RIPE NCC Database documentation update to support RIPE DB ver. 2.2.1

Marek Bukowy
Janne Snabb

Database Group
RIPE NCC

Document: RIPE–189
Date: January 14, 1999
Supplements: ripe–157

ABSTRACT

This document is a supplement to the original introduction to the use of the RIPE Network Management Database (the ”RIPE Database”), published as document RIPE-157. It contains description of new functionality implemented to the database software since version 2.0.4 to version 2.2.1 (current as of January 1999). Most of the recently implemented changes affect primarily advanced users. Questions and comments about this document should be sent to <ripe-dbm@ripe.net>.

Contents

Introduction

This is a supplement to the document *'RIPE NCC Database Documentation'*, published as ripe-157 with database version 2.0.4, available at `http://www.ripe.net/docs/ripe-157.html`.
It contains description of new functionality implemented to the database software since version 2.0.4 to version 2.2.1 (current as of January 1999).
The numbers in brackets state the version of the software in which a change has been introduced, eg. [since 2.2].
The sections 1.2 and 2.1 through 2.3 are meant to correspond to the original ripe-157 section numbers. The items that can be found there are those discussed in ripe-157, but now changed. For easy referencing, subsections of 1.2 have been also numbered.
The changes to section 2.2 have been listed in subsections that do *not* correspond to the layout of the section in document ripe-157.
Part II contains information that involves more than one section of the document ripe-157. Forward references to the new sections have been placed appropriately throughout part I.
Please, send all comments and questions to <ripe-dbm@ripe.net>.

Acknowledgments

The authors gratefully acknowledge the valuable comments by the following RIPE NCC staff: A.M.R. Magee, M. Kühne, J.L.S. Damas, A. van Aarst.
We would like also to thank J. Zsako, the author of the IETF draft *'PGP authentication for RIPE database updates'*, which was used as the basis for the implementation of PGP support.

Part I
Revised ripe-157 sections

Software Requirements

The new whois client is available from RIPE NCC ftp site at `ftp://ftp.ripe.net/tools/ripe-whois-tools-2.4.tar.gz`.
It reflects changes made to the RIPE database server.

1 Database contents

1.2 Description of the Database Objects

The `inetnum` object

```
inetnum:     [mandatory]  [single]     [primary/look-up key]
netname:     [mandatory]  [single]     [look-up key]
```

```
descr:        [mandatory]  [multiple]   [ ]
country:      [mandatory]  [multiple]   [ ]
admin-c:      [mandatory]  [multiple]   [inverse key]
tech-c:       [mandatory]  [multiple]   [inverse key]
rev-srv:      [optional]   [multiple]   [inverse key]
status:       [mandatory]  [single]     [ ]
remarks:      [optional]   [multiple]   [ ]
notify:       [optional]   [multiple]   [inverse key]
mnt-by:       [optional]   [multiple]   [inverse key]
mnt-lower:    [optional]   [multiple]   [ ]
changed:      [mandatory]  [multiple]   [ ]
source:       [mandatory]  [single]     [ ]
```

`status` attribute is now mandatory in `inetnum` objects. For details, please see document *'European Internet Registry Policies and Procedures'* (ripe-185 at the time of writing, available at `http://www.ripe.net/docs/ripe-185.html`). This document describes the policies and procedures applied when a Local Internet Registry assigns an address range.
The `rev-srv` attribute has become an inverse look-up key.

1.2.1 The Domain Registry

The `domain` object

```
domain:       [mandatory]  [single]     [primary/look-up key]
descr:        [mandatory]  [multiple]   [ ]
admin-c:      [mandatory]  [multiple]   [inverse key]
tech-c:       [mandatory]  [multiple]   [inverse key]
zone-c:       [mandatory]  [multiple]   [inverse key]
nserver:      [optional]   [multiple]   [inverse key]
sub-dom:      [optional]   [multiple]   [inverse key]
dom-net:      [optional]   [multiple]   [ ]
remarks:      [optional]   [multiple]   [ ]
notify:       [optional]   [multiple]   [inverse key]
mnt-by:       [optional]   [multiple]   [inverse key]
mnt-lower:    [optional]   [multiple]   [ ]
refer:        [optional]   [single]     [ ]
changed:      [mandatory]  [multiple]   [ ]
source:       [mandatory]  [single]     [ ]
```

[since 2.2] The `domain` objects may now have a `refer` attribute, thus enabling the new *referral mechanism*. That mechanism is described in detail in section 4 of part II.
The `nserver` and `sub-dom` attributes have become inverse look-up keys.

