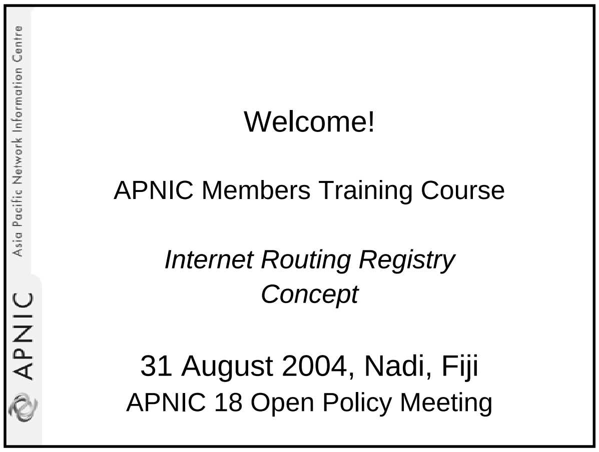
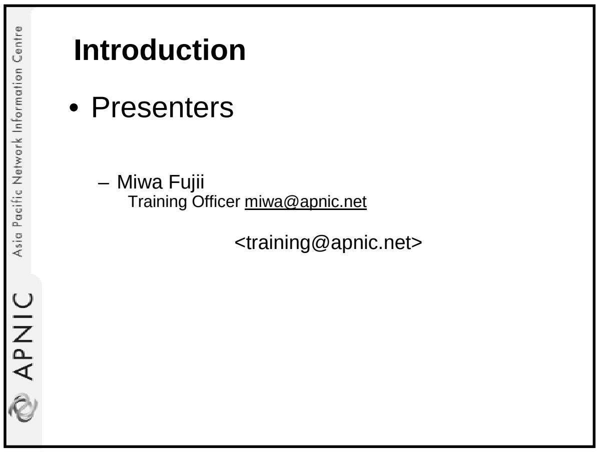


Welcome!

APNIC Members Training Course

Internet Routing Registry
Concept

31 August 2004, Nadi, Fiji
APNIC 18 Open Policy Meeting

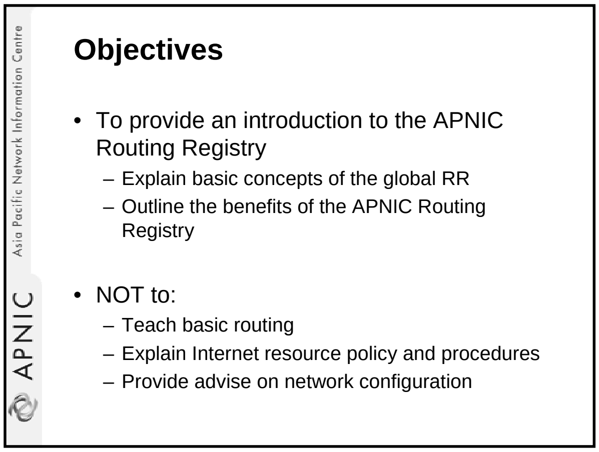


Introduction

- Presenters

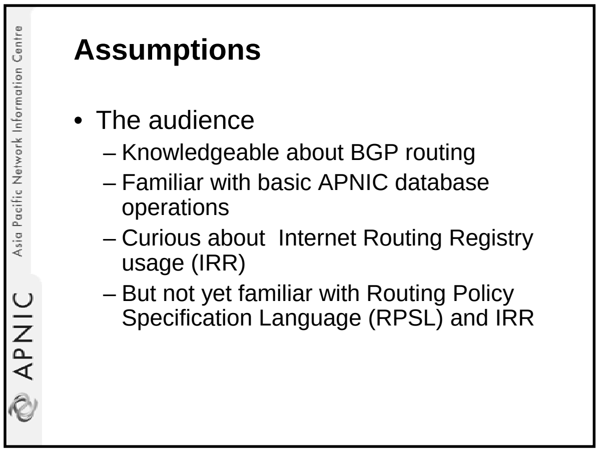
  – Miwa Fujii
  Training Officer miwa@apnic.net

  <training@apnic.net>



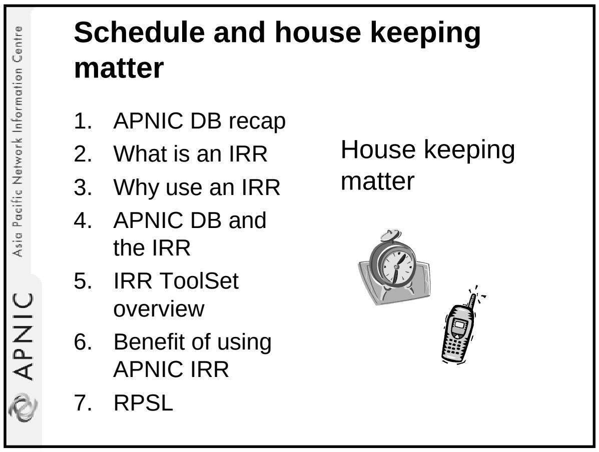
Objectives

- To provide an introduction to the APNIC Routing Registry
  - Explain basic concepts of the global RR
  - Outline the benefits of the APNIC Routing Registry
- NOT to:
  - Teach basic routing
  - Explain Internet resource policy and procedures
  - Provide advise on network configuration



Assumptions

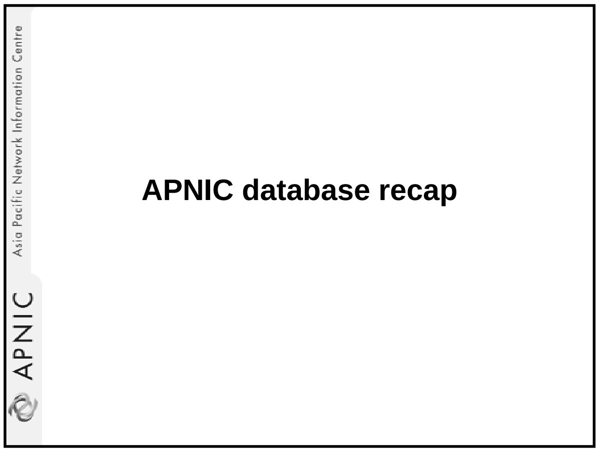
- The audience
  - Knowledgeable about BGP routing
  - Familiar with basic APNIC database operations
  - Curious about Internet Routing Registry usage (IRR)
  - But not yet familiar with Routing Policy Specification Language (RPSL) and IRR



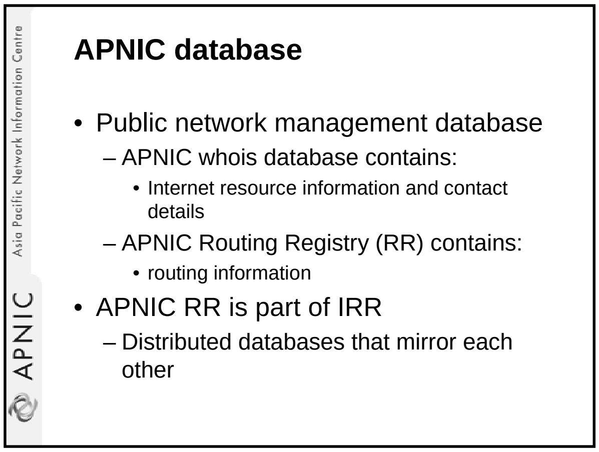
Schedule and house keeping matter

1. APNIC DB recap
2. What is an IRR
3. Why use an IRR
4. APNIC DB and the IRR
5. IRR ToolSet overview
6. Benefit of using APNIC IRR
7. RPSL

House keeping matter

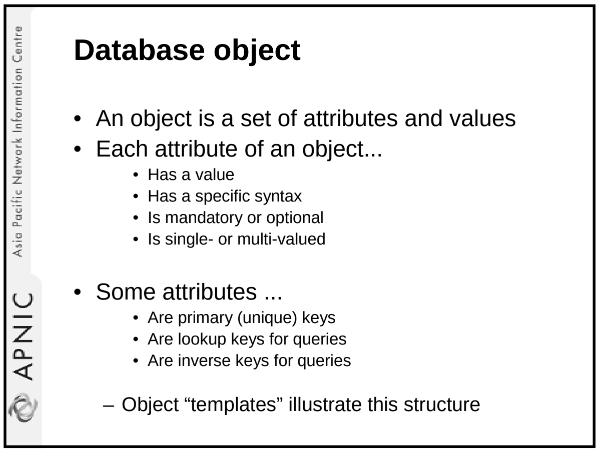


APNIC database recap



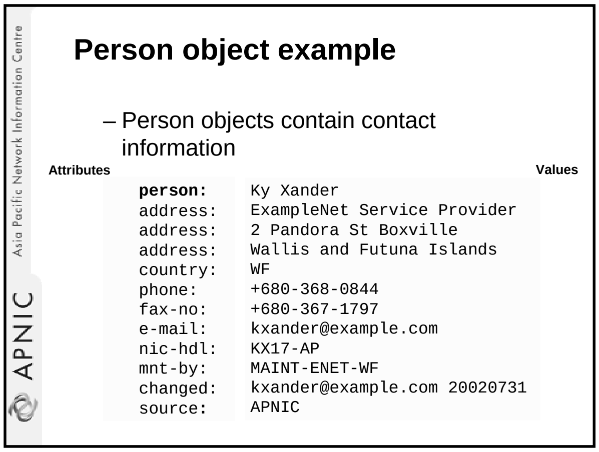
APNIC database

- Public network management database
  - – APNIC whois database contains:
    - Internet resource information and contact details
  - – APNIC Routing Registry (RR) contains:
    - routing information
- APNIC RR is part of IRR
  - – Distributed databases that mirror each other



Database object

- An object is a set of attributes and values
- Each attribute of an object...
    - Has a value
    - Has a specific syntax
    - Is mandatory or optional
    - Is single- or multi-valued
- Some attributes ...
    - Are primary (unique) keys
    - Are lookup keys for queries
    - Are inverse keys for queries
  - – Object "templates" illustrate this structure

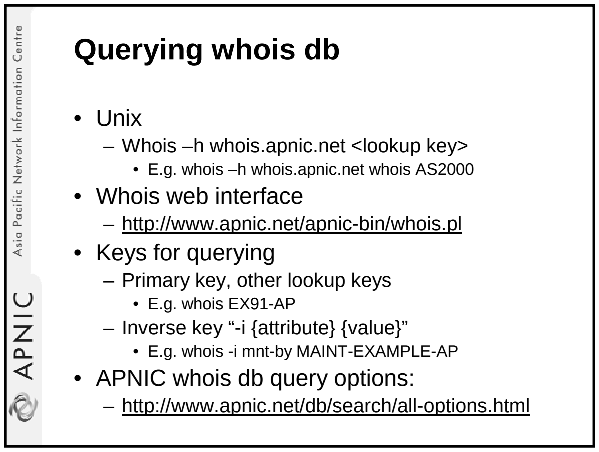


Person object example

- – Person objects contain contact information

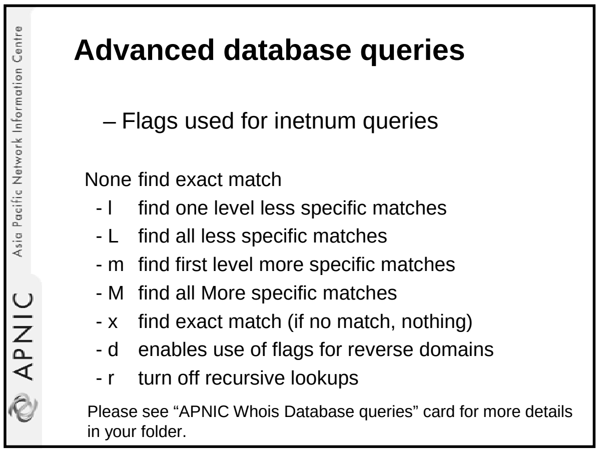
Attributes **Values**

```
person:    Ky Xander
address:   ExampleNet Service Provider
address:   2 Pandora St Boxville
address:   Wallis and Futuna Islands
country:   WF
phone:     +680-368-0844
fax-no:    +680-367-1797
e-mail:    kxander@example.com
nic-hdl:   KX17-AP
mnt-by:    MAINT-ENET-WF
changed:   kxander@example.com 20020731
source:    APNIC
```



Querying whois db

- Unix
  - Whois –h whois.apnic.net <lookup key>
    - E.g. whois –h whois.apnic.net whois AS2000
- Whois web interface
  - http://www.apnic.net/apnic-bin/whois.pl
- Keys for querying
  - Primary key, other lookup keys
    - E.g. whois EX91-AP
  - Inverse key “-i {attribute} {value}”
    - E.g. whois -i mnt-by MAINT-EXAMPLE-AP
- APNIC whois db query options:
  - http://www.apnic.net/db/search/all-options.html

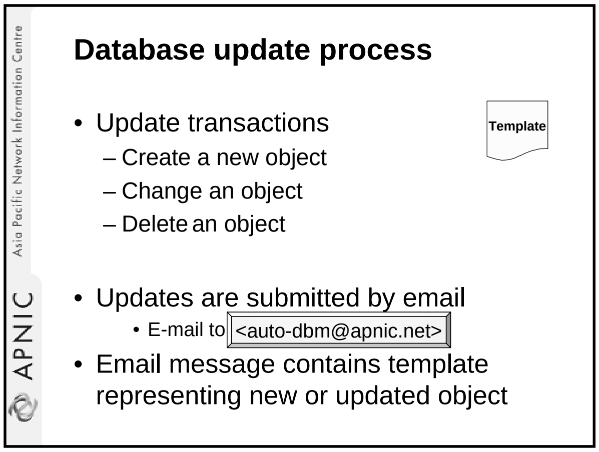


Advanced database queries

– Flags used for inetnum queries

None	find exact match
- l	find one level less specific matches
- L	find all less specific matches
- m	find first level more specific matches
- M	find all More specific matches
- x	find exact match (if no match, nothing)
- d	enables use of flags for reverse domains
- r	turn off recursive lookups

Please see “APNIC Whois Database queries” card for more details in your folder.

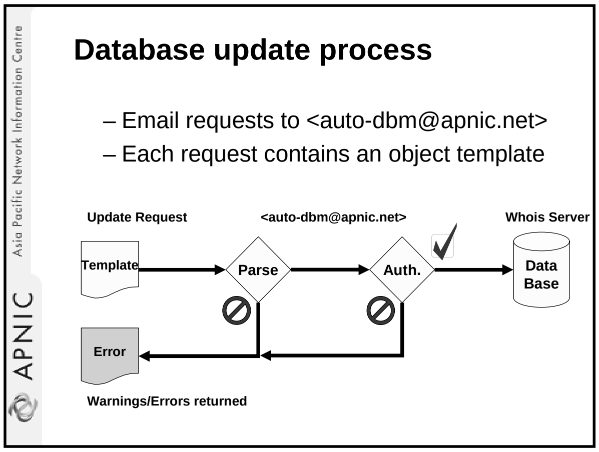


Database update process

- Update transactions
  - Create a new object
  - Change an object
  - Delete an object

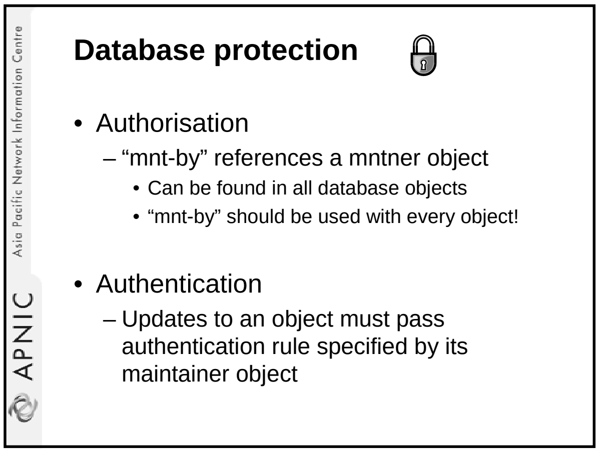
Template

- Updates are submitted by email
  - E-mail to <auto-dbm@apnic.net>
- Email message contains template representing new or updated object



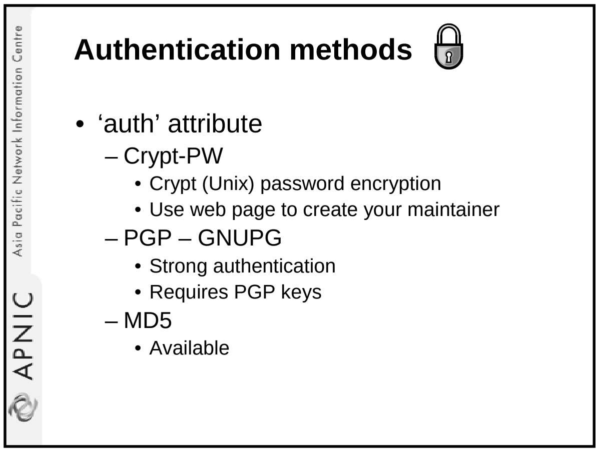
Database update process

– Email requests to <auto-dbm@apnic.net>
– Each request contains an object template



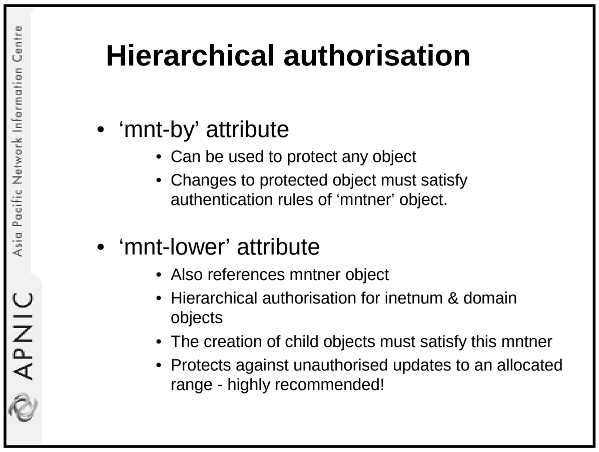
Database protection

- Authorisation
  - “mnt-by” references a mntner object
    - Can be found in all database objects
    - “mnt-by” should be used with every object!
- Authentication
  - Updates to an object must pass authentication rule specified by its maintainer object



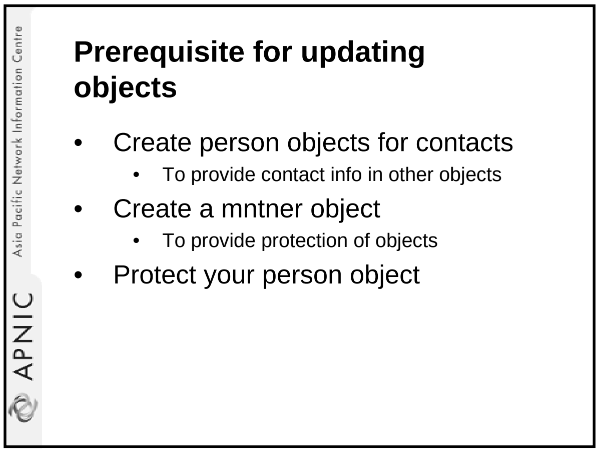
Authentication methods

- ‘auth’ attribute
  - Crypt-PW
    - Crypt (Unix) password encryption
    - Use web page to create your maintainer
  - PGP – GNUPG
    - Strong authentication
    - Requires PGP keys
  - MD5
    - Available



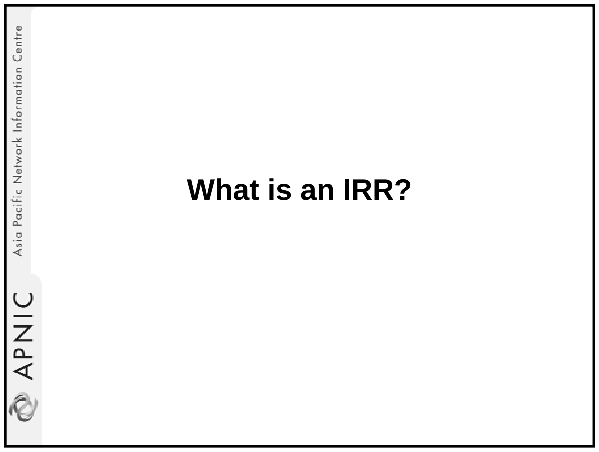
Hierarchical authorisation

- ‘mnt-by’ attribute
  - Can be used to protect any object
  - Changes to protected object must satisfy authentication rules of ‘mntner’ object.
- ‘mnt-lower’ attribute
  - Also references mntner object
  - Hierarchical authorisation for inetnum & domain objects
  - The creation of child objects must satisfy this mntner
  - Protects against unauthorised updates to an allocated range - highly recommended!

