

Internet Routing Registry Tool Set

Ambrose Magee
LM Ericsson Ltd.
Global Expertise Centre (Dublin)
<ambrose.magee@ericsson.com>

Introduction

- IRR Tool Set
- RtConfig
 - Invoking
 - Access List Filters
 - Generating Import Filters
 - AS Path Filters
 - Export Filters
 - Route Sets
 - Static Routes
 - Filters using Logical Operators

Introduction (Continued)

- RtConfig (continued)
 - Using Communities
 - Filters using AS-Sets
 - Suppressing Martians

Internet Routing Registry Tool Set

- Formerly known as RA Tool Set
- Several tools
 - RtConfig
 - prpath
 - prtraceroute
 - peval

IRR ToolSet Installation

- Some problems installing version 4.7.1 on Solaris 8
- GCC version
 - 2.95.3
- Library files
 - libarchive.so files

RtConfig

- Router configuration tool
- Formats: Juniper, Cisco, Gated, Rsd
- Can be invoked with a template file

Invoking RtConfig

- By default, connects to whois.radb.net
- Can connect to RIPE Database
- RtConfig -h whois.ripe.net -p 43 -protocol ripe [-s RIPE]

RtConfig: Common Command Line Options

- -help
- -version
- -s <source-list>
- -f <file name>

RtConfig: Other Options

- -config <config-format>
- -suppress-martian
- Cisco specific commands
 - -cisco_use_prefix_lists

RtConfig: Environmental Variables

- IRR_HOST
- IRR_PORT
- IRR_SOURCES
- Command line options take precedence

RtConfig: Generating Access List Filters

```
aut-num: AS100
as-name: AMRM-TEST-AS100
descr: IRR Tool Set training object
import: from AS200 2.2.2.1 at 1.1.1.1
        accept AS200
export: to AS200
        announce AS100
default: to AS200
.....
route: 130.100.100.0/24
descr: A route object for AM7-TEST.
origin: AS100
remarks: This is a TEST object. It has no operational effect whatsoever.
```

RtConfig: Generating Access List Filters 2

```
bash-2.03# RtConfig -h test-whois.ripe.net -p 43 -protocol ripe -s TEST
RtConfig> @RtConfig access_list filter AS100
!
no access-list 100
access-list 100 permit ip 130.100.100.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
RtConfig>
```

RtConfig: Generating Import Filters

- Generate a complete router configuration
- Export filters based on your published policy
- Import filters based on published policy of your peers

RtConfig: Generating Import Filters

```
aut-num:      AS100
as-name:      AMRM-TEST-AS100
descr:        IRR Tool Set training object
import:       from AS200 2.2.2.1   at 1.1.1.1
              accept AS200
export:       to AS200
              announce AS100
default:      to AS200
remarks:      AS100 is only accepting
              AS200 originated routes from AS200
. . . . .
```

RtConfig: Generating Import Filters (continued)

```
aut-num:      AS200
as-name:      AMRM-TEST-AS200
descr:        IRR Tool Set training object
import:        from AS100
                accept AS100
export:        to AS100
                announce AS200
default:      to AS300
remarks:      AS200 is only accepting
                AS100 originated routes from AS100
.....
```

RtConfig: Generating Import Filters (continued)

```
route:        130.100.100.0/24
descr:        A route object for AM7-TEST.
origin:       AS100
.....

route:        130.100.200.0/24
descr:        A route object for AM7-TEST.
origin:       AS200
.....
```

RtConfig

```
aut-num:    AS200
as-name:    AMRM-TEST-AS200
descr:      IRR Tool Set training object
import:      from AS100
              accept AS100
export:      to AS100
              announce AS200
default:     to AS300
remarks:     AS200 is only accepting
              AS100 originated routes from AS100
.....
```

