RIPE

RIPE Database Transition Plan

Tony Bates
Daniel Karrenberg
Marten Terpstra

Document-ID: ripe-123

October, 1994

ABSTRACT

This paper details a transition plan of the changes needed to move from the current RIPE Database to a RIPE Database that supports classless IP network numbers, authorisation of updates and changes to Routing Registry information.

1. Introduction

The RIPE database is about to undergo some major changes. These changes come from a set of documents produced by various RIPE working groups and the RIPE NCC. There are three major changes to the RIPE database that will affect the database in such a way, that a description of the changes and a clear transition plan is needed.

These three changes are:

- o Support for Classless Internet Addresses [1].
- o Authorisation and Notification of Changes [2].
- o Representation of IP Routing Policies in a Routing Registry [3].

All these three new features of the RIPE database will affect the working of the RIPE Database, from simple things like altered output, to more complex things like guarded attributes and objects [4]. The three changes will be dealt with in turn in this document.

If you wish to get a quick overview of how this affects you please refer to *Section 5*.

Each of the sections will have points in time attached to them, at which time a certain change will take effect. These points in time are labeled T1 and T2. For some changes that need to be made, a certain "flag day" is needed. On these days some changes will take place that are not backward compatible and must be performed in a "big bang" type of change. These dates are labeled B1 and

B2. All changes will be labeled with a Tn and/or a Bn.

Table 1 shows the current time estimates for each of these changes.

Date	Big Bang	Time
Now		T0
13-10-94	B1	T1
21-11-94	B2	T2

Table 1: Timescales for transition changes

In the sections describing a change, there will be a small section shortly describing the effect this will have on users of the RIPE database. For this document the users have been categorised in four groups: users querying the database, AS and community guardians, maintainers of "inetnum" objects and maintainers of other database information.

2. Support for Classless Internet Addresses

There are several aspects pertaining to the introduction of classless Internet addresses in the RIPE Database as document in [2].

2.1. Querying the RIPE database

To support classless addresses in the RIPE database, one different representation of Internet addresses will be supported for queries to the database. This notation is the prefix/length notation as explained in [1]. Correct queries to the database will be for example:

```
192.87.45.0/24
192.87.228.0/23
128.141.0.0/16
193.0.0.132/32
```

For backward compatibility, "old" style queries will still be supported. They are however internally to the database server transformed to a prefix/length notation, after which they are handled as above. This transformation could change the expected output. For example:

`192.87.45` will be rewritten internally to `192.87.45.0/32`
`128.141` will be rewritten to `128.141.0.0/32`
`193.0.0.0` will be rewritten `to 193.0.0.0/32`

Other type queries that are supported, but not encouraged are prefix/length queries, for which the prefix is not a full dotted quad:

`128.141/16` will be rewritten to `128.141.0.0/16`
`192.87.45/24` will be rewritten to `192.87.45.0/24`

For queries where the prefix and the length are incompatible (like 192.87.45.0/8), the length will be used to mask off all extra bits from the prefix. Below you can see the difference between queries for 193.1.209.0/24 and 193.1.209.0/8. In the second case, the query is rewritten to

193.0.0.0/8.

```
% whois -r 193.1.209.0/24
inetnum:     193.1.209.0
netname:     NCIR-LAN
descr:       Local Ethernet
descr:       National College of Industrial Relations
descr:       Sandford Road
descr:       Dublin 6
descr:       Ireland
country:     IE
admin-c:     Neil Armstrong
tech-c:      Neil Armstrong
connect:     RIPE NSF
aut-sys:     AS1213
changed:     mnorris@hea.ie 940823
source:      RIPE

% whois -r 193.1.209.0/8
inetnum:     193.0.0.0 - 193.255.255.255
netname:     RIPE-CBLK
descr:       European Address Block #1
country:     EU
admin-c:     DK58
tech-c:      TB230
tech-c:      MT2
remarks:     delegated
changed:     testdb@ripe.net 940707
source:      RIPE
```

The default behavior for the RIPE database will be to respond to the query with an exact match if possible, or otherwise the first less specific match it can find. In most cases this will mean that when queries for host addresses (193.0.0.132/32) one will receive the network block this host is part of as registered in the database (in this case probably 193.0.0.0/24).

```
% whois -r 193.0.0.132/32
inetnum:      193.0.0.0
netname:      RIPE-NCC
descr:        RIPE Network Coordination Centre
descr:        Amsterdam, Netherlands
country:      NL
admin-c:      Daniel Karrenberg
tech-c:       Marten Terpstra
tech-c:       Tony Bates
aut-sys:      AS3333
ias-int:      193.0.0.221 AS1104
ias-int:      193.0.0.157 AS1104
rev-srv:      ns.ripe.net
rev-srv:      ns.eu.net
notify:       ops@ripe.net
changed:      tony@ripe.net 940708
source:       RIPE
```