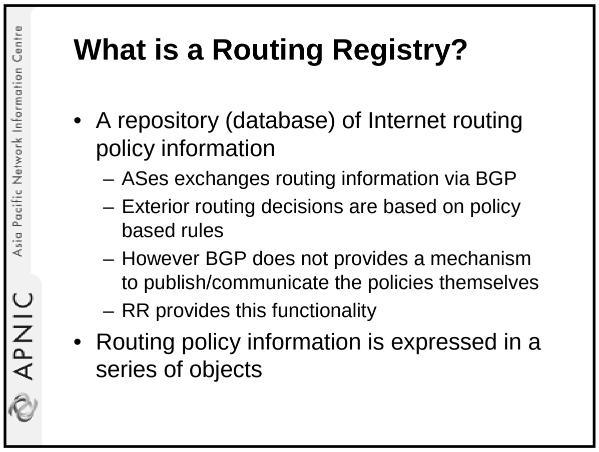


Prerequisite for updating objects

- Create person objects for contacts
  - To provide contact info in other objects
- Create a mntner object
  - To provide protection of objects
- Protect your person object

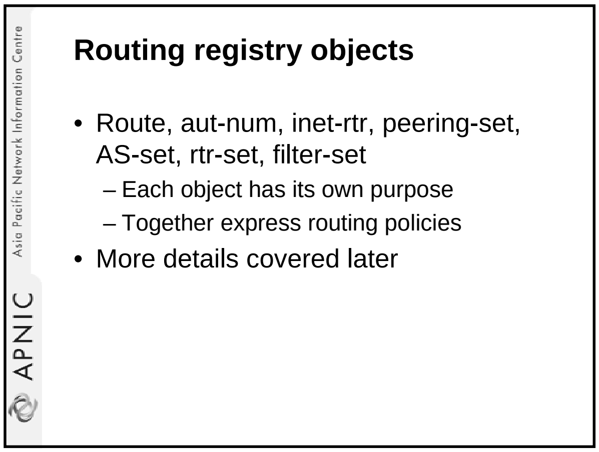


What is an IRR?



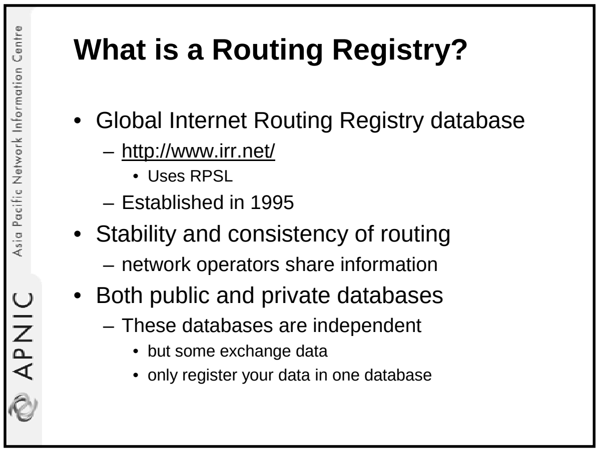
What is a Routing Registry?

- A repository (database) of Internet routing policy information
  - ASes exchanges routing information via BGP
  - Exterior routing decisions are based on policy based rules
  - However BGP does not provides a mechanism to publish/communicate the policies themselves
  - RR provides this functionality
- Routing policy information is expressed in a series of objects



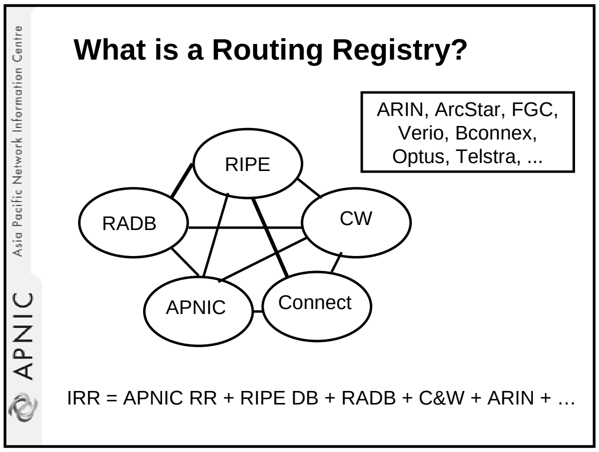
Routing registry objects

- Route, aut-num, inet-rtr, peering-set, AS-set, rtr-set, filter-set
  - Each object has its own purpose
  - Together express routing policies
- More details covered later



What is a Routing Registry?

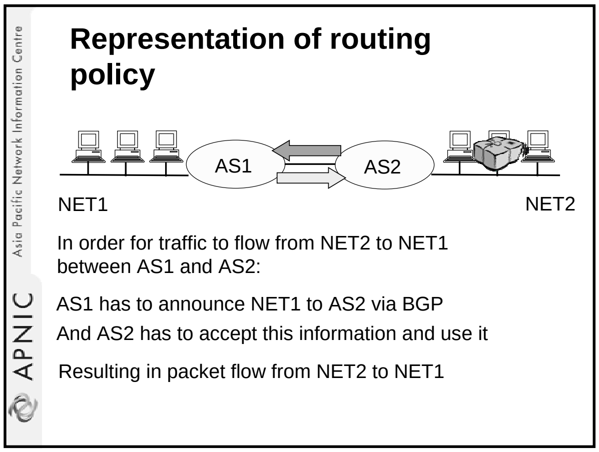
- Global Internet Routing Registry database
  - http://www.irr.net/
    - Uses RPSL
  - Established in 1995
- Stability and consistency of routing
  - network operators share information
- Both public and private databases
  - These databases are independent
    - but some exchange data
    - only register your data in one database



What is a Routing Registry?

ARIN, ArcStar, FGC, Verio, Bconnex, Optus, Telstra, ...

IRR = APNIC RR + RIPE DB + RADB + C&W + ARIN + …



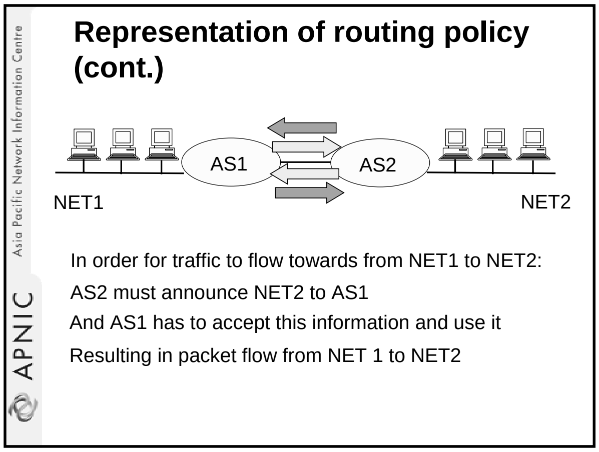
Representation of routing policy

In order for traffic to flow from NET2 to NET1 between AS1 and AS2:

AS1 has to announce NET1 to AS2 via BGP

And AS2 has to accept this information and use it

Resulting in packet flow from NET2 to NET1



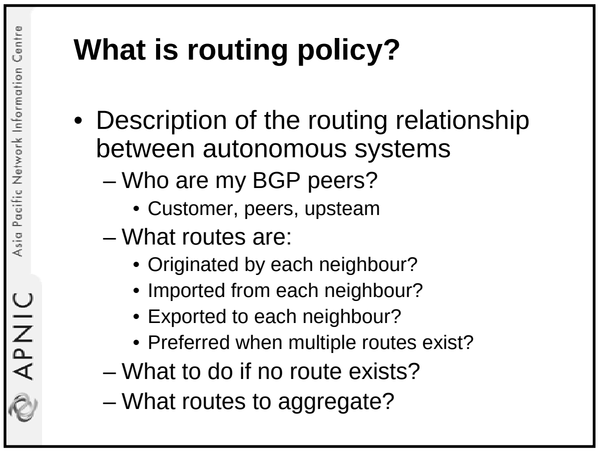
Representation of routing policy (cont.)

In order for traffic to flow towards from NET1 to NET2:

AS2 must announce NET2 to AS1

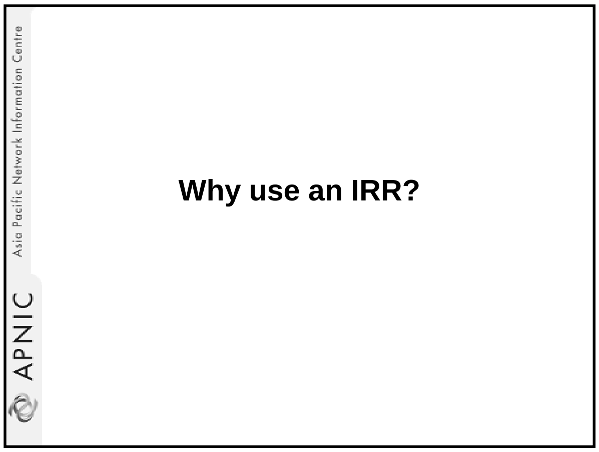
And AS1 has to accept this information and use it

Resulting in packet flow from NET 1 to NET2

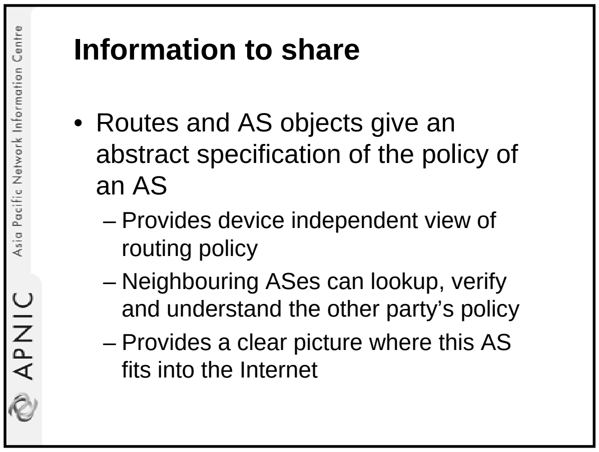


What is routing policy?

- Description of the routing relationship between autonomous systems
  - Who are my BGP peers?
    - Customer, peers, upsteam
  - What routes are:
    - Originated by each neighbour?
    - Imported from each neighbour?
    - Exported to each neighbour?
    - Preferred when multiple routes exist?
  - What to do if no route exists?
  - What routes to aggregate?

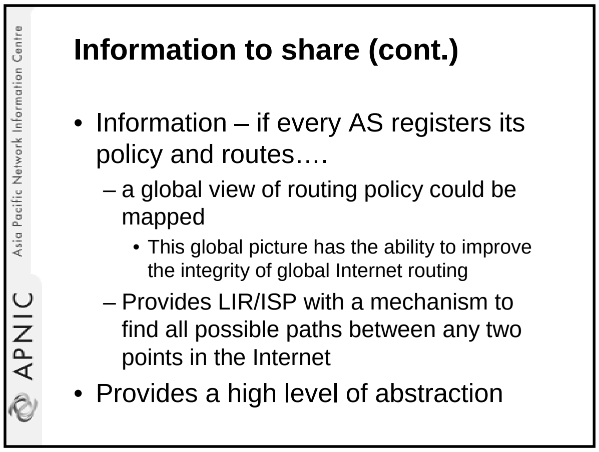


Why use an IRR?



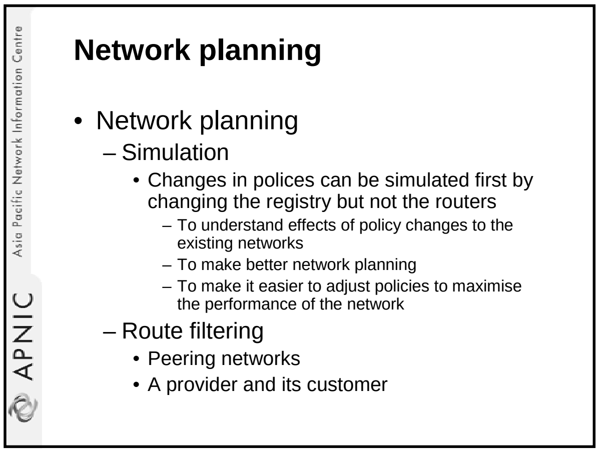
Information to share

- Routes and AS objects give an abstract specification of the policy of an AS
  - Provides device independent view of routing policy
  - Neighbouring ASes can lookup, verify and understand the other party’s policy
  - Provides a clear picture where this AS fits into the Internet



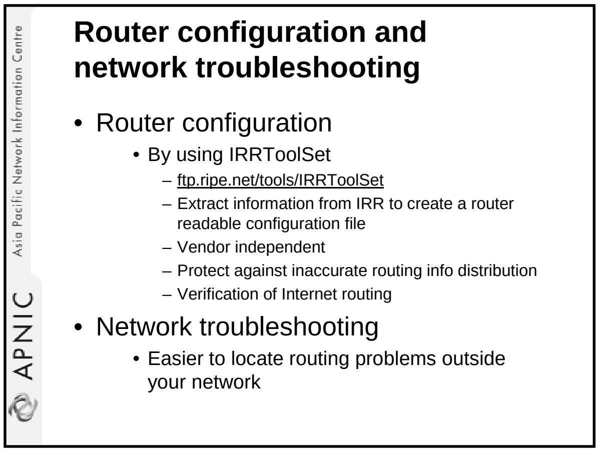
Information to share (cont.)

- Information – if every AS registers its policy and routes….
  - a global view of routing policy could be mapped
    - This global picture has the ability to improve the integrity of global Internet routing
  - Provides LIR/ISP with a mechanism to find all possible paths between any two points in the Internet
- Provides a high level of abstraction



Network planning

- Network planning
  - Simulation
    - Changes in polices can be simulated first by changing the registry but not the routers
      - To understand effects of policy changes to the existing networks
      - To make better network planning
      - To make it easier to adjust policies to maximise the performance of the network
  - Route filtering
    - Peering networks
    - A provider and its customer



Router configuration and network troubleshooting

- Router configuration
    - By using IRRToolSet
      - ftp.ripe.net/tools/IRRToolSet
      - Extract information from IRR to create a router readable configuration file
      - Vendor independent
      - Protect against inaccurate routing info distribution
      - Verification of Internet routing
- Network troubleshooting
    - Easier to locate routing problems outside your network

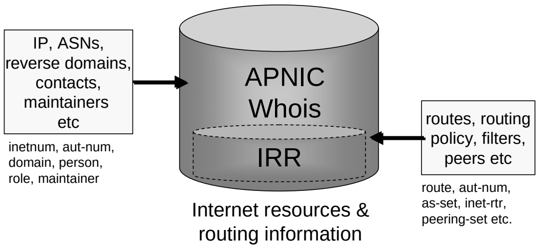
APNIC database and the IRR

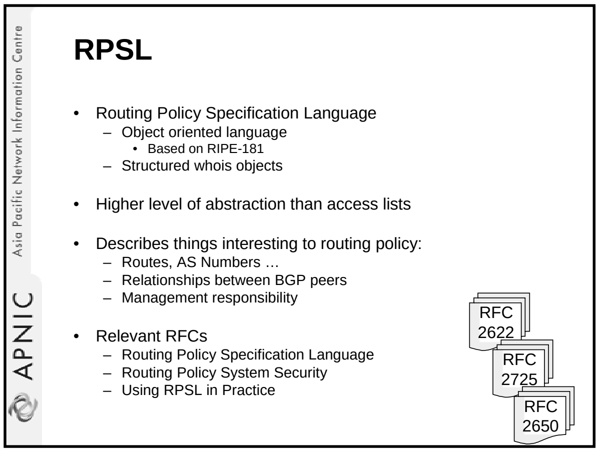
APNIC Database & the IRR

- APNIC whois Database
  - Two databases in one
- Public Network Management Database
  - “whois” info about networks & contact persons
    - IP addresses, AS numbers etc
- Routing Registry
  - contains routing information
    - routing policy, routes, filters, peers etc.
  - APNIC RR is part of the global IRR

Integration of Whois and IRR

- Integrated APNIC Whois Database & Internet Routing Registry

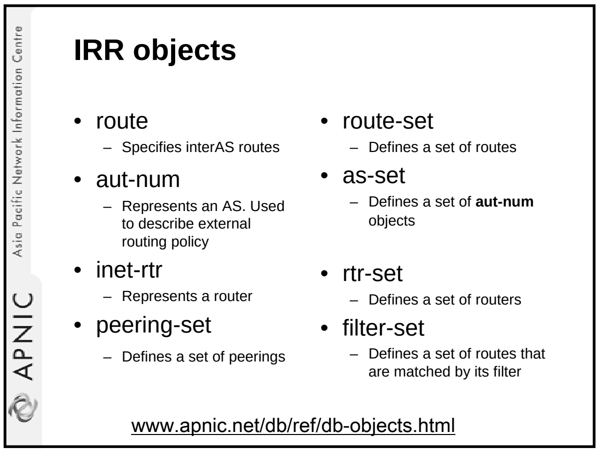

IP, ASNs, reverse domains, contacts, maintainers etc
inetnum, aut-num, domain, person, role, maintainer
APNIC Whois
IRR
Internet resources & routing information
routes, routing policy, filters, peers etc
route, aut-num, as-set, inet-rtr, peering-set etc.