1.2.2 The Routing Registry

[since 2.1.3] Two new attributes have been defined in `aut-num` and `route` objects to support *cross-notification*, added for tracking the announcements of routes.

Please see section 2.3.4 later in part I for a description of that mechanism.

The `aut-num` (Autonomous System, AS) object

```
aut-num:      [mandatory]  [single]     [primary/look-up key]
as-name:      [optional]   [single]     [ ]
descr:        [mandatory]  [multiple]   [ ]
as-in:        [optional]   [multiple]   [ ]
as-out:       [optional]   [multiple]   [ ]
cross-nfy:    [optional]   [multiple]   [inverse key]
cross-mnt:    [optional]   [multiple]   [inverse key]
interas-in:   [optional]   [multiple]   [ ]
interas-out:  [optional]   [multiple]   [ ]
as-exclude:   [optional]   [multiple]   [ ]
default:      [optional]   [multiple]   [ ]
guardian:     [optional]   [single]     [ ]
admin-c:      [mandatory]  [multiple]   [inverse key]
tech-c:       [mandatory]  [multiple]   [inverse key]
remarks:      [optional]   [multiple]   [ ]
notify:       [optional]   [multiple]   [inverse key]
mnt-by:       [mandatory]  [multiple]   [inverse key]
changed:      [mandatory]  [multiple]   [ ]
source:       [mandatory]  [single]     [ ]
```

The `route` object

```
route:        [mandatory]  [single]     [primary/look-up key]
descr:        [mandatory]  [multiple]   [ ]
origin:       [mandatory]  [single]     [primary key]
hole:         [optional]   [multiple]   [ ]
withdrawn:    [optional]   [single]     [ ]
comm-list:    [optional]   [multiple]   [ ]
advisory:     [optional]   [multiple]   [ ]
remarks:      [optional]   [multiple]   [ ]
cross-nfy:    [optional]   [multiple]   [inverse key]
cross-mnt:    [optional]   [multiple]   [inverse key]
notify:       [optional]   [multiple]   [inverse key]
mnt-by:       [mandatory]  [multiple]   [inverse key]
changed:      [mandatory]  [multiple]   [ ]
source:       [mandatory]  [single]     [ ]
```

1.2.3 Contact information

The `role` object

The previous edition of the documentation contained a typographical error, marking `role` attribute as optional. The `role` attribute is mandatory.

```
role:        [mandatory]  [single]     [primary/look-up key]
address:     [mandatory]  [multiple]   [ ]
phone:       [optional]   [multiple]   [ ]
fax-no:      [optional]   [multiple]   [ ]
e-mail:      [mandatory]  [multiple]   [look-up key]
trouble:     [optional]   [multiple]   [ ]
admin-c:     [mandatory]  [multiple]   [inverse key]
tech-c:      [mandatory]  [multiple]   [inverse key]
nic-hdl:     [mandatory]  [single]     [primary/look-up key]
remarks:     [optional]   [multiple]   [ ]
notify:      [optional]   [multiple]   [inverse key]
mnt-by:      [optional]   [multiple]   [inverse key]
changed:     [mandatory]  [multiple]   [ ]
source:      [mandatory]  [single]     [ ]
```

The key certificate object

A new type of object has been added to support PGP authentication:

```
key-cert:  [mandatory]  [single]     [primary/look-up key]
method:    [generated]  [single]     [ ]
owner:     [generated]  [multiple]   [ ]
fingerpr:  [generated]  [single]     [ ]
certif:    [mandatory]  [single]     [ ]
remarks:   [optional]   [multiple]   [ ]
notify:    [optional]   [multiple]   [inverse key]
mnt-by:    [mandatory]  [multiple]   [inverse key]
changed:   [mandatory]  [multiple]   [ ]
source:    [mandatory]  [single]     [ ]
```

The PGP support is described in section 3, part II of this document.