The RIPE NCC will also insert objects that describe a block delegation to an Internet Registry into the database. For example the block 193.1.0.0 - 193.1.255.255 delegated to HEANET, will have an associated entry in the database. When one would query a network number from this block that has not been allocated by HEANET yet, the server will respond with the object for the complete block. This means that less queries will result in a "No entries found" answer, but will give some indication of where that specific piece of address space currently resides. The same is true for address blocks 193.0.0.0 - 193.255.255.255 and 194.0.0.0 - 194.255.255.255 delegated to the RIPE NCC.

```
% whois -r 193.1.141.0/24
inetnum:      193.1.0.0 - 193.1.255.255
netname:      EU-BLOCK-193-1
descr:        HEAnet
country:      IE
admin-c:      Mike Norris
tech-c:       Mike Norris
remarks:      delegated
changed:      marten@ripe.net 930901
source:       RIPE
```

The database server will also support some extra features to display the complete list of less specific matches, a list of direct more specific matches, and a list of more specific matches including more specifics of more specifics. A flag to indicate "exact match only" is under investigation. The flags to enable these features can be found in the RIPE whois client manual page and will be available in the on-line "help" for the whois server. This can be requested by using "whois -h whois.ripe.net help".

All of these changes will take place at once when enabling the software that supports classless internet addresses at B1. Changes to tools that make use of the database server have to be made before that time. If you are using existing tools and see behavioral changes, retrieve the latest version if it is a RIPE NCC supplied tool, otherwise please make the authors of these tools aware of the changes mentioned in this document.

2.2. Changes to the guardian files

Currently all guardians of guarded attributes maintain files on the RIPE NCC database machine. These files are examined once per day and the associated guarded attribute added to all network objects listed in these files.

When a network block is found in the database , and not all of the networks in this block are mentioned in a guardian file, this block is split so that only the mentioned networks get the associated guarded attribute. This procedure can no longer work in a classless database for two reasons:

- o from a block of network numbers it is no longer clear where one network ends and the next one starts. The class A, B and C boundaries no longer apply, so the software cannot make decisions any longer on where to split a block if necessary;

- o a certain network number can appear more than once in the database in different block entries. In normal cases, any network number assigned through the NCC will appear at least three times in the database: once in the entry describing the allocation to an end-side or user, once as a block delegation from the NCC to a local registry, and once as a block delegation from IANA to the RIPE NCC. It is therefore impossible for the software to find out which of these blocks should be split and where this guarded attribute should be added.

The way this will be solved is to only add guarded attributes to objects if the entry in the guardian file exactly matches the "inetnum" value of an object in the database.

Assume we have a guardian file that contains two lines:

```
192.87.45.0
193.0.0.0 - 193.0.0.5
```

In this case, the guarded attribute associated with this file will NOT be added to entry "`192.87.45.0 - 192.87.46.0`", "`193.0.0.0 - 193.255.255.0`" or any other non-exact match in the database.

Since blocks are no longer split, any line in the guardian file that does not correspond to an exact match in the database, will cause a notification mail message to be sent to the guardian with the explanation that no match could be found in the database.

Because this is a direct effect of the enabling of the classless software, this change will have to be made at B1. This means that all guardians will have to check their files for non-matching entries and correct them.

The NCC will assist all guardians that have an account on the NCC database machine by generating a problem file for them. This file will list all entries in their guardian file that are either not present in the database, or entries that are appear in a bigger block in the database. Any non-exact match can therefore easily be spotted. The program producing these files will run every night up till B1. The file generated will be called "problems" and will be put in the home directory of each of the guardians. Guardians are strongly requested to change their guardian files to eliminate any non-exact match before B1. At point B1, non-exact matches can cause a loss of guarded attributes.

It should be noted that this transition is an interim measure to be used between T1 and T2. A completely new procedure to guard attributes and objects is outlined in [2] and a transition for that proposal is outlined in the next section.

3. Authorisation and Notification of Changes

The RIPE NCC has produced a paper that describes a mechanism to authorise and notify any changes to objects in the RIPE database [2]. Although these changes apply to all objects in the database, there are some transition issues when these mechanisms are used to replace the current guarded attribute and guarded object procedure.

