                Thu Nov  8 13:04:29
2012
ServerSMConfigChange      shalom                    Thu Nov  8 12:58:42
2012
ServerSMConfigChange      en-gedi-auth              Thu Nov  8 12:58:42
2012
ServerSMConfigChange      shalom-ext                Thu Nov  8 12:58:42
2012
ServerSMConfigChange      shalom-dr                 Thu Nov  8 12:58:42
2012
ServerSMConfigChange      dns-slave1                Thu Nov  8 12:58:42
2012
ServerSMConfigChange      dns-slave0                Thu Nov  8 12:58:42
2012
MasterSMHoldTimeout                                 Thu Nov  8 13:08:37
2012
zone_tool >
```

During the hold state, any reconfiguration requests will be honoured for either all SGs, or partially for one SG. That is what is tracked via the hold_sg and hold_sg_name fields displayed above. The hold_start and hold_stop fields are time stamps for start and end of a HOLD period. After a HOLD, and during READY any configuration change for named will immediately happen, followed by the HOLD rndc wait period.

The reconfig_all command reconfigures ALL SGs, replicas and slaves. reconfig_master, only the master DNS server, reconfig_replica_sg, the replica SG group consisting of the DMS master server and all replicas. And last but

not least reconfig_sg <sg-name> reconfigures one SG. These command configure all the name server configuration for the DNS servers involved in the DNS server network. For example for ISC BIND named, this is the named.conf contents.

The following is log of a zone_tool terminal session demonstrating the reconfig commands. Note how the hold_sg changes as one SG, replica SG, being individually reconfigured, and full reconfiguration. The scope of the reconfigure is stored if the Master SM is in HOLD, escalated if needed, and then performed after the MasterSMHoldTimeout event exits the HOLD state.



If the MasterSMHoldTimeOut event goes missing, the HOLD state will not be exited. This evidence of this is that hold_start and hold_stop are both in the past. A zone_tool reset_master command will be needed to restart the Master State Machine (MasterSM)

```
zone_tool > help reconfig_master

        Reconfigure master DNS server: reconfig_master

        Reconfigures the master DNS server via 'rndc reconfig'

zone_tool > help reconfig_replica_sg

        Reconfigure the Replica SG's DNS servers:

        reconfig_replica_sg

        Rsyncs DNSSEC key material to all DR replicas, and reconfigure all
the
        DR replica named processes.

zone_tool > reconfig_master
zone_tool > show_master_status

        MASTER_SERVER:     shalom-ext

        NAMED master configuration state:

        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      None
        hold_start:        Thu Nov  8 12:11:22 2012
        hold_stop:         Thu Nov  8 12:21:22 2012
        replica_sg_name:   anathoth-replica
        state:             HOLD
zone_tool > reconfig_replica_sg
zone_tool > show_master_status

        MASTER_SERVER:     shalom-ext

        NAMED master configuration state:

        hold_sg:           14
        hold_sg_name:      anathoth-replica
```

```
        hold_start:        Thu Nov  8 12:11:22 2012
        hold_stop:         Thu Nov  8 12:21:22 2012
        replica_sg_name:   anathoth-replica
        state:             HOLD
zone_tool > ls_pending_events
ServerSMCheckServer       shalom-ext                  Thu Nov  8 12:12:15
2012
ServerSMCheckServer       shalom                      Thu Nov  8 12:13:47
2012
ServerSMCheckServer       dns-slave0                  Thu Nov  8 12:14:23
2012
ServerSMCheckServer       shalom-dr                   Thu Nov  8 12:14:19
2012
ServerSMCheckServer       en-gedi-auth                Thu Nov  8 12:15:58
2012
MasterSMHoldTimeout                                   Thu Nov  8 12:21:22
2012
ServerSMCheckServer       dns-slave1                  Thu Nov  8 12:18:38
2012

zone_tool >
.
.
.
zone_tool > reconfig_sg anathoth-internal
zone_tool > ls_pending_events
ServerSMCheckServer       dns-slave1                  Thu Nov  8 12:28:33
2012
ServerSMCheckServer       shalom-ext                  Thu Nov  8 12:26:07
2012
ServerSMCheckServer       dns-slave0                  Thu Nov  8 12:26:11
2012
MasterSMHoldTimeout                                   Thu Nov  8 12:31:26
2012
ServerSMCheckServer       shalom                      Thu Nov  8 12:29:35
2012
ServerSMCheckServer       shalom-dr                   Thu Nov  8 12:31:43
2012
ServerSMCheckServer       en-gedi-auth                Thu Nov  8 12:33:26
2012
zone_tool > show_master_status

        MASTER_SERVER:     shalom-ext

        NAMED master configuration state:

        hold_sg:           8
        hold_sg_name:      anathoth-internal
        hold_start:        Thu Nov  8 12:21:26 2012
        hold_stop:         Thu Nov  8 12:31:26 2012
        replica_sg_name:   anathoth-replica
        state:             HOLD
zone_tool >
```

```
.
.
.
zone_tool > show_master_status

        MASTER_SERVER:     shalom-ext

        NAMED master configuration state:

        hold_sg:           HOLD_SG_NONE
        hold_sg_name:      anathoth-internal
        hold_start:        Thu Nov  8 12:31:32 2012
        hold_stop:         Thu Nov  8 12:41:32 2012
        replica_sg_name:   anathoth-replica
        state:             HOLD
zone_tool > reconfig_all
zone_tool > show_master_status

        MASTER_SERVER:     shalom-ext

        NAMED master configuration state:

        hold_sg:           HOLD_SG_ALL
        hold_sg_name:      None
        hold_start:        Thu Nov  8 12:31:32 2012
        hold_stop:         Thu Nov  8 12:41:32 2012
        replica_sg_name:   anathoth-replica
```

```
        state:              HOLD
zone_tool >
```

Named.conf and Zone Templating

On all the servers in the DMS system, the DNS server configuration is designed to use include files and per zone templates. The master and replica servers are bind9 only, but the DMS is designed to support different types of DNS servers (configured as slave servers) such as nsd3, as well as bind9.

To simplify the authentication for the DNS servers, the network connection between the master and replica/slave servers is encrypted and integrity protected by using IPSEC. This enables the bind9 ACLs to specified by IP address only, simplifying the configuration segments that need to be generated for the DNS replica/slave servers.

The include files are generated on the master, rsynced to all the servers, and then the servers are reconfigured via rndc or by a local daemon on the server stating the rsynced include file. If one of the servers gets compromised, it can be cut off by disabling its IPSEC connection or halting it.

/etc/bind/rsync-config	slaves and replicas	named.conf include segments
/etc/bind/master-config	master	Master named.conf include segment
/etc/net24/server-admin-config/bind9	master	named.conf segments for bind9 slaves. Seperate segments for controls, logging, options and local. zone_tool rsync_server_admin_config distributes these portions out.
/etc/net24/server-config-templates	master	Zone templates for replicas and slaves. See below.
/etc/net24/master-config-templates	master	zone templates for running master named.conf
/etc/net24/config-templates	master	rndc.conf templates for creating /etc/bind/rndc.conf, and TSIG key template for zone_tool tsig_key_generate
/var/lib/net24/dms-sg	master	Per SG include dirs for configuration segments to be rsynced.
/etc/bind/named-dr-replica.conf	replicas	Slave named configuration for replicating running master /var/lib/bind/dynamic zone database. Contains DNSSEC RRSIG and other non-database bind9 master data that should be replicated

/var/lib/net24/dms-sg	master	Per SG include dirs for configuration segments to be rsynced.
/var/lib/bind/dynamic	master and replicas	Named DNS dynamic database. Contains DNS cryptographic data that should be replicated between master servers. Replicated via Replica slave named process.
/var/cache/bind/slave	All slaves	Slave zone cache database.

All the directories listed above for the master should be manually synchronised with the all replicas for reliable fail over.

In all the following templates, the keys used are the ones given in the files. They are of the Python string %/sprintf form '%(key_name)s'

Master named.conf include templates in /etc/net24/master-config-templates are:

auto-dnssec-config.conf	DNSSEC dynamic DNS zone template
dynamic-config.conf	dynamic DNS zone template
server-acl.conf	template for server ACLs
slave-config.conf	Slave DNS zone template - not used
static-config.conf	Static zone template - not used

Server named.conf include templates in /etc/net24/server-config-templates and segments are:

bind9.conf	Bind9 slave zone template
bind9-replica.conf	Bind9 replica zone template

Nsd3 server zone config templates would have 'nsd3' in their name.

Administration server named.conf segments in /etc/net24/server-admin-config/bind9 are:

controls.conf	Controls segment of named.conf. Used to control rndc access
logging.conf	Logging named.conf segment. Configures named to log to local7 facility.
options.conf	Options include segment. Needs to be included as it is better to manually specify listen-on directives on each individual server
rndc-remote.key	Rndc remote key used in /etc/bind/rndc.conf on masters, and in controls.conf above.

The above segments are free form, and can be rearranged. No fields are filled in from net24dmd.

Miscellaneous templates in /etc/net24/config-templates are:

rndc.conf-header	Top of rndc.conf. Contains defailt settings and key includes
rndc.conf-server	Per server rndc.conf template
tsig.key	zone_tool tsig_key_generate TSIG key template

Netscript, Iptables and Filtering Incoming IPSEC

As the servers over seas and across other networks are connected to using IPSEC transport mode to secure the zone traffic for business confidentiality and competition reasons, and for ease of named.conf configuration, the incoming traffic has to be filtered on the DMS master server. The IPSEC Security Policy Database is not stateful, and is just IP address based when in comes to multiple ports being opened, especially many to one in both directions. If one of the slaves is compromised the SPD cannot prevent any one on that host from connecting back to any port on the Master DMS servers. The SPD can only really deal with many to one port type IPSEC relationships.

Netscript-2.4 is one of my tools for managing network configuration and iptables under Debian. It uses iptables-save/iptables-restore with roll back. It is inspired by my experience with programming routers, in terms of interface manipulation and IP filtering. It replaces ifupdown, which is probably fundamentally broken according to the router RFC1812 in terms of interface manipulation and addressing (pseudo-device concept for each address, whereas kernel complies with this.)

The network configuration is in /etc/netscript/network.conf. This file is a actually sourced as a shell script by /sbin/netscript. IP addresses are set in the eth0_IPADDR variable, with IF_AUTO specifying the interfaces brought up on boot.

The netscript command has the following help:

```
dms-chc: -root- [/etc/net24/conf-templates]
# netscript
Usage: netscript start|stop|reload|restart
       netscript ifup|ifdown|ifqos|ifreload
           {eth0|dms0|dms1|gre0|sit0|all}
       netscript compile [-fqh] [-b max-backup-level]
Usage: netscript ipfilter load|clear|fairq|flush|fwd|nofwd|reload|save
                           usebackup [backup-number]
       netscript ipfilter exec
Configure|FORWARD|INPUT|icmpfwd|icmphost|inbrdr|ingress|ingrssfwd|ipfwd|ipl
cl|laptopfw|log|martians|outbrdr|portscan|smb|snmp
                           [chain p1 p2 ...]
Usage: netscript ip6filter load|clear|fairq|flush|fwd|nofwd|reload|save
                           usebackup [backup-number]
       netscript ip6filter exec icmphost|laptopfw|log
                           [chain p1 p2 ...]
```

as well as man pages. The netscript ipfilter/ip6filter verbs are the ones used to save and load/reload the firewall configuration. The iptables files are saved as /etc/netscript/iptables{,.1,.2,.3} and /etc/netscript/ip6tables{,.1,.2,.3}.

The number of roll back files can be altered in the network.conf file. The netscript ipfilter exec creates a chain of the given name, with addresses and networks possibly given as variables in network.conf. ICMP host and router grooming packet chains are there. For ICMPv6 I did:

```
# netscript ip6filter exec icmphost
# netscript ip6filter exec log
```

which created chains I could hook into INPUT to groom ICMP for the host, and at the end of INPUT to log all no-accepted traffic. The log chain has a rate-limiter applied to save on runaway syslog messages.

iptables/ip6tables commands are used directly to configure the kernel Netfilter filters. The iptables -I (insert), -R (replace) arguments take a line number after the filter name. The line numbers can be printed by specifying --line-numbers to iptables -vnL <chain-name>

netscript ipfilter examples are:

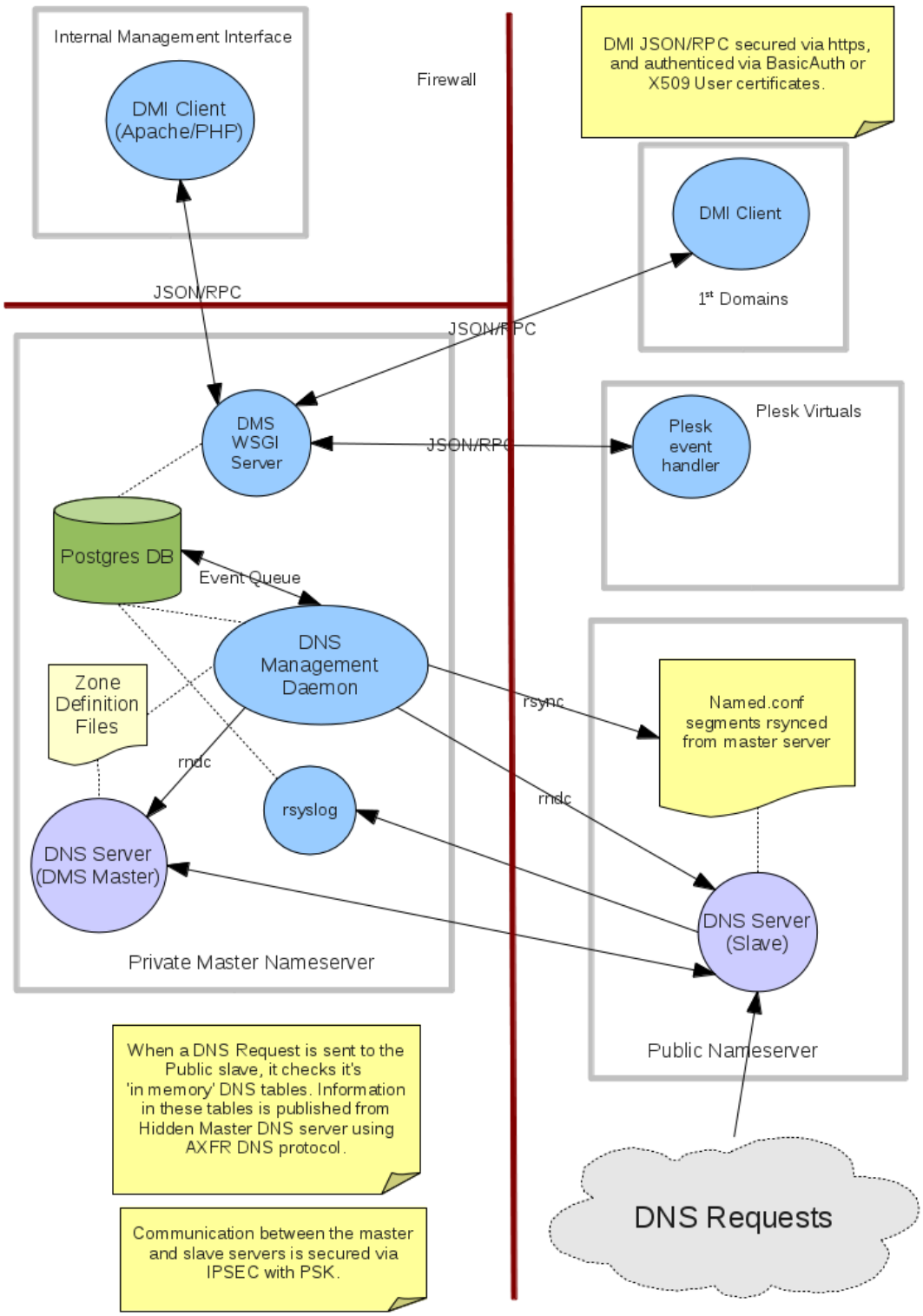
```
dms-chc: -root- [/etc/net24/conf-templates]
# netscript ip6filter save
Saving IPv6 filters...done.


dms-chc: -root- [/etc/net24/conf-templates]
# netscript ipfilter save
Saving IPv4 filters...done.


dms-chc: -root- [/etc/net24/conf-templates]
# netscript ipfilter usebackup 2
Loading IPv4 filters...done.

dms-chc: -root- [/etc/net24/conf-templates]
# netscript ip6filter usebackup 2
Loading IPv6 filters...done.

dms-chc: -root- [/etc/net24/conf-templates]
# netscript ip6filter reload
Loading IPv6 filters...done.

dms-chc: -root- [/etc/net24/conf-templates]
# netscript ipfilter reload
Loading IPv4 filters...done.

