

# The RIPE Database

Sign up today!

# Contents



- **What is RIPE?**
- **What is the RIPE Database?**
- **What information is maintained there?**
- **How to maintain the RIPE Database?**
- **Example entries**

## **Note:**

This document is a compilation of information provided by the RIPE NCC website (<http://www.ripe.net>) and is intended for **internal** usage only.

# What is RIPE?



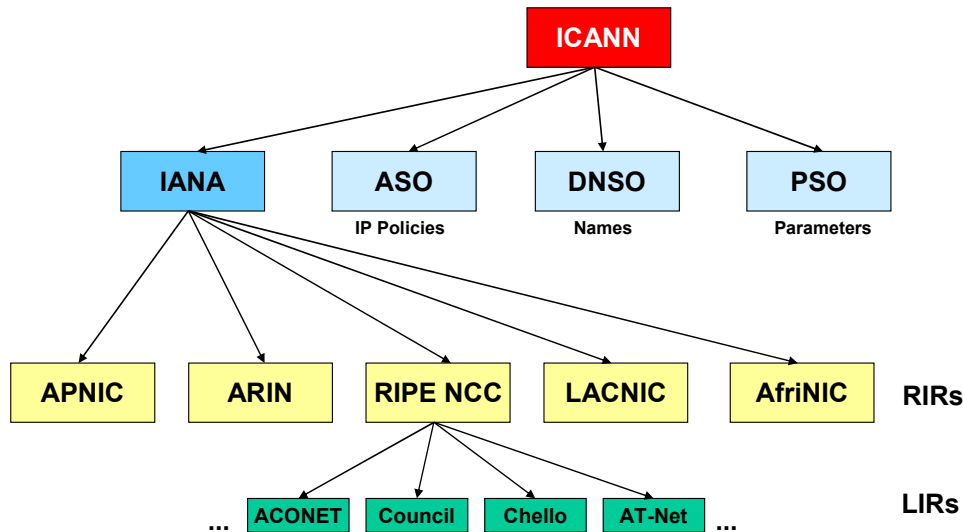
- **Réseaux IP Européens (RIPE)**
  - ♦ Open collaborative community of organizations and individuals, operating wide area IP networks in the "RIPE region"
- **RIPE NCC: Formal organization and RIR**
  - ♦ Founded in 1989
- **RIPE Region: Europe, Middle East, parts of Africa, Asia**

"**RIPE** is a collaborative organization open to all parties interested in wide area IP networks. No membership is required to participate in RIPE and activities are performed on a voluntary basis. The objective of RIPE is to ensure the administrative and technical co-ordination necessary to enable the operation of a pan-European IP network. RIPE is not a legal entity and as such has no legal power." (Source: RIPE FAQ)

The **RIPE Network Coordination Centre** or **RIPE NCC** is an association with a **formal** membership primarily comprised of Internet Service Providers representing 109 countries in its service region. The RIPE NCC services and co-ordination activities support the operation of the Internet infrastructure. The RIPE NCC is one of four Regional Internet Registries (RIR). RIPE NCC is a Regional Internet Registry (RIR) which assigns IP address blocks received by the IANA.

The **RIPE region** was originally only Europe but today also the Middle East, and parts of Africa and Asia are part of it.

# ICANN, RIRs, and LIRs



(C) Herbert Haas 2005/03/11

4

After foundation of the ICANN, the **Internet Assignment Numbers Authority (IANA)** is only responsible for IP address allocation to Regional Internet Registries (RIRs).

Other sub-organizations of the ICANN:

**Address Supporting Organization (ASO)**, which was founded by APNIC, ARIN, and RIPE NCC, and should oversee the recommendations of IP policies

**Domain Name Supporting Organization (DNSO)** is responsible for maintaining the DNS

**Protocol Supporting Organization (PSO)** is responsible for registration of various protocol numbers and parameters used by RFC protocols

Originally, all tasks of these sub-organizations were performed by the IANA only. Today the IANA only cares for address assignment to the RIRs.

The slide above shows a few of the long list of LIRs in Austria. These LIRs are those who are widely known by Internet users as "Internet Service Providers".