1.2.4 Attributes used in all objects

Layout of `changed` and `withdrawn` attributes

[since 2.1] The syntax for `changed` and `withdrawn` attributes has been changed to allow 4 digits in the year part of the date (8 digits in total rather than 6).
[since 2.2] Six digit dates supplied by the user in the `changed` and `withdrawn` attributes are always first converted to 8 digit format.
Existing 6 digit dates in customer data are left unchanged in the database.

Layout of other dates

[since 2.2] All the dates generated by the software are now in Y2K compliant YYYYMMDD format.
Date comparison has been fixed to correctly compare two dates when one is in 6 digit format and another is in 8 digit format. This bug made it previously impossible to use

6 digit dates in the `changed` attribute any more if there was already one such attribute with date specified with 8 digits.
Generally, all years specified with two digits are handled in the following way:

nn $\geq$ **70** becomes 19nn

nn $<$ **70** becomes 20nn

Thus, for example, year specified as 99 will be treated as 1999, and 69 will be understood as 2069.

2 Database queries and updates

2.1 Querying the RIPE Database

A number of changes has been made to the options of RIPE whois service.

Verbose template

[since 2.1] Another option for displaying templates of objects has been implemented, aside of `-t`.
`whois -v <object type>` shows a template with a description of the object and explanation of attributes.

`changed` attribute

[in 2.1.3] The default behaviour of the whois daemon was changed not to display the `changed` attributes in any objects as result of a query unless specifically requested to do so, by using the new `-c` whois query option.
[since 2.2] That functionality has been reverted to the previous state, i.e. all lines containing `changed` attributes are displayed.
The `-c` flag is accepted and quietly ignored. **There is no way to suppress display of the `changed` attributes now**.

Referral support

[since 2.2] The new `whois` client program understands the `-R` option used to partially suppress referral mechanism. Please see section 4 in part II for details.

2.1.3 WAIS support discontinued

The support for **WAIS** full-text search has been discontinued in favour of **Glimpse**.
The interface to the Glimpse full-text search of the RIPE database is available from the RIPE database home page at `http://www.ripe.net/db/`.

2.1.5 E-mail whois interface discontinued

The support for **e-mail** interface to whois service at <whois@ripe.net> has been discontinued.

2.2 Creating, Updating and Deleting an Object

2.2.1 Handling of the `changed` attribute

[in 2.1.3] Comparison of objects which is performed when processing updates and deletions ignored the `changed` attributes.
Thus, a deletion of an object supplied without `changed` attribute lines would be succesful, if other attributes matched.
On the other hand, if an updated object was sent that differed from the existing version only in the `changed` attributes, no operation would be performed and a NOOP would be returned.
[since 2.2.1] Now the original functionality has been restored. The `changed` attributes are again treated like any other attributes, so in order to delete an object, the object must be quoted with `changed` lines.

2.2.2 Duplicate person object warning

[since 2.2] While creating a new person object, a check is performed if there exists another person object with the same name in the database. A warning is issued if such object is found.
It is also checked if contact data in both objects are identical - address, phone and fax attributes are compared, case– and whitespace–insensitive. In case that another object contains identical data, it is mentioned in the warning text.

Example

Suppose that objects for the same person exist in the database, with nic-handles ABC123-RIPE and AB1234-RIPE, containing different information but assigned the same name. If you try to add a copy of ABC123-RIPE using AUTO-1 as a nic-handle (and thus requesting creation of a new object), the following warning will be issued:

```
WARNING:     Other person object(s) with the same name exists:
WARNING:     ABC123-RIPE(same contact data too)
WARNING:     AB1234-RIPE
```

2.2.3 Notifications

[since 2.1] Whenever an object is created with a `notify` attribute, notification is sent to that address, as well as the usual acknowledgement.
[since 2.1] New concise format for acknowledgements and notifications has been introduced. All acknowledgments are now of type LONGACK, so specifying LONGACK in subject line of the e-mail message is no longer necessary.