dms-chc: -root- [/etc/net24/conf-templates]
#
```

Overview

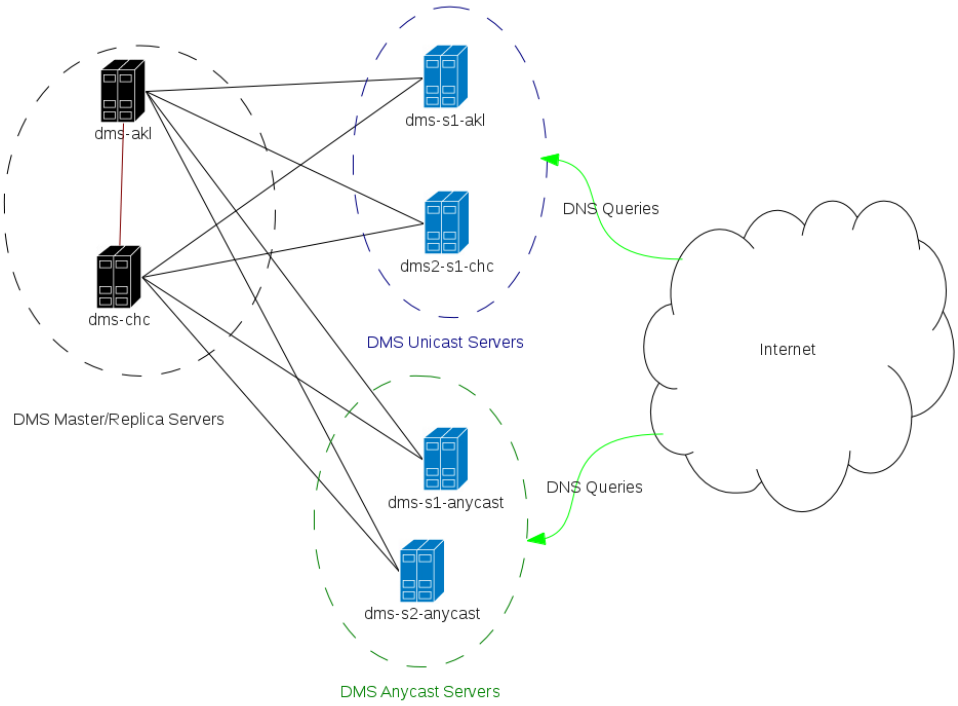
Software Architecture

Architecture Diagram

Only one of the DR replica pairs is shown below for clarity. The standby replica runs as a server of the replica SG group. PostgresQL Replication is also live to the DR replica, carried over the IPSEC connection between the DR master and replica servers. The master/replica servers use iptables/ip6tables to filter access to services carried over IPSEC.



Conceptual Network Diagram



The DMS is designed to support anycast and unicast servers, as per best practise guidlines

DMS Features

- IPv6 fully supported in back end and front end
- IPv6 DNS RRs (AAAA)
- Dynamic DNS configuration of Master server reduces need for reconfig and reload operations.
- DNS RRs supported include SOA NS A AAAA MX PTR TXT SPF RP SSHFP SRV NSAP NAPTR LOC KX IPSECKEY HINFO CERT DS. DNSSEC handled by bind9 master
- Auto DNSSEC via Bind9 dynamic DNS. Bind9 master server auto maintains zone DNSSEC operations records and signing. NSEC3 and NSEC supported. DNSSEC key management on Master server file system pending write of key management module. Key material directory is replicated via DR protocol (rsync) though. DMS is fully enabled to use DNSSEC for securing our core domains.
- Apex resource record (SOA and NS) management across all zones - can be turned off per zone.
- Auto reverse PTR generation
- Customer control of their own automated reverse DNS. Individual PTR records, and complete reverse zones. Useful for business IPv6 and IPv4 blocks. Enables on site use of IP PABX, intranet and email for SMBs on XDSL/Fibre.
- zone_tool command line administrative tool on master servers
- IPSEC secured communications between each of DR master replicas and slaves
- Modular design. For example, Racoon IPSEC can be replaced if needed.
- Multiple Slave DNS server software implementations. NL Netlabs nsd3 can be used as a slave server once backend code is completed, and a simple configuration monitoring/HUP daemon implemented to run on each slave.
- slave server/Server Groups (SG) support. Live migration of zones.
- Private SGs for internal Voyager/NET24 zones.
- Retention of deleted zones in database for aged auto-deletion later.
- Multiple Zone Instances per Zone. Roll forward and roll back changes. Again old ZIs aged for auto deletion above a threshold number.

- Templates used for generating name server configuration includes - master, replicas and slaves.
- Rsync to distribute name server configuration to servers.
- Central distribution of name server configuration segments.
- Hot standby master replica for DR purposes with manually controlled fail over. Includes automatic replica/slave server reconfiguration.
- WSGI JSON RPC over HTTPS API for mulitple front ends
- Security tags to control what front ends can see
- Zone reference metadata to tag the zone with the owner/customer entity ID. Set by DMI when a zone is created. Tag out of table in DB via foreign key for easy reference renaming.
- zone_tool has built in pager support and editor support via standard shell environment variables.
- zone_tool has a configurable restricted shell mode for Help Desk use
- RR Groups and RR comments supported in DB for use in text editor and in Web Admin DMI
- zone_tool has colourised diff support to display changes between different ZIs for a zone
- Vim can be used as zone tool editor, giving DNS colourised Zone file syntax high lighting.

Programming Language

The DMS backend software is written in Python 3.x, which is a good choice given the code base size of 22,000 lines. Python is well suited to larger projects, and fully object oriented, and very clear and systematically defined. Python 3.x was chosen over 2.x for future proofing.

The Python JSON RPC interface is implemented with Apache2 mod_wsgi, with a back end DNS Manager Daemon. zone_tool is a command line shell environment that implements all the functionality of the JSON RPC calls, as well as DMS systems configuration and management functionality. This common code functionality allows the JSON RPC calls to be called from a terminal, where a debugger can be used, for ease of development.

A state machine and event queue design is used, with state and event information recorded in PostgresQL. State machines exist for each:

- DNS zone to track life-cycle state of zone
- Master server configuration
- DNS replica/slave server configuration and reload cycles.

DNS server software

Decided to go forward using ISC Bind 9 as DNSSEC is on the way, and Bind 9 will be the software used to roll this out. Other implementations of DNS software exist, Netlabs NL NSD3 is one, but it looks more suited to a TLD registry and large site/domain use than for DNS Provider use for small zones.

The DNS server state machine classes are designed so that NL Netlabs nsd 3.x can be added latter on as a slave server. This is done achieved by the use of state machine design, object oriented code and modularity.

A Hidden Master DNS architecture is implemented, with a DR replica master server.

Backend Database.

PosgresQL 9.1+. PostgresQL has a significant history of high end functionality including transactions and stored procedures. Replication is also baked in as well.

DMI Server/Clients

1st Domains will communicate with DMS via the WSGI server, along with the Net24 front page. An administrative help desk DNS Management Interface is being implemented. To begin with, the DMS will be administered via

zone_tool by ssh into the Master DMS system.

Network protocols and security

DNS and logging traffic between the slave servers outside Net24 is be secured using IPSEC. Iptables filtering and IPSEC SAs are used to control the traffic that the slave servers accept from the network and Internet. IPSEC SAs exist for zone update and port 53 administrative traffic, and secure that traffic. Ie, DNS Traffic from the Master DNS server will be secured using IPSEC. This keeps all the cryptographic verbiage out of the DNS server configurations, and makes them a lot simpler to generate from templates. IP numbers and acls may need to be inserted in the named.conf files to identify the designation of administrative control and updates from the Master DNS server, but this is a lot easier that having to track of lot of configuration details about TSIG/SIG0 keys for each individual master-slave relationship, and where they are used....

Web UI Framework.

The Web GUI for the DMI will be rendered using ExtJS. Check logic, and business logic will be separated out and not mixed in (as much as possible) with the UI. This is basically a Mode View Controller programming model.

Python WSGI and JSON/RPC over HTTPS

The interface between the DMS Web servers and the DMS server is a Web service. The DMI servers talk JSON/RPC over HTTPS to the DMS running master server. Failover is handled by a CNAME dms-server.failover.vygr.net, which updated by the dms_promote_replica script. The HTTP connections are integrity protected by SSL, and use HTTP basic auth to authenticate to the URL attachment upon the DMS server, which is configured with that DMIs sectag.

The web service is set up as a Python3 WSGI script, running under apache mod_wsgi. WSGI is defined in PEP 3333 The WSGI scripts are configured to run in separate apache2 daemon processes. The hook point URLs are as follows:

/list_zone	Just for listing all zones. For Admin DMI use only
/admin_dms	Admin DMS access point. For systems administrators/NOC
/helpdesk_dms	Helpdesk DMS access point
/1stdomains_dms	1st Domains customer DMS access point.
/net24_dms	Net24 customer DMS access point

WSGI configuration files and directories

All of these files are in the /etc tree to comply with Debian configuration policy.

/etc/net24/dms-wsgi-apache.conf	Apache 2 Debian style include file
/etc/net24/wsgi-scripts	WSGI scripts
/etc/net24/htpasswd-dms	Apache 2 htpasswd file for DMS basic auththentication

The /etc/net24/dms-wsgi-apache.conf contains all the apache2 configuration for the URL hook points and basic authentication in Location directives, as well as defining the mod_wsgi daemon processes for each type of access. The list_zone access for the administrator and helpdesk DMIs uses a separate daemon to the rest as it allocates lots of memory each time it is run, and has to die off quickly to prevent resource consumption. The rest of the RPC

call profile is for only per zone or small listing requests each time, and these are configured into longer running daemon processes, with far more threads.

The RSS memory of the apache daemons should be monitored, as this is where a memory leak problem is most likely to occur in the system. Zone_tool hardly uses any memory, as it is typically not a long running process. Net24dmd uses server side data base cursors and self monitors its RSS usage. If the configured memory_exec_threshold of 250MB is exceeded, it will re exec() itself when once the event queue is empty, thus releasing all the sparsely allocated RSS memory.

Racoon and IPSEC

The IPSEC part of the application stack is done using racoon. Any IPSEC IKE daemon that does the job can be used, and another possibility would be Strongswan.

The reasons for using racoon are:

- compatibility cross platform, FreeBSD, NetBSD and Linux
- protocol compatibility, both IPv6 and IPv4
- historically the native IPSEC IKE and key management tools for NetBSD, FreeBSD, and Linux
- historically native IPv6 and IPv4, as they are from the KAME project.
- maturity
- strongswan later implementation. Originally Linux only. Not as fully tested with IPv6 as racoon/setkey Not too sure about its cross platform capabilities.

On the DMS, racoon-tool is used to wrap racoon and setkey. This is a Perl script that has been ported to FreeBSD, and is software I originally wrote to help make racoon more manageable for VPN network use. It was inspired in part by FreeSWAN, but it has a sensible set of defaults such as hmac-sha1, AES/3DES. Most things you need to configure for run of the mill IPSEC can be done with it. Can do PSK and X509 certificate authentication.

When installing in Debian, choose racoon-tool configuration mode. racoon-tool is made to operate as an Init.d script would. Its configuration is in /etc/racoon/racoon-tool.conf, with a directory /etc/racoon/racoon-tool.conf.d for configuration segments. Once the %default is set in racoon-tool.conf, it only takes a PSK added to /etc/racoon/psk.txt and 5 lines in a configuration segment to get one end of an IPSEC connection configured.

Racoon directories and files:

/etc/racoon	Racoon configuration
/etc/racoon/racoon-tool.conf	Master racoon-tool configuration file
/etc/racoon/racoon-tool.conf.d	Racoon-tool per segment configuration directory
/var/lib/racoon/racoon.conf	racoon-tool generated racoon.conf

The racoon-tool segment configuration directory could be used for a shell script to automate adding a new DNS slave server quickly.

racoon-tool has a full man page and has an entry README.Debian:

```
racoon-tool
-----------
racoon-tool is back.  It is a management script that simplifies looking
after
setkey SPD rules, and basic racoon.conf on a connection oriented basis.  It
now functions in transport mode and tunnel mode, with anonymous VPN
service,
and supports PSK/X509 authentication and IPv6. It should also function on
the
FreeBSD kernel.

Yes, racoon-tool is debian specific, upstream doesn't like it, it does
have all the features when compared to racoon.conf(5). If you're interested
in
using the latest and greatest feature in racoon, and advanced functionality
use /etc/racoon/racoon.conf directly.
```

Here is a sample racoon-tool.conf from dms-chc:

```
#
# Configuration file for racoon-tool
#
# See racoon-tool.conf(5) for details
#

global:
        # How to control the syslog level
        log: notify

connection(%default):
        # source address on this box
        src_ip: 2406:3e00:1001:1::2
        admin_status: disabled
```

with a racoon-tool.conf.d segment dms-akl.conf

```
peer(2406:1e00:1001:1::2):

connection(dms-akl):
        dst_ip: 2406:1e00:1001:1::2
        admin_status: enabled
```

racoon-tool will print out a summary of its sub commands if -h or no arguments are given. It follows the bread crumbs idea so that you can easily find your way through it. Useful sub commands are:

vlist	List all VPN connections
vup	start a VPN connection
vdown	stop a vpn connection
vreload	reload a connection
vmenu	Start VPN menu management mode. Lists all connections in SPD, and you can shut down VPN connections from here.
start	Initialize kernel SPD, and start racoon
stop	Stop racoon, and flush SPD
reload	Reload SPD and reload racoon
restart	Restart everything.

References

A reference is a customer and organisation identity string that is taken by the DMS when a domain is initially created. It is used to track the zones belonging to a customer for listing and for auto reverse record operations. Sample references are as follows:

reference	description
24866@1STDOMAINS-NZ	Customer ID for 1st Domains, made of account number and 1STDOMAINS-NZ
anathoth	Anathoth reference
NET24-NZ	Default Net24 reference
VOYAGERNET-NZ	Default Voyager reference

To the DMS they are just a string, conforming loosely to the format requirements for an email address or domain. They are case insensitive for comparison and searching purposes, but keep their case when saved. In the DB, they are assigned using foreign keys and ref_id - in other words, they are easily renamed. They are enough like an email address so that accountants and customers find them digestible for accounting purposes, but don't finish in a valid domain so that people know they are not straight email addresses.

The zone_tool commands for references are:

create_reference	create a reference
delete_reference	delete a reference
lsref/ls_reference	list references Can take wild cards, and multiple reference arguments
rename_reference	rename a reference
set_zone_reference	Set the reference for a zone to an existing reference.

The create_zone and ls zone_tool commands take a -r argument to either set/create the reference a new zone belongs to, and to show only the zones for a reference. The show_zone command will display the reference as a

REF: RR flag against the SOA record. The load_zone commands also take the -r argument, and recognize the ;!REF: SOA RR flag to set the reference for the zone being loaded.

Security Tags (sectags)

Sectags are used to control what is visible and changeable from a particular DMI front end. They are also used when evaluating whether an auto reverse operation can be carried out or not. Zone_tool and the administrative DMIs use the 'Admin' sectag if it is not set in the database, it is implied. The Sectags are NEVER changeable from a non-administrative DMI, and customer DMIs will never be configured with the WSGI calls to access them or list them. They will only be visible to a non-administrative DMI via the changed_by or locked_by attributes of a zone.

Each zone has a list of sectags attached. The DMIs do not use sectags at all with the WSGI JSON API. A separate administrative only call must be given to return the sectags for a zone. A zone may have multiple sectags, so that it can be accessed from 2 or more customer DMI front ends.

A zone has its initial sectag added when it is created. It is set to the sectag for the DMI it comes from. Authorisation is defined by the exact HTTPS URI for the JSON /RPC over http. The sectag is defined in the Python WSGI script attached to the URI.

As the DMS system is a central point for a lot of administrative information, it has to be more secure than the Web servers in front of it. Because of this the authentication and security code has to be separate from any part of the DMS implementation, widely used, and reviewed. Nginx requires authentication scripts to be written, and at the time of implementation could not define basic authentication in combination with URI Location access lists within its configuration file. With Apache, this code already exists, along with a comprehensive WSGI module that is similar to the Fast CGI model.

The zone_tool commands to do with sectags are as follows:

add_zone_sectag	Add a security tag to a zone
create_sectag	Create a new sectag
delete_sectag	Delete a sectag. It must not be attached to any zones
delete_zone_sectag	Delete a security tag from a zone
replace_zone_sectags	Replace the whole list of sectags for a zone
show_sectags	Show defined sectags
show_zone_sectags	Show the sectag list for a zone

Servers (replica & slave) and Server Groups (SGs)

Servers

In the DMS, servers (replicas and slaves) are defined with a human name, an IP address(v6 or v4), and the SG the server belongs to. Each server has a configuration state machine, which tracks if the the server is up to date for all the zones in its SG.

The commands to use with servers are create_server, delete_server, ls_slave, ls_server, show_server,

show_server_byaddr, enable_server, disable_server, rename_server, move_server_sg, set_server_address, set_server_ssh_address, reset_server, set_server_type. Type 'help <command> at the zone_tool prompt to get a full description of arguments and switches that are usable with these commands.

Other commands used when setting up a server are write_rndc_conf, and rsync_server_admin_config

When a slave is created in DMS, it is disabled for the creation of a new rndc config, and so that initial configuration of the slave via rsync can be sent, and the rsync/rndc hookup can be tested by using the rsync_admin_config command. Details of this are in Adding a DNS Slave to DMS

Some examples:

```
zone_tool > ls_server -jv
dns-slave0                    Thu Nov  8 14:22:55 2012                OK
        2001:470:c:110e::2                      111.65.238.10
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
dns-slave1                    Thu Nov  8 14:19:38 2012                OK
        2001:470:66:23::2                       111.65.238.11
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
en-gedi-auth                  Thu Nov  8 14:16:23 2012                OK
        fd14:828:ba69:6:5054:ff:fe39:54f9       172.31.12.2
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom                        Thu Nov  8 14:18:57 2012                OK
        fd14:828:ba69:1:21c:f0ff:fefa:f3c0      192.168.110.1
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom-dr                     Thu Nov  8 14:23:10 2012                OK
        2001:470:f012:2::3                      172.31.10.4
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
shalom-ext                    Thu Nov  8 14:20:35 2012                OK
        2001:470:f012:2::2                      172.31.10.2
        ping: 5 packets transmitted, 5 received, 0.00% packet loss
zone_tool > show_server en-gedi-auth
        server_name:     en-gedi-auth
        address:         fd14:828:ba69:6:5054:ff:fe39:54f9
        ctime:           Sat Feb 25 18:19:12 2012
        is_master:       False
        last_reply:      Thu Nov  8 14:16:23 2012
        mtime:           None
        server_id:       15
        server_type:     bind9
        sg_id:           8
        sg_name:         anathoth-internal
        ssh_address:     172.31.12.2
        state:           OK
        zone_count:      28
        retry_msg:
          None
zone_tool > disable_server en-gedi-auth
zone_tool > show_server en-gedi-auth
        server_name:     en-gedi-auth
        address:         fd14:828:ba69:6:5054:ff:fe39:54f9
        ctime:           Sat Feb 25 18:19:12 2012
```

```
        is_master:       False
        last_reply:      None
        mtime:           None
        server_id:       15
        server_type:     bind9
        sg_id:           8
        sg_name:         anathoth-internal
        ssh_address:     172.31.12.2
        state:           DISABLED
        zone_count:      28
        retry_msg:
          None
zone_tool > enable_server en-gedi-auth
zone_tool > enable_server en-gedi-auth
***   Event ServerSMEnable(892463) failed - ServerAlreadyEnabled:
      server already enabled
zone_tool > show_server en-gedi-auth
        server_name:     en-gedi-auth
        address:         fd14:828:ba69:6:5054:ff:fe39:54f9
        ctime:           Sat Feb 25 18:19:12 2012
        is_master:       False
        last_reply:      Thu Nov  8 14:24:37 2012
        mtime:           None
        server_id:       15
        server_type:     bind9
        sg_id:           8
        sg_name:         anathoth-internal
        ssh_address:     172.31.12.2
        state:           OK
        zone_count:      28
        retry_msg:
```