# Why a RIPE Database?



- **The purpose of the RIPE Database is to **assist** the operation of the Internet in the RIPE region**
  - ♦ IP address allocations & assignments
  - ♦ Routing policies
  - ♦ Reverse delegations (in-addr.arpa)
  - ♦ Forward domain names
  - ♦ Contact information
- **No white pages service!**
- **No commercial usage of the database!**

Any data is put into the database by those who operate IP networks within the RIPE region. The RIPE NCC supports the operation of the RIPE Database, but it is not responsible for its contents.

The data is accessible by a whois service (whois.ripe.net). Thus, the RIPE database is sometimes called the RIPE Whois Database.

The RIPE NCC does not own the data which is in the database and does not make any changes. Instead RIPE encourages users (LIRs) to change it by themselves or to ask the organization that created the object to do it.

The RIPE Database is not a "white-pages" service; person objects registered in the RIPE Database should be referenced from other objects in the database, either directly or through a role object. As the RIPE copyright document states:

*"Any use of this material to target advertising or similar activities is explicitly forbidden and will be prosecuted."*

Use of information from the RIPE for activities such as, but not restricted to, marketing, advertising and the sending of unsolicited commercial e-mail (UCE) is covered by the copyright.

<http://www.ripe.net/ripenncc/pub-services/db/copyright.html>

## Near-real-time Mirror



- **Improve the overall RIPE database performance by running a copy of the RIPE Database at your site**
  - ◆ Keeps up-to-date with the master server by regularly downloading the most recent updates
- **Requires authorization**
  - ◆ Contact [ripe-dbm@ripe.net](mailto:ripe-dbm@ripe.net) explaining why you want to become a NRTM

# Finding Information



- **Information Search: Whois Client**
  - ◆ Simple search
  - ◆ Advanced search
- **Whois client**
  - ◆ Originally only for UNIX OS
- **Whois server**
  - ◆ Telnet interface available
  - ◆ Port 43

Simple Search: <http://www.ripe.net/ripenc/db/whois/whois.html>

Advanced Search: <http://www.ripe.net/perl/whois>

# Managing Database



- The RIPE database consists of **objects**
- **Email Interface**
  - ♦ Create
  - ♦ Modify
  - ♦ Delete
- **auto-dbm@ripe.net** → RIPE robot
- **ripe-dbm@ripe.net** → bug reports & questions

The RIPE database is simply edited via an email interface which allows a user to create, modify, or delete objects. Any update is sent to the RIPE robot, reachable at [auto-dbm@ripe.net](mailto:auto-dbm@ripe.net).

# Structure of Database Objects



- List of attribute-value pairs
- Primary key is always the first attribute of the object
  - ◆ One exception: route object
- **Primary key** = attribute that uniquely identifies an object

Some attributes are specific to the type of object such as the status in **inetnum** object. Other attributes contain auxiliary [contact] information, such as admin-c, tech-c, mnt-by, remarks, etc, which are common to most objects.

The order of the attribute-value pairs is defined by the owner. There are different types of objects which are specified by the **class** attribute. The class attribute is always the first attribute

## Update an Object



- **Obtain a copy of the object as it appears in the RIPE database using whois**
- **Change the attributes that you need modified to their new values**
  - ◆ You cannot change key attributes
- **Finally, send the object as a plain-text e-mail to `auto-dbm@ripe.net`**
  - ◆ Follow any authentication requirements

The easiest way to update an object is to first obtain a copy of the object as it appears in the RIPE database. This can be done through a normal query, either using the Whois interface or the Whois web interface.

Once you have the object, change the attributes that you need modified to their new values. It is not possible to change key attributes such as a person name or NIC handle (nic-hdl) values.

Finally, send the object as a plain-text e-mail to `auto-dbm@ripe.net`. If the object is maintained (that is, if it has one or more "mnt-by:" attributes), then you must meet the authentication specified by one of the maintainers.

## Delete an Object



- **Get a copy of the object via whois**
- **Append a new line to the object:  
"delete: <reason>"**
- **Send it to: auto-dbm@ripe.net**
- **Don't change anything in the object,  
nor add or delete anything else**

To delete an object from the RIPE Database, you should first get this object by querying the RIPE Database.