RtConfig: Generating Import Filters (continued)

```
bash-2.03# RtConfig -h test-whois.ripe.net -p 43 -protocol ripe -s TEST
RtConfig> @RtConfig import AS100 1.1.1.1 AS200 2.2.2.1
!
access-list 100
access-list 100 permit ip 130.100.200.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
no route-map MyMap_200_1
!
route-map MyMap_200_1 permit 1
match ip address 100
!
router bgp 100
neighbor 2.2.2.1 route-map MyMap_200_1 in
RtConfig>
```

RtConfig: Import Filters using Prefix Lists

```
bash-2.03# RtConfig -h test-whois.ripe.net -p 43 -protocol ripe
-s TEST -cisco_use_prefix_lists
RtConfig> @RtConfig import AS100 1.1.1.1 AS200 2.2.2.1
!
no ip prefix-list pl100
ip prefix-list pl100 permit 130.100.200.0/24
ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map MyMap_200_1
!
route-map MyMap_200_1 permit 1
  match ip address prefix-list pl100
!
router bgp 100
neighbor 2.2.2.1 route-map MyMap_200_1 in
```

RtConfig: Generating AS Path Filters

- Several ways of doing this
- RtConfig Command: `as_path_access_list filter <filter>`
- This generates as-path filter directly
- <filter>: RPSL filter over AS Path regular expressions using operators AND, OR and NOT
- Example of <filter>: `<AS1> AND <AS2>`
- Practically no router configuration statements generated

RtConfig: Generating AS Path Filters (continued)

- AS Path regular expressions in aut-num object
 - import and export attributes
 - import: from AS1 accept <^AS1\$>
- Can use as-set names in AS Path Regular expression
 - import: from AS1 accept <^AS1 AS1:As-Customers*\$>
- More router configuration commands generated
 - e.g. neighbor statements, route-maps, etc.

RtConfig: AS Path Filters

```
bash-2.03# RtConfig -h test-whois.ripe.net -p 43 -protocol ripe
-s TEST
RtConfig> @RtConfig aspath_access_list filter <^AS100 .* AS200^>
!
no ip as-path access-list 1
ip as-path access-list 1 permit ^_100((_ [0-9]+))*_200^_
RtConfig>
```

N.B. The aut-num objects for AS100 and AS200 have no effect on the output.

RtConfig: AS Path Filters - Aut-num Objects

```
aut-num:      AS100
as-name:      AMRM-TEST-AS100
descr:        IRR Tool Set training object
import:        from AS200 2.2.2.1   at 1.1.1.1
               accept AS200
export:        to AS200
               announce AS100
default:       to AS200
remarks:
. . .
```

This is the same aut-num object for AS100 as before.

RtConfig: AS Path Filters - Aut-Num Objects

```
aut-num:      AS200
as-name:      AMRM-TEST-AS200
descr:        IRR Tool Set training object
import:        from AS100
               accept <^AS100 $>
export:        to AS100
               announce {130.100.200.0/24}
default:       to AS300
remarks:
. . .
```

N.B. AS Path regular expression in 'import' attribute.

RtConfig: AS Path Filters - Aut-Num Objects

```
bash-2.03# RtConfig -h test-whois.ripe.net -p 43 -protocol ripe
-s TEST
RtConfig> @RtConfig import AS200 2.2.2.1 AS100 1.1.1.1
!
no ip as-path access-list 1
ip as-path access-list 1 permit ^_100$
!
no route-map MyMap_100_1
!
route-map MyMap_100_1 permit 1
  match as-path 1
!
router bgp 200
neighbor 1.1.1.1 route-map MyMap_100_1 in
RtConfig>
```

RtConfig: AS Path Filters - AS-Set Objects

```
as-set:      AS10000:AS-Customers
descr:       Customers of AS10000
members:     AS10000, AS100, AS200, AS300
mbrs-by-ref: AMRM-RPSL-TEST-MNT
... .
```

This as-set object lists the customers of AS10000: AS100, AS200 and AS300 and AS10000.