```
            None
zone_tool >
```

Server Groups (SGs)

The Server Groups would ideally be of 4 servers each (configured as DNS slaves), handling about 100,000 zones. Once the first SG is full up, a second SG should be started. Rebalancing the SGs by moving zones between them is possible, but the registries would have to have their DNS server settings updated at the same time.

Each zone can have an alternate SG to its primary one, for the purposes of republishing the zone into a private SG consisting of RFC1989 and IPv6 fc::/7 site local addresses (IPv6 equivalent of IPv4 RFC 1918 private addressing).

Each SG has a configured:

- set of Apex name servers, and soa_mname for the purpose of setting the Apex NS records for Zones for which it is the primary SG.
- set of apex_ns records via the edit_apex_ns command, an soa_mname via set_config -g <soa-mname> soa_mname <mname>
- master_address of the master DMS server
- master_alt_address of the main DMS DR replica

```
zone_tool > show_replica_sg
        sg_name:              anathoth-replica
        config_dir:           /etc/bind/anathoth-master
        master_address:       2001:470:f012:2::2
        master_alt_address:   2001:470:f012:2::3
        replica_sg:           True
        zone_count:           14

        Replica SG named status:
        shalom-dr                     2001:470:f012:2::3

                OK
zone_tool >
```

SGs are created using the create_sg command. Other zone_tool commands are ls_sg, reconfig_sg, reconfig_replica_sg, refresh_sg, set_sg_config, set_sg_master_address, set_sg_alt_master_address, set_sg_replica_sg, show_sg, and show_master_sg.

Alternate SG for Zone

In addition to the primary SG that a zone belongs to, it may be a member of an alternate SG for the purposes of migration to a new set of DNS slave servers, or publication in a private SG. A zone never has its Apex records set to those of its alternate SG, the Apex records are always set from a zone's primary SG.

Moving Zones between SGs

Zones may be moved between SGs via the dupe_zone_alt_sg command, one hour later the swap_zone_sg command (to ensure that the zone is flooded to all servers to cover all possible values of NS records during apex changeover), followed by the delete_zone_alt_sg command 24 hours later.

The first two commands add the new SG as an alternate SG, and then swap the zone's primary and alt SGs

```
zone_tool > dupe_zone_alt_sg bad-thing.org

        Set the alternate sg for a zone:

        dupe_zone_alt_sg <zone> <sg-name>

        This is useful if you want to include an external zone on
        in (for example) a private internal SG group behind a firewall.

zone_tool > dupe_zone_alt_sg bad-thing.org anathoth-internal
zone_tool > show_zonesm bad-thing.org
        name:            bad-thing.org.
        alt_sg_name:     anathoth-internal
        auto_dnssec:     False
        ctime:           Thu Aug 23 14:54:07 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_by:       None
        mtime:           Thu Aug 30 09:11:45 2012
        nsec3:           True
        reference:       anathoth
        soa_serial:      2012082300
        sg_name:         anathoth-external
        state:           PUBLISHED
        use_apex_ns:     True
        zi_candidate_id: 102602
        zi_id:           102602
        zone_id:         101449
        zone_type:       DynDNSZoneSM

        zi_id:           102602
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Thu Aug 23 14:54:07 2012
        mtime:           Thu Aug 30 09:40:32 2012
        ptime:           Thu Aug 30 09:40:32 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.anathoth.net.
        soa_refresh:     600
        soa_retry:       600
```

```
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012082300
        soa_ttl:         None
        zone_id:         101449
        zone_ttl:        24h
.
.
1 hour later
.
zone_tool > swap_zone_sg bad-thing.org
***   Do really you wish to do this?
 --y/[N]> y

zone_tool > show_zonesm bad-thing.org
        name:            bad-thing.org.
        alt_sg_name:     anathoth-external
        auto_dnssec:     False
        ctime:           Thu Aug 23 14:54:07 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_by:       None
        mtime:           Thu Aug 30 09:11:45 2012
        nsec3:           True
        reference:       anathoth
        soa_serial:      2012082300
        sg_name:         anathoth-internal
        state:           PUBLISHED
        use_apex_ns:     True
        zi_candidate_id: 102602
        zi_id:           102602
        zone_id:         101449
        zone_type:       DynDNSZoneSM

        zi_id:           102602
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Thu Aug 23 14:54:07 2012
        mtime:           Thu Aug 30 09:40:32 2012
        ptime:           Thu Aug 30 09:40:32 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.anathoth.net.
        soa_refresh:     600
        soa_retry:       600
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012082300
        soa_ttl:         None
        zone_id:         101449
        zone_ttl:        24h
```

```
zone_tool >
```

Followed up 24 hours late by:

```
zone_tool > delete_zone_alt_sg bad-thing.org
zone_tool > show_zonesm bad-thing.org
        name:            bad-thing.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Thu Aug 23 14:54:07 2012
        deleted_start:   None
        edit_lock:       True
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        locked_by:       None
        mtime:           Thu Aug 30 09:11:45 2012
        nsec3:           True
        reference:       anathoth
        soa_serial:      2012082300
        sg_name:         anathoth-internal
        state:           PUBLISHED
        use_apex_ns:     True
        zi_candidate_id: 102602
        zi_id:           102602
        zone_id:         101449
        zone_type:       DynDNSZoneSM

        zi_id:           102602
        change_by:       grantma@shalom-ext.internal.anathoth.net/Admin
        ctime:           Thu Aug 23 14:54:07 2012
        mtime:           Thu Aug 30 09:40:32 2012
        ptime:           Thu Aug 30 09:40:32 2012
        soa_expire:      7d
        soa_minimum:     600
        soa_mname:       ns1.anathoth.net.
        soa_refresh:     600
        soa_retry:       600
        soa_rname:       matthewgrant5.gmail.com.
        soa_serial:      2012082300
        soa_ttl:         None
        zone_id:         101449
        zone_ttl:        24h
zone_tool >
```

Private SGs

Private SGs typically have a specified named.conf template directory, which has templates that restrict query access to its primary Zones. Such SGs are typically used for behind firewall or backend DNS information, which can either consist of RFC 1918 address space, or RFC 4193 IPv6 ULAs (fd13::/16 or fd14::/16). These SGs are usually configured with private master_address and master_alt_address records.

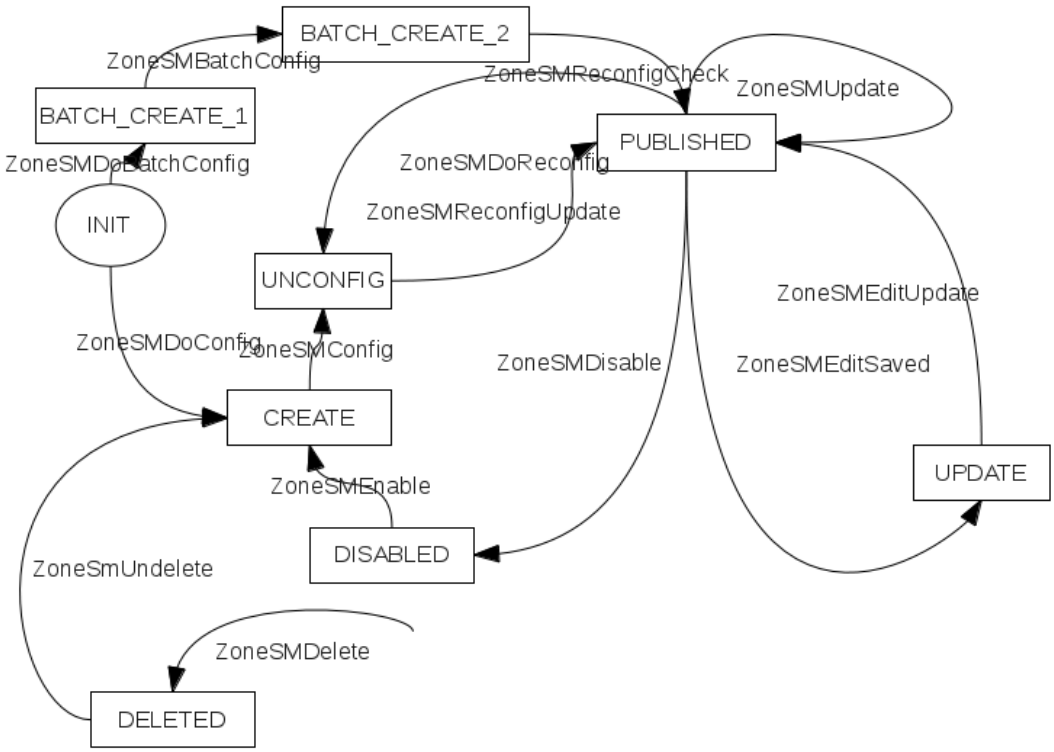
Replica SG

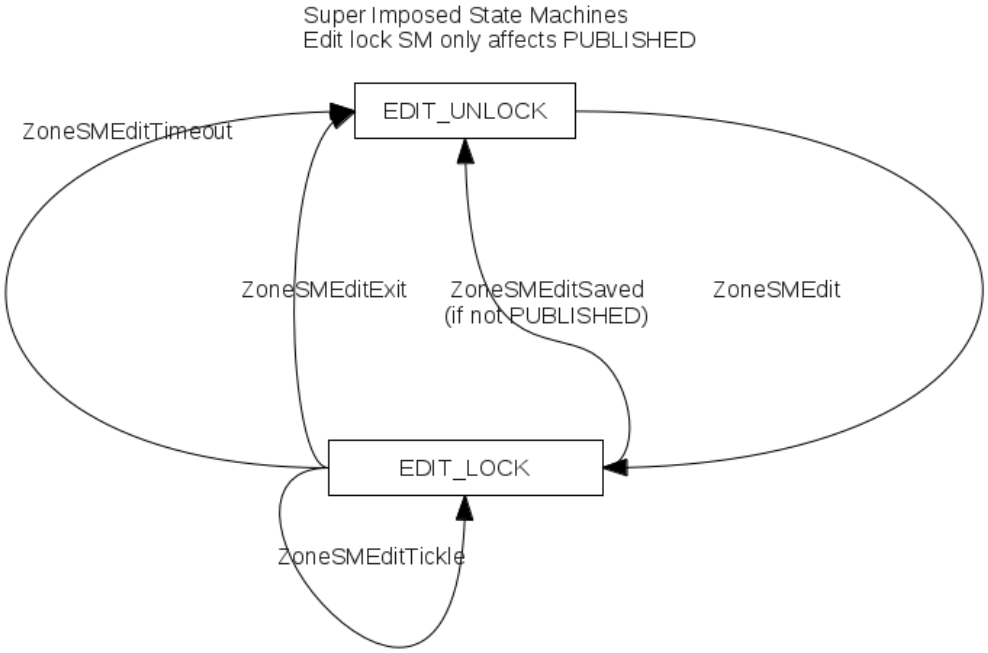
The replica_sg is a special SG covering all zones. It is used to replicate all zones to all DMS replicas (running named as DNS slaves), which save the information into what would be dynamic DNS directory if the replica became the Master server. PostgresQL is typically configured to replicate to each DMS replica alongside the named zone slave replication.

```
zone_tool > show_replica_sg
        sg_name:             anathoth-replica
        config_dir:          /etc/bind/anathoth-master
        master_address:      2001:470:f012:2::2
        master_alt_address:  2001:470:f012:2::3
        replica_sg:          True
        zone_count:          14

        Peer NAMED slave configuration state:
        shalom-dr                    2001:470:f012:2::3

                OK
zone_tool >
```

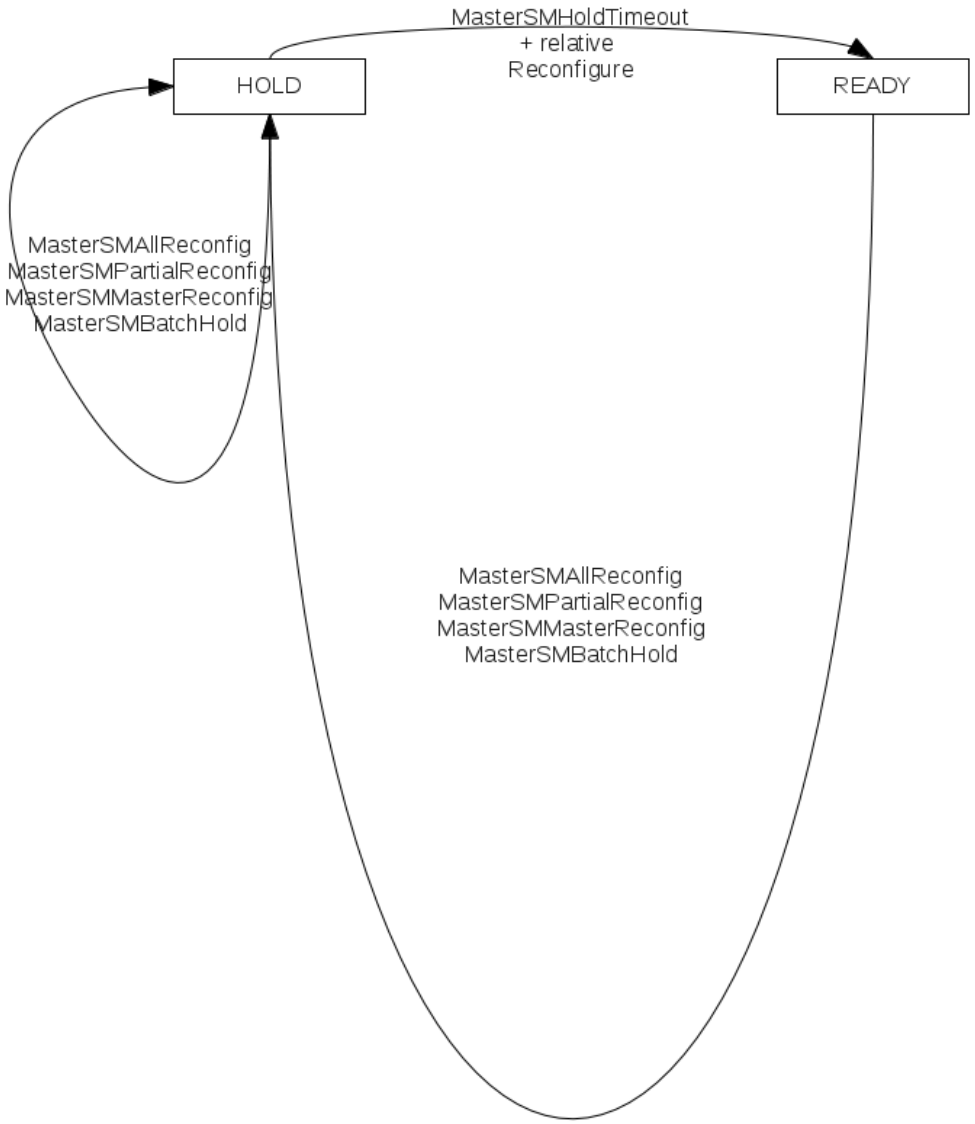
State Machine Diagrams

Zone State Machine

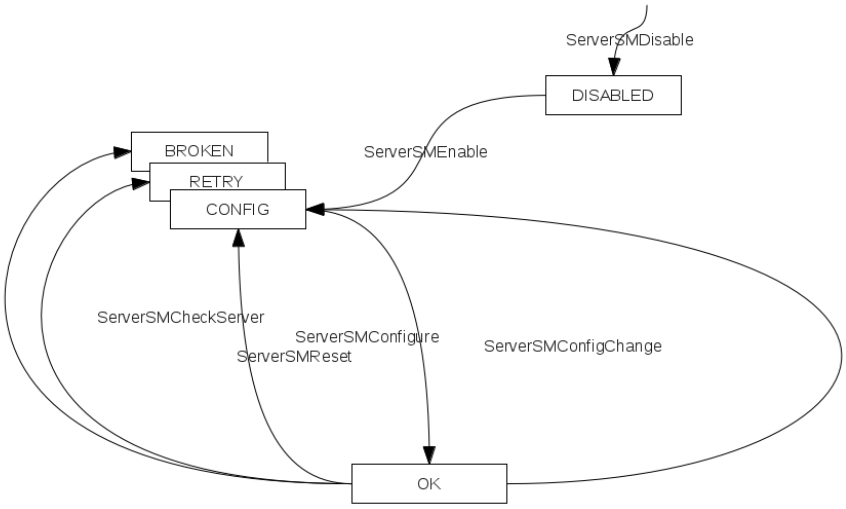




Master State Machine - drives DNS server configuration



Server State Machine



Vacuuming Deleted Zones and Old ZIs

There are 3 things in the DMS database that need daily ageing and cleaning done. They are Zone Instatnces (ZIs), deleted zones, and the syslog table. As it is PostgresQL, they are 'vacuumed'.

The default ages for the history are set in the DMS config table, shown by show_config, and mostly changed by set_config

The commands for vacuuming are:

vacuum_event_queue	Age the event queue
vacuum_zis	Age a zone's unpublished ZIs
vacuum_zones	Age deleted zones out of the DMS
vacuum_syslog	Age syslog messages
vacuum_pare_deleted_zone_zis	Pare deleted zone ZIs down to last published ZI
vacuum_all	Do all the the above, using default ages

The default ages for the above when the dms database was first installed are as follows:

Events	Anything older than event_max_age (120.0) days.
Zone Instances	Anything less than zi_max_age (90.0 days), down to zi_max_num limit (25) thereafter.

Deleted Zones	Anything older than 1000 years (zone_del_age 0.0 days)
Paring Deleted Zones	Anything older than zone_del_pare_age, 90.0 days
Log messages	Anything older than syslog_max_age (120.0) days.

Wrapping, Incrementing and Setting Zone serial numbers

Zone_tool also provides a way to manipulate a zones SOA serial number administratively. The serial numbers can be set, if it is greater than that in named, or incremented, or wrapped via the poke_ commands.

poke_zone_set_serial	Set a zone's SOA serial number, or increment it
poke_zone_wrap_serial	Wrap a zone's SOA serial number.

Zone_tool Shell Notes

Environment Variables

Zone_tool takes the standard environment variables, and uses them if you are not in restricted shell mode. You can use these to effect changes for your own login.

All these can be set globally in /etc/net24/net24.conf, using the appropriate setting names, in the [zone_tool] section

The usage of the variables for the pager and editor is determine by the privilege of the user account. The user must be a member of the sudo, root, or wheel groups by default (groups can be set via net24.conf admin_group_list) for the shell variables to be used.

Editor

Admin Shell Variable	restricted /etc/net24/net24.conf	restricted default	administrator default	description
VISUAL	'editor'	rvim, rnano	system default	zone_tool editor
EDITOR	'editor'	rvim, rnano	system default	zone_tool editor

Pager

zone_tool is aware of stdout not being a terminal, so if you redirect output from it, it will not try to put it through the pager.

Shell Variable	/etc//net24/net24.conf	restricted default	administrator default	description
PAGER (admin only)	pager (restricted)	less	system default	pager program
NET24_PAGER_ARGS	pager_args	-REX		pager arguments

Diff

Shell Variable	/etc/net24/net24.conf	default	description
NET24_DIFF		colordiff, then diff	diff program
NET24_DIFF_ARGS	diff_args	-uNw	diff program args

Tail

Shell variable	/etc/net24/net24.conf	default	description
NET24_TAIL		tail	tail program
NET24_TAIL_ARGS	tail_args	tail arguments	tail program

Technical Notes

Anycast Redux

Anycast Redux

Refer to RFCs 3528 and 4786. Also refer to http://dns.isc.org/f-root/ and http://www.isc.org/solutions/sns-anycast

- ISCs experience is that a combination of anycast and unicast DNS servers is the most reliable.
  Due to routing and load balancing instabilities, the unicast servers are required to fill in the holes of service.
  Like interference fringes from overlapping point wave sources.
- Small length TCP sessions mostly work.
- Keep Local node routing to one AS as mush as possible, due to trouble shooting difficulties.
- Global node routing has to be very stable.
- As soon as a DNS server can't keep content in sync with master, just shut down named, rather than withdrawing route.
- Turn off PMTU on anycast DNS servers
- Don't filter UDP fragments
- Set IPv6 MTU on anycast servers to 1280 bytes to avoid fragmentation.

Remember that DNS resolvers are v. good at handling non-responsive servers.

Also note that anycast address should at least be on a loopback interface.

Good idea for anycast/slave server to have 2 interfaces - one for query traffic, the other for admin and talking to master server. These should be connected to separate interfaces on upstream router. Avoids a DOS overflowing TX queue affecting admin of the server

Building Debian Packages

Prerequisites

1. Debian Wheezy VM
2. git installed
3. apt-get build-dep dms
4. git clone https://git.devel.net.nz/dms/dms.git

Building Package

This needs fleshing out. Just quick notes at the moment.

1. Merge master into deb-package branch

```
wheezy-dev$ git pull
wheezy-dev$ git merge v0.99w
```

2. Update debian/changelog

```
wheezy-dev$ git-dch --commit
```

3. Build package

```
wheezy-dev$ git-buildpackage -rfakeroot -uc -us --git-tag
```

4. Rebuild and index master repository structure

```
reprepro-devel-net include wheezy
/usr/src/debian/build-area/dms_0.99x-3_amd64.changes
```

   Note that you need to give the correct version number in the above command.
5. Rsync to repository server.