Now, append a new line to the object "delete: <reason>" and send it to *auto-dbm@ripe.net*.

**Example:** "delete: Not needed anymore" (without the quotes)

Note: Don't change anything in the object, nor add or delete anything else. This also applies to "changed:" attributes. A new "changed:" attribute should not be added.

# Protect an Object



- **Optionally protect your objects using a mntner (maintainer) object**
- **Add a "mnt-by:" attribute to your object(s)**
- **Now, any change of your objects issue a notification email**
- **Changes can be authenticated**
  - ◆ **Several authentication schemes supported (e.g. PGP)**

To add a **mntner** object to the RIPE Database, simply query the RIPE Database with "-t mntner". Then make a copy of the output and write the correct details into this object and send it to *auto-dbm@ripe.net*.

**Note:** If a person object is not referenced by a network (inetnum , aut-num, as-macro, domain, inet-rtr , etc.) object in the RIPE Database, it is not possible to have a mntner object protecting this person object.

The mntner has to be created with an authentication **other than PGP**. After that you can create the key-cert object protected with the new mntner. Upon creation of the mntner and the key-cert object (protected by your mntner), you can change the authentication to PGP.

RIPE NCC uses **GnuPG** to implement its PGP operations. A **key-cert** object holds the public part of your key in the RIPE Database.

The most straightforward way is to use gpg from the command line. Write your update to a file (say, update.txt). Sign this file with the command `gpg --clearsign update.txt`. You'll be required to enter the passphrase. Then gpg will create a file update.txt.asc which contains the signed version of update.txt. Mail update.txt.asc to *auto-dbm@ripe.net*. You can also use your mailer software facilities to do this which is mostly a menu entry.

## **If you get a "Notification of RIPE Database changes"**

The message you received has been automatically sent to you from the RIPE Database software. Its purpose is to inform you that some objects have been changed in the RIPE Database. You received the message because you were either listed in a "notify:" attribute of the object itself, or in a "mnt-nfy:" attribute of the mntner which protected the object.

If you did not create or update the object yourself, then someone else did this. The e-mail address from which the update came is usually mentioned in the notification message. Please contact that e-mail address for questions and clarifications about the update.

## Creating a route object



- **Three phases of authentication checks when creating a route object**
  - ◆ Check with the relevant aut-num object
  - ◆ Check with the "IP space"
  - ◆ Check with the "mnt-by:" attribute of the route object itself
- **All of these three phases must be passed individually**

If any of them fails, then the route creation is rejected.

# Inetnum Object



- **Information on IPv4 allocations and assignments**
- **Inetnum object represents range of IPv4 addresses**
- **Status attribute**
  - ◆ Specifies status of the address range
- **Rev-srv attribute**
  - ◆ DNS name server in RFC-1034 format

## Other Inetnum Attributes:

**Mnt-by**, **mnt-lower**, **mnt-routes** attributes

- Name of mntner object
- Mntner object authorizes & authenticates changes

**Mnt-by**

- Controls creation, modification & deletion of object

**Mnt-lower**

- Hierarchical authorization
- Controls creation of object directly below parent object

**Mnt-routes**

- Routing Policy System Security (RPSL)

# RIPE and DNS



- **RIPE NCC does not register domain names**
- **Some country-code TLD registrars require RIPE entries**
- **Main reasons for DNS object entries in the RIPE database:**
  - ◆ **Protection of DNS delegations: Changing a domain requires changing the RIPE DB object first**
  - ◆ **DNS restricts contact to a single email address (SOA record), while RIPE supports a number of contact addresses**
  - ◆ **LIRs are usually familiar with the RIPE DB format, hence use it also for DNS related information**

Currently, the RIPE Database contains non-authoritative information about some domains but this information will be moved to the databases of the national country-code domain registries.

The RIPE NCC does not register the generic domain names, i.e. domain names ending with ".net", ".org", etc.

Some country-code Top Level Domain (ccTLD) registrars require that information about the registered owners of domain names must be stored in the RIPE Database.