2.2.4 Referential integrity checks

[since 2.2] New checks have been implemented to ensure integrity of the database.
While *creating or modifying* an object which has references to other objects in its `admin-c`, `tech-c`, `zone-c`, `cross-nfy`, `mnt-by`, `cross-mnt`, or `origin` attributes,

these referenced objects are checked to see if they exist in the database. The update is refused if reference to nonexisting object is found.
While *deleting* `person`, `role`, `mntner` or `aut-num` object a reverse lookup is performed to find objects referencing the object being deleted. If any objects are found that reference the object in question, then the delete operation is refused and an error message is issued stating the number and types of objects referencing the object being deleted.
Currently, only the RIPE database is checked for references – other sources (RADB etc) are not searched.

Object reordering principle

If an update contains one or more AUTO-n nic-handles, a reordering of objects takes place to allow using those nic-handles as references in other objects of the same update. The objects are then processed in the following order:

1. objects that do not contain AUTO-n nic-handles at all
2. `person` objects to be created with `nic-hdl:  AUTO-n`
3. other objects containing AUTO-n references

Replacing a contact person with a new one

Therefore, the correct way for replacing a contact person using the AUTO-n feature is to send in **two** messages.

1. The first message:
   - creating the new `person` object with AUTO-1,
   - updating the object referencing the newly created person by AUTO-1.
2. The second message:
   - deleting the old person object.

The messages must be processed in the correct order. The easiest way to ensure that is to wait until the notification of first update is returned before sending the second message.

Replacing a contact person with an existing one

In case that no AUTO-n are used, eg. when no new person objects are being created, the reordering principle does not apply. The update can be performed by sending only one message. However, the deletion(s) still have to be placed at the end of the message.

2.2.5 Refined parsing of subject line keywords

The code for parsing subject lines of mail messages sent to <auto-dbm@ripe.net> has been substantially changed.
[since 2.1] A minor bug was fixed, thus enabling the intended way of parsing subject line keywords only if they appeared as complete words, eg. 'new object' and not 'My Apple Newton'. Also made to be case-insensitive.
[since 2.2] The code for parsing subject line keywords has been fully rewritten.
Keywords will trigger options ONLY if they are alone in subject line, i.e. no non-keyword words are found. Otherwise, the **whole** subject line will be ignored and a warning message will be issued.

Recognized keywords

The following keywords are recognized:

NEW	make sure the object does not exist
HELP	send help text
HOWTO	send (the same) help text
ASSIGN (*)	make sure the inetnum does not exist
LONGACK (**)	send long acknowledgment

(*) **ASSIGN** is now obsolete. This functionality has been dropped, since a general **NEW** keyword exists. If **ASSIGN** is found in subject line, it will be ignored and a warning message will be issued, but all other valid keywords will be processed.

(**) **LONGACK** is now the format of all acknowledgments, so the keyword is silently ignored.

Multiple keywords

In addition, to prevent ambiguity when multiple keywords are specified, a list of allowed combinations of keywords has been defined. If the combination of recognized keywords is not one of:

ASSIGN NEW
ASSIGN LONGACK
ASSIGN NEW LONGACK
LONGACK NEW
HELP HOWTO

then the **whole** subject line will be ignored and a warning message will be issued.