```
rsync-deb-repo
```

Master Server Install from Source Repository

> This may be a bit out of date, but useful for details hidden by Debian Packaging

Packages to install

FreeBSD 9.0

Recompile kernel with IPSEC support

> FreeBSD kernel cannot filter outgoing TCP connections over IPSEC in ipfw. These connections invariably get blocked. IPv6 Code hooked into IPFW kernel to allow this to happen is incomplete...

1. Install kernel sources

1. Create IPSEC kernel configuration in /usr/src/sys/amd64/conf

```
# pwd

# cp GENERIC IPSEC
# echo -ne "\n# IPSEC support
options         IPSEC           #IP security
device          crypto          # Cryptography device
options         IPSEC_DEBUG     # IPSEC debugging
# PF extras
device pf
device pflog
device pfsync
" >> IPSEC
# echo -ne "\n# PostgresQL SysV Optimizations
options         SYSVSHM
options         SYSVSEM
options         SYSVMSG
options         SHMMAXPGS=65536
options         SEMMNI=40
options         SEMMNS=240
options         SEMUME=40
options         SEMMNU=120
" >> IPSEC
# cd /usr/src
# make -j 4 KERNCONF=IPSEC buildkernel
# make KERNCONF=IPSEC installkernel
```

freebsd-update will need /boot/kernel.old (GENERIC kernel) mv-ed to /boot GENERIC for its update process to work correctly.

```
# cd /boot
# mv kernel.old GENERIC
```

Also add */boot/kernel/* to *UpdateIfUnmodified* in */etc/freebsd-update.conf* to stop *freebsd-update* moving the custom kernel out of the way.

```
# Paths which start with anything matching an entry in an
UpdateIfUnmodified
# statement will only be updated if the contents of the file have not been
# modified by the user (unless changes are merged; see below).
UpdateIfUnmodified /etc/ /var/ /root/ /.cshrc /.profile /boot/kernel/
```

Base software install

Ports collection will have to be used, and a Net24 ports collection installed. This is because the www/mod_wsgi3 as shipped by BSD ports does not work! We have our own port

Install the following:

portmaster

```
# cd /usr/ports/ports-mgmt/portmaster
# make install
# cd /usr/ports
```

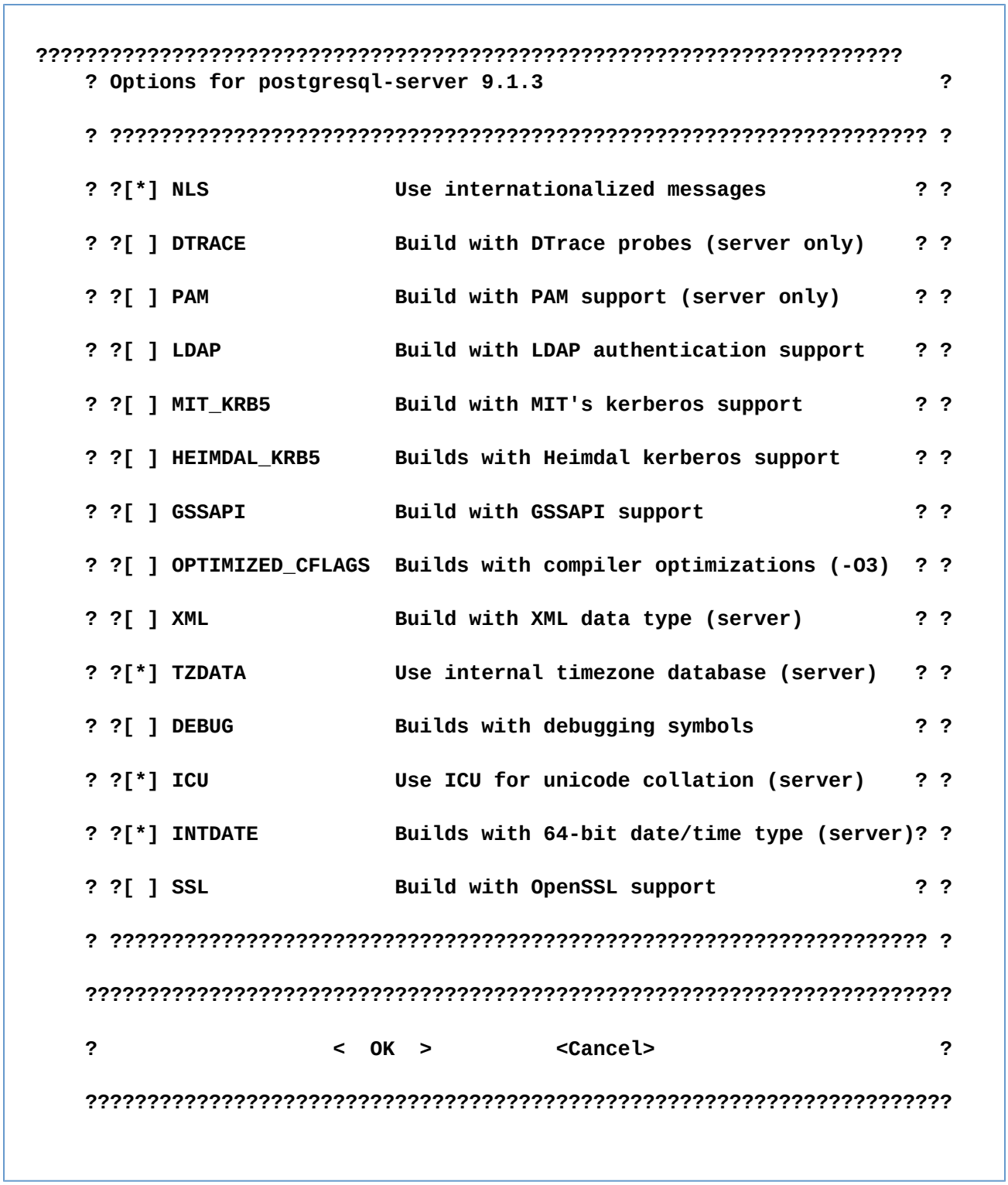
git

```
# cd /usr/ports
# portmaster -PG devel/git
```

PostgresQL 9.1

```
# cd /usr/ports
# portmaster -P databases/postgresql91-server
```

Options for PostgresQL. ICU is for compatibility reasons with Linux PGSQL.



Build ICU library with threading support

```
????????????????????????????????????????????????????????????????????
    ? Options for icu 4.8.1.1_1                                        ?
    ? ?????????????????????????????????????????????????????????????? ?
    ? ?     [*] THREADS  Build thread-safe version of the library      ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?                                                                ? ?
    ? ?????????????????????????????????????????????????????????????? ?
    ????????????????????????????????????????????????????????????????????
    ?                   <  OK  >          <Cancel>                     ?
    ????????????????????????????????????????????????????????????????????
```

Python 3.2

```
portmaster -P lang/python32
```

Python32 config options used. THREADS needed for mod_wsgi:

```
????????????????????????????????????????????????????????????????????????
    ? Options for python32 3.2.2_3                                        ?
    ? ?????????????????????????????????????????????????????????????????? ?
    ? ?     [*] THREADS   Enable thread support                         ? ?
    ? ?     [*] UCS4      Use UCS4 for unicode support                  ? ?
    ? ?     [*] PYMALLOC  Use python's internal malloc                  ? ?
    ? ?     [*] IPV6      Enable IPv6 support                           ? ?
    ? ?     [*] FPECTL    Enable floating point exception handling      ? ?
    ? ?                                                                 ? ?
    ? ?                                                                 ? ?
    ? ?                                                                 ? ?
    ? ?                                                                 ? ?
    ? ?                                                                 ? ?
    ? ?                                                                 ? ?
    ? ?                                                                 ? ?
    ? ?                                                                 ? ?
    ? ?                                                                 ? ?
    ? ?????????????????????????????????????????????????????????????????? ?
    ????????????????????????????????????????????????????????????????????????
    ?                    <  OK  >          <Cancel>                      ?
    ????????????????????????????????????????????????????????????????????????
```

Apache 2.2

```
cd # /usr/ports
# portmaster -P www/apache22
```

Enable THREADS, PGSQL, SQLITE IPV6, leave rest of config as is. Threads is needed for mod_wsgi.

Sqlite, accept default options.

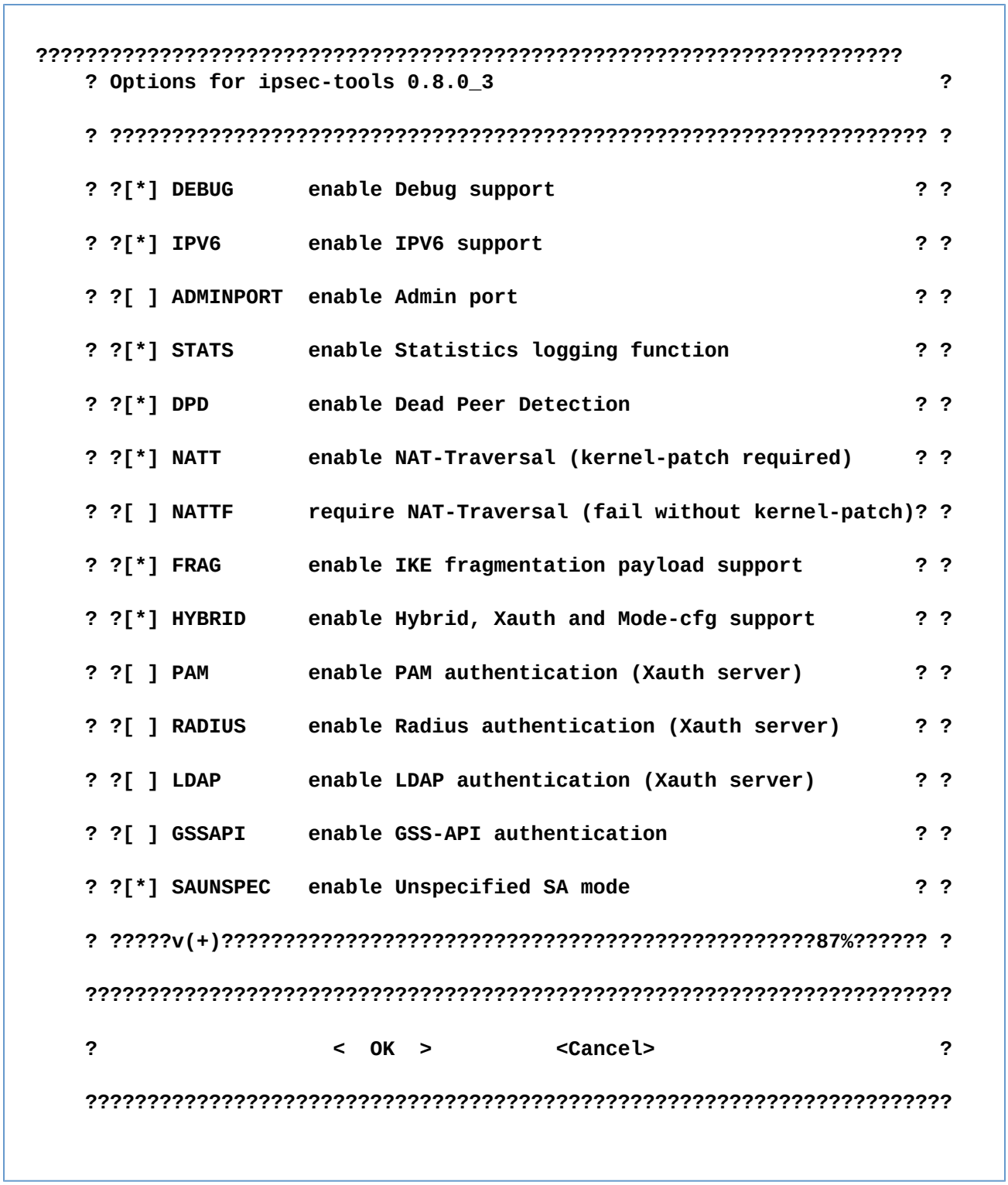
Pcre, accept default options

apache22_enable="YES"

racoon

```
# cd /usr/ports
# portmaster -P security/ipsec-tools
```

ipsec-tools configuration options:



also without IDEA and RC5

curl

```
# cd /usr/ports
# portmaster -PG ftp/curl
```

Install Net24 BSD ports collection. You may need to set up /root/.gitconfig and /root/.netrc

```
# cd /usr/ports
# git clone https://get.devel.net.nz/freebsd-ports/net24.git
# echo  "SUBDIRS += net24" > Makefile.local
# make index
```

racoon-tool

```
# cd /usr/ports
# portmaster -P net24/security/racoon-tool
```

Mod WSGI www/mod_wsgi3

```
# cd /usr/ports
# portmaster -P net24/www/mod_wsgi3
```

Bind 9.9

```
# cd /usr/ports
# portmaster -P dns/bind99
```

Install options:

```
# This file is auto-generated by 'make config'.
# No user-servicable parts inside!
# Options for bind98-base-9.8.0.2
_OPTIONS_READ=bind98-base-9.8.0.2
WITH_SSL=true
WITH_LINKS=true
WITH_XML=true
WITH_IDN=true
WITH_REPLACE_BASE=true
WITH_LARGE_FILE=true
WITH_SIGCHASE=true
WITH_IPV6=true
WITH_THREADS=true
WITHOUT_DLZ_POSTGRESQL=true
WITHOUT_DLZ_MYSQL=true
WITHOUT_DLZ_BDB=true
WITHOUT_DLZ_LDAP=true
WITHOUT_DLZ_FILESYSTEM=true
WITHOUT_DLZ_STUB=true
```

rsyslog

```
# cd /usr/ports
# portmaster -P sysutils/rsyslog5
```

lsof

```
# cd /usr/ports/sysutils/lsof
# make install
```

rsync
Accept config defaults

```
# cd /usr/ports/net/rsync
# make install
```

Debian Linux testing

Install from Debian 6.0 business card CD, Expert install. On downloading release information offered a choice to install Squeeze(stable), Wheezy(Testing) or Sid(unstable). Choose Wheezy and follow prompts.

Just do:

Useful sys admin tools

```
# aptitude install screen vim-nox colordiff ssh lsof strace at less sudo
telnet-ssl dnsutils \
  curl
```

```
# aptitude install git postgresql-9.1 python3.2 python3.2-dev apache2
libapache2-mod-wsgi-py3 \
 rsyslog-pgsql bind9 curl racoon libpq-dev build-essential haveged ntp
```

Haveged is installed as it helps a lot with entropy material for */dev/random* on a Linux VM, by finding randomness in the memory allocation and LDT tables etc.

Choose racoon-tool as the mode of control of the IKE daemon. With rsyslog-pgsql, choose not to configure with dbconfig-common as rsyslog database will be part of dms database.

FreeBSD Configuration

This first lot of configuration is to bring a FreeBSD master up to the same level as basically installed Debian Wheezy Server, with kernel tuning.

Setting up /etc/rc.conf etc for all the services just installed:

Rsyslog 5

Add

```
# Rsyslog 5
syslogd_enable="NO"
rsyslogd_enable="YES"
# avoid warnings about rsyslogd running in compatibility mode
rsyslogd_flags="-c5"
# Compatibility with newsyslog(8) - syslog PID file hard coded.
rsyslogd_pidfile="/var/run/syslog.pid"
```

to /etc/rc.conf. Copy /etc/syslog.conf to /usr/local/etc/rsyslog.conf, and add the following 3 lines to the top for basic functionality:

```
$ModLoad immark.so   # provides --MARK-- message capability
$ModLoad imuxsock.so # provides support for local system logging
$ModLoad imklog.so   # kernel logging
```

Stop syslogd and start rsyslogd:

```
/etc/rc.d/syslogd  stop
/usr/local/etc/rc.d/rsyslogd  start
```

NTP

Add

```
server ntp.net24.net.nz iburst maxpoll 9
server ntp2.net24.net.nz iburst maxpoll 9
```

to /etc/ntp.conf, commenting out the default NTP servers and

```
ntpd_enable="YES"
ntpd_sync_on_start="YES"
```

to /etc/rc.conf and then

```
# /etc/rc.d/ntpd start
```

PostgresQL

Add these settings for PostgresQL. This sets up the machine to be in NZ, just the same as a PostgresQL install under Debian or Ubuntu for cross system compatibility.

/etc/login.conf:
edit /etc/login.conf, and add the following to the bottom of the default section:

```
:charset=UTF-8:\
:lang=en_NZ.UTF-8:
```

Don't forget to add the '\' to the end of the last line to continue, and then run

```
# cap_mkdb /etc/login.conf
```

Initialize the postgresql DB using:

```
# su - pgsql
$ initdb -D /usr/local/pgsql/data --locale=en_NZ.UTF-8 --encoding=UTF8
```

Then add

```
postgresql_enable="YES"
#  # optional
#  postgresql_data="/usr/local/pgsql/data"
#  postgresql_flags="-w -s -m fast"
postgresql_initdb_flags="--encoding=UTF8 --locale=en_NZ.UTF-8"
#  postgresql_class="default"
```

to /etc/rc.conf and

```
/usr/local/etc/postgresql start
```

BIND9

Just add

```
named_enable="YES"
```

to /etc/rc.conf and

```
# /etc/rc.d/named start
```

Apache 2.2

Just add

```
apache22_enable="YES"
```

to /etc/rc.conf and

```
# /usr/local/etc/rc.d/apache22 start
```

IPSEC

Add

```
# Enable racoon via racoon-tool
racoon_tool_enable="YES"
```

to /etc/rc.conf and

```
# /usr/local/etc/rc.d/racoon_tool start
```

Debian Kernel Tuning

This is to increase SYSV shared memory and semaphore limits.