RPSL

- Routing Policy Specification Language
  - Object oriented language
    - Based on RIPE-181
  - Structured whois objects
- Higher level of abstraction than access lists
- Describes things interesting to routing policy:
  - Routes, AS Numbers …
  - Relationships between BGP peers
  - Management responsibility
- Relevant RFCs
  - Routing Policy Specification Language
  - Routing Policy System Security
  - Using RPSL in Practice

RFC 2622
RFC 2725
RFC 2650
APNIC
Asia Pacific Network Information Centre

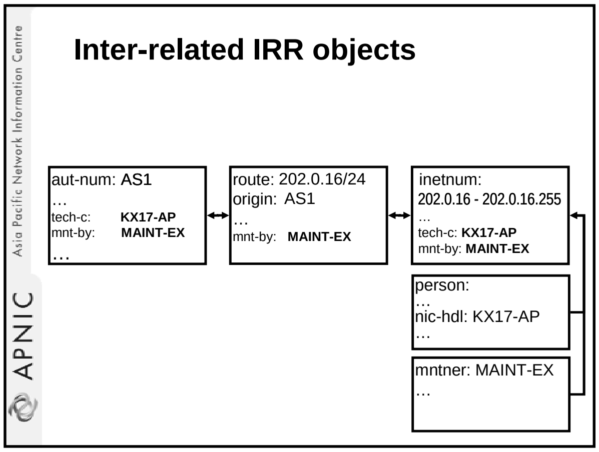


IRR objects

- route
  - Specifies interAS routes
- aut-num
  - Represents an AS. Used to describe external routing policy
- inet-rtr
  - Represents a router
- peering-set
  - Defines a set of peerings
- route-set
  - Defines a set of routes
- as-set
  - Defines a set of **aut-num** objects
- rtr-set
  - Defines a set of routers
- filter-set
  - Defines a set of routes that are matched by its filter

www.apnic.net/db/ref/db-objects.html

APNIC
Asia Pacific Network Information Centre



Inter-related IRR objects

```
aut-num: AS1
…
tech-c:     KX17-AP
mnt-by:     MAINT-EX
…
```

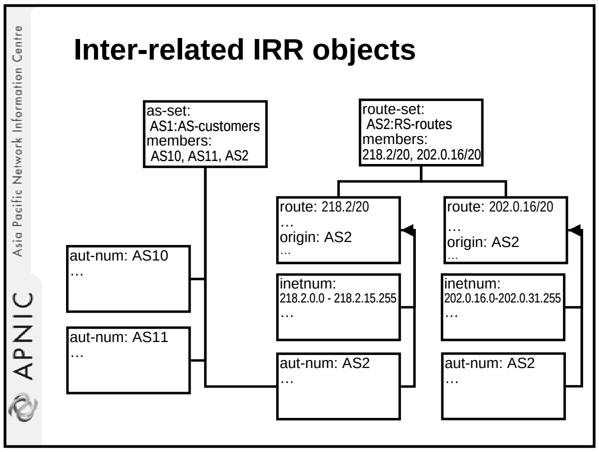
```
route: 202.0.16/24
origin:  AS1
…
mnt-by:   MAINT-EX
```

```
inetnum:
202.0.16 - 202.0.16.255
…
tech-c: KX17-AP
mnt-by: MAINT-EX
```

```
person:
…
nic-hdl: KX17-AP
…
```

```
mntner: MAINT-EX
…
```

APNIC
Asia Pacific Network Information Centre



Inter-related IRR objects

as-set:
AS1:AS-customers
members:
AS10, AS11, AS2

route-set:
AS2:RS-routes
members:
218.2/20, 202.0.16/20

aut-num: AS10
…

aut-num: AS11
…

route: 218.2/20
…
origin: AS2
…

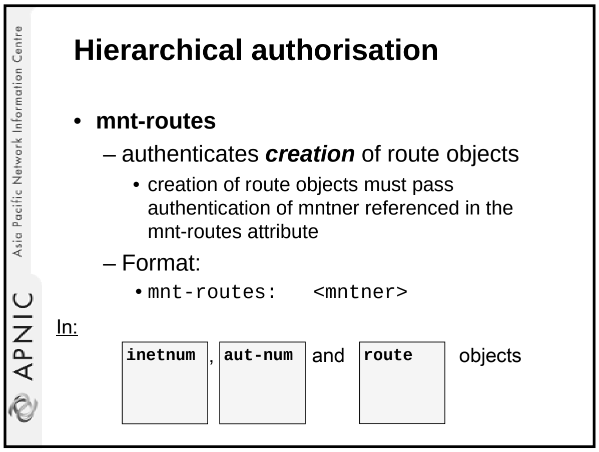
inetnum:
218.2.0.0 - 218.2.15.255
…

aut-num: AS2
…

route: 202.0.16/20
…
origin: AS2
…

inetnum:
202.0.16.0-202.0.31.255
…

aut-num: AS2
…

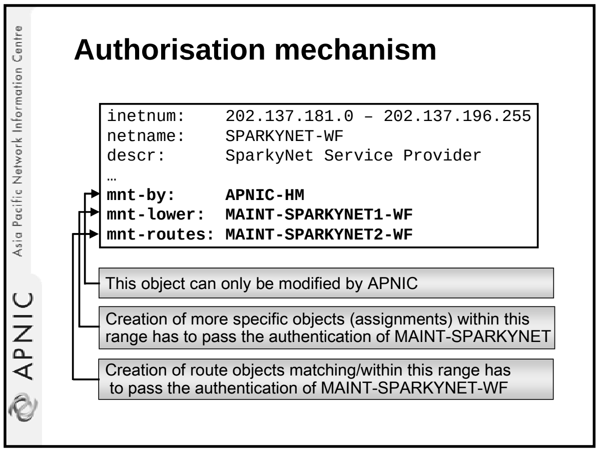


Hierarchical authorisation

- **mnt-routes**
  - authenticates ***creation*** of route objects
    - creation of route objects must pass authentication of mntner referenced in the mnt-routes attribute
  - Format:
    - `mnt-routes:   <mntner>`

In:

`inetnum` , **`aut-num`** and **`route`** objects



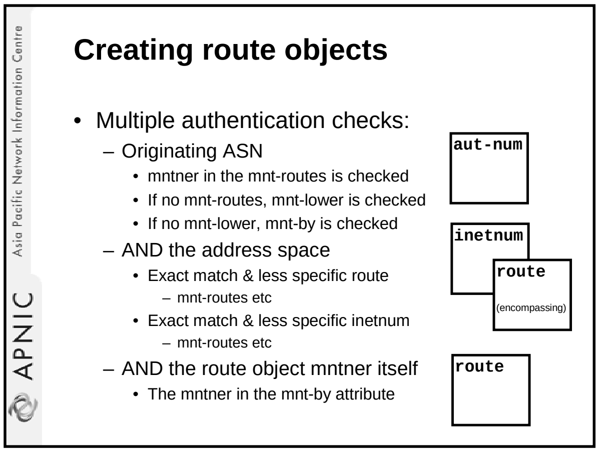
Authorisation mechanism

```
inetnum:     202.137.181.0 – 202.137.196.255
netname:     SPARKYNET-WF
descr:       SparkyNet Service Provider
…
mnt-by:      APNIC-HM
mnt-lower:   MAINT-SPARKYNET1-WF
mnt-routes:  MAINT-SPARKYNET2-WF
```

This object can only be modified by APNIC

Creation of more specific objects (assignments) within this range has to pass the authentication of MAINT-SPARKYNET

Creation of route objects matching/within this range has to pass the authentication of MAINT-SPARKYNET-WF



Creating route objects

- Multiple authentication checks:
  - Originating ASN
    - mntner in the mnt-routes is checked
    - If no mnt-routes, mnt-lower is checked
    - If no mnt-lower, mnt-by is checked
  - AND the address space
    - Exact match & less specific route
      - mnt-routes etc
    - Exact match & less specific inetnum
      - mnt-routes etc
  - AND the route object mntner itself
    - The mntner in the mnt-by attribute

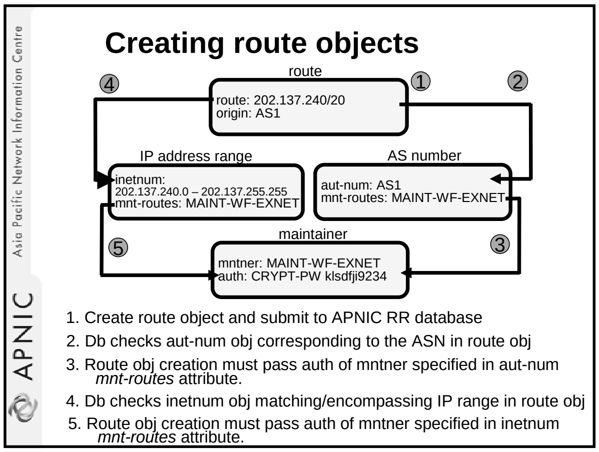
`aut-num`

`inetnum`

`route`

(encompassing)

`route`



Creating route objects

route

route: 202.137.240/20
origin: AS1

IP address range

inetnum:
202.137.240.0 – 202.137.255.255
mnt-routes: MAINT-WF-EXNET

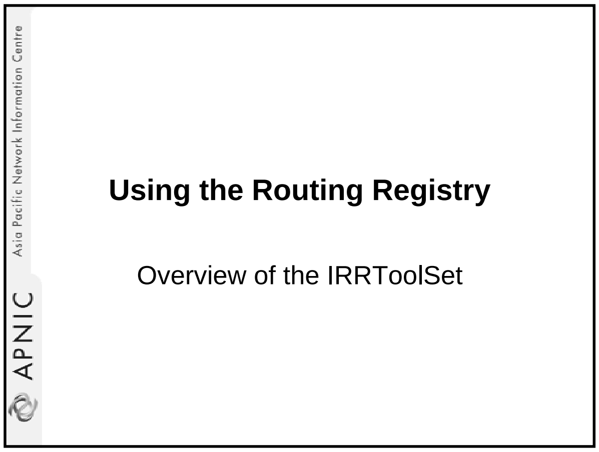
AS number

aut-num: AS1
mnt-routes: MAINT-WF-EXNET

maintainer

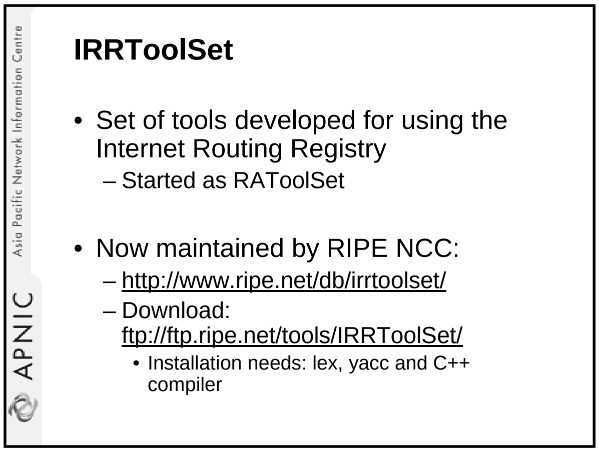
mntner: MAINT-WF-EXNET
auth: CRYPT-PW klsdfji9234

1. Create route object and submit to APNIC RR database
2. Db checks aut-num obj corresponding to the ASN in route obj
3. Route obj creation must pass auth of mntner specified in aut-num *mnt-routes* attribute.
4. Db checks inetnum obj matching/encompassing IP range in route obj
5. Route obj creation must pass auth of mntner specified in inetnum *mnt-routes* attribute.



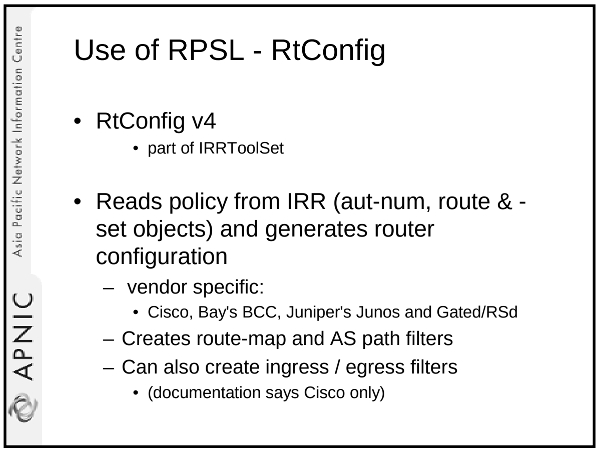
Using the Routing Registry

Overview of the IRRToolSet



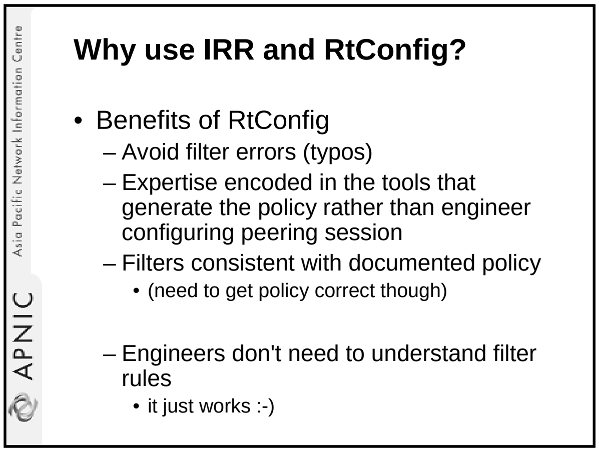
IRRToolSet

- Set of tools developed for using the Internet Routing Registry
  - Started as RAToolSet
- Now maintained by RIPE NCC:
  - http://www.ripe.net/db/irrtoolset/
  - Download: ftp://ftp.ripe.net/tools/IRRToolSet/
    - Installation needs: lex, yacc and C++ compiler



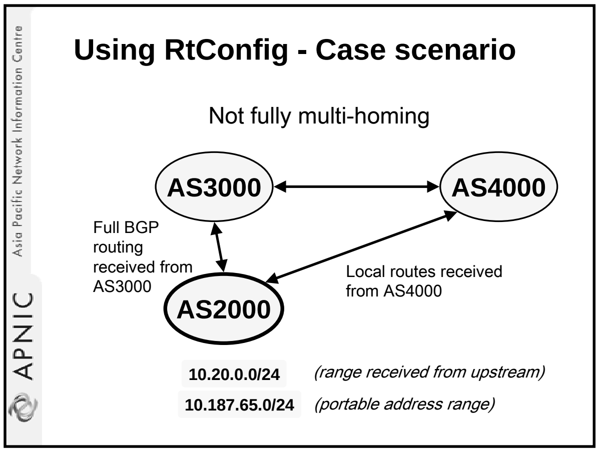
Use of RPSL - RtConfig

- RtConfig v4
    - part of IRRToolSet
- Reads policy from IRR (aut-num, route & -set objects) and generates router configuration
  - vendor specific:
    - Cisco, Bay's BCC, Juniper's Junos and Gated/RSd
  - Creates route-map and AS path filters
  - Can also create ingress / egress filters
    - (documentation says Cisco only)



Why use IRR and RtConfig?

- Benefits of RtConfig
  - Avoid filter errors (typos)
  - Expertise encoded in the tools that generate the policy rather than engineer configuring peering session
  - Filters consistent with documented policy
    - (need to get policy correct though)
  - Engineers don't need to understand filter rules
    - it just works :-)



Using RtConfig - Case scenario

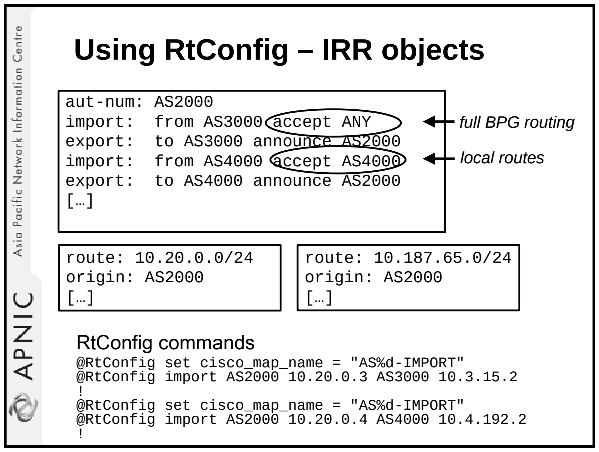
Not fully multi-homing

AS3000
AS4000
Full BGP routing received from AS3000
AS2000
Local routes received from AS4000

10.20.0.0/24 *(range received from upstream)*

10.187.65.0/24 *(portable address range)*

Asia Pacific Network Information Centre
APNIC



Using RtConfig – IRR objects

```
aut-num: AS2000
import:  from AS3000 accept ANY
export:  to AS3000 announce AS2000
import:  from AS4000 accept AS4000
export:  to AS4000 announce AS2000
[…]
```

full BPG routing

local routes

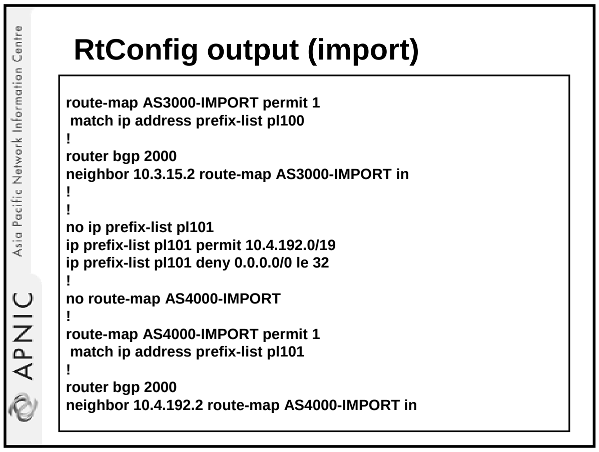
```
route: 10.20.0.0/24
origin: AS2000
[…]
```

```
route: 10.187.65.0/24
origin: AS2000
[…]
```

RtConfig commands

```
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.4 AS4000 10.4.192.2
!
```

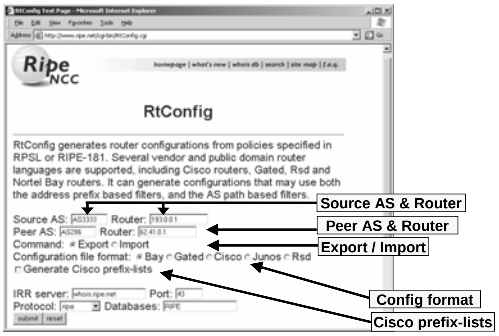
Asia Pacific Network Information Centre
APNIC



RtConfig output (import)

```
route-map AS3000-IMPORT permit 1
 match ip address prefix-list pl100
!
router bgp 2000
neighbor 10.3.15.2 route-map AS3000-IMPORT in
!
!
no ip prefix-list pl101
ip prefix-list pl101 permit 10.4.192.0/19
ip prefix-list pl101 deny 0.0.0.0/0 le 32
!
no route-map AS4000-IMPORT
!
route-map AS4000-IMPORT permit 1
 match ip address prefix-list pl101
!
router bgp 2000
neighbor 10.4.192.2 route-map AS4000-IMPORT in
```

RtConfig – web prototype



RtConfig Test Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.ripe.net/cgi-bin/RtConfig.cgi
Ripe NCC
homepage | what's new | whois db | search | site map | f.a.q.
submit
reset

RtConfig

RtConfig generates router configurations from policies specified in RPSL or RIPE-181. Several vendor and public domain router languages are supported, including Cisco routers, Gated, Rsd and Nortel Bay routers. It can generate configurations that may use both the address prefix based filters, and the AS path based filters.