RtConfig: AS Path Filters - AS-Set Objects

```
aut-num:      AS20000
as-name:      AMRM-TEST-AS20000
descr:        IRR Tool Set training object
import:        from AS10000      192.16.151.10 at 192.71.19.20
               accept <^AS10000 AS10000:AS-Customers*$>
export:        to AS10000
               announce ANY AND NOT RS-MARTIANS
... .
```

N.B. AS Path regular expression in 'import' attribute.

N.B. Router 192.71.19.20 in AS20000 imports routes from router 192.16.151.10 in AS10000

RtConfig: AS Path Filters - AS-Set Objects

```
RtConfig> @RtConfig import AS20000 192.71.19.20 AS10000 192.16.151.10
!
no ip as-path access-list 1
ip as-path access-list 1 permit ^_10000(_(100|200|300|10000))*$
!
no route-map MyMap_10000_1
!
route-map MyMap_10000_1 permit 1
  match as-path 1
!
router bgp 20000
neighbor 192.16.151.10 route-map MyMap_10000_1 in
RtConfig>
```

RtConfig: AS Path Filters - Outbound

```
aut-num:      AS10000
as-name:      AMRM-TEST-AS10000
descr:        IRR Tool Set training object
import:        from AS200
               accept AS200 OR AS100
export:        to AS200
               announce {192.16.131.0/22, 192.16.151.0/24}
export:        to AS20000 192.71.19.20 at 192.16.151.10
               announce <^AS10000 AS10000:AS-Customers*$>
. . . . .
```

We can also use AS Path regular expressions and as-sets in the export attribute.

RtConfig: AS Path Filters - Outbound

```
as-set:        AS10000:AS-Customers
descr:          Customers of AS10000
members:        AS10000, AS100, AS200, AS300
mbrs-by-ref:    AMRM-RPSL-TEST-MNT
. . . . .
```

RtConfig: AS Path Filters - Outbound

```
RtConfig> @RtConfig export AS10000 192.16.151.10 AS20000 192.71.19.20
!
no ip as-path access-list 1
ip as-path access-list 1 permit ^_10000(_(100|200|300|10000))*$
!
no route-map MyMap_20000_1
!
route-map MyMap_20000_1 permit 1
  match as-path 1
!
router bgp 10000
neighbor 192.71.19.20 route-map MyMap_20000_1 out
RtConfig>
```

RtConfig: Export Filters

```
aut-num:      AS10000
as-name:      AMRM-TEST-AS10000
descr:        IRR Tool Set training object
import:       from AS200
              accept AS200 OR AS100
export:       to AS200
              announce {192.16.131.0/22, 192.16.151.0/24}
export:       to AS20000
              announce AS10000:AS-Customers
... .
```

Router 192.16.151.10 in AS10000 exports routes to
router 192.71.19.20 in AS20000.
AS10000:AS-Customers is the same as-set as before.

RtConfig: Export Filters

```
RtConfig> @RtConfig export AS10000 192.16.151.10 AS20000 192.71.19.20
!  
no access-list 101  
access-list 101 permit ip 130.100.100.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 101 permit ip 130.100.200.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 101 permit ip 192.16.128.0 0.0.0.0 255.255.252.0 0.0.0.0  
access-list 101 permit ip 192.16.151.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 101 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255  
!  
no route-map MyMap_20000_1  
!  
route-map MyMap_20000_1 permit 1  
  match ip address 101  
!  
router bgp 10000  
neighbor 192.71.19.20 route-map MyMap_20000_1 out  
RtConfig>
```

RtConfig: Route-Sets

```
route-set:      RS-AS10000  
descr:          AS10000 address prefixes  
members:        192.16.131.0/22, 192.16.151.0/24  
mbrs-by-ref:    AMRM-RPSL-TEST-MNT  
. . . .
```

This object describes routes, not route objects.