```
# echo "# Shared memory settings for PostgreSQL

# Note that if another program uses shared memory as well, you will have to
# coordinate the size settings between the two.

# Maximum size of shared memory segment in bytes
#kernel.shmmax = 33554432

# Maximum total size of shared memory in pages (normally 4096 bytes)
#kernel.shmall = 2097152


kernel.shmmax = 134217728
kernel.shmall = 32768
" > /etc/sysctl.d/30-postgresql-shm.conf

# service procps start
```

Shared memory settings also have to be adjusted in postgresql.conf

Debian Setup

NTP

Add

```
server ntp.net24.net.nz iburst maxpoll 9
server ntp2.net24.net.nz iburst maxpoll 9
```

to /etc/ntp.conf, commenting out the default Debian NTP server pool servers.

Now Debian and FreeBSD servers are configured similarly from this stage.

Get DMS Master Source code

In you home directory:

```
$ git clone https://git.devel.net.nz/dms/dms-2011.git
```

PostgresQL set up

The procedures here will create PostgresQL setups that are cross compatible. Using Unicode in the DB (above) makes the dumps compatible with Debian's PGSQL install, and creating the pgsql super user on Debian means that access control statements in the dumps work on FreeBSD and Debian.

FreeBSD set up

Record data base passwords for later on, when setting up *net24.conf*

```
# sh ./setup_scripts/create-freebsd-db.sh
```

For development, you may also want to create a super user account for yourself:

```
$ createuser -sEP <your-login>
```

Debian setup

Same as for Free BSD, except su to the postgres user, and add a pgsql super user:

```
# sh ./setup_scripts/create-debian-db.sh
```

and of course for development

```
$ createuser -sEP <your-login>
```

Again, record data base passwords for later on, when setting up *net24.conf*

PostgresQL network and user account mapping

Under FreeBSD, the files pg_hba.conf, postgresql.conf, and pg_ident.conf are found in /usr/local/pgsql/data. On

Debian they are in /etc/postgresql/9.1/main, the configuration directory for the main DB cluster. Peer mapping is used instead of trust, as it is far more secure. MD5 authentication is used on localhost as configuring Unix sockets is a lot harder to do in Python SQLAlchemy. Also, at some point in the future, the master DB may be run as a cluster for scalability.

> On Debian, do not disable the administrative access in pg_hba.conf for the postgres user across a unix socket. All sorts of maintenance and system cron jobs won't work then!

Production

Edit /usr/local/pgsql/data/pg_hba.conf to be:

```
# TYPE  DATABASE        USER            CIDR-ADDRESS            METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
```

This turns on password checking for localhost IP access, and sets Unix socket connections from psql to have no passwords from the command line.

Development

On Debian, to make work easier and to enable Python DB stuff to work with less fuss add the following to /etc/postgresql/9.1/main/pg_ident.conf:

```
# MAPNAME       SYSTEM-USERNAME         PG-USERNAME
net24           <login>                 pgsql
net24           <login>                 <login>
```

And edit /etc/postgresql/main/pg_hba.conf to be:

```
# TYPE  DATABASE        USER            CIDR-ADDRESS            METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
map=net24
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
```

This turns on password checking for localhost IP access, and sets Unix socket connections

from psql to have no passwords from the command line.

Of course you can also set the following environment variables in .profile/.bashrc

```
PGDATABASE="dms"
PGUSER="pgsql"
export PGUSER PGDATABASE
```

Postgresql.conf settings

On Debian, set listen_addresses to ip6-localhost,localhost, and on both system types set shared_buffers to 64MB.

DR Postgresql.conf settings

For reference see the PostgesQL wiki

For DR, add external interface address to *listen_addresses* , set *max_wal_senders* to 3, set *wal_keep_segments* to 256 (4GB WAL logs), and set *hot_standby* to on. Do this on both machines
as the *recovery.conf* file in the PostgresQL cluster data directory is what determines whether postgresql comes up in standby mode or not.

Create the DR user on the master as shell user *postgres* or *pgsql* (FreeBSD):

```
postgres $ psql -c "CREATE USER ruser WITH REPLICATION PASSWORD
'SomethingSimplyDuplex';"
```

Rsync the contents of the data directory, after stopping PostgresQL on the master:

```
root@master # rsync -a /var/lib/postgresql/9.1/main/
root@dr:/var/lib/postgresql/9.1/main
```

On the DR server, in the main cluster data directory create the file *recovery.conf*

```
primary_conninfo = 'host=master port=5432 user=ruser
password=SomethingSimplyDuplex'
standby_mode = on
```

Add the replication user *ruser* to the PG cluster's pg_hba.conf on both master and DR servers:

```
host    replication     ruser           2001:db8::1/128  md5
```

Now do this on the DR

```
# chown postgres:postgres /var/lib/postgresql/9.1/main/recovery.conf
# chmod 640 /var/lib/postgresql/9.1/main/recovery.conf
# ls -l /var/lib/postgresql/9.1/main/recovery.conf
-rw-r----- 1 postgres postgres 108 May 14 17:06
/var/lib/postgresql/9.1/main/recovery.conf
```

When the DR DB is promoted to read/write via the *prctl promote* command, the *recovery.conf* file will be renamed. *recovery.done*

Restart postgresql to make the new settings take effect.

- service postgresql restart for Debian
- /usr/local/etc/rc.d/postgresql restart for FreeBSD

Load database schema and functions

Run psql as DB superuser and load the DB net24dmd schema onto the fresh dms database created above. Its also a good time to load the seed configuration settings as well.

```
$ psql -U pgsql dms
dms=# \i sql/dms-2011.sql
dms=# \i sql/dms-2011-seed-config.sql
```

This contains all stored procedures and triggers etc for the database, created by *pg_dump -s -U pgsql dms*

Install Python stack

As *root* from the *dms-2011* directory, run *./setup_scripts/bootstrap-python-packages.sh*

Create system users

The following will create the 2 system/pseudo users *net24dms* and *net24dmi* the DMS software will run as.

For FreeBSD:

```
# sh ./setup_scripts/create-freebsd-users.sh
```

Debian:

```
# sh ./setup_scripts/create-debian-users.sh
```

Don't forget to add users who need access to zone_tool to the net24dms group so that they can read the *net24.conf* file that also stores the database password.

Install software

As root on FreeBSD:

```
# gmake install
```

Also update */etc/mtree/BIND.chroot.dist* for the /etc/namedb/dynamic directory:

```
/set type=dir uname=root gname=wheel mode=0755
.
    dev             mode=0555
    ..
    etc
        namedb
            dynamic uname=bind gname=net24dmd mode=2775
            ..
            master
            ..
            slave   uname=bind
            ..
            working uname=bind
            ..
        ..
    ..
```

This will set the permissions required for net24dmd to work whenever named is restarted...

As root on Debian:

```
# make install
```

Edit *net24.conf* and test

Edit *net24.conf* in */etc/net24* on Debian, and */usr/local/etc/net24* on FreeBSD to set up DB passwords recorded from above. Also note that logging settings can also be adjusted here. Each different program and mod_wsgi has their own section, that overrides the DEFAULT section.

Test using:

```
$ zone_tool
Welcome to the zone_tool program.  Type help or ? to list commands.

zone_tool > show_config
        auto_dnssec:     false
        default_ref:     net24
        default_ssg:     net24-one
        default_stype:   bind9
        edit_lock:       false
        event_max_age:   120.0
        inc_updates:     false
        nsec3:           false
        soa_expire:      7d
        soa_minimum:     24h
        soa_mname:       ns1.net24.net.nz. (net24-one)
        soa_refresh:     7200
        soa_retry:       7200
        soa_rname:       soa.net24.net.nz.
        use_apex_ns:     true
        zi_max_age:      90.0
        zi_max_num:      25
        zone_del_age:    90.0
        zone_ttl:        24h
zone_tool > show_apex_ns
        ns1.net24.net.nz.
        ns2.net24.net.nz.
zone_tool > show_sectags
        1ST_DOMAINS
        Admin
        NET24
zone_tool > ls_sg
        net24-one                       1
/usr/local/etc/net24/slave-config-templates
zone_tool >
```

Configuring BIND named

Generate the following keys and rndc.conf using zone_tool:

```
# zone_tool generate_tsig_key rndc-key hmac-md5 rndc-local.key
# zone_tool generate_tsig_key remote-key hmac-md5 rndc-remote.key
# zone_tool generate_tsig_key update-ddns hmac-sha256 update-session.key
# zone_tool write_rndc_conf
```

Go to the named */etc* directory (*/etc/bind* on Debian and */etc/namedb* on FreeBSD)
and copy the rndc-remote.key key to the */etc/net24/slave-admin-config* directory.

Debian:

```
# cd /etc/bind
# cp rndc-remote.key /etc/net24/slave-admin-config/bind9
```

FreeBSD:

```
# cd /etc/namedb
# cp rndc-remote.key /usr/local/etc/net24/slave-admin-config/bind9
```

Edit named.conf to set up the options and include statements for the master server. Named.conf segments can be found in _etc/master-named.conf-segments, off an example Debian system.

Options settings. On FreeBSD don't forget to comment out the listen { 127.0.0.1; }; statement. For FreeBSD you will have to change */etc/bind* to */etc/namedb*:

```
//
// Net24 DMS ACL set up for master server
//
// ACLs need to be configured here to use in options...

// include public SG ACL
include "/etc/bind/master-config/master-slave-acl.conf";

options {
        // OS bind options here
        // .
        // .
        // .

        // On multi-homed box, where bind is not on primary
        // hostname and IP use the following to stop named twittering to
itself
        // as it thinks it is not the master server!
        //server-id  "full.host.name.on.internet.";
        //hostname  "full.host.name.on.internet.";

        // we want to do this....
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };


        // secure this name server for use on internet
        recursion no;
        //allow-recursion {
```

```
//        localhost;
//};

// Slave and AXFR settings
allow-transfer {
        localhost;
};
transfers-in 10;
transfers-out 150;
transfers-per-ns 50;

allow-query {
        any;
};

// Notify only from port 53
notify-source  * port 53;
notify-source-v6 * port 53;
// notify to SOA mname server?
notify-to-soa no;

// DNSSEC related options
```

```
        key-directory "keys";
};
```

Master server include zone setup. Add this to */etc/bind/named.conf.local* on Debian, and paste into */etc/namedb/named.conf* on FreeBSD. For FreeBSD you will have to change */etc/bind* to */etc/namedb*:

```
// local rndc key
include "/etc/bind/rndc-local.key";
controls {
        inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { "rndc-key"; };
};
include "/etc/bind/update-session.key";

include "/etc/bind/master-config/master-config.conf";
```

1. Restart named and make sure it works.
2. Start *net24dmd* in debug mode and make sure it runs

   ```
   # net24dmd -d 1
   ```

   It should start and keep running, not detaching from the terminal.
3. In another terminal, do a *zone_tool reconfig_master*. This should rewrite the ACL file in */etc/bind/master-config*
4. Create a zone, and check that you can AXFR it. Delete it once the check has been performed.

   ```
   # zone_tool create_zone test.blam
   # dig +noall +answer -t AXFR test.blam. @localhost
   test.blam.  86400 IN SOA ns1.net24.net.nz. soa.net24.net.nz.
    2012032200 7200 7200 604800 86400
   test.blam.  86400 IN NS ns1.net24.net.nz.
   test.blam.  86400 IN NS ns2.net24.net.nz.
   test.blam.  86400 IN SOA ns1.net24.net.nz. soa.net24.net.nz.
    2012032200 7200 7200 604800 86400
   zone_tool delete_zone test.blam
   ```

   This zone may take about 10 minutes to turn up. Try typinp *show_configsm* at the zone_tool prompt. That will show the next time the ConfigSM will cycle, allowing the zone to be published.

Enabling *net24dmd* at boot

FreeBSD

Copy etc/freebsd/init/net24dmd.init to /usr/local/etc/rc.d/net24dmd, and set *net24dmd_enable* in */etc/rc.conf*.

```
cp etc/freebsd/init/net24dmd.init /usr/local/etc/rc.d/net24dmd
```

Edit /etc/rc.conf and add:

```
# Net24 DMS
net24dmd_enable="YES"
```

To start daemon:

```
# /usr/local/etc/rc.d/net24dmd start
```

Debian

Copy *etc/debian/init/net24dmd.init* to */etc/init.d/net24dmd*, and copy *etc/debian/init/net24dmd.default* to */etc/default/net24dmd*, and run *insserv /etc/init.d/net24dmd*.

```
# cp etc/debian/init/net24dmd.init /etc/init.d/net24dmd
# chmod 755 /etc/init.d/net24dmd
# cp etc/debian/init/net24dmd.default /etc/default/net24dmd
# insserv /etc/init.d/net24dmd
```

Edit */etc/default/net24dmd* to enable net24dmd on boot.

```
# defaults file for net24dmd

# start net24dmd from init.d script?
#  only allowed values are "true", and "false"
NET24DMD_ENABLE=true
```

Cron jobs

FreeBSD

Just create a cron job to run *zone_tool vacuum_all* daily, It does not have to be done as root, though that is probably the easiest.

Same as for FreeBSD.

WSGI Setup.

Debian

Edit /etc/apache2/sites-available/default and insert 'include /etc/net24/dms-wsgi-apache.conf'

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        include /etc/net24/dms-wsgi-apache.conf

        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel info

        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Reload apache2 with

```
# service apache2 reload
```

Create WSGI accounts, and mind that you record the passwords for later:

```
# htpasswd -c /etc/apache2/htpasswd-dms admin-dms
# htpasswd /etc/apache2/htpasswd-dms net24-dms
# htpasswd /etc/apache2/htpasswd-dms 1stdomains-dms
# chown root:www-data /etc/apache2/htpasswd-dms
# chmod 640 /etc/apache2/htpasswd-dms
```

Check that it works:

```
# curl -X POST -H 'Content-Type: application/json' -u admin-dms -d
"@testing/test.jsonrpc" \
http://dns-master1.grantma-imac/admin_dms
```

It should spew a lot of JSON content.

FreeBSD

Enable virtual hosts by uncommenting the include at the bottom of */usr/local/etc/apache22/httpd.conf*, and editing the include file to comment out example sites. Then:

```
cat etc/vhost-net24dms-freebsd >>
/usr/local/etc/apache22/extra/httpd-vhosts.conf
```

and edit it to set ServerName and log file names as above.

Create WSGI accounts, and mind that you record the passwords for later:

```
# htpasswd -c /usr/local/etc/apache22/htpasswd-dms admin-dms
# htpasswd /usr/local/etc/apache22/htpasswd-dms net24-dms
# htpasswd /usr/local/etc/apache22/htpasswd-dms 1stdomains-dms
# chown root:www /usr/local/etc/apache22/htpasswd-dms
# chmod 640 /usr/local/etc/apache22/htpasswd-dms
```

Check that it works:

```
# curl -X POST -H 'Content-Type: application/json' -u admin-dms -d
"@testing/test.jsonrpc" \
http://dns-master1.grantma-imac/admin_dms
```

It should spew a lot of JSON content.

Rsyslog

FreeBSD

```
# mkdir /usr/local/etc/rsyslog.d
# mkdir /var/spool/rsyslog
```

Edit */etc/usr/local/etc/rsyslog.conf* and add the following after the top 3 modules lines, unashamedly adapted from

Debian:

```
# provides UDP syslog reception
#$ModLoad imudp
#$UDPServerRun 514

# provides TCP syslog reception
#$ModLoad imtcp
#$InputTCPServerRun 514


###########################
#### GLOBAL DIRECTIVES ####
###########################

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

#
# Set the default permissions for all log files.
#
$FileOwner root
$FileGroup wheel
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022

#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /usr/local/etc/rsyslog.d/*.conf
```

Then create the file */usr/local/etc/rsyslog.d/00network.conf*:

```
# provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514

# Sample Clients
#$AllowedSender UDP, [2001:470:c:110e::2]
#$AllowedSender TCP, [2001:470:c:110e::2]
#$AllowedSender UDP, [2001:470:66:23::2]
#$AllowedSender TCP, [2001:470:66:23::2]
```

and the file */usr/local/etc/rsyslog.d/pgsql.conf*, setting the rsyslog database password:

```
$ModLoad ompgsql
# local7.* /var/log/local7.log
local7.* :ompgsql:localhost,dms,rsyslog,ScrabyBee
```

then

```
# chmod 600 /usr/local/etc/rsyslog.d/pgsql.conf
```

and restart rsyslog, checking */var/log/messages* for errors. Also do:

```
# ps axc | grep rsyslogd
20736   0  S       0:00.01 rsyslogd
# lsof -p 20736
```

and you should see a connection listed to *postgresql*. Check */var/log/messages* for postgresql error messages.

Debian

Basically the same except that the file structures are already there!

Create the file */etc/rsyslog.d/00network*:

```
# provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514

# Sample Clients
#$AllowedSender UDP, [2001:470:c:110e::2]
#$AllowedSender TCP, [2001:470:c:110e::2]
#$AllowedSender UDP, [2001:470:66:23::2]
#$AllowedSender TCP, [2001:470:66:23::2]
```

and */etc/rsyslog.d/pgsql.conf*, setting the database passwordfor rsyslog:

```
### Configuration file for rsyslog-pgsql
### Changes are preserved

$ModLoad ompgsql
# local7.* /var/log/local7.log
local7.* :ompgsql:localhost,dms,rsyslog,ScrapyBee
```

Restart rsyslog, and check /var/log/syslog for error messages. Also do:

```
# ps axc | grep rsyslogd
20736   0  S       0:00.01 rsyslogd
# lsof -p 20736
```

and you should see a connection listed to *postgresql*. Check */var/log/syslog* for postgresql error messages.

Master Server Bind Logging Setup

Add the following to */etc/namedb* or */etc/bind* as logging.conf, and include it:

```
// Logging
logging {
        channel master_server {
                // Sends log messages to master server.
                syslog local7;
                severity info;
        };
        // Lots of notifies bounce around, giving heaps of refused messages
        // that are basically noise
        category notify {null;};
        // Both below are default bind options. Here for 'normality'
        category default { master_server; default_syslog; default_debug; };
        category unmatched { null; };
};
```

Restart named and check the system events table in the dms database. Log messages should start appearing in it.

Master Server Firewall Setup

IPsec SPD is not stateful, and for 2 way traffic, it is easier just to set it up to allow all traffic in both directions. System IP filtering on the DMS master server should be used to protect the master server. It is possible to detect IPSEC traffic in iptables and IPFW2, and filter that incoming traffic statefully.

Due to an incomplete IPv6 IPFW2 filtering implementation on FreeBSD, it is not possible to turn the filtering on for IPv6 as outgoing TCP connections (eg rndc and named) end up being denied if IPSEc filtering is enabled. It all works for IPv4....

Here is a Sample iptables set up for Linux. Notice the INPUT rule diverting all incoming IPSEC traffic into the ipsec-in rule, which ends in a log chain that DROPs disallowed traffic. There are also a couple of rules for local system admin traffic as one of the slaves is a internal host in this example. The latter is not typical of a large scale setup.