Source AS: AS3333 Router: 193.0.0.1
Peer AS: AS286 Router: 62.41.0.1
Command: ● Export ○ Import
Configuration file format: ● Bay ○ Gated ○ Cisco ○ Junos ○ Rsd
□ Generate Cisco prefix-lists

IRR server: whois.ripe.net Port: 43
Protocol: ripe Databases: RIPE

Source AS & Router

Peer AS & Router

Export / Import

Config format

Cisco prefix-lists

http://www.ripe.net/cgi-bin/RtConfig.cgi

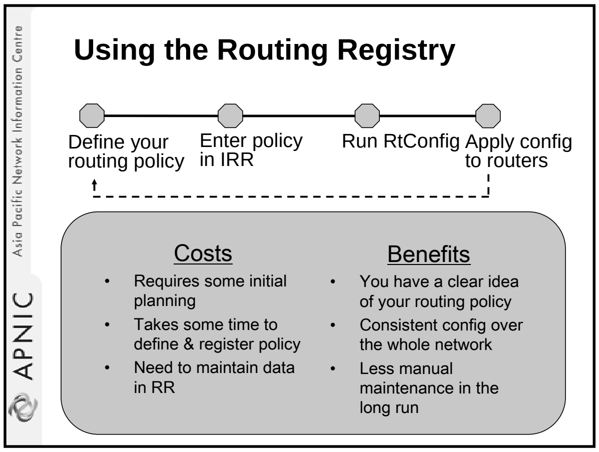
RtConfig - web prototype

- Let’s try the same command we tried in the previous slide:

@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2

- You can select different configuration format depends on a vendor
  - Select “Cicso”
  - Select “Bay”
  - Select “Junos”

Benefit of using IRR



Using the Routing Registry

Define your routing policy

Enter policy in IRR

Run RtConfig

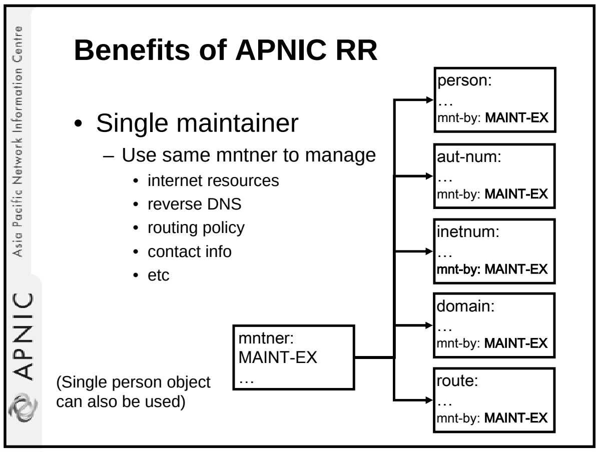
Apply config to routers

Costs

- Requires some initial planning
- Takes some time to define & register policy
- Need to maintain data in RR

Benefits

- You have a clear idea of your routing policy
- Consistent config over the whole network
- Less manual maintenance in the long run



Benefits of APNIC RR

- Single maintainer
  - Use same mntner to manage
    - internet resources
    - reverse DNS
    - routing policy
    - contact info
    - etc

(Single person object can also be used)

```
mntner:
MAINT-EX
…
```

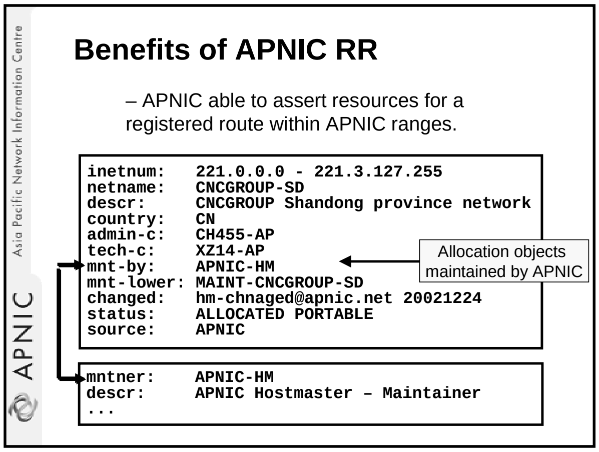
```
person:
…
mnt-by: MAINT-EX
```

```
aut-num:
…
mnt-by: MAINT-EX
```

```
inetnum:
…
mnt-by: MAINT-EX
```

```
domain:
…
mnt-by: MAINT-EX
```

```
route:
…
mnt-by: MAINT-EX
```


Asia Pacific Network Information Centre
APNIC

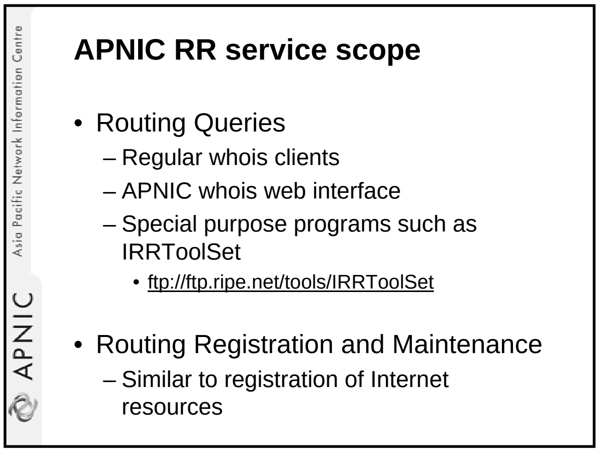
Benefits of APNIC RR

– APNIC able to assert resources for a registered route within APNIC ranges.

```
inetnum:   221.0.0.0 - 221.3.127.255
netname:   CNCGROUP-SD
descr:     CNCGROUP Shandong province network
country:   CN
admin-c:   CH455-AP
tech-c:    XZ14-AP
mnt-by:    APNIC-HM
mnt-lower: MAINT-CNCGROUP-SD
changed:   hm-chnaged@apnic.net 20021224
status:    ALLOCATED PORTABLE
source:    APNIC
```

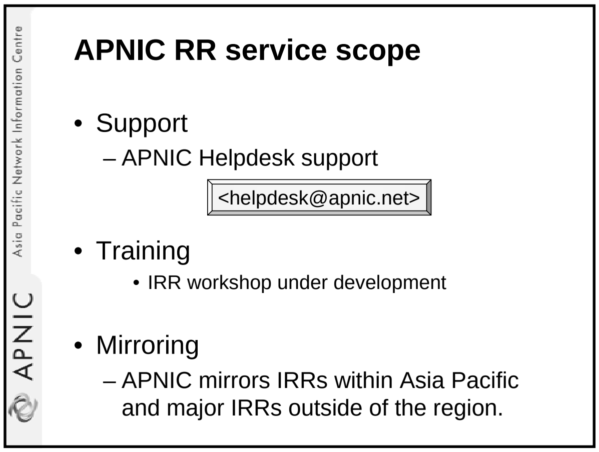
Allocation objects maintained by APNIC

```
mntner:    APNIC-HM
descr:     APNIC Hostmaster – Maintainer
...
```



APNIC RR service scope

- Routing Queries
  - Regular whois clients
  - APNIC whois web interface
  - Special purpose programs such as IRRToolSet
    - ftp://ftp.ripe.net/tools/IRRToolSet
- Routing Registration and Maintenance
  - Similar to registration of Internet resources

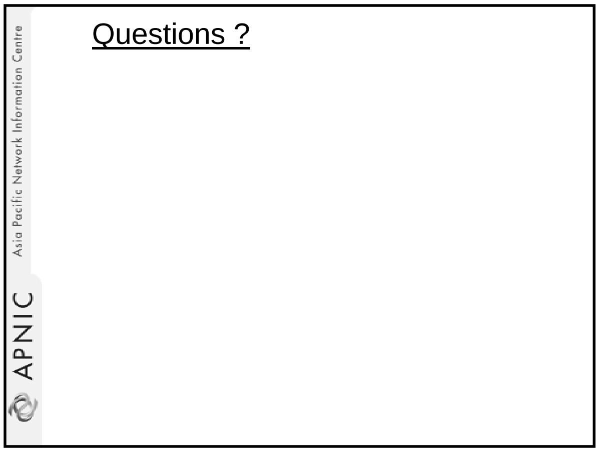


APNIC RR service scope

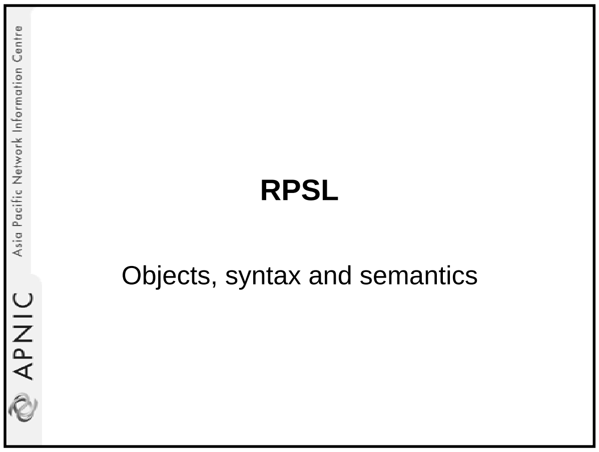
- Support
  - APNIC Helpdesk support

<helpdesk@apnic.net>

- Training
  - IRR workshop under development
- Mirroring
  - APNIC mirrors IRRs within Asia Pacific and major IRRs outside of the region.

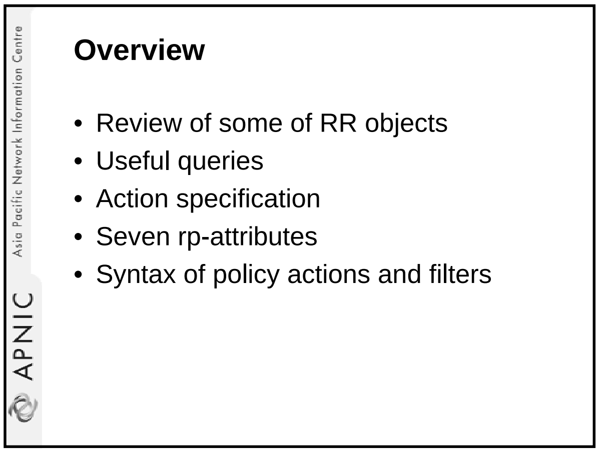


Questions ?



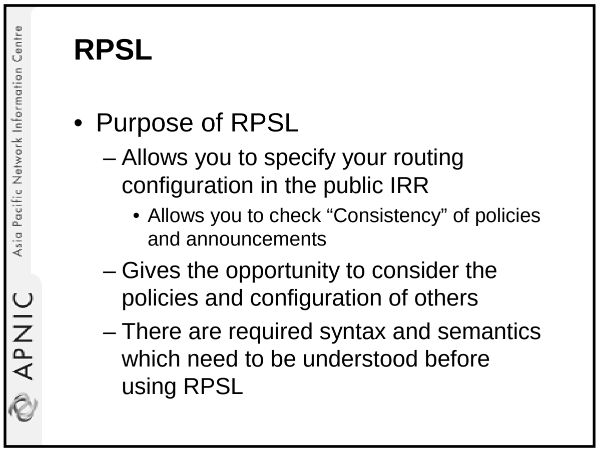
RPSL

Objects, syntax and semantics



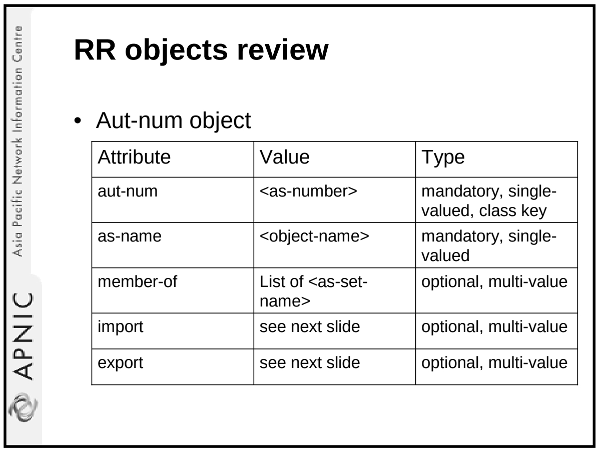
Overview

- Review of some of RR objects
- Useful queries
- Action specification
- Seven rp-attributes
- Syntax of policy actions and filters



RPSL

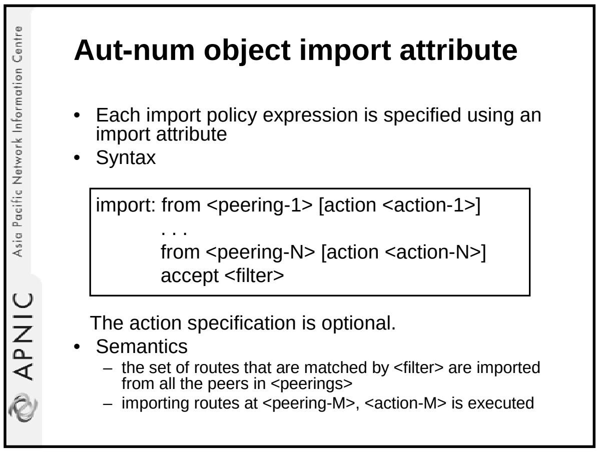
- Purpose of RPSL
  - Allows you to specify your routing configuration in the public IRR
    - Allows you to check “Consistency” of policies and announcements
  - Gives the opportunity to consider the policies and configuration of others
  - There are required syntax and semantics which need to be understood before using RPSL



RR objects review

- Aut-num object

Attribute	Value	Type
aut-num	<as-number>	mandatory, single-valued, class key
as-name	<object-name>	mandatory, single-valued
member-of	List of <as-set-name>	optional, multi-value
import	see next slide	optional, multi-value
export	see next slide	optional, multi-value



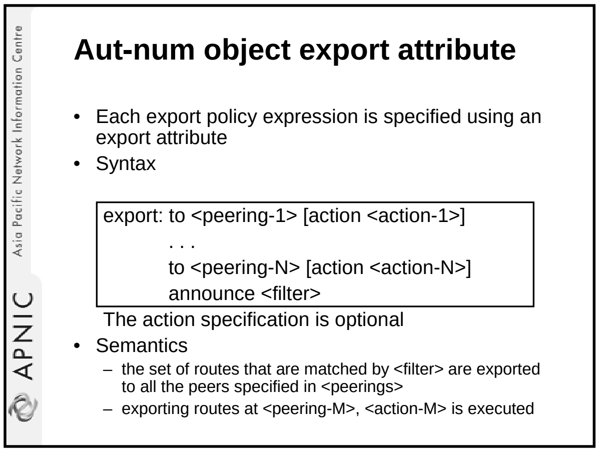
Aut-num object import attribute

- Each import policy expression is specified using an import attribute
- Syntax

```
import: from <peering-1> [action <action-1>]
           . . .
           from <peering-N> [action <action-N>]
           accept <filter>
```

The action specification is optional.

- Semantics
  - the set of routes that are matched by <filter> are imported from all the peers in <peerings>
  - importing routes at <peering-M>, <action-M> is executed



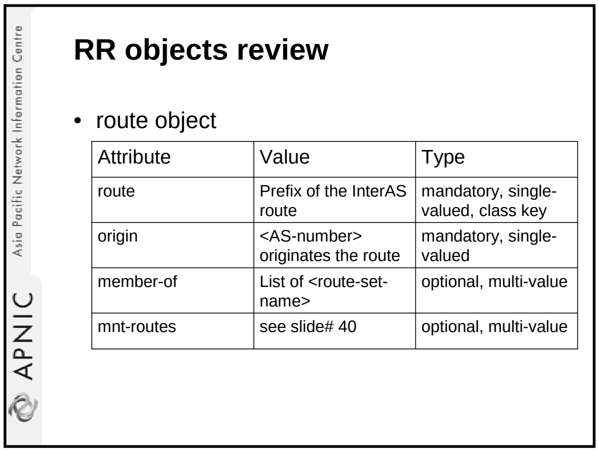
Aut-num object export attribute

- Each export policy expression is specified using an export attribute
- Syntax

```
export: to <peering-1> [action <action-1>]
           . . .
           to <peering-N> [action <action-N>]
           announce <filter>
```

The action specification is optional

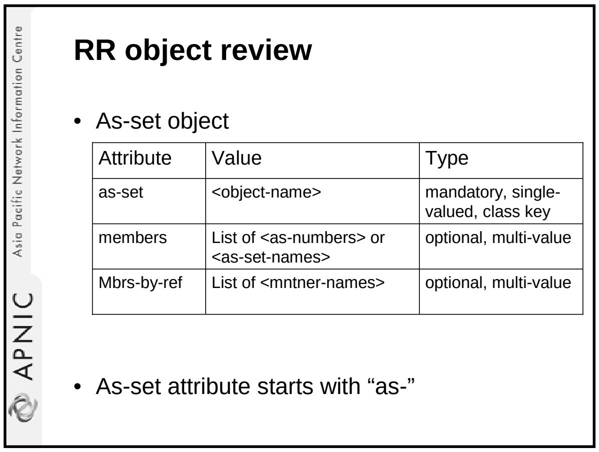
- Semantics
  - the set of routes that are matched by <filter> are exported to all the peers specified in <peerings>
  - exporting routes at <peering-M>, <action-M> is executed



RR objects review

- route object

Attribute	Value	Type
route	Prefix of the InterAS route	mandatory, single-valued, class key
origin	<AS-number> originates the route	mandatory, single-valued
member-of	List of <route-set-name>	optional, multi-value
mnt-routes	see slide# 40	optional, multi-value

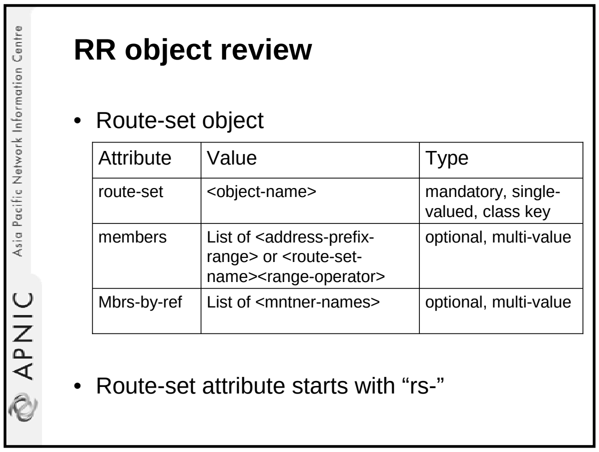


RR object review

- As-set object

Attribute	Value	Type
as-set	<object-name>	mandatory, single-valued, class key
members	List of <as-numbers> or <as-set-names>	optional, multi-value
Mbrs-by-ref	List of <mntner-names>	optional, multi-value

- As-set attribute starts with “as-”

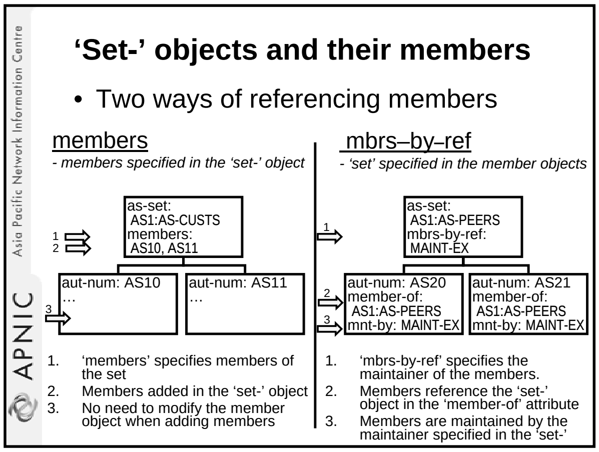


RR object review

- Route-set object

Attribute	Value	Type
route-set	<object-name>	mandatory, single-valued, class key
members	List of <address-prefix-range> or <route-set-name><range-operator>	optional, multi-value
Mbrs-by-ref	List of <mntner-names>	optional, multi-value