RtConfig: Route-Sets

```
$ RtConfig -h test-whois.ripe.net -p 43 -protocol ripe -s TEST
RtConfig> @RtConfig access_list filter RS-AS10000
!
no access-list 100
access-list 100 permit ip 192.16.128.0 0.0.0.0 255.255.252.0 0.0.0.0
access-list 100 permit ip 192.16.151.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
RtConfig>
```

RtConfig: Static Routes

```
aut-num:      AS10000
as-name:      AMRM-TEST-AS10000
descr:        IRR Tool Set training object
import:       from AS200
              accept AS200 OR AS100
import:       protocol STATIC into BGP4
              from AS10000
              action aspath.prepend(AS1);
              accept RS-AS10000
... .
```

N.B. 'import' attribute

RtConfig: Static Routes

- 'protocol STATIC into BGP4'
 - injects static routes into BGP4 routing protocol
- action aspath.prepend(AS1);
- action
 - set or modify value of route attribute
 - actions are terminated by ';'

RtConfig: Static Routes

```
RtConfig> @RtConfig static2bgp AS10000 192.16.151.10
!
no access-list 100
access-list 100 permit ip 192.16.128.0 0.0.0.0 255.255.252.0 0.0.0.0
access-list 100 permit ip 192.16.151.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
no route-map MyMap_10000_1
!
route-map MyMap_10000_1 permit 1
 match ip address 100
 set as-path prepend 1
!
router bgp 10000
 redistribute static route-map MyMap_10000_1
RtConfig>
```

RAToolSet and Static Routes

- Route object has 'inject' attribute
- RtConfig does not look at this
- 'inject' attribute configures static routes
- RtConfig generates policy to redistribute into BGP

RtConfig: Filters using Logical Operators

```
aut-num:      AS200
as-name:      AMRM-TEST-AS200
descr:        IRR Tool Set training object
import:        from AS100
                accept <^AS100 $>
export:        to AS100
                announce AS200 OR AS300 OR AS10000 OR ANY
default:      to AS300
.....
```

RtConfig: Filters using Logical Operators

```
aut-num:      AS10000
as-name:      AMRM-TEST-AS10000
descr:        IRR Tool Set training object
import:       from AS200
              accept AS200
export:       to AS200
              announce AS300 OR AS10000 OR ANY
. . . . .
```

RtConfig: Filters using Logical Operators

```
RtConfig> @RtConfig access_list filter AS200 OR AS10000
!
no access-list 100
access-list 100 permit ip 130.100.200.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 100 permit ip 192.16.128.0 0.0.0.0 255.255.252.0 0.0.0.0
access-list 100 permit ip 192.16.151.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
RtConfig>
```

This configuration is the union of the policies of AS200 and AS10000.

RtConfig: Using Communities

```
aut-num:      AS1 0000
as-name:      AMRM-TEST-AS10000
descr:        IRR Tool Set training object
import:        from AS200
               accept AS200
export:        to AS200
               announce AS300 OR AS10000 OR ANY
export:        to AS20000
               announce AS200 AND NOT {0.0.0.0/0}
               AND <^AS200*$>
               AND NOT community(200:1)

.....
```

RtConfig: Using Communities

```
RtConfig> @RtConfig export AS10000 192.16.151.10 AS20000 192.71.19.20
!
no access-list 100
access-list 100 permit ip 130.100.200.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
no ip as-path access-list 1
ip as-path access-list 1 permit ^(_200)*$
!
ip bgp-community new-format
!
no ip community-list 1
ip community-list 1 deny 200:1
ip community-list 1 permit internet
```

RtConfig: Using Communities

```
no route-map MyMap_20000_1
!
route-map MyMap_20000_1 permit 1
  match as-path 1
  match community 1
  match ip address 100
!
router bgp 10000
neighbor 192.71.19.20 route-map MyMap_20000_1 out
RtConfig>
```

RtConfig: Filters using AS-Sets

```
as-set:          AS10000:AS-Customers
descr:           Customers of AS10000
members:         AS10000, AS100, AS200, AS300
mbrs-by-ref:     AMRM-RPSL-TEST-MNT
... .
```

This as-set object lists the customers of AS10000: AS100, AS200 and AS300 and AS10000.