```
# Completed on Sun Mar  4 16:30:11 2012
# Generated by ip6tables-save v1.4.12.2 on Sun Mar  4 16:30:11 2012
*filter
:INPUT ACCEPT [66:5920]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [33:3448]
:ipsec-in - [0:0]
:log - [0:0]
-A INPUT -m policy --dir in --pol ipsec -j ipsec-in
-A ipsec-in -m state --state RELATED,ESTABLISHED -j ACCEPT
-A ipsec-in -p udp -m udp --sport 500 --dport 500 -j ACCEPT
-A ipsec-in -p ipv6-icmp -m icmp6 --icmpv6-type 129 -j ACCEPT
-A ipsec-in -p udp -m state --state NEW -m udp --dport 53 -j ACCEPT
-A ipsec-in -p udp -m state --state NEW -m udp --dport 514 -j ACCEPT
-A ipsec-in -p tcp -m state --state NEW -m tcp --dport 53 -j ACCEPT
-A ipsec-in -p tcp -m tcp --dport 53 -m state --state NEW -m frag --fragid
0 --fragfirst -j ACCEPT
-A ipsec-in -s fd14:828:ba69::/48 -p tcp -m tcp --dport 22 -m state --state
NEW -j ACCEPT
-A ipsec-in -s fd14:828:ba69::/48 -p tcp -m tcp --dport 80 -m state --state
NEW -j ACCEPT
-A ipsec-in -j log
-A log -m limit --limit 3/sec -j LOG --log-prefix "Def log:  - "
--log-tcp-options --log-ip-options
-A log -p icmp -j DROP
-A log -j REJECT --reject-with icmp6-port-unreachable
COMMIT
# Completed on Sun Mar  4 16:30:11 2012
```

The above and a FreeBSD IPFW2 example are in the *etc/firewall* directory of the dms-2011 git archive.

Net24 DMS Debian Install Documentation

```
Net24 DMS System
----------------

The DNS Management System (DMS) is designed to have a master/replica master
setup. It is a complex setup, requiring the hand configuration of database,
DNS
server, and other components. If your setup does not require one of the
components such as quagga, dms-wsgi, or etckeeper, just skip that section.

etckeeper and ssh
-----------------

etckeeper is a tool to keep the contents of /etc in a git VCS.  When
combined
with ssh and the appropriate git remote setup with cron, it allows the /etc
of
```

```
the other machine in the master/replica DR pair to be kept on its
alternate,
and vice-versa.  This protects aginst the /etc on the master being updated,
the
replica being missed, and then finding that things aren't working on the
replica when the master dies, with no record of the updates needed to
machine
configuration.

For information on etckeeper usage, see /usr/share/doc/etckeeper/README.gz

Example for diffing/checking out /etc/racoon/racoon-tool.conf from other
machine:

dms3-master:/etc# git diff dms4-dr/master racoon/racoon-tool.conf
dms3-master:/etc# git checkout dms4-dr/master racoon/racoon-tool.conf
dms3-master:/etc# git checkout HEAD racoon/racoon-tool.conf

1) etckeeper installation.  Before installing etckeeper, you need to add a
.gitignore to /etc to prevent /etc/shadow and other secrets files from
ending
up in etckeeper for security reasons. The contents of the seed
/etc/.gitignore
file is:

-----
# Ignore these files for security reasons
krb5.keytab
shadow
shadow-
racoon/psk.txt
ssl/
ssh/moduli
ssh/ssh_host_*

# Ignore these DMS dirs as the files are daemon generated
bind/master-config/
bind/rsync-config/

-----

You probably have to purge etckeeper removing the initial /etc git archive
if
you are reading this, create the .gitignore file, and reinstall etckeeper:

# dpkg --purge etckeeper
# vi /etc/.gitignore
# aptitude install etckeeper

1) Set up ssh.  As root on both boxes, turn off the following settings in
sshd_config:

RSAAuthentication no
```

```
PubkeyAuthentication no
RhostsRSAAuthentication no
HostbasedAuthentication no
ChallengeResponseAuthentication no
PasswordAuthentication no
GSSAPIAuthentication no
X11Forwarding no
UsePAM no

Then add the following to /etc/ssh/sshd_config, and adjust your network
and administrative sshd authentication settings:

-----
UsePAM no
AllowTcpForwarding no
AllowAgentForwarding no
X11Forwarding no
PermitTunnel no
AllowGroups sudo root
# Section for DMS master/replica servers
Match Address 2001:db8:f012:2::3/128,2001:db8:ba69::3/128
        PubkeyAuthentication yes
        # PermitRootLogin forced-commands-only
        # The above only works with commands given in authorized_keys
        PermitRootLogin without-password
        ForceCommand /usr/sbin/etckeeper_git_shell
# Section for administrative access
Match Address 2001:db8:ba69::/48,192.0.2.0/24,201.0.113.2/32
        PermitRootLogin yes
        GSSAPIAuthentication yes
        PubkeyAuthentication yes
        MaxAuthTries 10
        X11Forwarding yes
        AllowTcpForwarding yes
        AllowAgentForwarding yes
-----

Reload sshd on both servers:

# service ssh reload

Create a passwordless  ssh key on both servers as root, and copy the public
part of the key to /root/.ssh/authorized_keys.

# mkdir /root/.ssh
# ssh-keygen -f /root/.ssh/id_gitserve_rsa -t rsa -q -N ''
# vi /root/.ssh/config

and set contents of ssh config as follows, changing Host as appropriate:

-----
Host dms3-dr*
IdentityFile ~/.ssh/id_gitserve_rsa
```

```
-----

It is also a good idea to set up a /etc/hosts file entries on each server.

Set up /root/.ssh/authorized_keys:

-----
# mkdir /root/.ssh
# cat - > /root/.ssh/authorized_keys
-----
Cut and paste /root/.ssh/id_gitserve_rsa.pub from other machine into above,
finishing with ^D.  Then do vice-versa, to make the other direction
functional.

Check that things work on both hosts:

# ssh -l root dms4-dr
Rejected
Connection to dms4-dr closed.

etc.

Note: Stopping ssh and running sshd from the commandline '/usr/sbin/sshd
-d' on
one, and then using 'ssh -vl root' on the other (and vice versa) is very
useful
for connection debugging.

2) Git remote set up to pair up /etc archives.

As root do:

dms3-master# git remote add dms4-dr ssh://dms4-dr.devel.net.nz/etc

and vice versa

Check that both work by executing:

dms3-master:/etc# git fetch --quiet dms4-dr

and vice versa

3) Set up crond.

Edit the file /etc/cron.d/dms-core, uncomment the line for git fetch, and
set
the remote name:

-----
# Replicate etckeeper archive every 4 hours
7 */4 * * * root  cd /etc && /usr/bin/git fetch --quiet dms4-dr
-----
```

Do test each cron command by running it from the root command line.

racoon IPSEC set up

The DMS system uses IPSEC to authenticate server access to the master servers,
encrypting and/or integrity protecting the outgoing zone transfers, rndc and
configuration rsync traffic.

This is only covering basic PSK set up. For X509 etc, see /usr/share/doc/racoon/README.Debian

On each machine, dpkg-reconfigure racoon, and choose the "racoon-tool" configuration method. Edit /etc/racoon/racoon-tool.conf, and add the machines
source IP address:

```
-----
connection(%default):
       src_ip: 192.168.102.2
       admin_status: disabled
-----
```

Add the other replica server and each DNS as a separate configuration fragment
in /etc/racoon/racoon-tool.conf.d, named after the machine's short hostname:

```
-----
peer(192.168.102.2):

connection(dms4-dr-eth1):
        dst_ip: 192.168.102.2
 # defaults to esp
        # encap: ah
        admin_status: enabled
-----
```

For the replica servers, if you want to inspect/control traffic select ah IPSEC
encapsulation. Note, racoon-tool sets up a transport mode IPSEC connection if
no src_range/dst_range parameters are given.

Transport mode does not encrypt ICMP traffic, as that can complicate UDP/TCP
connection issues extensively.

Also enter a separate PSK in /etc/racoon/psk.txt for each IPSEC connection.

Each server has IPSEC configured and active to both the replica servers (master and DR). The master and replica have IPSEC configured as well. Both
replica servers and 2 slaves should be PSK keyed with each other if DNSSEC authentication is to be used for the majority of slaves. This ensures that the
DNSSEC CERT records can be progated for use.

Useful racoon-tool commands are:

```
# racoon-tool vlist
# racoon-tool spddump
# racoon-tool saddump
# racoon-tool vup <connection name>
# racoon-tool vdown <connection name>
# racoon-tool reload
# racoon-tool restart.
```

Test the connection by pinging the far end - tests unencrypted reachability,
and then telnet/netcat the different TCP ports used across the link. This will involve ports 873 (rsync), 953 (rndc/named), 53 (named) to each slave,
and port 53 on the masters (from slave). Between both the replica servers (master and DR), port 5432 (postgresql) has to be reachable, as well as port 22
(ssh). Port 80 (http) for apt-get updates may also be involved.

PostgresQL DB Setup and Master/Replica Configuration

DB user and DB creation only has to happen on the initial master server, as it
will be 'mirrored' to the replica once DB replication is established. The replica server will configured to run in 'hot-standby' mode so that we can verify mirroring by read-only means using zone_tool.

Though the master and replica can run the PGSQL dms cluster on port 5433 or other port, it is reccommended to swap the ports with the main cluster, and revert the main cluster to manual start up.

Edit postgresql.conf /etc/postgresql/9.1/main and /etc/postgresql/9.1/dms, and swap the settings for 'port =', making dms port 5432.

Edit /etc/postgresql/9.1/main/start.conf, and set it to manual.

Stop posgresql, and start it, (restart will probably result in failure due to a
port clash...):

```
# pg_ctlcluster 9.1 main stop
# service postgresql stop
```

```
# service postgresql start

Use etckeeper to migrate the configuration to the replica:

dms3-master:/etc# etckeeper commit

dms4-dr:/etc#  git fetch dms3-master
dms4-dr:/etc/# git checkout dms3-master/master postgresql/9.1/main
postgresql/9.1/dms
dms4-dr:/etc# pg_ctlcluster 9.1 main stop
dms4-dr:/etc# service postgresql stop
dms4-dr:/etc# service postgresql start

On the master, set the DB passwords for the dms user and the ruser (they
will
be copied to the replica when mirroring is started):

root@dms3-master:/home/grantma# psql -U pgsql dms
psql (9.1.4)
Type "help" for help.

dms=# \password ruser
Enter new password:
Enter it again:
dms=# \password dms
Enter new password:
Enter it again:
dms=# \q

Note: The pgsql database super user exists for cross OS/distro
compatibility
reasons.

Record the 2 passwords you have just set for reference.  Put the ruser
password in /etc/net24/pgpassfile on both machines, which is in the
standard
PGSQL format (see section 31.14 in "PostgreSQL 9.1.4 Documentation").

NB: You will have to alter the machine name and password. Use vi or vim as
root to prevent permissions and ownership alteration.

Also edit /etc/net24/net24.conf, and set the dms db_password for zone_tool
on
both machines as zone_tool uses password access unless user is in
pg_ident.conf

Connecting Replica and Starting Replication
-------------------------------------------

On the master, and replica, set the replication address in pg_hba.conf:

dms3-master:/root# dms_admindb -r dms4-dr.devel.net.nz
dms4-dr:/root# dms_admindb -r dms3-master.devel.net.nz
```

Set up PGSQL recovery.conf, and start replica DB:

```
dms4-dr:/root# service postgresql stop
dms4-dr:/root# dms_pg_basebackup dms3-master.devel.net.nz
dms4-dr:/root# dms_write_recovery_conf dms3-master.devel.net.nz
dms4-dr:/root# service postgresql start
```

Note: The above is flipping DB replica functionality from the default DB as
master

Check that replication is running by seeing if zone_tool can access default configuration settings:

```
dms4-dr:/root# zone_tool show_config
root@dms4-dr:/home/grantma# zone_tool show_config
        auto_dnssec:       false
        default_ref:       net24
        default_sg:        net24-one
        default_stype:     bind9
        edit_lock:         false
        event_max_age:     120.0
        inc_updates:       false
        nsec3:             false
        soa_expire:        7d
        soa_minimum:       24h
        soa_mname:         ns1.net24.net.nz. (net24-one)
        soa_refresh:       7200
        soa_retry:         7200
        soa_rname:         soa.net24.net.nz.
        syslog_max_age:    120.0
        use_apex_ns:       true
        zi_max_age:        90.0
        zi_max_num:        25
        zone_del_age:      0.0
        zone_del_pare_age: 90.0
        zone_ttl:          24h
```

Master/Replica rsyncd setup

Both the machines will have to rsync from one another, depending on which is
running as the DR replica. So we are setting up rsync client passwords, and
rsyncd configuration on one, and using the same settings on the other machine.

Add the following to /etc/rsyncd.conf

```
-----
hosts allow = 2001:db8:f012:2::2/128 2001:db8:f012:2::3/128
```

```
secrets file = /etc/rsyncd.secrets

[dnsconf]
        path = /etc/bind/rsync-config
        uid=bind
        gid=bind
        comment = Slave server config area
        auth users = dnsconf
        use chroot = yes
        read only = no

[dnssec]
        path = /var/lib/bind/keys
        uid=bind
        gid=net24dmd
        comment = DNSSEC key data area
        auth users = dnssec
        use chroot = yes
        read only = no
-----

adjusting IP addresses as needed. And also set up the /etc/rsyncd.secrets
file:

-----
dnsconf:SuperSecret
dnssec:PlainlyNotSecret
-----

making it only readable by root:

# chown root:root /etc/rsyncd.secrets
# chmod 600 /etc/rsyncd.secrets

and set the passwords /etc/net24/rsync-dnssec-password and
/etc/net24/rsync-dnsconf-password using vi to preserve permissions.

and enable the rsyncd daemon in /etc/default/rsync, and start the service.

# service rsync start

Use etckeeper to mirror the config to the replica:

dms3-master:/etc# etckeeper commit

dms4-dr:/etc#  git fetch dms3-master
dms4-dr:/etc/# git checkout dms3-master/master rsyncd.secrets rsyncd.conf
/etc/default/rsync net24/rsync-dnsconf-password net24/rsync-dnssec-password
dms4-dr:/etc/# chmod 600 /etc/rsyncd.secrets

And start rsyncd on the replica as well.

Check that you can connect to the rsync port on one from the other machine,
```

```
and vice-versa.

root@dms4-dr:/home/grantma# telnet dms3-master rsync
Trying 192.168.101.2...
Connected to dms3-master.devel.net.nz.
Escape character is '^]'.
@RSYNCD: 30.0
^]c

telnet> c
Connection closed.
root@dms4-dr:/home/grantma#

Lets create the master sg, and disabled replica servers (DMS master and
DR),
and check that the DR slave named config can be rsynced.

dms3-master:/etc/# zone_tool
zone_tool > create_sg -p net24-master /etc/net24/server-config-templates
2001:db8:f012:2::2 2001:db8:f012:2::3
zone_tool > create_server -g net24-master dms4-dr 2001:db8:f012:2::2
zone_tool > create_server -g net24-master dms3-master 2001:db8:f012:2::3
zone_tool > rsync_server_admin_config dms4-dr no_rndc
zone_tool >

dms4-dr:/etc/# zone_tool
zone_tool > rsync_server_admin_config dms3-master no_rndc
zone_tool >

Look in /var/log/syslog on the rsyncd server to debug issues.