- Route-set attribute starts with “rs-”



'Set-' objects and their members

- Two ways of referencing members

members

- members specified in the 'set-' object

```
as-set:
 AS1:AS-CUSTS
members:
 AS10, AS11
```

```
aut-num: AS10
…
```

```
aut-num: AS11
…
```

1. 'members' specifies members of the set
2. Members added in the 'set-' object
3. No need to modify the member object when adding members

mbrs–by–ref

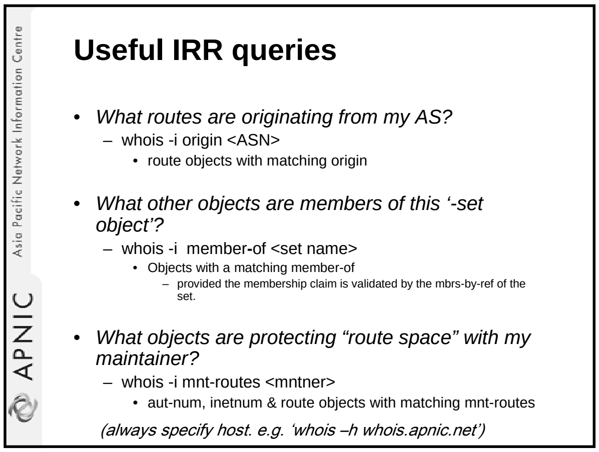
- 'set' specified in the member objects

```
as-set:
 AS1:AS-PEERS
mbrs-by-ref:
 MAINT-EX
```

```
aut-num: AS20
member-of:
 AS1:AS-PEERS
mnt-by: MAINT-EX
```

```
aut-num: AS21
member-of:
 AS1:AS-PEERS
mnt-by: MAINT-EX
```

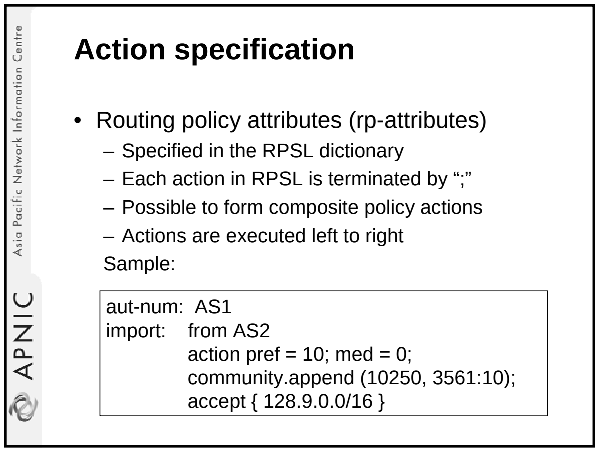
1. 'mbrs-by-ref' specifies the maintainer of the members.
2. Members reference the 'set-' object in the 'member-of' attribute
3. Members are maintained by the maintainer specified in the 'set-'



Useful IRR queries

- *What routes are originating from my AS?*
  - whois -i origin <ASN>
    - route objects with matching origin
- *What other objects are members of this '-set object'?*
  - whois -i member-of <set name>
    - Objects with a matching member-of
      - provided the membership claim is validated by the mbrs-by-ref of the set.
- *What objects are protecting "route space" with my maintainer?*
  - whois -i mnt-routes <mntner>
    - aut-num, inetnum & route objects with matching mnt-routes

(always specify host. e.g. 'whois –h whois.apnic.net')

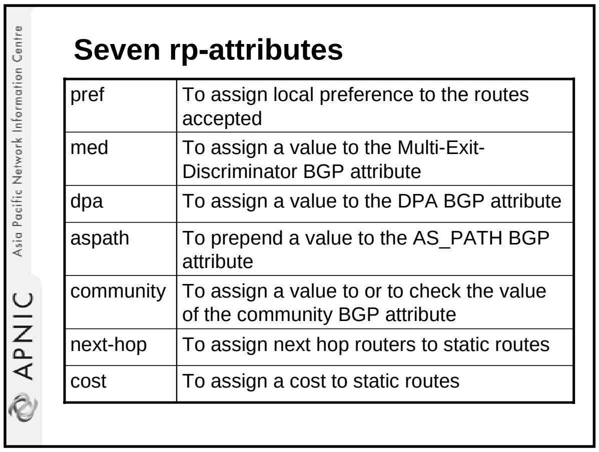


Action specification

- Routing policy attributes (rp-attributes)
  - Specified in the RPSL dictionary
  - Each action in RPSL is terminated by ";"
  - Possible to form composite policy actions
  - Actions are executed left to right

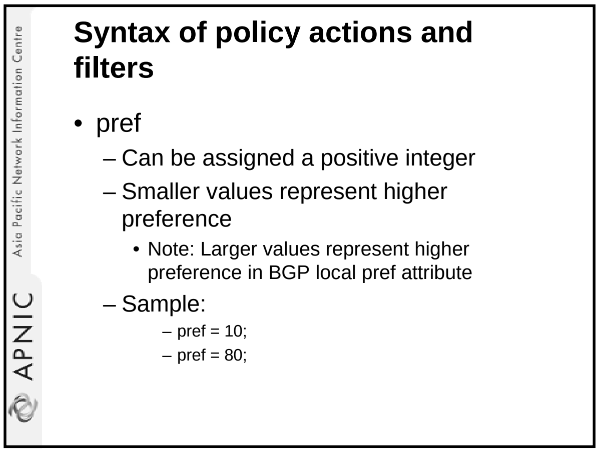
Sample:

```
aut-num:  AS1
import:   from AS2
          action pref = 10; med = 0;
          community.append (10250, 3561:10);
          accept { 128.9.0.0/16 }
```



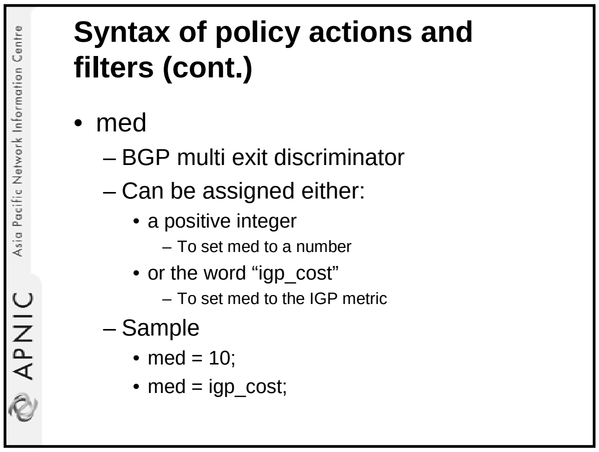
Seven rp-attributes

pref	To assign local preference to the routes accepted
med	To assign a value to the Multi-Exit-Discriminator BGP attribute
dpa	To assign a value to the DPA BGP attribute
aspath	To prepend a value to the AS_PATH BGP attribute
community	To assign a value to or to check the value of the community BGP attribute
next-hop	To assign next hop routers to static routes
cost	To assign a cost to static routes



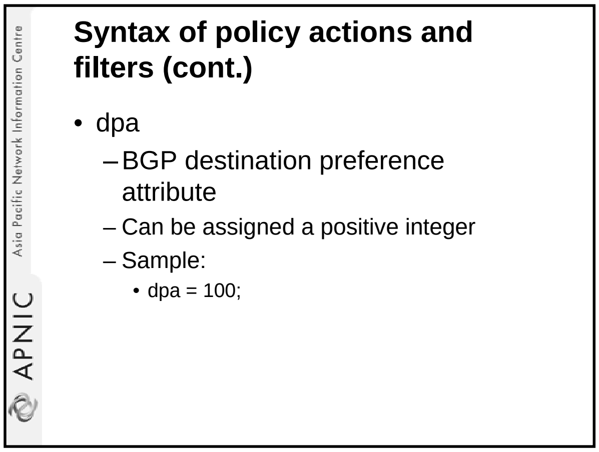
Syntax of policy actions and filters

- pref
  - Can be assigned a positive integer
  - Smaller values represent higher preference
    - Note: Larger values represent higher preference in BGP local pref attribute
  - Sample:
    - pref = 10;
    - pref = 80;



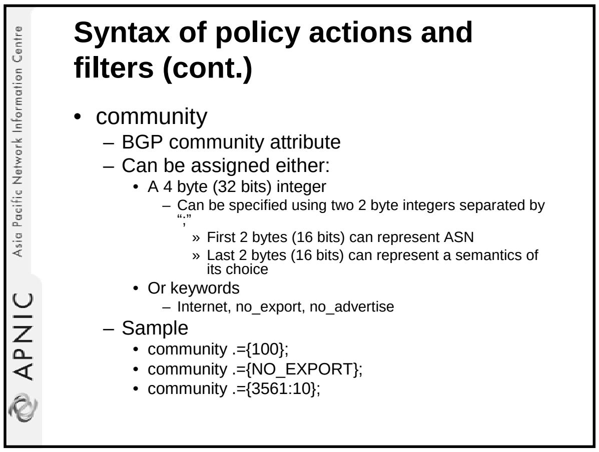
Syntax of policy actions and filters (cont.)

- med
  - BGP multi exit discriminator
  - Can be assigned either:
    - a positive integer
      - To set med to a number
    - or the word “igp_cost”
      - To set med to the IGP metric
  - Sample
    - med = 10;
    - med = igp_cost;



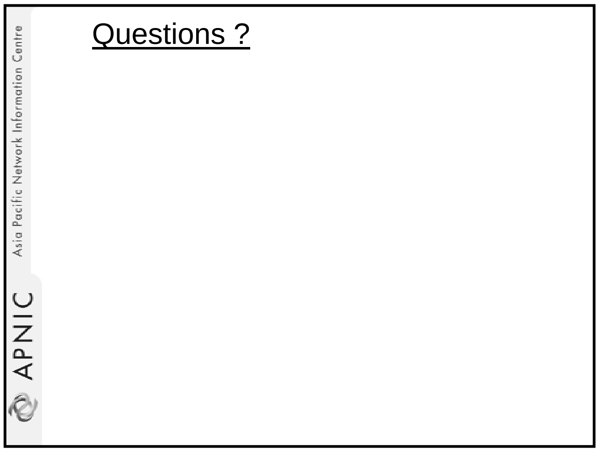
Syntax of policy actions and filters (cont.)

- dpa
  - BGP destination preference attribute
  - Can be assigned a positive integer
  - Sample:
    - dpa = 100;

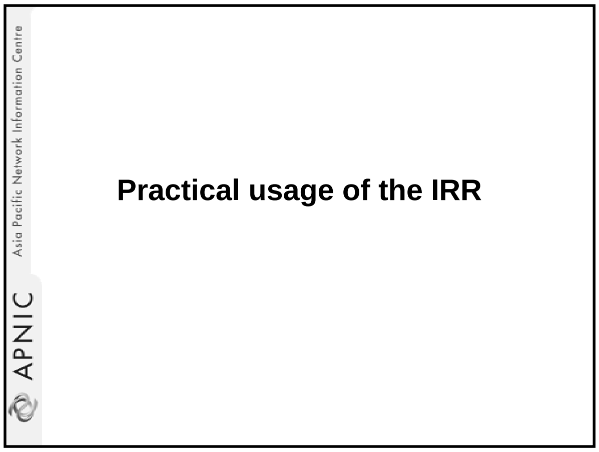


Syntax of policy actions and filters (cont.)

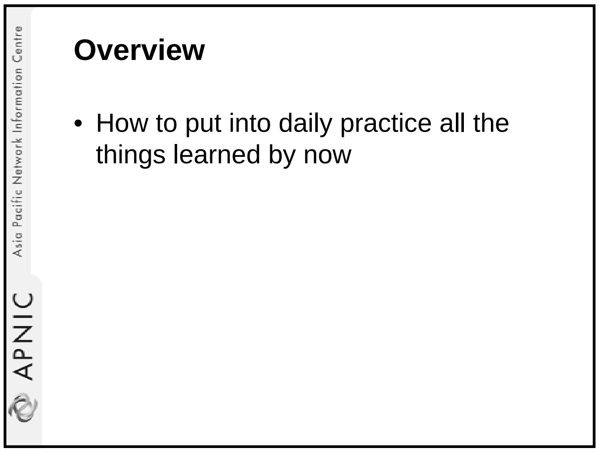
- community
  - BGP community attribute
  - Can be assigned either:
    - A 4 byte (32 bits) integer
      - Can be specified using two 2 byte integers separated by “;”
        - » First 2 bytes (16 bits) can represent ASN
        - » Last 2 bytes (16 bits) can represent a semantics of its choice
    - Or keywords
      - Internet, no_export, no_advertise
  - Sample
    - community .={100};
    - community .={NO_EXPORT};
    - community .={3561:10};



<u>Questions ?</u>

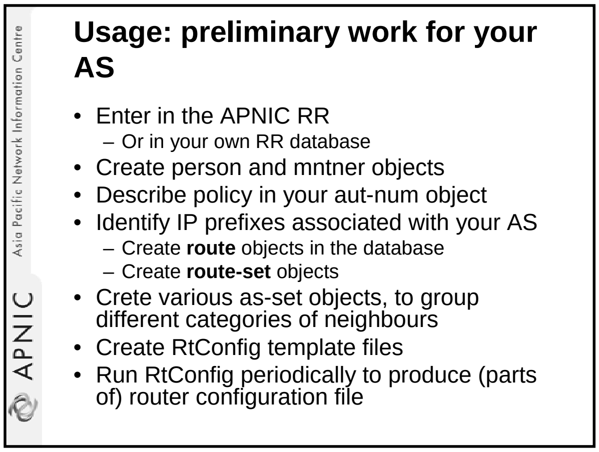


Practical usage of the IRR



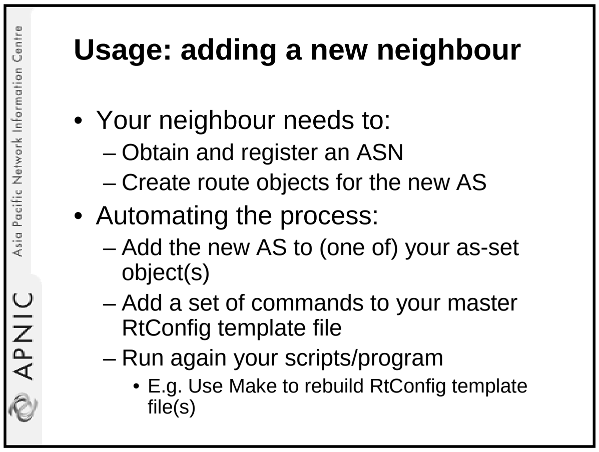
Overview

- How to put into daily practice all the things learned by now



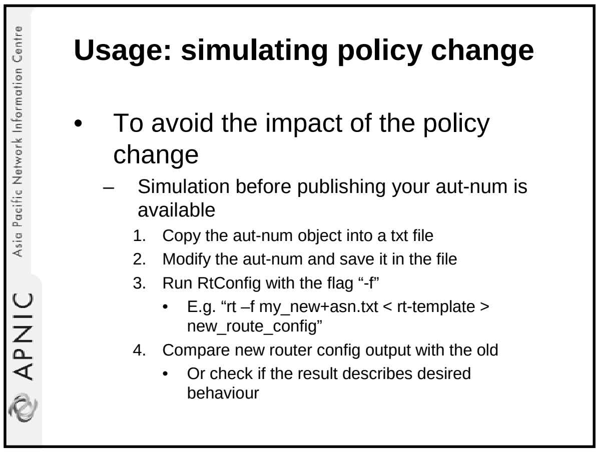
Usage: preliminary work for your AS

- Enter in the APNIC RR
  - Or in your own RR database
- Create person and mntner objects
- Describe policy in your aut-num object
- Identify IP prefixes associated with your AS
  - Create **route** objects in the database
  - Create **route-set** objects
- Crete various as-set objects, to group different categories of neighbours
- Create RtConfig template files
- Run RtConfig periodically to produce (parts of) router configuration file



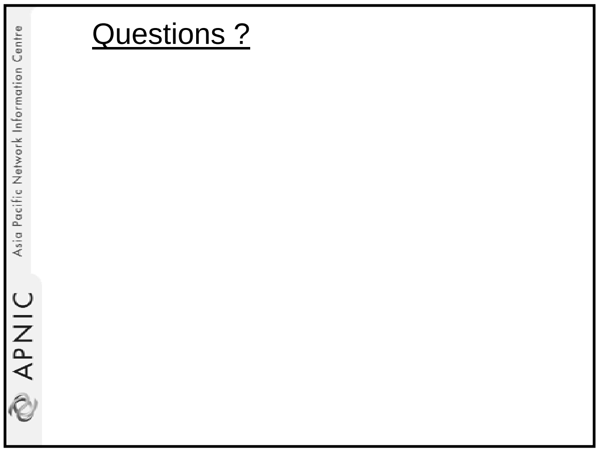
Usage: adding a new neighbour

- Your neighbour needs to:
  - Obtain and register an ASN
  - Create route objects for the new AS
- Automating the process:
  - Add the new AS to (one of) your as-set object(s)
  - Add a set of commands to your master RtConfig template file
  - Run again your scripts/program
    - E.g. Use Make to rebuild RtConfig template file(s)

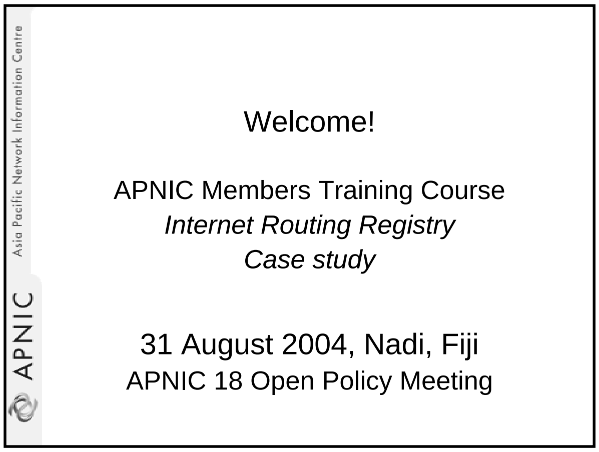


Usage: simulating policy change

- To avoid the impact of the policy change
  - Simulation before publishing your aut-num is available
    1. Copy the aut-num object into a txt file
    2. Modify the aut-num and save it in the file
    3. Run RtConfig with the flag “-f”
       - E.g. “rt –f my_new+asn.txt < rt-template > new_route_config”
    4. Compare new router config output with the old
       - Or check if the result describes desired behaviour



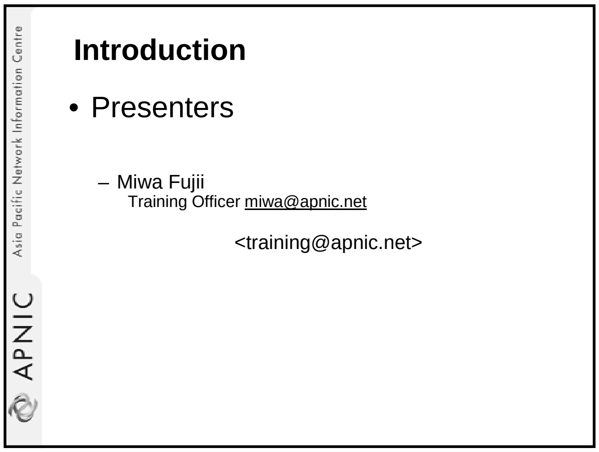
Questions ?