Examples of keyword usage

If a mail message is sent to <auto-dbm@ripe.net> saying

```
Subject: NEW person object for John
```

then the following warning will be issued:

```
Warning: unknown keywords found in subject line:
person object for John
Thus, all keywords in subject line were ignored.
```

If a mail message contains:

```
Subject: NEW LONGACK ASSIGN
```

then the following warning will be issued:

```
Warning: obsolete keyword ASSIGN found in subject line was ignored.
```

There will be no warning for LONGACK, but it will not change the layout of the acknowledgment in any way. The NEW functionality will be activated.
Finally, if a mail message contains:

```
Subject: NEW HELP
```

and no objects in the message body, then the following warnings will be issued:

```
*** No objects were found ***
Warning: this combination of keywords in subject line is not allowed.
Thus, all keywords in subject line were ignored.
```

2.3 Object Protection

2.3.3 Authentication schemes

A new scheme has been added for PGP support. The valid schemes are now:
`NONE, MAIL-FROM, CRYPT-PW` and `PGPKEY-<key-id>`.
Please note that `PGPKEY` differs from the convention used with schemes `MAIL-FROM` and `CRYPT-PW` in that the first blank-delimited word already contains the authentication information, so instead of saying eg. `auth: CRYPT-PW <argument>` for PGP you say `auth: PGPKEY-<argument>`.
See section 3 in part II for details on using PGP.

2.3.4 Cross notification in the Routing Registry

[since 2.1.3] A new feature has been added to track the registration and removal of route objects which overlap existing routes.
Although it does not prevent unintended changes, it still provides some protection of data by notifying administrators of affected routes.

Feedback to the sender

Whenever a route object is added to the routing registry, a check is done to see if the given prefix overlaps with that in any other route object in the RR. If so, a notification will be sent to the sender of the route object.
This will always happen and is not dependent on any notification attributes described below.

Feedback to other parties

Two new optional, multiple attributes called `cross-nfy` and `cross-mnt` have been added to the route and aut-num objects.
Both attributes contain a contact reference, the difference being that

`cross-nfy` contains a NIC-handle pointing to a person or role object, the email address of which will be used for notification;

`cross-mnt` contains a mntner ID referencing the maintainer to be notified.

These attributes are used to request notifications about the addition or removal of route objects which overlap the prefixes of the already registered route objects.
The `cross-nfy` or `cross-mnt` attributes may be used in:

`route` object – to get a notification for any overlaps with the prefix specified in that route object;

`aut-num` object – to get a notification for overlaps with any of the prefixes announced in route objects in which the given AS number is specified in the `origin` attribute.

A notification will be sent to mailbox(es) listed in `cross-nfy` and mailbox(es) listed in the `mnt-nfy` attributes of the maintainers referenced by the `cross-mnt` whenever an overlapping route is **added** or **removed**.
In case of an intentional overlap, the former will indicate the introduction of multihoming, and the latter, its cancellation. It is expected, however, that the former will often indicate a mistake, and the latter its correction. Either way, it will be useful to know about the registration and elimination of overlapping announcements.
Currently, only RIPE database is searched for overlaps. Mirror and remote databases, such as RADB, are not checked.

Example of cross-notification

Supposing that the following `aut-num`, `mntner` and `route` objects exist in the database:

```
aut-num:    AS3333
.......
cross-mnt:  RIPE-NCC-MNT


mntner:     RIPE-NCC-MNT
.......
mnt-nfy:    as-maint@ripe.net


route:      193.0.0.0/24
.......
origin:     AS3333
```

If an update containing a `route` object, even originating from some other AS, but overlapping the 193.0.0.0/24 prefix, is sent to <auto-dbm@ripe.net>:

```
route:      193.0.0.128/25
.......
origin:     AS12345
```

then a notification is sent to <as-maint@ripe.net>, because the specified route overlaps with one of the routes originating from AS3333.

Part II
New features of the RIPE database

3 PGP authentication support

[since 2.2] Support for authentication with PGP signatures has been added. This is the most secure way of protecting objects from un-authorised modification. The current implementation supports DSS/Diffie-Hellman and RSA algorithms.