RtConfig: Filters using AS-Sets

```
RtConfig> @RtConfig access_list filter AS10000:AS-Customers
!  
no access-list 101  
access-list 101 permit ip 130.100.100.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 101 permit ip 130.100.200.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 101 permit ip 192.16.128.0 0.0.0.0 255.255.252.0 0.0.0.0  
access-list 101 permit ip 192.16.151.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 101 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255  
RtConfig>
```

RtConfig: Supressing Martians

- 'Martian' routes
 - loopback
 - private address space
- Command Line Option
- Cisco configurations only

RtConfig: -supress_martian

```
RtConfig> @RtConfig access_list filter AS10000:AS-Customers
!  
no access-list 100  
access-list 100 deny ip host 0.0.0.0 any  
access-list 100 deny ip 127.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255  
access-list 100 deny ip 10.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255  
access-list 100 deny ip 172.16.0.0 0.15.255.255 255.240.0.0 0.15.255.255  
access-list 100 deny ip 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255  
access-list 100 deny ip 192.0.2.0 0.0.0.255 255.255.255.0 0.0.0.255  
access-list 100 deny ip 128.0.0.0 0.0.255.255 255.255.0.0 0.0.255.255  
access-list 100 deny ip 191.255.0.0 0.0.255.255 255.255.0.0 0.0.255.255  
access-list 100 deny ip 192.0.0.0 0.0.0.255 255.255.255.0 0.0.0.255  
access-list 100 deny ip 223.255.255.0 0.0.0.255 255.255.255.0 0.0.0.255  
access-list 100 deny ip 224.0.0.0 31.255.255.255 224.0.0.0 31.255.255.255  
access-list 100 deny ip 169.254.0.0 0.0.255.255 255.255.0.0 0.0.255.255  
access-list 100 permit ip 130.100.100.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 100 permit ip 130.100.200.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 100 permit ip 192.16.128.0 0.0.0.0 255.255.252.0 0.0.0.0  
access-list 100 permit ip 192.16.151.0 0.0.0.0 255.255.255.0 0.0.0.0  
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255  
RtConfig>
```

RtConfig: Configuring a Router

- Generates a configuration file for the router
- Uses inet-rtr object
- Configures import and export policies

RtConfig: Configuring a Router

```
inet-rtr:      Amsterdam.ripe.net
local-as:      AS3333
ifaddr:        192.87.4.28 masklen 24
ifaddr:        193.0.0.222 masklen 27
ifaddr:        192.16.183.128 masklen 24
ifaddr:        193.0.15.130 masklen 24
peer:          BGP4 192.87.4.20 asno(AS286)
peer:          BGP4 192.87.4.18 asno(AS1103)
peer:          BGP4 192.87.4.24 asno(AS1103)
peer:          BGP4 193.0.0.221 asno(AS1104)
peer:          BGP4 192.87.4.35 asno(AS1128)
peer:          BGP4 193.0.15.129 asno(AS1755)
peer:          BGP4 192.16.183.32 asno(AS1888)
peer:          BGP4 192.87.4.27 asno(AS1890)
peer:          BGP4 192.87.4.19 asno(AS2121)
peer:          BGP4 193.0.0.219 asno(AS2122)
peer:          BGP4 192.16.183.64 asno(AS3317)
```

RtConfig: Configuring a Router

```
aut-num:       AS3333
as-name:       RIPE-NCC-AS
descr:         RIPE NCC
descr:         European Regional Internet Registry
import:        from AS286
                action pref=120;
                accept ANY
import:        from AS702
                action pref=120;
                accept ANY
import:        from AS1103
                action pref=120;
                accept ANY
.....
export:        to AS286
                announce AS3333
export:        to AS702
                announce AS3333
export:        to AS1103
                announce AS3333
```