Welcome!

APNIC Members Training Course
Internet Routing Registry
Case study

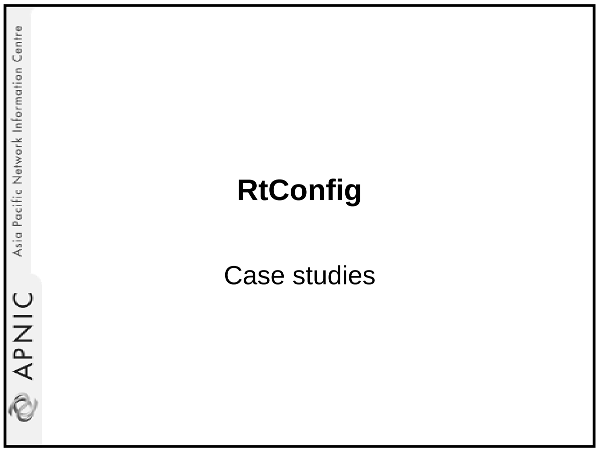
31 August 2004, Nadi, Fiji
APNIC 18 Open Policy Meeting



Introduction

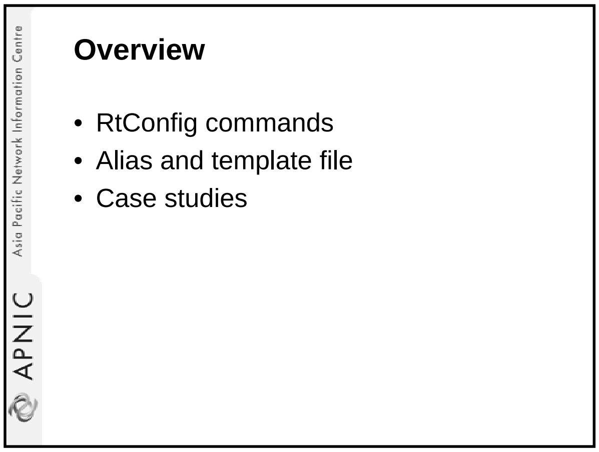
- Presenters
  - Miwa Fujii
    Training Officer miwa@apnic.net

<training@apnic.net>



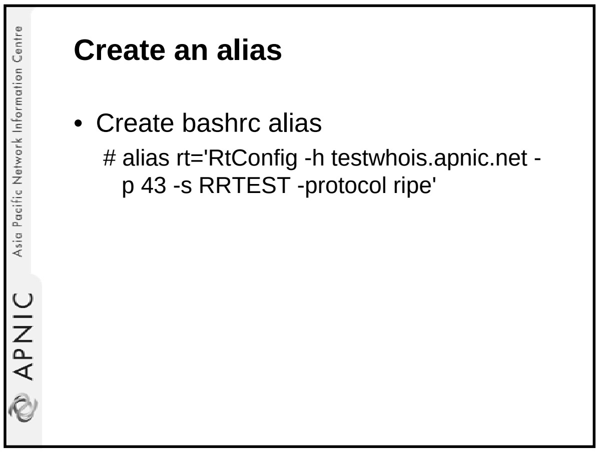
RtConfig

Case studies



Overview

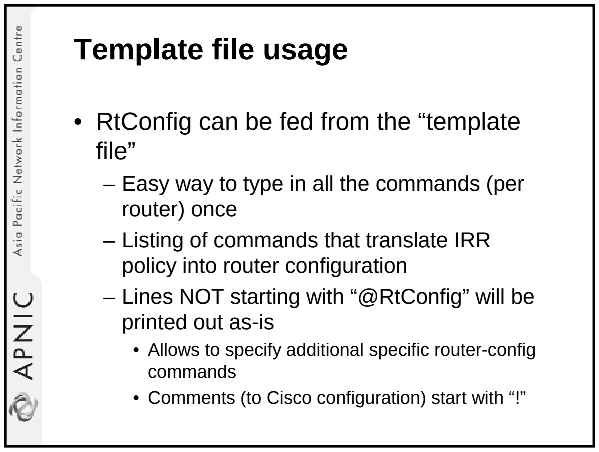
- RtConfig commands
- Alias and template file
- Case studies



Create an alias

- Create bashrc alias

```
# alias rt='RtConfig -h testwhois.apnic.net -
  p 43 -s RRTEST -protocol ripe'
```



Template file usage

- RtConfig can be fed from the “template file”
  - Easy way to type in all the commands (per router) once
  - Listing of commands that translate IRR policy into router configuration
  - Lines NOT starting with “@RtConfig” will be printed out as-is
    - Allows to specify additional specific router-config commands
    - Comments (to Cisco configuration) start with “!”

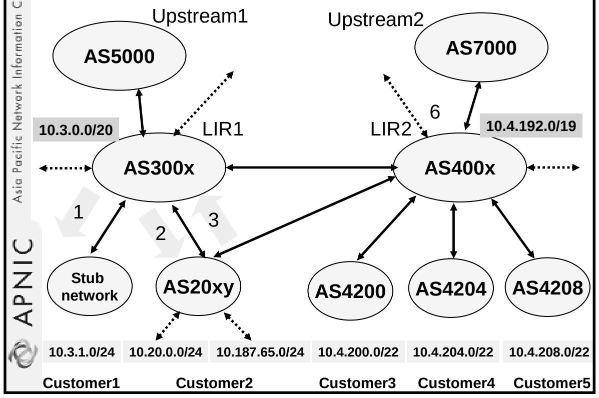
Template file usage (cont.)

- Create a template file as a txt file
  - e.g. case1-rt-file.txt

```
@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2
```

- Use the following command line (alias is handy)
  - rt < case1-rt-file.txt > output-case1.txt
- RtConfig output can be stored into the specified file name

Experimental Setup: AS Relations



Case Studies Overview

- AS3000 has a number of routing requirements:
  - Static routes injected at customer edge
  - BGP peering with customers
  - BGP peering with peers
  - BGP peering with upstream providers
- Learning objectives
  - how to express routing policies via RPSL
  - how to extract information from IRR by using RtConfig

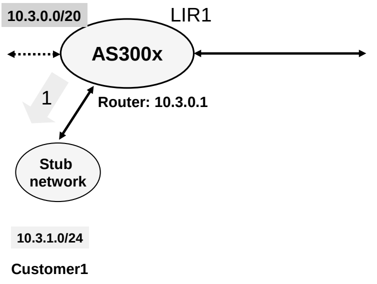
Case study overview (cont.)

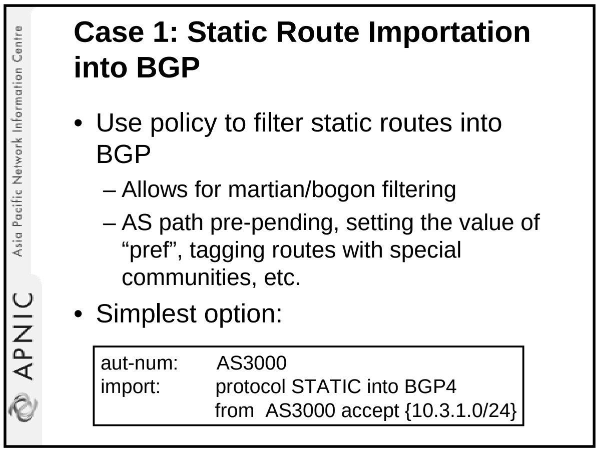
- Case 1: Provider inserting static routes
- Case 2: Provider setup for various BGP customers
- Case 3: Multi-homed customer set-up

Case study overview (cont.)

- AS “families” used in the examples
  - AS300x = {AS3000, AS3001, AS3007, …}
  - Similar “position”, policies of different complexity
    - Enables keeping track of changes throughout the course
- Neighbour = customer, peer, upstream…

Case 1: Static route importation into BGP

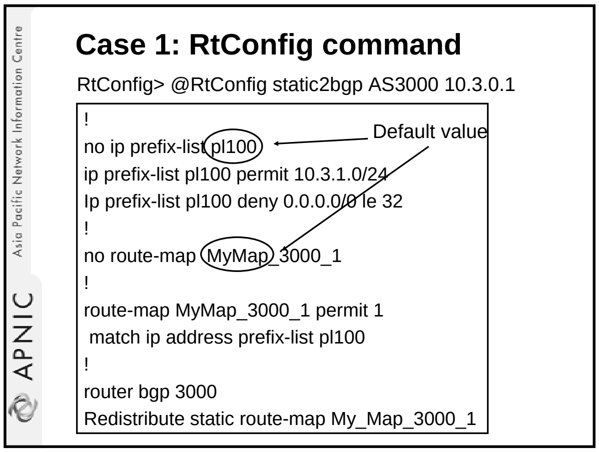




Case 1: Static Route Importation into BGP

- Use policy to filter static routes into BGP
  - Allows for martian/bogon filtering
  - AS path pre-pending, setting the value of "pref", tagging routes with special communities, etc.
- Simplest option:

```
aut-num:     AS3000
import:      protocol STATIC into BGP4
             from  AS3000 accept {10.3.1.0/24}
```

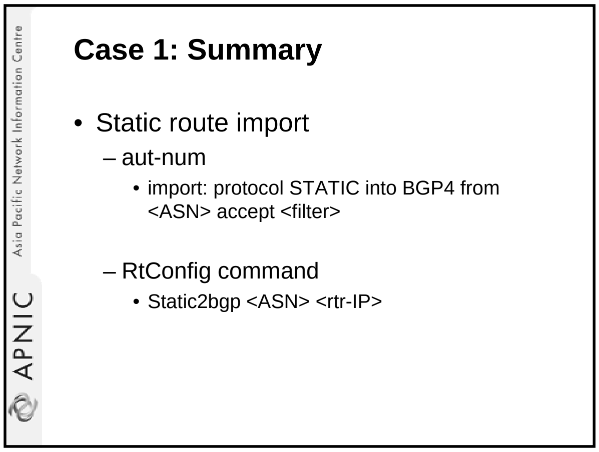


Case 1: RtConfig command

RtConfig> @RtConfig static2bgp AS3000 10.3.0.1

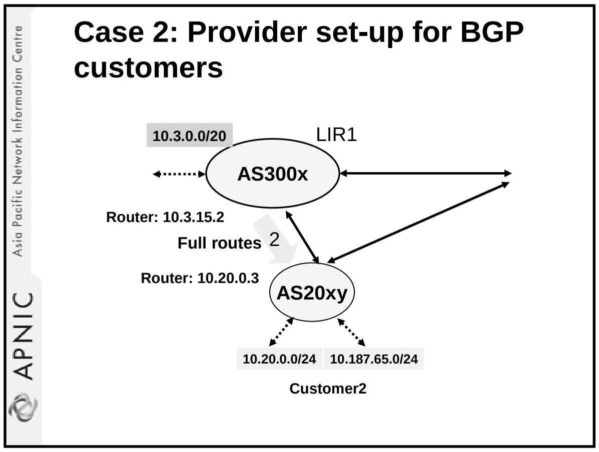
```
!
no ip prefix-list pl100
ip prefix-list pl100 permit 10.3.1.0/24
Ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map MyMap_3000_1
!
route-map MyMap_3000_1 permit 1
 match ip address prefix-list pl100
!
router bgp 3000
Redistribute static route-map My_Map_3000_1
```

Default value

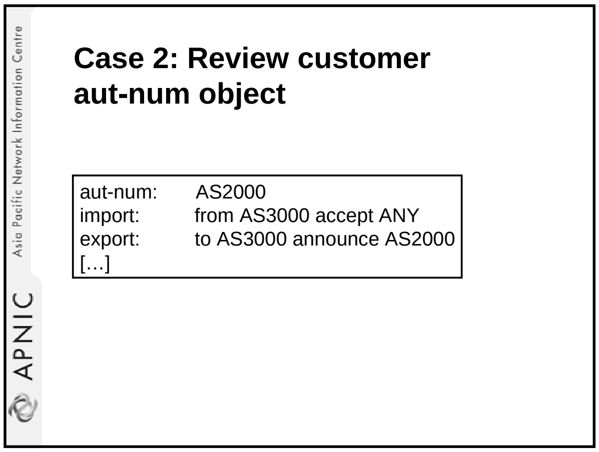


Case 1: Summary

- Static route import
  - aut-num
    - import: protocol STATIC into BGP4 from <ASN> accept <filter>
  - RtConfig command
    - Static2bgp <ASN> <rtr-IP>

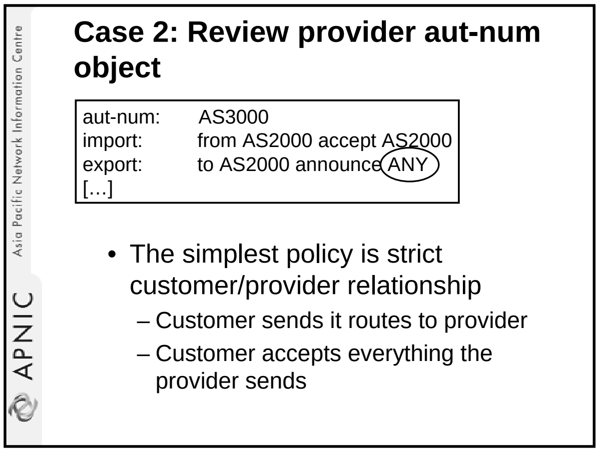


Case 2: Provider set-up for BGP customers



Case 2: Review customer aut-num object

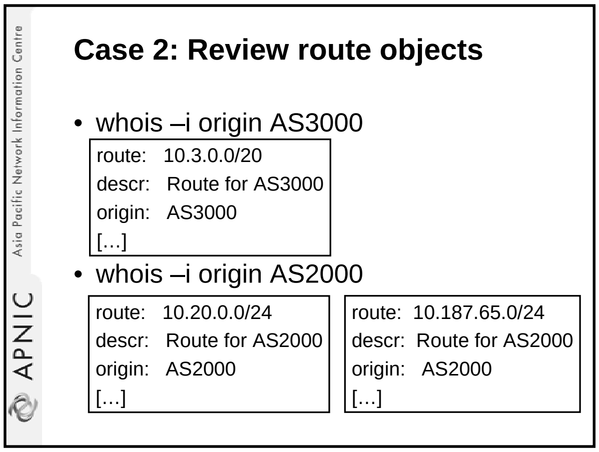
```
aut-num:     AS2000
import:      from AS3000 accept ANY
export:      to AS3000 announce AS2000
[…]
```



Case 2: Review provider aut-num object

```
aut-num:     AS3000
import:      from AS2000 accept AS2000
export:      to AS2000 announce ANY
[…]
```

- The simplest policy is strict customer/provider relationship
  - Customer sends it routes to provider
  - Customer accepts everything the provider sends



Case 2: Review route objects

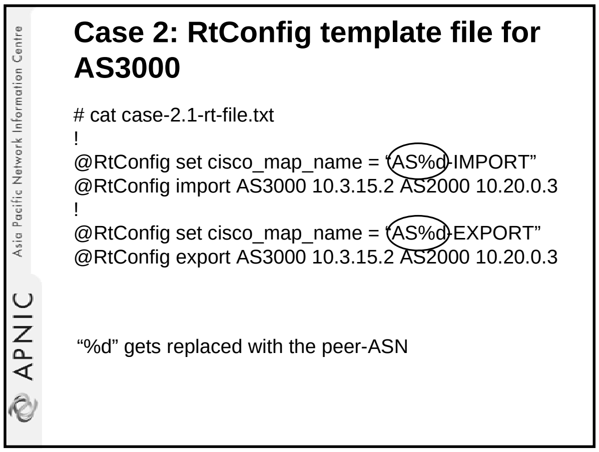
- whois –i origin AS3000

```
route:   10.3.0.0/20
descr:   Route for AS3000
origin:   AS3000
[…]
```

- whois –i origin AS2000

```
route:   10.20.0.0/24
descr:   Route for AS2000
origin:   AS2000
[…]
```

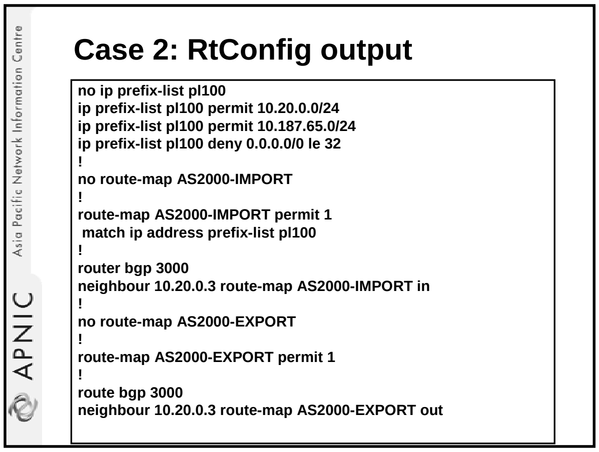
```
route:  10.187.65.0/24
descr:  Route for AS2000
origin:   AS2000
[…]
```



Case 2: RtConfig template file for AS3000

```
# cat case-2.1-rt-file.txt
!
@RtConfig set cisco_map_name = “AS%d-IMPORT”
@RtConfig import AS3000 10.3.15.2 AS2000 10.20.0.3
!
@RtConfig set cisco_map_name = “AS%d-EXPORT”
@RtConfig export AS3000 10.3.15.2 AS2000 10.20.0.3
```

“%d” gets replaced with the peer-ASN



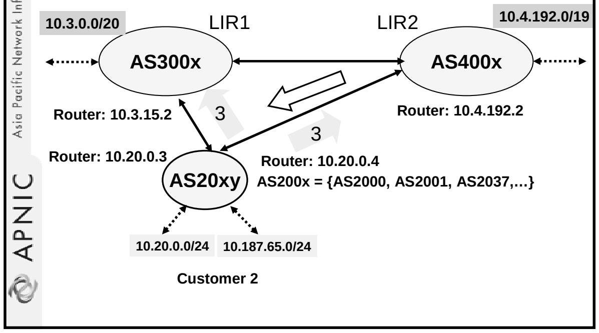
Case 2: RtConfig output

```
no ip prefix-list pl100
ip prefix-list pl100 permit 10.20.0.0/24
ip prefix-list pl100 permit 10.187.65.0/24
ip prefix-list pl100 deny 0.0.0.0/0 le 32
!
no route-map AS2000-IMPORT
!
route-map AS2000-IMPORT permit 1
 match ip address prefix-list pl100
!
router bgp 3000
neighbour 10.20.0.3 route-map AS2000-IMPORT in
!
no route-map AS2000-EXPORT
!
route-map AS2000-EXPORT permit 1
!
route bgp 3000
neighbour 10.20.0.3 route-map AS2000-EXPORT out
```