The general use of the authentication and notification will be available from T1.

3.1. Guarded Attributes

As explained in [4] the current procedure for guarded attributes is that guardians maintain files on the NCC database machine, from which guarded attributes are added to objects to the database. Although this has worked reasonably well for over a year, it can be solved with a more general authentication and notification scheme. This scheme is explained in [2].

The introduction of "route" objects in the RIPE database (see below) removes the necessity of guarded attributes in their current form. The primary reason for having guarded attributes (authenticity guarantees because of their operational impact) can be solved by the general authentication scheme.

The transition issues involved are of a bootstrap nature. This transition plan foresees the generation of routes from existing network numbers. This will be explained in more detail in the next section. When these newly generated routes are to be used they should be properly maintained using the proposed mechanism. To bootstrap this procedure and to ensure that all routes are properly maintained, the NCC will generate an example "mntner" object [2] for all guardians that have an account on the NCC database machine. The information for this object will be placed in a file in each guardian's home directory. The file name will be "mntner".
The information in this object will be derived as best as possible from the current guardian information. These objects will be generated once only. The guardian can then change this object in whatever way he/she wishes. The guardian can even decide to already submit the modified "mntner" object generated for them to the database.

At B2, all routes that will be generated from the current network numbers will be maintained by the maintainer mentioned in the file generated in the associated guardian accounts. To ensure that all generated routes are properly maintained, only the "mntner" file in each of the guardian's accounts will be used to tag the newly generated routes. The "mntner" object found in this file will also be automatically entered into the database. Guardians that currently maintain multiple files on multiple accounts can decide to only register with a single "mntner" object. They will have to ensure that the "mntner" objects in all their accounts are equal.

For more details see below. The current guardian procedure will be turned off at B2. At that time, the newly generated route objects should all be maintained, and the need for guardians in the current meaning will have disappeared. All guardian accounts and files will be removed shortly after

B2.

3.2. Guarded Objects

The RIPE database currently has support for guarded objects. These are objects that cannot be updated using the automatic procedure, but need to be manually checked by NCC staff and are then forwarded to the database. Currently the only guarded objects are "community" and "aut-num". This mechanism was put in place to avoid mistakes when updating operationally critical objects like autonomous system objects. It is fairly obvious that this procedure can be replaced using the authentication mechanism described in [2].

At T1, the maintainer mechanism will be enabled and current guarded attributes can start using this mechanism to replace the manual intervention by the NCC currently needed. For a certain time, guarded objects will still be accepted using the old mechanism (mailed to **ripe-dbm@ripe.net** for checking), but whenever a "mnt-by" attribute is present in any guarded object, these objects can be updated automatically. This does however mean it will have to pass the authorisation as specified by the maintainer.

4. Transition to RIPE-81++

There are several important aspects pertaining to the transition of the objects documented in RIPE-81++ [3]. This also includes the related "inet-rtr" object [5].

4.1. Separation of routing from allocation information

This represents a significant change to both the database objects and to actual data in the database. The two database objects affected will be the existing "inetnum" [8] object and the "route" object as detailed in [3]. This will result in some information being moved from the "inetnum" [6] object to "route" object and some information being deleted from the "inetnum" object. If we look at an existing "inetnum" object:

```
inetnum:   192.87.45.0
netname:   RIPE-NCC
descr:     RIPE Network Coordination Centre
descr:     Amsterdam, Netherlands
country:   NL
admin-c:   Daniel Karrenberg
tech-c:    Marten Terpstra
connect:   RIPE NSF WCW
aut-sys:   AS3333
comm-list: SURFNET
ias-int:   192.87.45.80  AS1104
ias-int:   192.87.45.6   AS2122
ias-int:   192.87.45.254 AS2600
rev-srv:   ns.ripe.net
rev-srv:   ns.eu.net
notify:    ops@ripe.net
changed:   tony@ripe.net 940110
source:    RIPE
```

The routing information contained in the "inetnum" object will now be distributed over two objects like so:

```
inetnum:   192.87.45.0
netname:   RIPE-NCC
descr:     RIPE Network Coordination Centre
descr:     Amsterdam, Netherlands
country:   NL
admin-c:   Daniel Karrenberg
tech-c:    Marten Terpstra
rev-srv:   ns.ripe.net
rev-srv:   ns.eu.net
notify:    ops@ripe.net
changed:   tony@ripe.net 940110
source:    RIPE

route:       192.87.45.0/24
descr:       RIPE Network Coordination Centre
origin:      AS3333
comm-list:   SURFNET
remarks:     ias-int: 192.87.45.80  AS1104
remarks:     ias-int: 192.87.45.6   AS2122
remarks:     ias-int: 192.87.45.254 AS2600
changed:     dfk@ripe.net 940427
source:      RIPE
```

