

Mirroring the RIPE Database

Abstract

This document describes how to access a RIPE Near Real Time Mirror (NRTM) Database data stream from the RIPE NCC.

Intended Audience

This reference manual is for users who have signed a contract with the RIPE NCC to receive a RIPE NRTM Database data stream. It also serves as general information to any user interested in this service.

Conventions Used in This Document

We use <label> for a placeholder or to indicate syntax.

We use [option] to indicate an optional text or command argument.

We use a **bold** font to indicate an object type.

We use "attribute:" to indicate an attribute of an object.

Table of Contents

Mirroring the RIPE Database

Introduction

Near Real Time Mirroring (NRTM) is a mechanism that allows any authorised client to receive a stream of available data from a database on a server operated by the RIPE NCC. The client can receive this stream of data modifications from the server in near real time. The server will determine what data is made available to the client. Some data will be excluded from the stream or modified before sending. This is described below. The client pulls available data from the server by submitting requests. These requests can be periodic or continuous. The client can be initialised to a known state and the data stream can then be requested to continue with updates from that point onwards. However, updates are only available from the server as a data stream for two weeks. If the client has requested no updates for this period, then the client will need to be re-initialised.

Permission must be obtained from the Database Administration to access the RIPE NRTM Database. The service is only available to a user on a single IP address, which must be specified in the contract. The RIPE NRTM Database server will only accept requests to pull data from a registered IP address.

If you wish to use this service and have not already obtained permission for an NRTM data stream please contact our Customer Services department by emailing ripe-dbm@ripe.net.

After permission has been obtained, you may want an initial setup, and then configure your client to pull the data from the NRTM stream.

NOTE that it is not possible for users to mirror any of the RIPE NCC mirrored databases (for instance, other RIR databases). If you need a NRTM data stream from any other source you must contact that source directly.

NRTM Protocols

Currently there are two versions of the NRTM protocol available, Version 2 and Version 3. There is backwards compatibility between the two versions.

The RIPE Database server generates a serial number every time it processes an update in the database. When generating these serial numbers, the server describes all modifications to the database in terms of two atomic operations: deletion and addition.

Version 2 sends some data from every serial in the data stream. Where the real data is excluded, dummy data must be sent to maintain the sequence of serials.

Version 3 adds the serial number after the operation command for each update in the data stream. This eliminates the need to send dummy objects to keep serials in sync.

Currently, either version can be used. It is possible to switch between the versions. We would recommend using Version 3. This allows you to process a more condensed data stream, leading to a cleaner database without the clutter of excessive dummy objects.

We anticipate deprecating Version 2 in the near future.

Excluded Data

The RIPE NRTM Database made available by this service does not include the full set of data contained in the RIPE Database. It specifically excludes:

- Personal data (**person** and **role** objects)
- Organisation details (**organisation** object)
- Security data (**mntner** object)
- References to personal data (NIC Handles)

For both Version 2 and 3 of the NRTM protocol the **mntner** and **organisation** objects are replaced in the data stream by dummy data. These are place-holder objects. They still contain the primary key and other objects make references to these objects.

In Version 2 of the NRTM protocol, the **person** and **role** objects are also replaced in the data stream by dummy data. However, the same dummy **person** object replaces all **person** and **role** objects.. All references in other objects to **person** and **role** objects will refer to this single dummy **person** object.

In Version 3 of the NRTM protocol, the **person** and **role** objects are completely omitted from the data stream. Because the serials are included with each object in the data stream, it is not necessary to maintain place-holders.

Initial Setup

An initial data set is available from our ftp site. It is available in two formats:

- Individual data files
- Single data file

The individual data files contain data for each object type in separate text files. The excluded data (described above) is not available from these files. The set of files can be accessed here:

<ftp://ftp.ripe.net/ripe/dbase/split>

All the data from the "optional" attributes has been removed from the files for the **organisation** and **mntner** objects. All of the data from the "mandatory" attributes has been replaced with standard dummy data, except for the primary key value. So, for a **mntner** object, only the "mntner:" attribute will contain any real data. The files for **person** and **role** objects contain only a single dummy **person** object with the NIC Handle 'DUMY-RIPE'.