Setting up rsyslog on Master and Replica
----------------------------------------

On the master, create the file /etc/rsyslog.d/00network.conf with the
contents:
-----
# provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514

#$AllowedSender UDP, [2001:470:c:110e::2]
#$AllowedSender TCP, [2001:470:c:110e::2]
#$AllowedSender UDP, [2001:470:66:23::2]
#$AllowedSender TCP, [2001:470:66:23::2]
#$AllowedSender UDP, [fd14:828:ba69:1:21c:f0ff:fefa:f3c0]
#$AllowedSender TCP, [fd14:828:ba69:1:21c:f0ff:fefa:f3c0]
-----
```

All replica and slave DNS servers will have to be entered into this file.

Also alter the file /etc/rsyslog.d/pgsql and change the contents to:

```
-----
### Configuration file for rsyslog-pgsql
### Changes are preserved

$ModLoad ompgsql
local7.* /var/log/local7.log
local7.* :ompgsql:/var/run/postgresql,dms,rsyslog,
-----
```

The default configuration propagated to the DMS servers uses local7 as the named logging facility.

Setting up Bind9 master DNS server

Copy the bind configuration segments from
/usr/share/doc/dms-core/examples/bind
to the /etc/bind directory:

```
root@dms3-master:/etc/bind# cp /usr/share/doc/dms-core/examples/bind .
root@dms3-master:/etc/bind# chgrp bind named-dr-replica.conf
named.conf.options named.conf.local
```

Create all the required TSIG rndc and dynamic DNS update keys, and generate required /etc/bind/rndc.conf:

```
root@dms3-master:/etc/bind# zone_tool generate_tsig_key -f update-ddns
hmac-sha256 update-session.key
root@dms3-master:/etc/bind# zone_tool generate_tsig_key -f rndc-key
hmac-md5 rndc-local.key
root@dms3-master:/etc/bind# zone_tool generate_tsig_key -f remote-key
hmac-md5 rndc-remote.key
root@dms3-master:/etc/bind# zone_tool write_rndc_conf -f
root@dms3-master:/etc/bind# cp -a rndc-remote.key
/etc/net24/server-admin-config/bind9
root@dms3-master:/etc/bind# cp rndc-remote.key /etc/bind/rsync-config
```

Add a line to /etc/default/bind9 to get rid of the default rndc.key to stop rndc complaining:

```
-----
# Get rid of default bind9 rndc.key, that debian install scripts always
# generate  Stops rndc complaining:
rm -f /etc/bind/rndc.key

# run resolvconf?
RESOLVCONF=no
```

```
# startup options for the server
OPTIONS="-u bind"
-----

Create 2 empty named.conf include files that will be overwritten by
net24dmd,
on both machines (master and replica):

# touch /etc/bind/master-config/server-acl.conf
# touch /etc/bind/master-config/zones.conf
# chown net24dmd:bind /etc/bind/master-config/*

Restart named to make sure all is good:

root@dms3-master:/etc/bind# service bind9 stop
root@dms3-master:/etc/bind# service bind9 start
root@dms3-master:/etc/bind# rndc status
version: 9.8.1-P1
CPUs found: 1
worker threads: 1
number of zones: 5
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running

Enable net24dmd, the dynamic DNS update and DMS event daemon by editing
/etc/default/net24dmd, setting NET24DMD_ENABLE=true, and start it:

root@dms3-master:/etc# vi /etc/bind/net24dmd
root@dms3-master:/etc# service net24dmd start
[ ok ] Starting net24dmd: net24dmd.
root@dms3-master:/etc# service net24dmd status
[ ok ] net24dmd is running.

Enable the master server so that the server SM can monitor named on the
machine:

root@dms3-master:/etc# zone_tool enable_server dms3-master

This means that when net24dmd is started, it will set up an index in the
Master
SM in the DB to the Master server in the ServerSM table (important for
keeping
track of where the master is for human output and ServerSM functionality -
uses
machines actual network addresses cf.  master_address and
master_alt_address in
```

```
replica SG)

And make sure you can create a domain:

root@dms3-master:/etc/bind# zone_tool create_zone foo.bar.org
root@dms3-master:/etc/bind# zone_tool show_zone foo.bar.org
$TTL 24h
$ORIGIN foo.bar.org.

;
; Zone:      foo.bar.org.
; Reference: net24
; zi_id:     1
; ctime:     Mon Jul  2 11:30:26 2012
; mtime:     Mon Jul  2 11:31:03 2012
; ptime:     Mon Jul  2 11:31:03 2012
;


;| Apex resource records for foo.bar.org.
;!REF:net24
@                       IN      SOA             ( ns1.net24.net.nz. ;Master
NS
                                                soa.net24.net.nz. ;RP email
                                                2012070200   ;Serial
yyyymmddnn
                                                7200         ;Refresh
                                                7200         ;Retry
                                                604800       ;Expire
                                                86400
;Minimum/Ncache
                                                )
                        IN      NS              ns2.net24.net.nz.
                        IN      NS              ns1.net24.net.nz.


root@dms3-master:/etc/bind# zone_tool show_zonesm foo.bar.org
        name:            foo.bar.org.
        alt_sg_name:     None
        auto_dnssec:     False
        ctime:           Mon Jul  2 11:30:26 2012
        deleted_start:   None
        edit_lock:       False
        edit_lock_token: None
        inc_updates:     False
        lock_state:      EDIT_UNLOCK
        mtime:           Mon Jul  2 11:30:26 2012
        nsec3:           False
        reference:       net24
        soa_serial:      2012070200
        sg_name:         net24-one
        state:           PUBLISHED
        use_apex_ns:     True
```

```
        zi_candidate_id: 1
        zi_id:           1
        zone_id:         1
        zone_type:       DynDNSZoneSM
        zi_id:           1
        ctime:           Mon Jul  2 11:30:26 2012
        mtime:           Mon Jul  2 11:31:03 2012
        ptime:           Mon Jul  2 11:31:03 2012
        soa_expire:      7d
        soa_minimum:     24h
        soa_mname:       ns1.net24.net.nz.
        soa_refresh:     7200
        soa_retry:       7200
        soa_rname:       soa.net24.net.nz.
        soa_serial:      2012070200
        soa_ttl:         None
        zone_id:         1
        zone_ttl:        24h

root@dms3-master:/etc/bind# dig -t AXFR +noall +answer foo.bar.org
@localhost
foo.bar.org.  86400 IN SOA ns1.net24.net.nz. soa.net24.net.nz. 2012070200
7200 7200 604800 86400
foo.bar.org.  86400 IN NS ns1.net24.net.nz.
foo.bar.org.  86400 IN NS ns2.net24.net.nz.
foo.bar.org.  86400 IN SOA ns1.net24.net.nz. soa.net24.net.nz. 2012070200
7200 7200 604800 86400
root@dms3-master:/etc/bind# zone_tool delete_zone foo.bar.org

Reflect the bind directory to the DR via etckeeper:

root@dms3-master:/etc# etckeeper commit

root@dms4-dr:/etc# git fetch dms3-master
root@dms4-dr:/etc# git checkout dms3-master/master bind
root@dms4-dr:/etc# git checkout dms3-master/master default/bind9

Setting UP DR bind9 slave server on Replica
-------------------------------------------

Edit /etc/net24/server-admin-config/bind9/controls.conf and add each
masters IP
address to the uncommented inet  allow line.  IPv4 address will have to be
prefixed with '::ffff:' as by default Linux binds  v6 sockets to IPv4.

Rsync the admin config from the master to the DR replica, not doing any
rndc
reconfig:

root@dms3-master:/etc# zone_tool rsync_server_admin_config dms4-dr no_rndc

Copy the /etc/net24/server-admin-config/bind9 directory to
/etc/bind/rsync-config
```

```
root@dms3-master:/etc# cp -a /etc/net24/server-admin-config/bind9/*
/etc/bind/rsync-config
root@dms3-master:/etc# chown bind:bind /etc/bind/rsync-config/*

Reflect the bind directory to the DR via etckeeper:

root@dms3-master:/etc# etckeeper commit

root@dms4-dr:/etc# git fetch dms3-master
root@dms4-dr:/etc# git checkout dms3-master/master
net24/server-admin-config

To apply permissions on master to replica:

root@dms4-dr:/etc# git checkout dms3-master/master .etckeeper
root@dms4-dr:/etc# etckeeper init
root@dms4-dr:/etc# etckeeper commit

On the replica, Edit /etc/default/bind9, adding '-c
/etc/bind/named-dr-replica.conf' to OPTIONS, and restart named.

root@dms4-dr:/etc# service bind9 restart

On the master, enable the DR replica server in the replica SG:

root@dms3-master:/etc# zone_tool enable_server dms4-dr

Check by switching between master and replica:

root@dms3-master:/etc# dms_master_down

root@dms4-dr:/etc/# dms_promote_replica

root@dms3-master:/etc# dms_start_as_replica dms4-dr.devel.net.nz

Wait for synchronisation to be shown 15 - 20 minutes:

root@dms4-dr:/etc# zone_tool show_replica_sg -v
        sg_name:             net24-replica
        config_dir:          /etc/net24/server-config-templates
        master_address:      192.168.101.2
        master_alt_address:  192.168.102.2
        replica_sg:          True
        sg_id:               2
        zone_count:          0

        Slave Servers:
        dms4-dr                     192.168.102.2

                OK
        dms3-master                 192.168.101.2
```

```
                OK

and switch back as above.

Importing Zones to DMS system
-----------------------------

Set the default settings shown in zone_tool show_config on the DMS master.

root@dms3-master:/etc# zone_tool show_config
root@dms3-master:/etc# zone_tool set_config soa_mname ns1.foo.bar.net
root@dms3-master:/etc# zone_tool set_config soa_rname soa.foo.bar.net
root@dms3-master:/etc# zone_tool set_config default_sg foo-bar-net
root@dms3-master:/etc# zone_tool set_config default_ref foo-bar-net
root@dms3-master:/etc# zone_tool show_apex_ns
root@dms3-master:/etc# zone_tool edit_apex_ns

Create all required reverse zone on the master, setting the zone_tool
create_zone inc_updates flag argument so that auto reverse zone records can
be
created and managed.

root@dms3-master:/etc# zone_tool create_zone 2001:2e8:2012::/32 inc_updates

Import all the zones. First of all, load the apex zone which contains the
ns1/ns2 records with no_update_apex_ns, then load all the rest. Its an idea
to
have a look at the edit_lock flag at the same time for those top zone(s).
Note
that zone_tool load_zones requires all files to be named by full domain
name.

root@dms3-master:/some/dir/with/zone/files# zone_tool load_zone foo.bar.net
foo.bar.net no_use_apex_ns edit_lock
root@dms3-master:/some/dir/with/zone/files# zone_tool load_zones *


Setting up WSGI on apache
-------------------------

Include the /etc/net24/dms-wsgi-apache.conf fragment into the file
/etc/apache2/sites-available/default-ssl

Set the apache log level to info, delete the cgi-bin section, and set up
the
SSL certificates.

Create the htpasswd file /etc/net24/htpasswd-dms, and set the passwords for
admin-dms, net24-dms, 1stdomains-dms WSGI users.

Use a2ensite and a2dissite to enable the SSL default site

# a2dissite default
```

```
# a2ensite default-ssl
```

Reload apache2

```
# service apache2 reload
```

Reflect configuration as above to DR partner server

Check that it functions by using curl on the master server:

```
# cd /tmp
# cp -a /usr/share/doc/dms-core/examples/wsgi-json-testing .
# cd wsgi-json-testing
```

Edit json-test.sh so that it works for you, re URLs and user/password.
test4.jsonrpc uses list_zone, so try that first to check WSGI is live.

```
# ./json-test.sh test4
```

It may take a while before anything shows up if you have imported tens of thousands of zone. Full error information will be shown in the configured apache error log /var/log/apache2/error.log. You can also try some of the other example tests as well after editing them for the current setup.

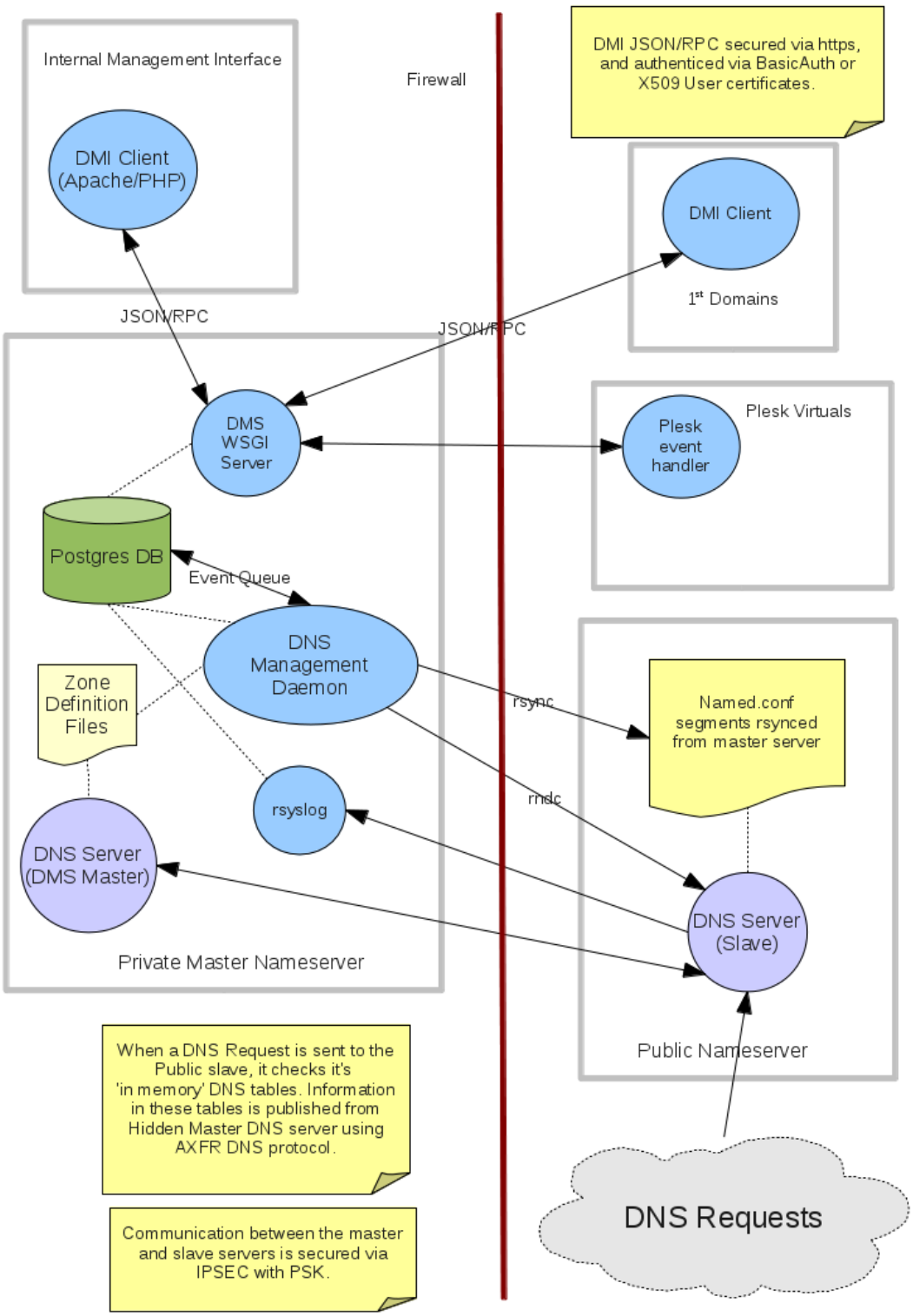
NOTE: Also try some of the readonly tests on the other DR partner server to make sure WSGI is functional there.