The general idea is all routing based information is moved from attributes within the "inetnum" object to the "route" object (1). It is important to note the direct effect this have on the "inetnum" object data. Table 2 below gives a full list of the transitioned attributes. It should also be noted that in an effort to clean up the "inetnum" object at the same time certain attributes will be obsoleted. The effect column has an T for transitioned and an O for obsoleted.

inetnum attribute	effect	route attribute
aut-sys	T	origin
comm-list	T	comm-list
ias-int	T	remark
connect	O	
gateway	O	
routpr-l	O	
bdrygw-l	O	
nsf-in	O	
nsf-out	O	

Table 2: Affected attributes

As can be seen a simple translation can be made and this will be the last major transition step at time T2.

---

(1) The "ias-int" attributes are preserved as remarks. See details of "inet-rtr" object for reasoning behind this.

4.2. Auto-Generation of "Route" and New "Inetnum" objects

Clearly, the transitioning of data will require a major flag day where the translation of attributes detailed in Table 2 takes place. This is expected to occur at big bang B2.

At point B2 the RIPE NCC will automatically generate both new "route" and "inetnum" objects in line with Table 2 and load them into the RIPE database. It should be noted that these changes will only take place to "inetnum" objects with have an associated "aut-sys" attribute. All "inetnum" objects not containing the aut-sys attribute will have both T and O attributes removed. This may mean the potential loss of comm-list information if there is no corresponding aut-sys attribute for the existing "inetnum" object.

Between time T1 and T2 the NCC will generate for each guardian account both the translated "inetnum" and "route" objects in the form of files known as:

new.inetnum

This file will contain a full list of "to be generated" "inetnum" objects.

new.route

This file will contain a full list of "to be generate" "route" objects.

These files will be re-written every night and are purely given to give each guardian of what will happen to any associated object referencing a guarded attribute after B2.

The new "route" objects will be automatically split into CIDR [7] compliant aggregates. Below is a simple example of an automatic object split from the current "inetnum" to a new "inetnum" and "route" object. The original "inetnum" is as follows:

```
inetnum:     193.12.128.0 - 193.12.132.0
netname:     SE-COMVIS-NET
descr:       ComputerVision Sweden AB
country:     SE
admin-c:     Anders Andersson
tech-c:      Anders Andersson
connect:     SWIP
aut-sys:     AS1257
changed:     uffe@swip.net 930707
source:      RIPE
```

The will be translated to an new "inetnum" object as follows:

```
inetnum:    193.12.128.0 - 193.12.132.0
netname:    SE-COMVIS-NET
descr:      ComputerVision Sweden AB
country:    SE
admin-c:    Anders Andersson
tech-c:     Anders Andersson
changed:    uffe@swip.net 930707
source:     RIPE
```

Notice all attributes noted in Table 2 are now removed. The following two (yes two, because this entry was not CIDR [7] aligned) "route" objects are also generated:

```
route:      193.12.128.0/22
descr:      SE-COMVIS-NET
origin:     AS1257
mnt-by:     AS1257-MNT
changed:    ripe-dbm@ripe.net 940829
source:     RIPE

route:      193.12.132.0/24
descr:      SE-COMVIS-NET
origin:     AS1257
mnt-by:     AS1257-MNT
changed:    ripe-dbm@ripe.net 940829
source:     RIPE
```

You also see that mnt-by attributes have been added to the "route" objects. This information has been derived from the "maintainer" files in each guardian account (see above for details of this). In this case the guardian of AS1257 named his maintainer object "AS1257-MNT". All other administrative attributes are preserved in both the "inetnum" and "route" objects which exception of the changed field which in the "route" object will just represent the actual date the split was done.

At the time B2, you will also see both the new "inetnum" and "route" objects in the database. If "inetnum" entries are sent into the database with obsoleted attributes the entries will be accepted with the obsoleted attributes removed and a warning sent back indicating the obsoleted attributes have been removed.

The auto-generation will happen at a set time at B2. This is a one-time operation and all users of the RIPE database should be aware of this major change.

It should be noted that the generation of routes as proposed above may not necessarily represent the correct routing information currently used by the various autonomous systems. This is mainly because the generation of routes is still based on the classful entries currently in the database. Guardians of all autonomous systems should after B2 check all the routes generated for them and make the necessary changes to better reflect their current external routing.