The single file is a merged data set of all the individual data files in a single text file. It contains the excluded **organisation** and **mntner** objects as dummy objects and a single **person** and **role** dummy object. This can be accessed here:

<ftp://ftp.ripe.net/ripe/dbase/ripe.db.gz>

All of these files are produced on a daily basis. Only the most recent file is available from the ftp site. Each database update has a serial number and these serials define the data pulled from an NRTM data stream. The latest update serial included in the data files is held in a separate text file on the ftp site:

<ftp://ftp.ripe.net/ripe/dbase/RIPE.CURRENTSERIAL>

You will need this serial number to know which serial to start pulling from the NRTM data stream. Note that only update serials from the previous 14 days can be pulled from the RIPE NRTM Database server. If you have not pulled any data from the NRTM stream for more than 14 days, you will need to download the above text files again, re-initialise your database and start from the new serial number. You may want to keep track of the serials that you pull from the NRTM data stream. If you have any gaps that are older than 14 days, then you will need to download the full text files again.

Accessing the Data Stream

To receive data from the NRTM data stream you must first connect to the host server and then pull selected data from it.

The client must connect to the host `nrtm.db.ripe.net` at port 4444.

When sending data to a client using the Version 2 protocol, the server will send one of two strings ("ADD" or "DEL") followed by two line-break characters (`\n\n`) and then the corresponding object. This will either be the object as it was before deletion or the object as it should appear after being created or modified. If an object already existing in the database follows the "ADD" string, the client software should treat this as a modification.

With the Version 3 protocol the serial follows the operation string ("ADD" or "DEL") on the same line. All other returned data is the same as for Version 2.

After connecting to the server, the client can request data by using the "-g" query flag. The arguments to this query flag are:

```
-g <source> :<NRTM_Protocol_version_#>:<first>-<last>
```

where

- <source> is the string that identifies the server database that is being mirrored (only RIPE is available).
- <NRTM_protocol_version_#> is the version of the mirroring protocol that the <source> supports (this should be 2 or 3 for the source RIPE).
- <first> is the lowest serial number requested.
- <last> is the most recent serial number requested. If the keyword "LAST" is used, this tells the server to send all updates up to the most recent one available at the time of the request.

This is an example of accessing the NRTM data stream using telnet and pulling the serials from 11012700 to the latest available.

```
$ telnet nrtm.db.ripe.net 4444
Trying 193.0.0.135...
Connected to whois.ripe.net.
Escape character is '^]'.
-g RIPE:3:11012700-LAST
```

```
% Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html
```

```
%START Version: 3 RIPE 11012700-11012701
```

```
ADD 11012700
```

```
person:      Lord Nagios
address:     NowhereLand
phone:       +31111111111
nic-hdl:     RIPE124-RIPE
e-mail:      bit-bucket@ripe.net
mnt-by:      NAGIOS-MNT
remarks:     Thu Mar 6 13:22:38 2008
changed:     bit-bucket@ripe.net 20070124
source:      RIPE
```

```
DEL 11012701
```

```
inetnum:     193.0.0.0 - 193.255.255.255
netname:     some name
descr:       RIPE
country:     NL
admin-c:     RIPE124-RIPE
tech-c:      RIPE124-RIPE
status:      ALLOCATED PA
notify:      bit-bucket@ripe.net
mnt-by:      NAGIOS-MNT
```

changed: bit-bucket@ripe.net 20040614
source: RIPE

A client may request a persistent connection by including the "-k" query flag with a mirroring request ("-g" query flag). In this case, the last argument is ignored and the server supplies the new objects as soon as they are processed. The client is responsible for closing the connection. A persistent connection for mirroring does not time out, but the connection may break if there is a server error.

A client can use the "-q sources" flag with the mirror server to retrieve information regarding available mirroring possibilities. Please see the Section on 'Other Server Features' in the RIPE Database Query Reference Manual for more details.[1]

At the beginning of the data stream, the server will send the following string:

```
%START Version: NRTM_Protocol_version_# source first-last
```

For example:

```
%START Version: 3 RIPE:1539595-1539597
```

After the last piece of data is sent to the client, the server will send the string:

```
%END source
```

to signal the end of transmission.

For example:

```
%END RIPE
```

References

[1] RIPE Database Query Reference Manual:
<http://www.ripe.net/db/support/query-reference-manual.pdf>