There are a couple of reasons why DNS information should be stored in the RIPE database. For instance, you can find out which reverse zones a nameserver serves with an inverse whois query like "whois -r -i nserver <server></server>"

# Reverse DNS



- **The RIPE NCC strongly encourages and advises that all address space which is in use has reverse mapping set up**
  - ♦ **But only registered networks are reverse-delegated by RIPE**
  - ♦ **Only requests from a Local Internet Registry (LIR) are accepted**
  - ♦ **Networks smaller than /24 are not reverse-delegated**
  - ♦ **It is not possible to reverse delegate on non-octet boundaries (e. g. /17, /18, /19)**

Reverse delegation is becoming increasingly important for organizations in order to verify the origin of requests to their servers by looking up the domain name associated with the IP address making the request. This is especially important for services such as FTP and E-Mail.

Customers may not be able to access services if reverse lookup on their host IP numbers is not setup.

Furthermore the feature of reverse lookups is useful for Internet fault-finding and for web servers being able to produce more useful log files.

The 'in-addr.arpa.' domain is the reverse tree for IPv4 addresses. The name derives from 'Inverse (IP) address', and 'ARPA' was one of the organizations behind the creation of the Internet.

The RIPE NCC cannot directly reverse delegate networks smaller than /24 from provider aggregatable blocks. This is due to the fact that RIPE sits high in the reverse delegation tree and would not have enough administrative resources.

Therefore, the LIR holding the block from which the space originates should request reverse delegation of the enclosing /24, and then further delegate to their customers as required.

**Exception:** The RIPE NCC will directly delegate to reverse zones smaller than /24 which correspond to "Provider Independent" IP addresses and do not originate from a LIR IP allocation.

# Changing a Delegation



- Query for the *domain* object in the RIPE DB via whois
- Copy-and-paste the resulting *domain* object into an email
- Change the nserver lines as required to match the new set of nameservers
- Add a *password* field to the end of the object if there is a maintainer on it (i.e. a mnt-by line) which requires a password
- Send the mail to <auto-inaddr@ripe.net>
  - ♦ Include your Registry ID in the mail
  - ♦ Sign the object (use ASCII armouring) if it is protected by a *mntner* using PGPKEY authorisation

## Changing a Delegation:

Query for the *domain* object in the RIPE DB using the whois utility. If the nameservers were originally registered using an *inetnum* object this will not give any result. In that case just delete the rev-srv lines from the *inetnum* object (via < auto-dbm@ripe.net >) and then follow the procedure for a new delegation.

Copy-and-paste the resulting *domain* object into an email.

Change the nserver lines as required to match the new set of nameservers.

Add a *password* field to the end of the object if there is a maintainer on it (i.e. a mnt-by line) which requires a password. Don't leave a blank line between the object and the password line.

Send the mail to <auto-inaddr@ripe.net>. Be sure to include your Registry ID in the mail. Also, remember to sign the object (use ASCII armouring) if it is protected by a *mntner* using PGPKEY authorisation.

## Deleting a Delegation

Query for the *domain* object in the RIPE DB via whois (you can also use your local whois client if you have one available). If the nameservers were registered using an *inetnum* object this will not give any result. In that case, simply ignore the rest of this page. Delete the rev-srv lines from the *inetnum* object (via < auto-inaddr@ripe.net >) and then contact < inaddr@ripe.net > giving as many details as possible of the deletion required.

Copy-and-paste the resulting *domain* object into an email.

Add a line like delete: REASON FOR DELETION to the bottom of the object. Don't leave a blank line between the object and the delete line !

Add a password field to the end of the object if there is a maintainer on it (i.e. a mnt-by line) which requires a password. Again, don't leave a blank line between the object and the password line.

Send the mail to [auto-inaddr@ripe.net](mailto:auto-inaddr@ripe.net) . Be sure to include your registry ID in the mail. Also, remember to sign the object (use ASCII armouring) if it is protected by a *mntner* using PGPKEY authorisation.