4.3. The "inet-rtr" object

The "inet-rtr" object is an object which can be used to describe any router within an autonomous system [5]. This object can be used as of time T1. One important piece of the "inet-rtr" object is that it will replace the "ias-int" attribute used in the old "inetnum" object. We encourage all users of the "ias-int" attribute to register new "inet-rtr" object from time T1 onwards. In an effort to preserve any loss of "ias-int" information when the auto-generation of "route"'s and "inetnum"'s takes place the ias-int information will be included as a "remarks" attribute in the generated "route" object. Here is a simple example:

```
route:        192.87.4.0/24
descr:        EUR-IP
origin:       AS1755
remarks:      ias-int: 192.87.4.17 AS1755
remarks:      ias-int: 192.87.4.18 AS1103
remarks:      ias-int: 192.87.4.19 AS2121
remarks:      ias-int: 192.87.4.20 AS286
remarks:      ias-int: 192.87.4.21 AS1104
remarks:      ias-int: 192.87.4.22 AS1755
remarks:      ias-int: 192.87.4.24 AS1103
remarks:      ias-int: 192.87.4.35 AS1128
remarks:      ias-int: 192.87.4.27 AS1890
remarks:      ias-int: 192.87.4.28 AS3333
changed:      ripe-dbm@ripe.net 940829
source:       RIPE
```

We realise this is non optimal and encourage the use of the "inet-rtr" object from time T1 onwards.

4.4. New "RIPE-81++" Syntax

All proposed "ripe-81++" syntax will be accepted after B1 at time T1, with the exception of the changes to "inetnum" and "route" objects which will not be accepted until B2 at time T2 (see section "Auto-Generation of "Route" and New "Inetnum" objects" above for reasoning behind this). The direct effect of this is you will see different whois output. The most important aspect will be the addition of syntactic sugar to the existing "as-in" and "as-out" attributes as well as the ability to use the attributes as well. To highlight this change if we look at a current "aut-num" object a whois query would produce the following:

```
aut-num:   AS1104
descr:     NIKHEF-H
descr:     Science Park Watergraafsmeer
descr:     Amsterdam, The Netherlands
as-in:     AS1103 100 AS1103
as-in:     AS1890 100 AS1890 AS2004 AS288
as-in:     AS1888 100 AS1888
as-in:     AS2122 200 AS2122 AS2600
as-in:     AS2600 100 AS2600
as-in:     AS3333 100 AS3333
as-out:    AS1103 AS1104
as-out:    AS1890 AS1104
as-out:    AS1888 AS1104
as-out:    AS2122 AS1104
as-out:    AS2600 AS1104
as-out:    AS3333 AS1104
default:   AS1103 100
guardian:  ripe-op@nikhef.nl
admin-c:   Rob Blokzijl
tech-c:    Marten Terpstra
remarks:   peers with AS2122 and AS2600 are RR test peers
notify:    tony@ripe.net
notify:    marten@ripe.net
changed:   tony@ripe.net 940426
source:    RIPE
```

As of B1 the same query will produce:

```
aut-num:     AS1104
descr:       NIKHEF-H
descr:       Science Park Watergraafsmeer
descr:       Amsterdam, The Netherlands
as-in:       from AS1103 100 accept AS1103
as-in:       from AS1890 100 accept AS1890 AS2004 AS288
as-in:       from AS1888 100 accept AS1888
as-in:       from AS2122 200 accept AS2122 AS2600
as-in:       from AS2600 100 accept AS2600
as-in:       from AS3333 100 accept AS3333
as-out:      to AS1103 announce AS1104
as-out:      to AS1890 announce AS1104
as-out:      to AS1888 announce AS1104
as-out:      to AS2122 announce AS1104
as-out:      to AS2600 announce AS1104
as-out:      to AS3333 announce AS1104
default:     AS1103 100
guardian:    ripe-op@nikhef.nl
admin-c:     Rob Blokzijl
tech-c:      Marten Terpstra
remarks:     peers with AS2122 and AS2600 are RR test peers
notify:      tony@ripe.net
notify:      marten@ripe.net
changed:     tony@ripe.net 940426
source:      RIPE
```

This is an important subtle change and anyone using tools should be aware of this change. To provide some backward compatibility it is possible to use the "-S" flag to the whois server to produce output without the added syntax information. A whois client supporting this flag will also be made available at B1.