RtConfig: Configuring a Router

```
$ RtConfig -h whois.ripe.net -p 43 -protocol ripe -s RIPE
RtConfig> configureRouter Amsterdam.ripe.net
configureRouter Amsterdam.ripe.net
RtConfig> @RtConfig configureRouter Amsterdam.ripe.net
!
no route-map MyMap_286_1
!
route-map MyMap_286_1 permit 1
  set local-preference 880
!
router bgp 3333
neighbor 192.87.4.20 route-map MyMap_286_1 in
!
no access-list 100
access-list 100 permit ip 193.0.0.0 0.0.0.0 255.255.248.0 0.0.0.0
access-list 100 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
```

RtConfig: Configuring a Router

```
no route-map MyMap_286_2
!
route-map MyMap_286_2 permit 1
  match ip address 100
!
router bgp 3333
neighbor 192.87.4.20 route-map MyMap_286_2 out
!
no route-map MyMap_1103_3
!
route-map MyMap_1103_3 permit 1
  set local-preference 880
!
router bgp 3333
neighbor 192.87.4.18 route-map MyMap_1103_3 in
.....
```

Other Tools

- CIDRAdvisor
- prtraceroute
- prpath
- peval
- Route Object Edito (roe)
- Aut-num Object Editor (aoe)

prtraceroute

- Path and policy information
- Example:
 - host: whois.ripe.net
 - port: 43
 - protocol ripe
 - destination netnod.se

prtraceroute

```
prtraceroute to netnod.se (212.32.191.62), 30 hops max, 12 byte packets
 1 [AS1239] 193.95.167.177 (193.95.167.177)  14.412 ms  73.871 ms  30.111 ms
 2 [AS1239] serial3-3-2.cr002.ddm.esat.net (193.95.169.113)  94.994 ms  102.389 ms  66.964 ms
 3 [AS1239] fe0-0.tr001.ddm.esat.net (193.95.138.1)  102.389 ms  66.964 ms  117.000 ms
 4 [AS2110] atm1-0-1.tr001.cwt.esat.net (193.120.29.101)  103.772 ms  117.000 ms  117.000 ms
 5 [AS1239] vlan9.rt001.cwt.esat-x.com (193.95.129.161)  82.357 ms  347.600 ms  117.000 ms
 6 [AS1239] 193.95.129.4 (193.95.129.4)  118.859 ms  119.21 ms  511.088 ms
...
22 [AS1257] betala-gw.norrnod.se (194.165.224.17)  186.955 ms  166.587 ms  201.65 ms
23 [AS1257] dmz-gw.norrnod.se (194.165.224.130)  181.663 ms  201.65 ms  201.65 ms
24 [AS1257] netnod.se (212.32.191.62)  211.93 ms  186.883 ms  *
```

Path taken:

AS1239 AS2110 AS1239 AS702 AS1849 AS702 AS1299 AS3301 AS1257

prtraceroute

```
24 AS1257 netnod.se          destination -> internal
23 AS1257 dmz-gw.norrnod.se   internal -> internal
22 AS1257 betala-gw.norrnod.se internal -> export
...
6 AS1239 193.95.129.4          import: 1 -> internal
5 AS1239 vlan9.rt001.cwt.esat-x.com internal -> !export
4 AS2110 atm1-0-1.tr001.cwt.esat.net import: 2 -> !export
3 AS1239 fe0-0.tr001.ddm.esat.net import: 1 -> internal
2 AS1239 serial3-3-2.cr002.ddm.esat.net internal -> internal
1 AS1239 193.95.167.177       internal -> internal
0 AS1239 ratoolset            internal -> source
```

prtraceroute

- Output in reverse order
- Routes are advertised and propagated like this
- 6: import 1 -> internal
 - first preferred hop from current AS to destination
- 5: AS1239 has no registered export or default policy to accept routes being exported from the next hop

prpath

- `prpath [options] [[<as-no>] <destination-specification>]`
- Enumerates list of paths between <as-no> and <destination-specification>.