3.1 Supplying public keys to the server

A new `key-cert` object has been introduced to allow supplying of PGP public keys and storing them on the server. The object template has the following attributes:

```
key-cert:  [mandatory]  [single]     [primary/look-up key]
method:    [generated]  [single]     [ ]
owner:     [generated]  [multiple]   [ ]
fingerpr:  [generated]  [single]     [ ]
certif:    [mandatory]  [single]     [ ]
remarks:   [optional]   [multiple]   [ ]
notify:    [optional]   [multiple]   [inverse key]
mnt-by:    [mandatory]  [multiple]   [inverse key]
changed:   [mandatory]  [multiple]   [ ]
source:    [mandatory]  [single]     [ ]
```

The `key-cert` attribute is defined as `PGPKEY-<id>` where `<id>` is the PGP key ID of the public key included in the object in the usual eight digit hex format without ”`0x`” prefix.
The public key should be supplied in the `certif` attribute. Usually this is easily done by exporting the key from your local key ring in ASCII armored format and prepending each line of the key with a string ”`certif:`”. Remember to include all the lines of the exported key; also the begin and end markers and the empty line which separates the header from the key body.
Please note that RPSL line continuation (whitespace or plus at the beginning of a continuation line) is not currently supported and thus can’t be used when supplying `key-cert` objects.
The attributes marked as `generated` (`method`, `owner` and `fingerpr`) are generated by the software and they can be omitted by the user when supplying a key. The server always derives these attributes from the actual key and ignores user input.
The other attributes (`remarks`, `notify`, `mnt-by`, `changed` and `source`) have their usual meanings as defined in RIPE-157.

This is an example of a valid `key-cert` object as it might appear in the database:

```
key-cert: PGPKEY-4B8AE00D
method:   PGP
owner:    Joe User <joe@example.net>
fingerpr: 9D 82 4B B8 38 56 AE 12  BD 88 73 F7 EF D3 7A 92
certif:   -----BEGIN PGP PUBLIC KEY BLOCK-----
certif:   Version: 2.6.3ia
certif:
certif:   mQA9AzZizeQAAAEBgJsq2YfoInVOWlLxalmR14GlUzEd0WgrUH9iXjZ
certif:   a/uqWiLnvN59S4rgDQAFEbQeSm9lIFRoZSBVc2VyIDxqb2VAZXhhbXB
certif:   iQBFAwUQNmLN5ee83n1LiuANAQFOFQGAmowlUYtF+xnWBdMNDKBiOSy
certif:   YvpKr05Aycn8Rb55E1onZL5KhNMYU/gd
certif:   =nfno
certif:   -----END PGP PUBLIC KEY BLOCK-----
mnt-by:   EXAMPLE-MNT
changed:  joe@example.net 19981117
source:   TEST
```

Please note that this example also shows the attributes which are automatically generated by the software. Only `key-cert`, `certif`, `mnt-by`, `changed` and `source` need to be specified when adding a new object.
If you do not already have a maintainer object to be used in the mandatory `mnt-by` attribute, you need to create a new `mntner` with some other authentication method (for example CRYPT-PW), then create the `key-cert` object which references the maintainer just created, and after that you can change the maintainer to use PGP authentication with the `key-cert` object as an authentication key.
These `key-cert` objects can be queried for in the usual ways with whois by asking for a specific key as defined in the `key-cert` attribute.
The RIPE NCC does not guarantee that a key belongs to any specific entity; we are not a certificate authority. Anyone can supply any public keys with any ownership information to the database and these keys can be used to protect other objects by checking that the update comes from someone who knows the corresponding secret key.
Please also note that signatures in the keys are ignored. We kindly ask you to limit the number of key signatures to a minimum.

3.2 Changes to the maintainer object

A new value for the `auth` attribute of `mntner` object has been introduced. PGP authentication can be activated by setting `auth` to `PGPKEY-<id>` where `<id>` is the PGP key ID to be used for authentication. This string is the same one which is used in the corresponding `key-cert` object's `key-cert` attribute.
Remember that if you have multiple `auth` attributes in a maintainer or if you have multiple `mnt-by` attributes in an object, all possible authentication methods are combined by a logical OR which means that any single one of the specified authentication

methods can be used. There is no security advantage in using PGP authentication with an object which can be updated also with MAIL-FROM or NONE authentication.
This is an example of a valid maintainer object which uses PGP authentication:

```
mntner:   EXAMPLE-MNT
descr:    Example maintainer
admin-c:  JOE1-RIPE
upd-to:   joe@example.net
auth:     PGPKEY-4B8AE00D
mnt-by:   EXAMPLE-MNT
changed:  joe@example.net 19981117
source:   RIPE
```