Case 2: Summary

- aut-num contains policy specifications
  - In import and export attributes
- RtConfig commands per neighbour
  - set cisco_map_name = “AS%d-IMPORT”
  - import <ASN1> <rtr1> <ASN2> <rtr2>
  - set cisco_map_name = “AS%d-EXPORT”
  - Export <ASN1> <rtr1> <ASN2> <rtr2>
- Use RtConfig template file for efficiency

Case 3: Multihomed customer setup



10.3.0.0/20

LIR1

LIR2

10.4.192.0/19

AS300x

AS400x

Router: 10.3.15.2

3

3

Router: 10.4.192.2

Router: 10.20.0.3

AS20xy

Router: 10.20.0.4

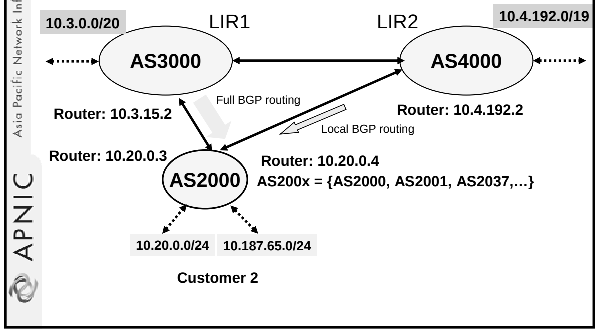
AS200x = {AS2000, AS2001, AS2037,…}

10.20.0.0/24

10.187.65.0/24

Customer 2

Case 3.1: “Partial” multihoming



10.3.0.0/20

LIR1

LIR2

10.4.192.0/19

AS3000

AS4000

Full BGP routing

Router: 10.3.15.2

Router: 10.4.192.2

Local BGP routing

Router: 10.20.0.3

AS2000

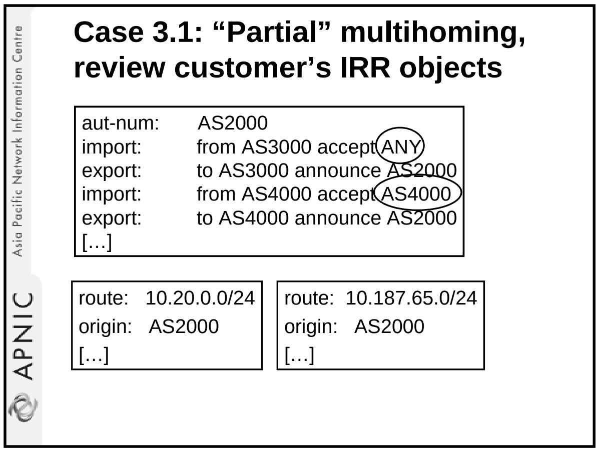
Router: 10.20.0.4

AS200x = {AS2000, AS2001, AS2037,…}

10.20.0.0/24

10.187.65.0/24

Customer 2

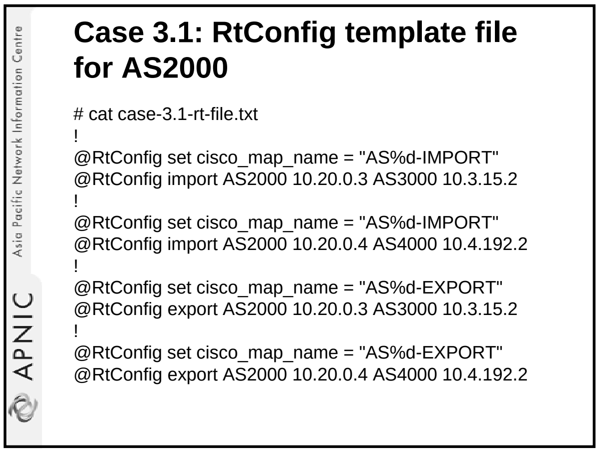


Case 3.1: “Partial” multihoming, review customer’s IRR objects

```
aut-num:    AS2000
import:     from AS3000 accept ANY
export:     to AS3000 announce AS2000
import:     from AS4000 accept AS4000
export:     to AS4000 announce AS2000
[…]
```

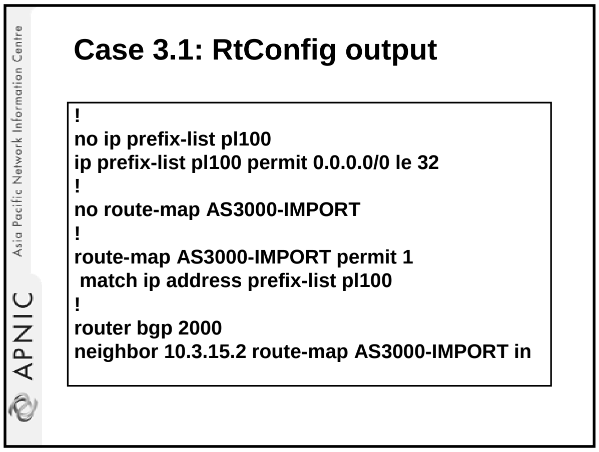
```
route:   10.20.0.0/24
origin:  AS2000
[…]
```

```
route:  10.187.65.0/24
origin:  AS2000
[…]
```



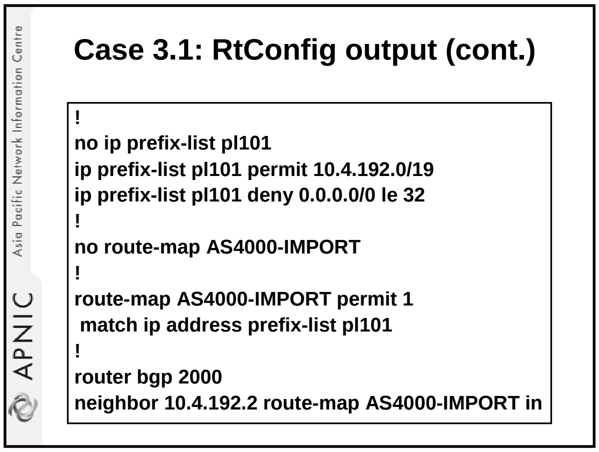
Case 3.1: RtConfig template file for AS2000

```
# cat case-3.1-rt-file.txt
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2000 10.20.0.4 AS4000 10.4.192.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2000 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2000 10.20.0.4 AS4000 10.4.192.2
```



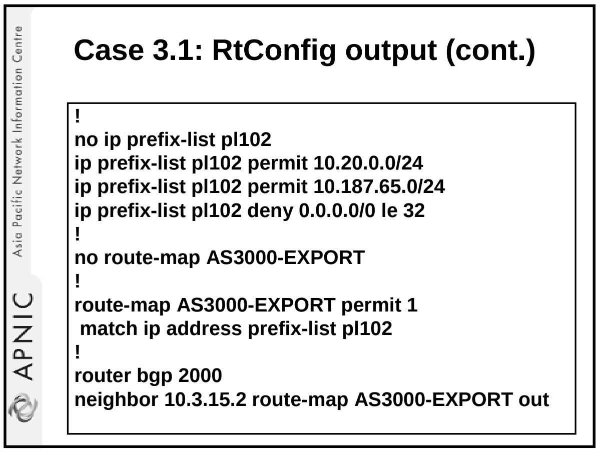
Case 3.1: RtConfig output

```
!
no ip prefix-list pl100
ip prefix-list pl100 permit 0.0.0.0/0 le 32
!
no route-map AS3000-IMPORT
!
route-map AS3000-IMPORT permit 1
 match ip address prefix-list pl100
!
router bgp 2000
neighbor 10.3.15.2 route-map AS3000-IMPORT in
```



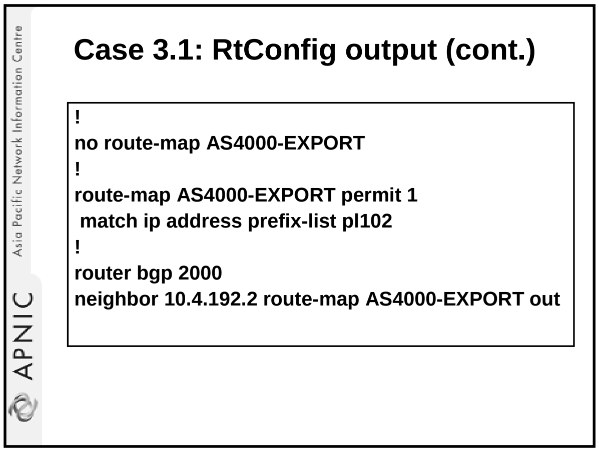
Case 3.1: RtConfig output (cont.)

```
!
no ip prefix-list pl101
ip prefix-list pl101 permit 10.4.192.0/19
ip prefix-list pl101 deny 0.0.0.0/0 le 32
!
no route-map AS4000-IMPORT
!
route-map AS4000-IMPORT permit 1
 match ip address prefix-list pl101
!
router bgp 2000
neighbor 10.4.192.2 route-map AS4000-IMPORT in
```



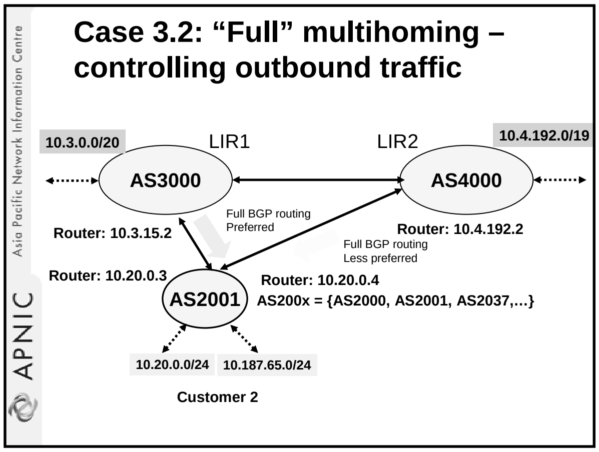
Case 3.1: RtConfig output (cont.)

```
!
no ip prefix-list pl102
ip prefix-list pl102 permit 10.20.0.0/24
ip prefix-list pl102 permit 10.187.65.0/24
ip prefix-list pl102 deny 0.0.0.0/0 le 32
!
no route-map AS3000-EXPORT
!
route-map AS3000-EXPORT permit 1
 match ip address prefix-list pl102
!
router bgp 2000
neighbor 10.3.15.2 route-map AS3000-EXPORT out
```

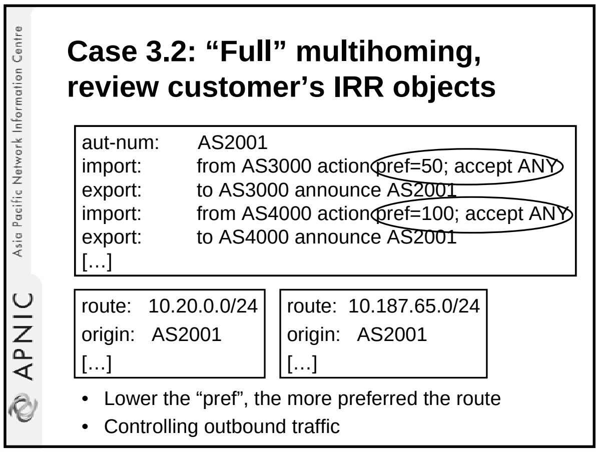


Case 3.1: RtConfig output (cont.)

```
!
no route-map AS4000-EXPORT
!
route-map AS4000-EXPORT permit 1
 match ip address prefix-list pl102
!
router bgp 2000
neighbor 10.4.192.2 route-map AS4000-EXPORT out
```



Case 3.2: "Full" multihoming – controlling outbound traffic



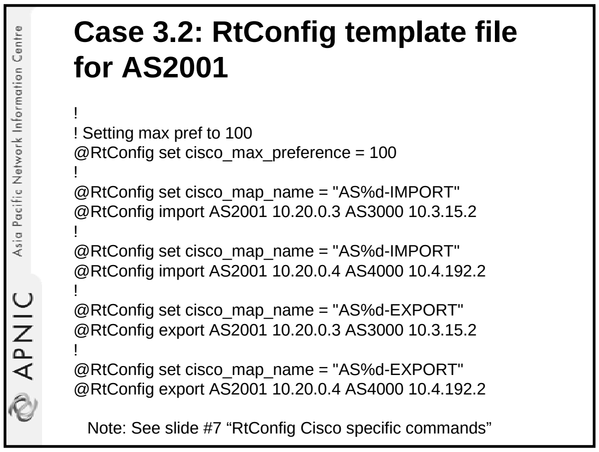
Case 3.2: "Full" multihoming, review customer's IRR objects

```
aut-num:    AS2001
import:     from AS3000 action pref=50; accept ANY
export:     to AS3000 announce AS2001
import:     from AS4000 action pref=100; accept ANY
export:     to AS4000 announce AS2001
[…]
```

```
route:   10.20.0.0/24
origin:   AS2001
[…]
```

```
route:  10.187.65.0/24
origin:   AS2001
[…]
```

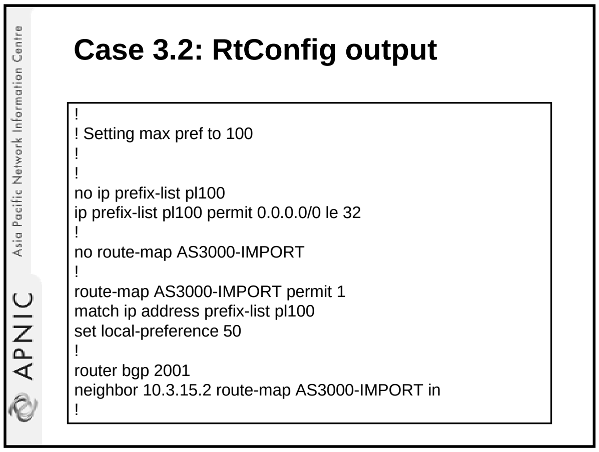
- Lower the "pref", the more preferred the route
- Controlling outbound traffic



Case 3.2: RtConfig template file for AS2001

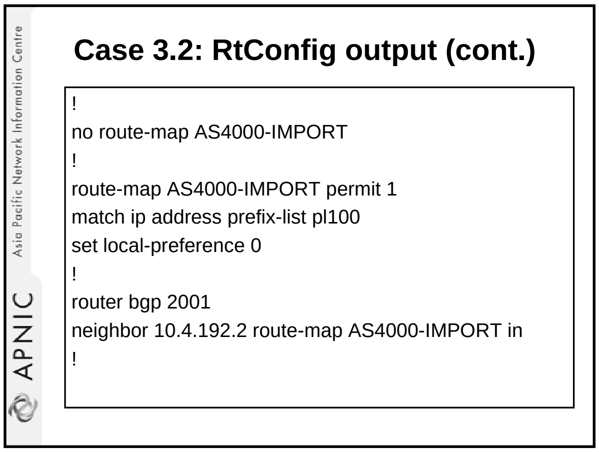
```
!
! Setting max pref to 100
@RtConfig set cisco_max_preference = 100
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2001 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-IMPORT"
@RtConfig import AS2001 10.20.0.4 AS4000 10.4.192.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2001 10.20.0.3 AS3000 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2001 10.20.0.4 AS4000 10.4.192.2
```

Note: See slide #7 "RtConfig Cisco specific commands"



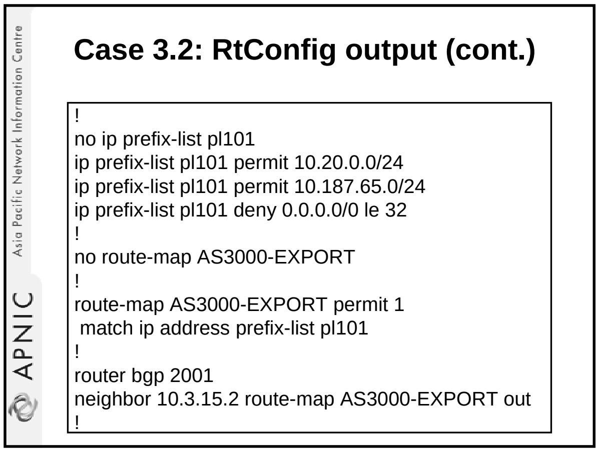
Case 3.2: RtConfig output

```
!
! Setting max pref to 100
!
!
no ip prefix-list pl100
ip prefix-list pl100 permit 0.0.0.0/0 le 32
!
no route-map AS3000-IMPORT
!
route-map AS3000-IMPORT permit 1
match ip address prefix-list pl100
set local-preference 50
!
router bgp 2001
neighbor 10.3.15.2 route-map AS3000-IMPORT in
!
```



Case 3.2: RtConfig output (cont.)

```
!
no route-map AS4000-IMPORT
!
route-map AS4000-IMPORT permit 1
match ip address prefix-list pl100
set local-preference 0
!
router bgp 2001
neighbor 10.4.192.2 route-map AS4000-IMPORT in
!
```



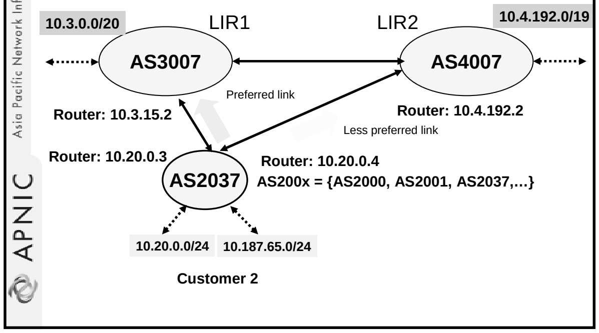
Case 3.2: RtConfig output (cont.)

```
!
no ip prefix-list pl101
ip prefix-list pl101 permit 10.20.0.0/24
ip prefix-list pl101 permit 10.187.65.0/24
ip prefix-list pl101 deny 0.0.0.0/0 le 32
!
no route-map AS3000-EXPORT
!
route-map AS3000-EXPORT permit 1
 match ip address prefix-list pl101
!
router bgp 2001
neighbor 10.3.15.2 route-map AS3000-EXPORT out
!
```

Case 3.2: RtConfig output (cont.)

```
!
no route-map AS4000-EXPORT
!
route-map AS4000-EXPORT permit 1
match ip address prefix-list pl101
!
router bgp 2001
neighbor 10.4.192.2 route-map AS4000-EXPORT out
```

Case 3.3: Controlling inbound traffic



Case 3.3: Controlling inbound traffic

- Prepending your ASN on the less preferred link, in the “export” attribute
  - E.g. AS2037 connected to AS3007 and AS4007

```
aut-num:   AS2037
Remarks:  More preferred link to A3007;
export:     to AS4007 action aspath.prepend
                 (AS2037, AS2037); announce
              AS2037
export:     to AS3007 announce as2037
```