prpath

```
# prpath -h whois.ripe.net -p 43 -protocol ripe AS3333 AS1213
Route:
  NLRI: don't care
  Origins: AS1213
AS3333 AS5615 AS1200 AS2686 AS2529 AS1290 AS786 AS1213
AS3333 AS6728 AS1200 AS2686 AS2529 AS1290 AS786 AS1213
AS3333 AS8210 AS1200 AS2686 AS2529 AS1290 AS786 AS1213
```

peval

- peval is a low level policy evaluation tool
- can be used to write router configuration generators

peval

```
route:      130.100.100.0/24
descr:      A route object for AM7-TEST.
origin:     AS100
.....
```

```
route:      130.100.200.0/24
descr:      A route object for AM7-TEST.
origin:     AS200
.....
```

peval

```
route:      192.16.131.0/22
descr:      A route object for AM7-TEST.
origin:     AS10000
.....
```

```
route:      192.16.151.0/24
descr:      A route object for AM7-TEST.
origin:     AS10000
.....
```

```
as-set:     AS10000:AS-Customers
descr:      Customers of AS10000
members:    AS10000, AS100, AS200, AS300
.....
```

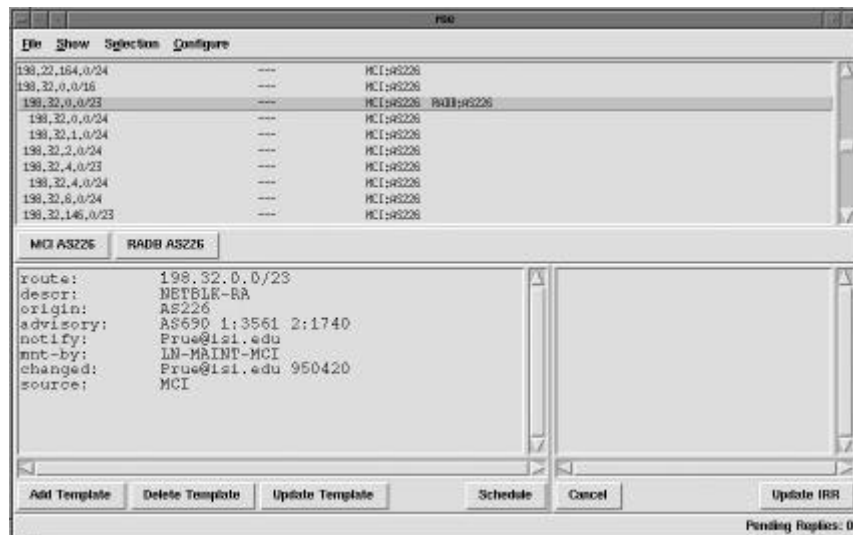
peval

```
$ peval -h test-whois.ripe.net -p 43 -protocol ripe -s TEST
peval> AS10000
({192.16.151.0/24, 192.16.128.0/22})
peval> AS10000 AND AS200
NOT ANY
peval> AS200
({130.100.200.0/24})
peval> AS10000 OR AS200
({192.16.151.0/24, 192.16.128.0/22, 130.100.200.0/24})
peval> AS10000:AS-Customers
({192.16.151.0/24, 192.16.128.0/22, 130.100.200.0/24, 130.100.100.0/24})
peval>
```

Route Object Editor (roe)

- GUI lists routes registered by an autonomous system
- roe -h whois.ripe.net -p 43 -protocol ripe -as AS 8674
- Extra option: -s RIPE
- Can be used to update route objects

Example of Route Object Editor



Aut-Num Object Editor (aoe)

- Does not work in current version (4.7.1)
- Fixed in a future release

CIDRAdvisor

- examines an autonomous system
 - address prefixes of the
 - polices registered in the IRR
- Identifies aggregates
- Aggregates do not violate policy constraints.

CIDRAdvisor

```
$ CIDRAdvisor -h whois.ripe.net -p 43 -protocol ripe -s RIPE -as AS8674
```

Best possible aggregation:

192.36.134.0/23

Components:

192.36.134.0/24 (AS8674)

192.36.135.0/24 (AS8674)

Acknowledgements

- Cengiz Alaettinoglu, Packet Design
- K. Petrusha, RIPE NCC
- Ericsson Global Expertise Centre (Dublin)