5. Visible Changes and Actions Required by User Group

The effects of the B1/B2 changes and the actions needed by the different groups of RIPE database users are summarised below. This is intended as a check list for members of those user groups. It should be noted most readers of this document will belong to more than one group.

5.1. Routing Registry Guardians

This group comprises those who maintain the information in the Routing Registry part of the database: "aut-num" and "community" guardians and guardian file maintainers. This group will see the most changes and need to take some actions.

5.1.1. Between now and B1

Action
: Change entries in guardian files to exactly match single database objects.

5.1.2. After B1 and before B2

Action
: Check and update the "mntner" objects generated in the guardian accounts, thus establishing a mapping between guardian files and "mntner" objects.

Action
: Check the file of generated "route" objects and modified "inetnum" objects in the guardian account for errors and report those to the NCC.

5.1.3. After B2

Change
: Guardian file mechanism is no longer needed and disabled.

Change
: Route objects created.

Action
: Clean up guardian accounts.

Action
: Align any locally stored objects with the split objects.

Action
: Check "route" objects for remarks containing "ias-int" attributes and create "inet-rtr" objects where necessary.

Action
: Check the auto-generated routes and make the necessary changes to better reflect external routing policy with these routes.

5.2. Local Internet Registries

This is the group who maintains "inetnum" objects. They will see the consequences of the "inetnum"/"route" split and will be able to use the new authorisation scheme.

5.2.1. Between now and B1

No changes or actions before B1.

5.2.2. After B1 and before B2

Change
: Now possible to store multiple "inetnum" objects covering the same address space at different levels.

Change
: All representations described in [1] can be used for object submission.

Change
: New authorisation now available.

Action
: Coordinate authorisation on objects maintained by more than one party.

Action
: Prepare for "inetnum"/"route" split especially when storing data locally.

Action
: Check "inetnum" objects for "ias-int" attributes and create "inet-rtr" objects where necessary.

5.2.3. After B2

Change
: Some attributes obsoleted and some moved to "route" object.

Action
: Align any locally stored objects with the split objects.

Action
: Stop submitting obsolete/moved attributes.

5.3. Maintainers of other Objects

This group will basically only see added authorisation features and the need for coordination of this when multiple parties maintain an object.

5.3.1. Between now and B1

No changes or actions before B1.

5.3.2.

Change
: New authorisation now available.

Action
: Coordinate authorisation on objects maintained by more than one party.

5.3.3. After B2

No changes or actions after B2.

5.4. Users of RIPE DB Information

This is the group who queries the RIPE whois server and/or uses the database files.

5.4.1. Between now and B1

Action

Check and adapt tools for the effects of B1: queries will return less specific matches if no exact match can be found.

5.4.2. After B1 and before B2

Change

Network number queries can be classless and return less specific matches if no exact match is found.

Change

Ability to obtain less and more specific matches on network number queries.

Action

Check and prepare tools for the effects of B2: "route"/"inetnum" split, obsoleted attributes, new attributes, syntactic sugar returned by default in "aut-num".

5.4.3. After B2

Change

Effects of "route"/"inetnum" split will be visible.

6. Results

The results from this three phase transition will the following achievements.

1) A fully supported classless database (as of B1).
2) A new and improved authorisation and maintenance mechanism (as of B2).
3) Full support of ripe-81++ features (completed by B2 and most available from B1).

As with any transition of this type some change of data is unavoidable and a number of "big bang" days need to be factored in. However, the end result is a clear improvement to both the RIPE allocation and routing registry. Throughout the whole transition period the RIPE NCC will be available to make emergency changes to data if needed and deemed to be critical. All queries should be directed towards **`ncc@ripe.net`**.

7. References

[1] Bates, T., Karrenberg, D., Terpstra, M., "Support for Classless Internet Addresses in the RIPE Database", ripe-121, October 1994.

[2] Karrenberg, D., Terpstra, M., "Authorisation and Notification of Changes in the RIPE Database", ripe-120, October 1994.

[3] Bates et al, "Representation of IP Routing Policies in a Routing Registry (ripe-81++), ripe-181, October 1994.

[4] Bates, T., "Support of Guarded fields within the RIPE Database", ripe-117, July 1994.

[5] Bates, T., "Specifying an ‘Internet Router’ in the Routing Registry", ripe-122, October 1994.

[6] Lord, A., Terpstra, M., "RIPE Database Template for Networks and Persons", ripe-119, October 1994.

[7] Fuller, V et al, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC1519, September 1993.

[8] Karrenberg, D., "RIPE Database Template for Networks", ripe-050, April 1992.