```
 -- Matthew Grant <matthew.grant@net24.co.nz>  Mon, 18 Jun 2012 12:44:48
+1200
```

Original Architecture Documentation

DMS and DNS Architecture (2011)

Architecture



Languages

Looked quickly at comparison of language strengths, from PHP, Python, Perl, Ruby to Java. Python fits in best for the net24dmd. Has right mix of low level with library, combined with the string flexibility of Perl, dynamic typing, scalability, and a coherent data syntax and memory model. A lot of Pythons odd things like the 'self' parameter in

class methods is built around the internals that make it more scalable than the syntactical fuzziness of Perl. It also does not have the obtuse execution requirements of Java, and is flexible enough on the outside to be completely unix daemon-like.

The thing we will have to watch is the code duplication between the PHP in net24dms, and net24dmd. If there is going to be a lot of common object code, it is better writing net24dms on top of the classes and framework in net24dmd.

Language for DNS Manager Daemon

Python 3.x has been chosen as 1st candidate with a fallback to latest python 2.7/2.8 if that proves to be too difficult. This will provide a longer life cycle for the code before it has to be updated for a newer Python. The ability to thread which enables a robust task per child-thread execution model, and Python SQLAlchemy Database ORM are key considerations for this.

DNS server software

Decided to go forward using ISC Bind 9 as DNSSEC is on the way, and Bind 9 will be the software used to roll this out. Other implementations of DNS software exist, Netlabs NL NSD3 is one, but it looks more suited to a TLD registry and large site/domain use than for DNS Provider use for small zones.

Bind 9.7.3 has just been released, which is a bug fix version, which includes bug fixes for Dynamic update DNSSEC master mode. I will get this installed on the test master server for anathoth.net and anathoth.org

The DNS server state machine classes will be designed so that NL Netlabs nsd 3.x can be added latter on. The design should also be able to accept a machine running openDNSSEC zone signing daemon being added later on. This will be achieved by the use of state machine design, and code modularity. A war will declared on spaghetti and noodles.

Backend Database.

PosgresQL will be used for this. I have used both MySQL and PostgresQL, and PostgresQL has proven to be the most flexible due to the maturity of its transactional design.

DMI Server/Clients

The old PHP API will be adapted and used for this. Some code may be reusable if its API is basically is what is needed. Otherwise, it is probably better to start again.

Hidden Master Management software

Will use state machine and event queue design, with state and event information recorded in PostgresQL. State machines will exist for each:

- DNS zone to track life-cycle state of zone
- DNS server to track its availability, and another to track its name server reload cycles.

Event Queue

Shall be able to process multiple events at once, up to a specified maximum number of threads. Python has a Global memory lock, but must threads will block on outside file descriptors, or during C API calls. Concurrency in the processing of events will be handled via the event state machine. There will be a hierarchy of result codes for each event, with some indicating a retry later on. This will be done to accommodate resources being temporarily unavailable.

An event queue for the DMD will exist in the database. This will be used to post events to create a zone, delete a zone, reload name server configuration etc. Each event will processed it its own python thread, and its parameters status, and results recorded in the event record. The event table structure will contain a parameters column and results column with input and output encoded in JSON that maps easily to python native data structures.

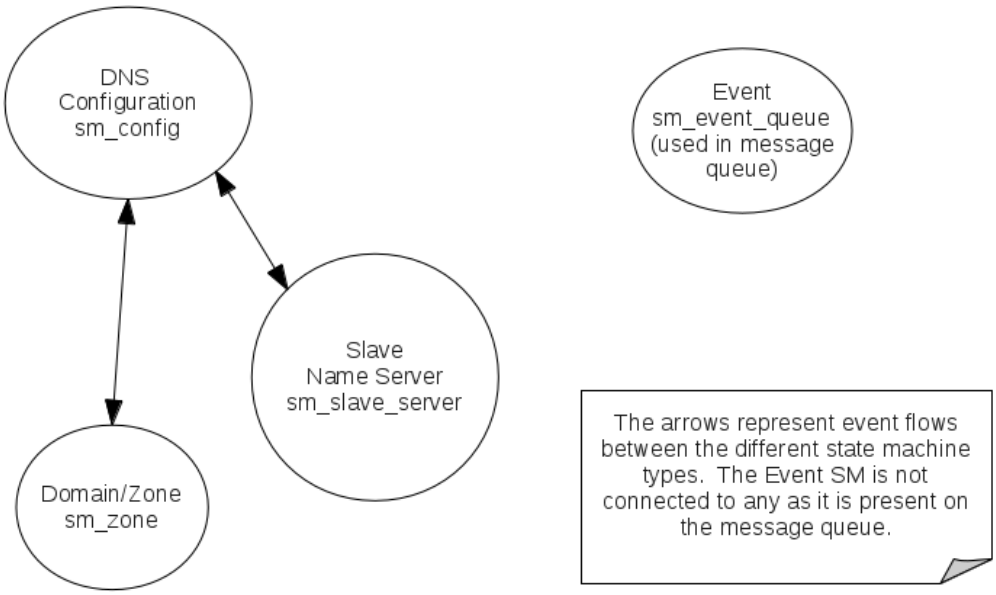
Event creation, processing start and completion time would be recorded in separate columns, along with host name information. This can then be combined with the databased name server log messages, and DMD log output so that problems can start to be diagnosed from the DMD console web page.

State Machine Types

There are 5 state machine types we need for DMS.

#sm_event - An Event state machine that is used on the message queue for net24dmd,
#sm_zone - a zone state machine for configuring a domain into the hidden master name server,
#sm_masterns - a master name server state machine for its configuration and control,
#sm_slavens - a slave name serve state machine for its configuration and control,
#sm_zonestatus - a zone status state machine for indicating to users what is happening with their zones,
#sm_config - an overall configuration state machine for the net24dmd program.

They are related as follows:



Each state machine class is instantiated as a callable Python object, taking an event name as an argument. Internally, each transition is implemented as a hidden method, indexed into a state/event dictionary structure that emulates a 2D array. The state of the object can be queried via public methods.

Logging.

Bind will be set up to log significant information to rsyslog, which will then be feed into the management database. The DMD daemon will also log status, information and debug messages into the same tables for administration and triage purposes.

Network and Host

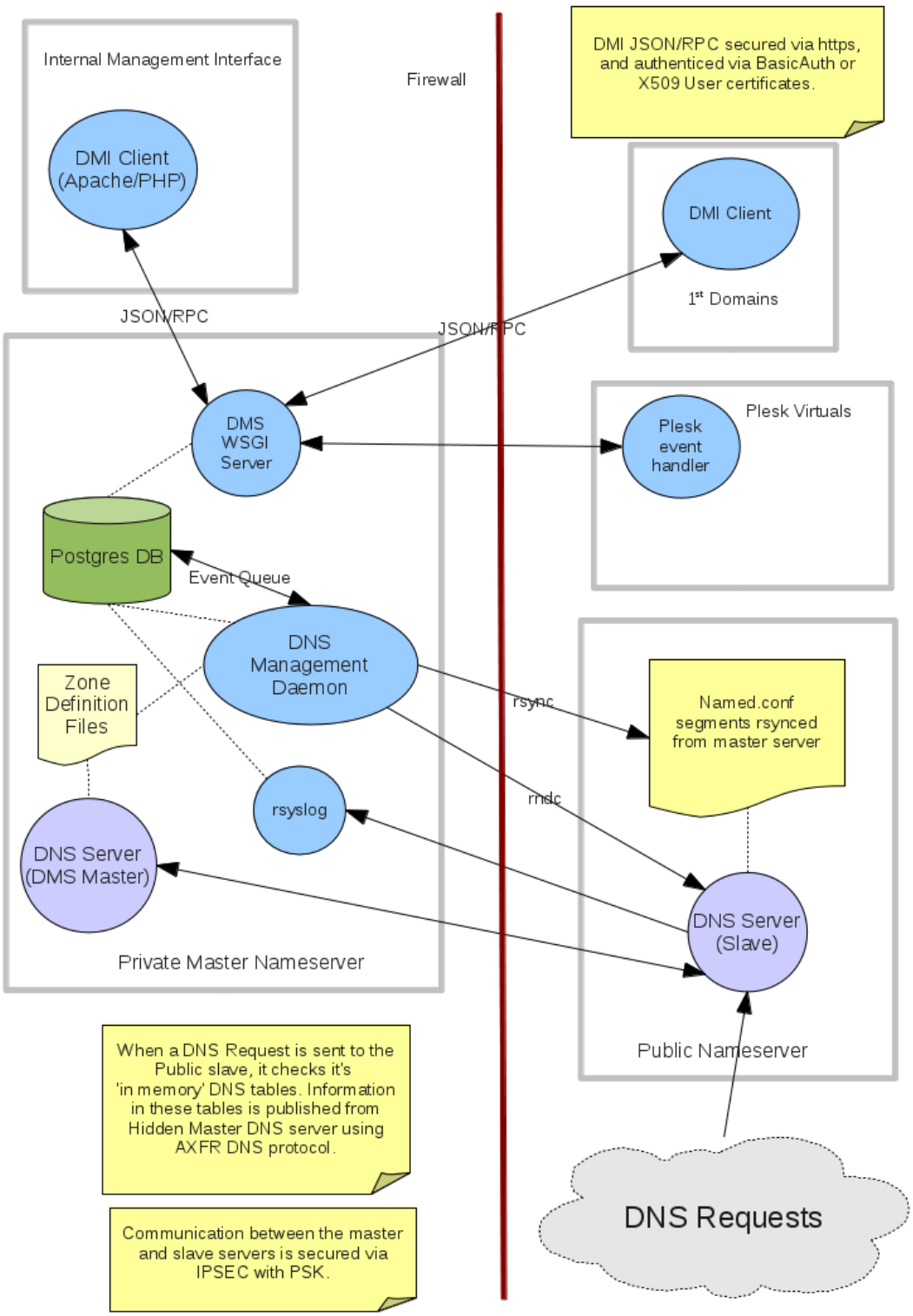
DNS Master server will be a hidden master behind the firewall at Net24.

Network protocols and security

DNS and logging traffic between the slave servers outside Net24 will be secured using IPSEC with PSK keying. ipfw filtering and IPSEC SAs will be used to control the traffic that the slave servers accept from the network and Internet. IPSEC SAs will distinguish between public DNS traffic, and zone update and port 53 administrative traffic, and secure the latter. Ie, DNS Traffic from the Master DNS server will be secured using IPSEC. This keeps all the cryptographic verbiage out of the DNS server configurations, and makes them a lot simpler to generate from templates. IP numbers and acls may need to be inserted in the named.conf files to identify the designation of administrative control and updates from the MAster DNS server, but this is a lot easier that having to track of lot of configuration details about TSIG/SIG0 keys for each individual master-slave relationship, and where they are used....

Web UI Framework.

The Web GUI for the DMI will be rendered using ExtJS. Check logic, and business logic will be separated out and not mixed in (as much as possible) with the UI. This is basically a Mode View Controller programming model.



Configuration State Machine

Configuration State Machine

Writing out configuration sections

There are 2 directories in the net24 CONFFIG_DIR, 'master-config-templates' and 'slave-config-templates'. The master directory contains per zone type templates for bind9, and the slave directory per name server type templates for publishing to slave servers over rsync. The configuration files are assembled into various directories, /etc/(bind|namedb)/master-config on the master server, and (/usr/local/net24/var|var/lib/net24)/default-ssg-config. Multiple slave server group capability will be added later.

Seeding DNS Zone File Contents for Dynamic DNS

When a Dynamic DNS zone is first configured, the contents of the zone is published into the bind9 dynamic directory, before the server is configured, which then generates a dyndns-engine update to take it to the PUBLISHED state. That way, there is far less work when getting the zone to work. Any how, for the server to accept the dynamic configuration, a seed file of the apex records for the zone has to be present, so this is an obvious thing to do.

Grouping of Slave Servers

When running multiple slave servers, the zone does not need to be published on all of them. 4-6 servers are sufficient for any one zone. So when we get 20 servers going, it would pay to balance the load across them by spreading the DNS domains around.

Config State Machine

This drives the whole process. After a zone is enabled/disabled, or added/removed, an event is posted on the event queue which triggers the DNS server configuration update process. The main config state machine synchronously executes the master server state machine to configure it, writes each slave server group configuration out, and then triggers each slave server state machine to propagate the updates.

Features and Functional Requirements

DMS/XCMS framework

This functionality previously found in the DMS program, and has become a representation in the database, achieved through record states. It is manipulated using the XCMS PHP framework generated from Anton's XCMS tool.

1. ~~Delayed record changes~~
2. ~~Delayed record removal~~
3. Ability to disable records
4. ~~GEO-DNS?~~
5. Tags / Descriptions
6. ~~Better support for huge zones~~
7. ~~Add/Remove zone affected records only rather then remove all and create new records for every single change.~~
8. ~~Add flag to convert all RDNs in response to FQDNs (something like OutputFQDNs~~)
9. ~~Zone rollback (and forward)~~
10. ~~Server Failover~~
11. Add more record types (LOC, SRV (must not point to cname?), SFP TXT)
12. DNSSEC support (future - no UI currently)

DMI

1. Ajax when adding / editing records
2. Tags / Descriptions
3. Different views: list view, grouped by tags, grouped by record type.
4. Zone Dump Page

5. ~~DNSSEC Key data dump page(s)~~
6. ~~Record Stats~~
7. ~~Record visibility check~~

Uncategorised

- ~~Blackout hosted page/site~~** ~~customer can edit live~~
  - ~~dns record changed automatically~~
  - ~~alerting via SMS (buy credits?), email~~
  - ~~maintainence windows~~
- ~~Geodns~~** ~~repond with ip based on source of query~~
- ~~Reporting~~** ~~top zones by query~~
  - ~~recent changes~~
  - ~~most modified zone~~
- Versioning
  - Undo
  - roll back multiple versions
- ~~Record scheduling~~** ~~change / add / delete~~
- Custom TTLs - Admin function only
  - Reset ttl after x hours/days
- ~~Front page~~** ~~dashboard overview~~
- Access control
  - zone level logins (login with zone name for full read or write access)
  - master administrator user account for each customer with sub user access control. Create sub users with permissions levels on assigned zones. (assign zones to a a user and then set access level eg. read / write / modify / other types)
    **
- ~~API for customers~~** ~~Access to API is controlled by IP address. Customer defines system which will access API. Allow single IP only (not network) set limited number.~~
  - ~~API function is control by access controls area. A user is given APU access what the user can do is defined by access control for that user~~
- IPv6 - Must Support
- ~~Wizards~~** ~~zone importer from file~~
  - ~~zone importer from another name server; wizard will show config needed on server to allow access for zone tansfer~~
  - ~~SPF Record Wizard Creator~~
- ~~UI Functions~~** ~~Zone cloning - create a new zone from existing zone~~
  - ~~Zone templates - creata a new zone from a template~~
  - ~~Clone record - create duplicate of existing record~~
  - ~~Record visibility checker - click on record and ajaz will perform query on several international dns servers, check the visibility and report OK or FAIL and will display TTL information from queried name server. This allows customer to see when record will be visible by this name server.~~
- UI Feel
  - Ajax driven zone management page for add / remove and changes
- ~~Query Tools~~** ~~whois~~
  - ~~Dig~~

- Scalability
  - ~~System must handle million+ zones~~
  - ~~Single set of name servers may not be sufficient. Only X number of zones per server, then provision new server. with different name server names. Example:~~

```
ns1.us.globaldns.net
ns1.au.globaldns.net
ns1.nz.globaldns.net
ns1.uk.globaldns.net
..
ns23.us.globaldns.net
ns23.au.globaldns.net
ns23.nz.globaldns.net
ns23.uk.globaldns.net
```

    ~~Customers don't care, since they use private name server names (their own hostnames for nameservers)~~
  - ~~Single server should be able to handle 100,000 zones based on load from 1stdomains and RegisterDirect system.~~
- Reliability
  - System must provide 100% uptime. DNS is critical and can not fail.
  - DoS - a zone could get DoS'd and affect all other zones on same systems. What can be done to minimise this? If this is DDoS this could be very hard to combat. The best protection against a DoS is to prevent it occuring in the first place, this means not hosting zones which could be subject to DoSs. Eg. Porn zones, militant sites, warez, IRC, crackers, hackers zones etc.

Setting up FreeBSD Slave DNS server

Setting up FreeBSD Slave DNS server

Procedure for setting up FreeBSD as a DNS slave server. This is using FreeBSD built in bind 9.6.x as no special features are required to serve DNSSEC signed zones.

Install FreeBSD

Proceed as normal, install the standard system with kernel sources for later rebuilding kernel for IPSEC. This can be done later on via sysinstall, adding FreeBSD base source, and kernel sources.

Add useful packages.

Quickest way is to go via sysintall, Configure|Packages, for shells/bash, security/sudo, sysutils/screen (useful for doing builds over long distance ssh), editors/vim-lite, net/rsync etc.

IPSEC enabling host.

IPSEC Tools for racoon/setkey

For ipsec-tools install from /usr/ports to specially 0.7.x. Select ADMINPORT (this will be used in the future by racoon-tool), IPv6, STATS, DEBUG(useful when things don't want to talk), DPD, NATT, FRAG, HYBRID, SAUNSPEC.

IPSEC kernel

Configuring FreeBSD kernel for IPSEC

Follow instructions for building a custom kernel:

Custom kernel install

Do the following:

```
# cd /usr/src/sys/amd64/conf
# cp GENERIC IPSEC
```

and append the following to the bottom

```
# IPSEC support
options  IPSEC         #IP security
device  crypto  # Cryptography device
device          enc             # To allow use to filter and use tcpdump
options  IPSEC_DEBUG # IPSEC debugging
# PF extras
device pf
device pflog
device pfsync
```

which turns on kernel IPSEC and pf network filtering.

Do:

```
# cd /usr/src
# screen
# make buildkernel KERNCONF=IPSEC
# make installkernel KERNCONF=IPSEC
```

Reboot of course to pick up new kernel.

racoon-tool

Install the net24 ports tree, and Makefile.local:

```
# screen
# cd /usr/ports
# git clone https://git.devel.net.nz/freebsd-ports/net24.git
# echo "SUBDIR += net24" > Makefile.local
# make index
```

and go and get 2 cups of tea.

```
# cd net24/security/racoon-tool
# make install
```

Setting Up PostgresQL 9.0+

Setting Up Postgresql 9.0+

Install Postgresql from the ports tree, or apt repository, version 9.0 or later.

On FreeBSD, edit /etc/login.conf, and add the following to the bottom of the default section:

```
:charset=UTF-8:\
:lang=en_NZ.UTF-8:
```

Don't forget to add the '\' to the end of the last line to continue, and then run cap_mkdb /etc/login.conf

Initialize the postgresql DB using:

```
# su - pgsql
$ initdb -D /usr/local/pgsql/data --locale=en_NZ.UTF-8 --encoding=UTF8
$ createuser dms
$ createuser -s <your-login>
$ createdb dms
$ psql template1
$ ALTER USER dms ENCRYPTED PASSWORD  'ScoobyDoo';
$ ALTER USER <your-login> ENCRYPTED PASSWORD 'Something-Super-Secret';
```

Then load the dms database schema from sql/dms-2011.sql

```
$ psql -U pgsql < sql/dms-2011.sql
```

That schema also contains all the stored procedures and constraints/foreign-keys for the dms database.

Development DB access

To make work easier and to enable Python DB stuff to work with less fuss add the following to /usr/local/pgsql/pg_ident.conf:

```
# MAPNAME       SYSTEM-USERNAME         PG-USERNAME
net24           <login>                 pgsql
net24           <login>                 <login>
```

And edit /usr/local/pgsql/data/pg_hba.conf to be:

```
# TYPE  DATABASE        USER            CIDR-ADDRESS            METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     ident
map=net24
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
```

This turns on password checking for localhost IP access, and sets Unix socket connections from psql to have no passwords from the command line.

Of course you can also set the following environment variables in .profile/.bashrc

```
PGDATABASE="dms"
PGUSER="pgsql"
export PGUSER PGDATABASE
```

Notes on SQL source code.

The schema was created by using pg_dump:

```
pg_dump -U pgsql -s dms > sql/dms-2011.sql
```

Each stored procedure is defined in a separate SQL file in the sql subdirectory, and will redefine/replace the old stored procedure when loaded using \i from psql:

```
$ psql -U pgsql dms
dms=# \i sql/<stored-procedure-file>.sql
```

After doing this, create a new dump of the database schema:

```
$ pg_dump -U pgsql -s dms > sql/dms-2011.conf
```

and commit it to git. This method of working with the SQL means that the schema dump will create the latest up to date database 'format'.

All the tables sequences, functions etc are created as user pgsql, so that they are owned by the postgresql super user pgsql, and the user dms has been granted SELECT,INSERT,DELETE,UPDATE on tables, and USAGE on sequences.

Web User Interfaces

Web User Interfaces

These will be based on Sencha ExtJS, which is a Javascript application framework. An 'Outlook' like UI would be used consisting of a view dominating the page from the RHS, with a Tab Index on the LHS. The tabs of this would be at the top, and search results would be displayed in a tree like fashion below.

Customer UI

The tabbed views would consist of the Welcome page, Domains, Events, Preferences, and Tools. The Welcome page would have a summary of whats been happening, customers domain status and other summary information. The Events screen would show the recent events, error conditions, and other status information. The Preferences screen would contain some preferences for their account, their zone limit, and the number of zones they have, and a change password.

All panels including user preferences to link off top left tab for high UI consistency. This will help a lot with customer understanding of UI and verbal guidance of customers over the phone.

DNS Zone editing tab to be laid out like a modern spreadsheet, which is a UI paradigm that most customers would understand easily.

Zone search results would display below tab index on left, and screen on right would not change until zone clicked on.

Right panel display programming would be better not connected to tabs and view in left column, except for hints either way about what should be displaying on right. This will make code better structured.

Each DNS Zone would have a per zone records limit of about 1000 which can be ajusted from Help Desk/Admin UI.

Administrative UI

Same as above, except there would be 2 domain tabs, one per customer, and the other all domains we host. Also has tab for all customers, and their details. The customer details should also be accessible from the DNS cusomter zone view probably.

Integration with internal DNS registry DB? - could be useful if it could connect across to that one.

Administrative/favorites tab would be a good idea. Favorites would display as per search results.

Help Desk UI

This would be an adjusted version of the Administrative UI, with a different role access level. Phase one development it will just be a help desk UI with a different access role.