3.3 Using authentication when sending updates

PGP signed updates can be sent to the server simply by signing the body of the message which contains the updates and sending it to the server. Remember to use ASCII armoring.
Multiple PGP-signed and non-signed parts can be supplied in a single update message; each part gets processed separately. You can supply several objects which are protected by different PGP keys in a single update message, but you cannot use any ”magic” references like AUTO-1 nic-handles between these parts. Also, the software doesn't currently support recursive PGP decoding. If you sign an already signed message, only the outermost PGP block gets checked.
Please note that there is no MIME support in the current DB software. It just ignores MIME headers and feeds the body of the message to the update engine without any decoding. Thus, you cannot currently use PGP/MIME multipart/signed messages (as specified in RFC 2015) to send updates. The signature must be supplied in the traditional way by signing the text file which contains the objects.
PGP parts with invalid signatures are rejected in all cases, even if the object is not protected by PGP authentication.

3.4 Other PGP details

General information about PGP is available at `http://www.pgpi.com/`.
This implementation is based on the specification done by the RIPE Database Security Task Force, which is also available as IETF draft for possible use in the IETF's RPS working group authentication procedures.

3.5 Legal issues

Please note that encryption technology is subject to legal restrictions in some countries. PGP signatures are based on public key encryption. Consult a lawyer if you are uncertain about your local situation.

4 Referral mechanism for domains

[since 2.2] Since the RIPE database does not contain authoritative data for domains, a mechanism has been implemented to provide redirection of domain name queries to appropriate database servers, if a domain name is not found in the RIPE database itself.
The referral mechanism is implemented via new attribute `refer` in the domain object. That attribute is single and optional:
`refer: <type> <host> [<port>]`
where

`<type>` indicates which style of whois service is provided:

RIPE	whoisd software understanding RIPE options;
InterNIC	InterNIC whoisd software;
SIMPLE	other service understanding query of the form `<domain name>`.

`<host>` is the DNS name of the whois service.

`<port>` is the TCP port number (optional: 43 is the default).

4.1 Domain name stripping

When no domain object is found in the database with the name specified in the query, the domain name is stripped towards higher level domains (xxx.yyy.zzz becoming yyy.zzz) and the lookup is repeated until a domain object is found or search string becomes empty.

4.2 Querying remote hosts

When a higher level domain object is found and it contains a `refer` attribute, the proper referral mechanism is invoked.
A whois query is made to the referred host and its answer is displayed, preceeded by a note that this data does not come from the RIPE database but from a remote server. When the query to the remote host fails, an error message is given.

4.3 Displaying local copy

If requested specifically by a new `-R` option to whois client, the database still strips the domain name until it finds a matching object, but shows this local object instead of handing the query over to the remote server.

4.4 Backward compatibility

Temporarily, the old behaviour of the database software is still provided for domains falling under higher level domains that do not set their domain objects to support referral.
If the domain object found does not contain a `refer` attribute, then the query result is displayed just as it was in the previous software version:

```
% No entries found for the selected source(s).
```

5 Miscellaneous

5.1 RIPE Copyright and related issues

[since 2.1] A copyright message has been added to all source files to better assert the rights of the RIPE community, but the copyright message itself has not changed.

Setting up a near-real-time mirror or a web-interface to the RIPE database requires signing the Acceptable Use Policy agreement. Please, contact <ripe-dbm@ripe.net> for more information.

We do take reports on abuse of the data in the RIPE database seriously. If you have complaints or suspicions, please contact <ripe-dbm@ripe.net>.

The queries made to the database are constantly monitored. In case a suspicious activity is seen, queries originating from the abuser will be blocked.