# Practical Reverse Delegations



- **Don't use the *rev-srv* lines in the *inetnum* object anymore**
- **Only use the *nserver* lines in the domain object instead!**
- **Don't include *ns.ripe.net* as a secondary nameserver for a /24 reverse delegation**
- **You *must* include *ns.ripe.net* as a secondary nameserver for any /16 reverse delegation**

The *rev-srv* lines are intended to indicate the reverse servers for a particular block of IP addresses. They serve the same function as the *nserver* lines in the domain object for a reverse zone.

Previously, delegation requests could be made using either an *inetnum* object or a domain object. But several problems had been discovered with this, one being inconsistency of the two possible sources. Requests are now only accepted via domain objects, so it's probably best to register reverse servers only in the domain object.

Any *rev-srv* lines in a *inetnum* object in the RIPE DB should be deleted.

## **ns.ripe.net must be listed as a secondary for all /16 domains**

IANA (the Internet Assigned Numbers Authority) has made the RIPE NCC responsible for the 'quality of service' for the RIPE NCC's branch of the *in-addr.arpa* tree. One way for the RIPE NCC to provide this service is to require that *ns.ripe.net* is secondary for all /16 allocations in the RIPE NCC service region.

Having a RIPE nameserver as secondary NS this assures that there will always be a nameserver with good visibility available for each reverse /16 zone.

# Primary RIPE Zones



- **The RIPE NCC is primary for the following zones**
  - ♦ 80.in-addr.arpa.
  - 81.in-addr.arpa.
  - 193.in-addr.arpa.
  - 194.in-addr.arpa.
  - 195.in-addr.arpa.
  - 62.in-addr.arpa.
  - 212.in-addr.arpa.
  - 213.in-addr.arpa.
  - 217.in-addr.arpa.
  - ♦ 6.0.1.0.0.2.ip6.int.
  - 7.0.1.0.0.2.ip6.int.

RIPE is also primary for 13[2-5].24.in-addr.arpa., which are allocated for cable networks only.

# Example Query



## ■ Whois -h whois.ripe.net

```
inetnum:      193.0.0.0 - 193.0.1.255
netname:      RIPE-NCC
descr:        RIPE Network Coordination Centre
descr:        Amsterdam, Netherlands
country:      NL
admin-c:      NN32-RIPE
tech-c:       CREW-RIPE
tech-c:       OPS4-RIPE
status:       ASSIGNED PI
remarks:      used to be two different /24 inetnum objects
remarks:      until 19990305 (ripe-ncc & ripe-meeting)
mnt-by:       RIPE-NCC-MNT
mnt-lower:    RIPE-NCC-MNT
changed:      <hostmaster@ripe.net> 20010119
source:       RIPE
```



## ■ Whois 193.0.0.0

```
route:          193.0.0.0/21
descr:         RIPE-NCC
origin:        AS3333
mnt-by:        RIPE-NCC-MNT
changed:       GeertJan.deGroot@ripe.net 19960812
changed:       ripe-dbm@ripe.net 19970822
changed:       ripe-dbm@ripe.net 19970822
changed:       OKolkman@ripe.net 19980102
changed:       ripe-dbm@ripe.net 19980225
changed:       joao@ripe.net 19980720
changed:       joao@ripe.net 20000908
source:        RIPE
```

**Note:** The "*changed*" line is not updated automatically. You must add a new line when you do a new update. Remember to keep the previous "changed" lines also. If there are many "changed" lines, keep at least the first (the oldest) and the last (the newest) one.



## ■ Whois 193.0.0.0 (continued)

```
role: RIPE NCC Hostmaster Team
address: RIPE Network Coordination Centre
address: Singel 258
address: 1016 AB Amsterdam
address: The Netherlands
phone: +31 20 535 4444
fax-no: +31 20 535 4445
e-mail: hostmaster@ripe.net
admin-c: SAWA1-RIPE
tech-c: PT4253-RIPE
nic-hdl: CREW-RIPE
mnt-by: RIPE-NCC-HM-MNT
changed: hostmaster@ripe.net 20011024
source: RIPE
```

## More Information



- <http://www.ripe.net/ripencc/pub-services/db/>
- Contact [ripe-dbm@ripe.net](mailto:ripe-dbm@ripe.net)