Case 3.3: RtConfig template file for AS2037

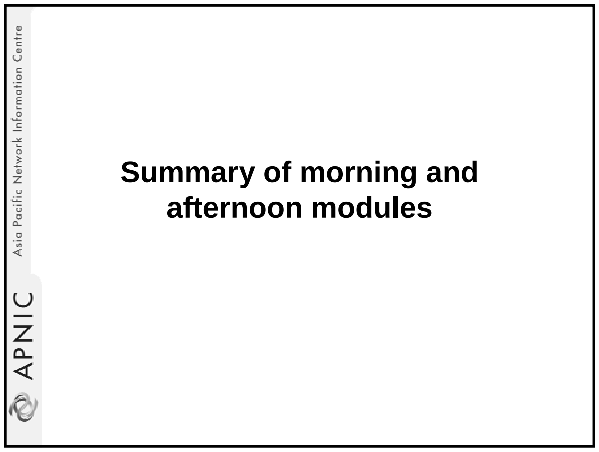
```
!
! The Export Policy - THIS IS the only IMPORTANT bit! –
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2037 10.20.0.3 AS3007 10.3.15.2
!
@RtConfig set cisco_map_name = "AS%d-EXPORT"
@RtConfig export AS2037 10.20.0.4 AS4007 10.4.192.2
```

Case 3.3: RtConfig output

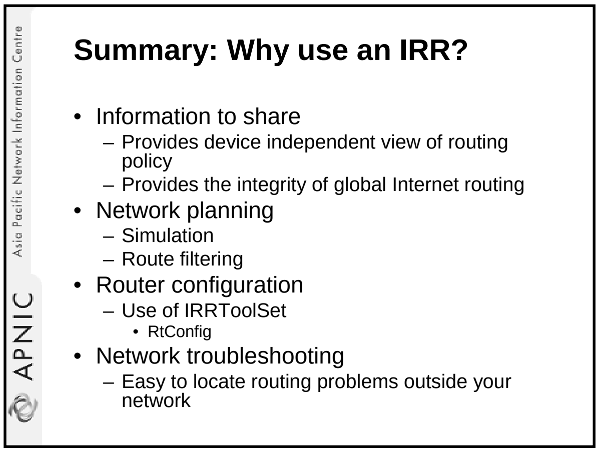
```
!
! The Export Policy - THIS IS the only IMPORTANT bit! –
!
!
no route-map AS3007-EXPORT
!
route-map AS3007-EXPORT permit 1
!
router bgp 2037
neighbor 10.3.15.2 route-map AS3007-EXPORT out
!
!
no route-map AS4007-EXPORT
!
route-map AS4007-EXPORT permit 1
set as-path prepend 2037 2037
!
router bgp 2037
neighbor 10.4.192.2 route-map AS4007-EXPORT out
```

Case 3: Summary

- Policy expressions to achieve/control multihoming
  - Outbound traffic
    - Set the value of local preference
      - **"action pref=NN" in the "import" attribute of aut-num object**
  - Inbound traffic
    - Modify as-path length
      - **"action aspath.prepend (ASN)" in the "export" attribute of aut-num object**
- RtConfig template file has to contain a set of commands for each neighbour

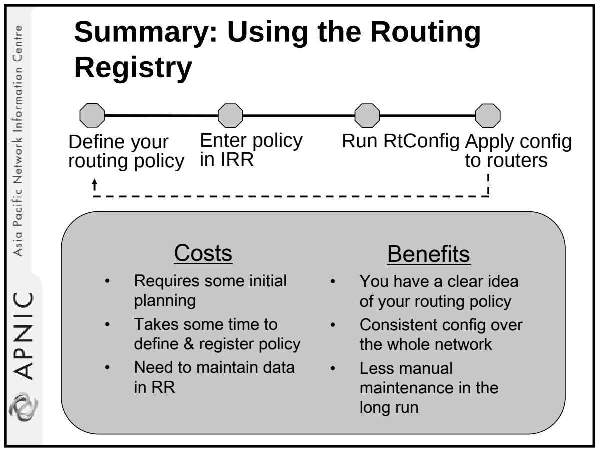


Summary of morning and afternoon modules



Summary: Why use an IRR?

- Information to share
  - Provides device independent view of routing policy
  - Provides the integrity of global Internet routing
- Network planning
  - Simulation
  - Route filtering
- Router configuration
  - Use of IRRToolSet
    - RtConfig
- Network troubleshooting
  - Easy to locate routing problems outside your network



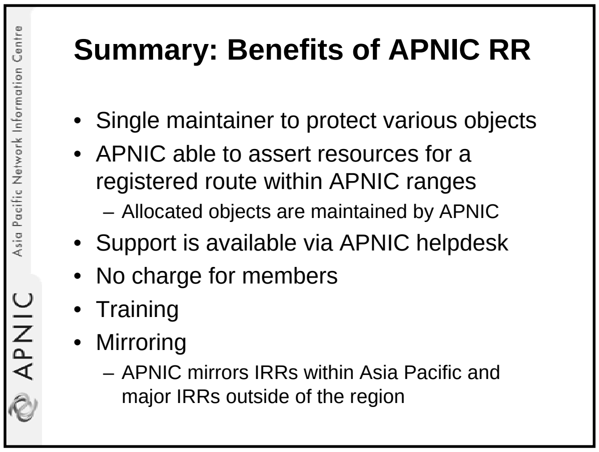
Summary: Using the Routing Registry

Costs

- Requires some initial planning
- Takes some time to define & register policy
- Need to maintain data in RR

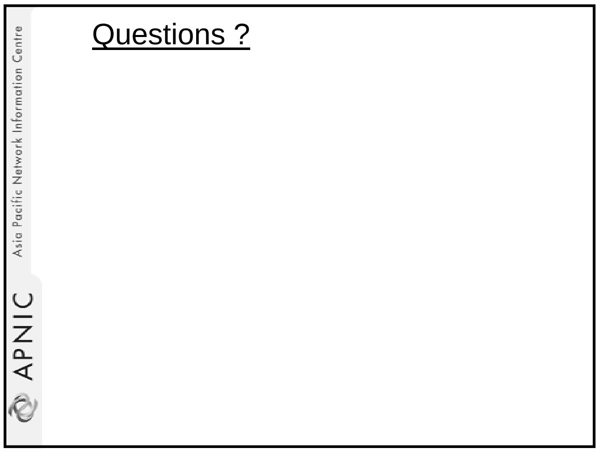
Benefits

- You have a clear idea of your routing policy
- Consistent config over the whole network
- Less manual maintenance in the long run

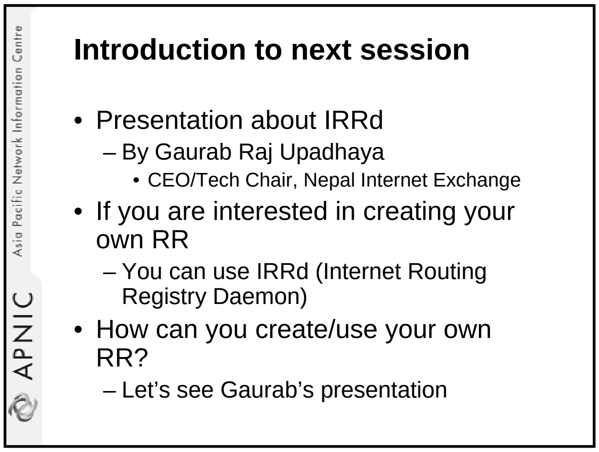


Summary: Benefits of APNIC RR

- Single maintainer to protect various objects
- APNIC able to assert resources for a registered route within APNIC ranges
  - Allocated objects are maintained by APNIC
- Support is available via APNIC helpdesk
- No charge for members
- Training
- Mirroring
  - APNIC mirrors IRRs within Asia Pacific and major IRRs outside of the region



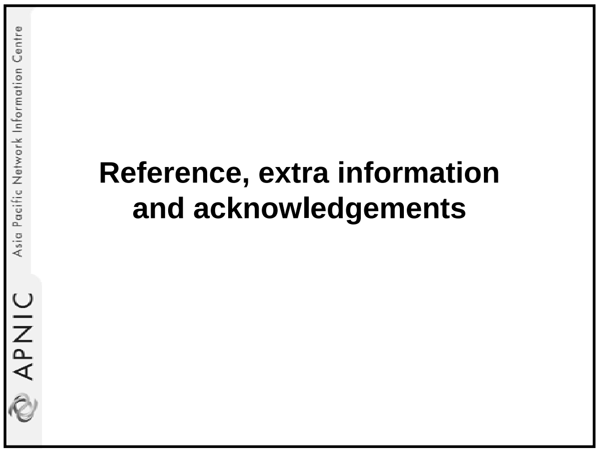
Questions ?



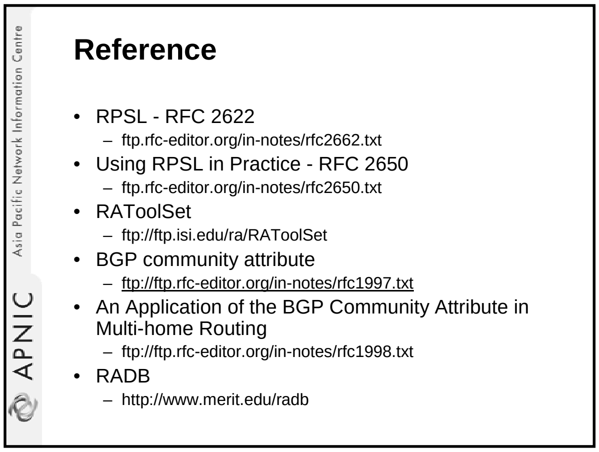
Introduction to next session

- Presentation about IRRd
  - By Gaurab Raj Upadhaya
    - CEO/Tech Chair, Nepal Internet Exchange
- If you are interested in creating your own RR
  - You can use IRRd (Internet Routing Registry Daemon)
- How can you create/use your own RR?
  - Let’s see Gaurab’s presentation

Thank you!

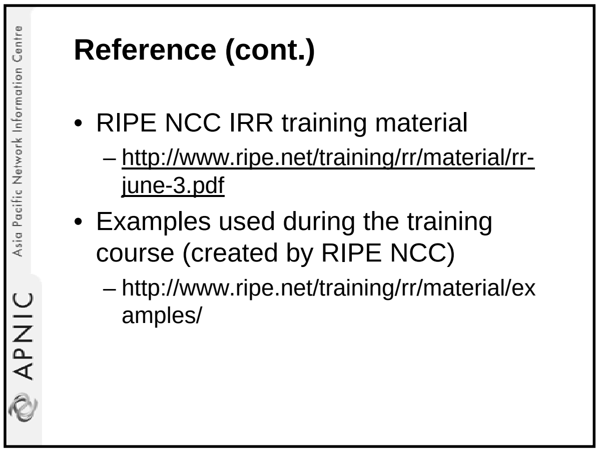


Reference, extra information and acknowledgements



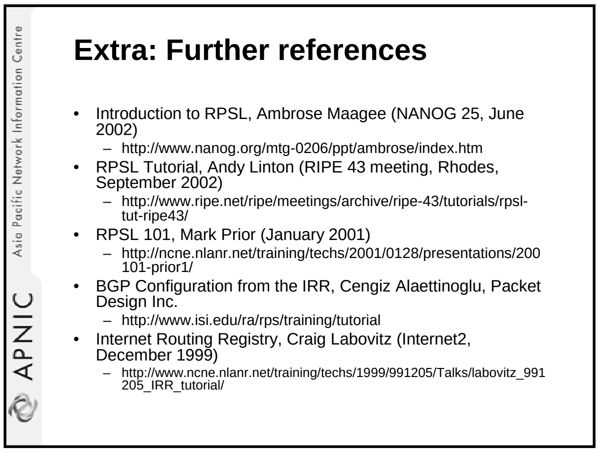
Reference

- RPSL - RFC 2622
  - ftp.rfc-editor.org/in-notes/rfc2662.txt
- Using RPSL in Practice - RFC 2650
  - ftp.rfc-editor.org/in-notes/rfc2650.txt
- RAToolSet
  - ftp://ftp.isi.edu/ra/RAToolSet
- BGP community attribute
  - ftp://ftp.rfc-editor.org/in-notes/rfc1997.txt
- An Application of the BGP Community Attribute in Multi-home Routing
  - ftp://ftp.rfc-editor.org/in-notes/rfc1998.txt
- RADB
  - http://www.merit.edu/radb


Asia Pacific Network Information Centre
APNIC

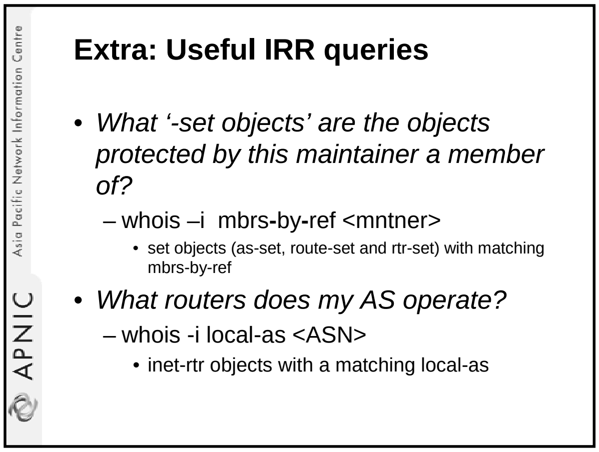
Reference (cont.)

- RIPE NCC IRR training material
  - http://www.ripe.net/training/rr/material/rr-june-3.pdf
- Examples used during the training course (created by RIPE NCC)
  - http://www.ripe.net/training/rr/material/examples/


Asia Pacific Network Information Centre
APNIC

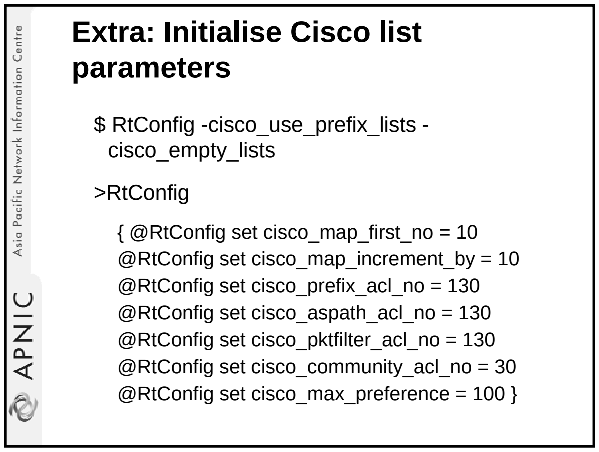
Extra: Further references

- Introduction to RPSL, Ambrose Maagee (NANOG 25, June 2002)
  - http://www.nanog.org/mtg-0206/ppt/ambrose/index.htm
- RPSL Tutorial, Andy Linton (RIPE 43 meeting, Rhodes, September 2002)
  - http://www.ripe.net/ripe/meetings/archive/ripe-43/tutorials/rpsl-tut-ripe43/
- RPSL 101, Mark Prior (January 2001)
  - http://ncne.nlanr.net/training/techs/2001/0128/presentations/200101-prior1/
- BGP Configuration from the IRR, Cengiz Alaettinoglu, Packet Design Inc.
  - http://www.isi.edu/ra/rps/training/tutorial
- Internet Routing Registry, Craig Labovitz (Internet2, December 1999)
  - http://www.ncne.nlanr.net/training/techs/1999/991205/Talks/labovitz_991205_IRR_tutorial/


Asia Pacific Network Information Centre
APNIC

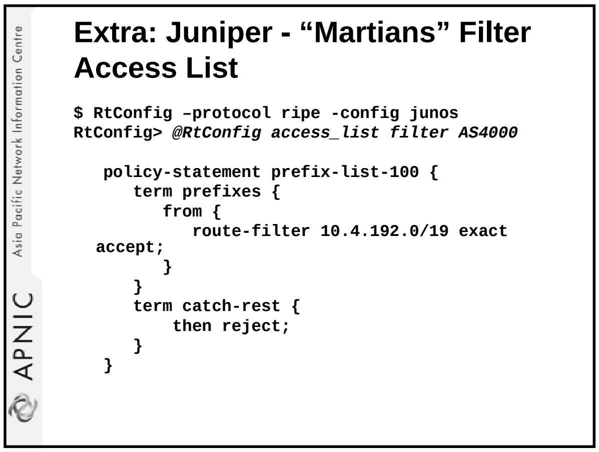
Extra: Useful IRR queries

- *What '-set objects' are the objects protected by this maintainer a member of?*
  - whois –i mbrs-by-ref <mntner>
    - set objects (as-set, route-set and rtr-set) with matching mbrs-by-ref
- *What routers does my AS operate?*
  - whois -i local-as <ASN>
    - inet-rtr objects with a matching local-as



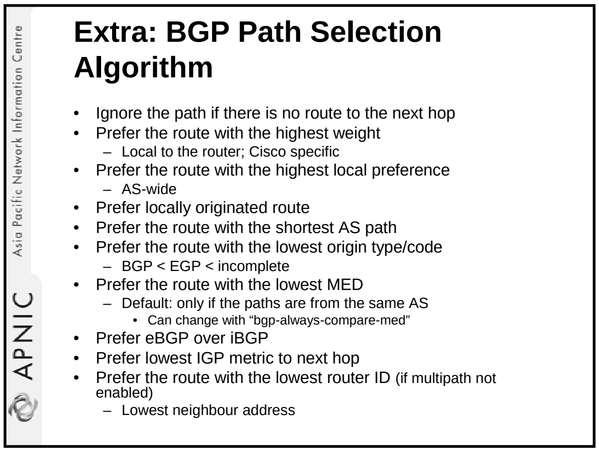
Extra: Initialise Cisco list parameters

```
$ RtConfig -cisco_use_prefix_lists -
  cisco_empty_lists

>RtConfig

   { @RtConfig set cisco_map_first_no = 10
   @RtConfig set cisco_map_increment_by = 10
   @RtConfig set cisco_prefix_acl_no = 130
   @RtConfig set cisco_aspath_acl_no = 130
   @RtConfig set cisco_pktfilter_acl_no = 130
   @RtConfig set cisco_community_acl_no = 30
   @RtConfig set cisco_max_preference = 100 }
```



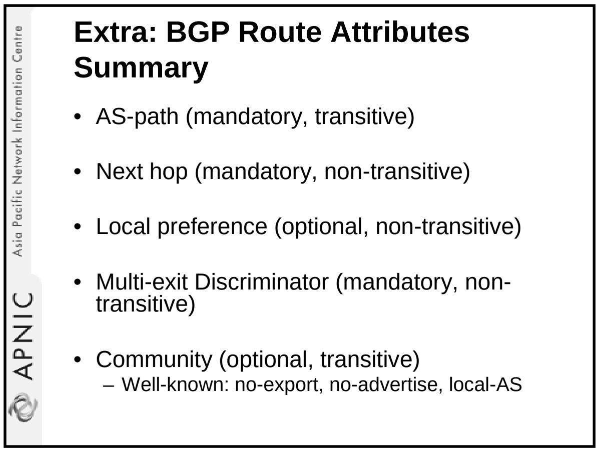
Extra: Juniper - “Martians” Filter Access List

```
$ RtConfig –protocol ripe -config junos
RtConfig> @RtConfig access_list filter AS4000

   policy-statement prefix-list-100 {
      term prefixes {
         from {
            route-filter 10.4.192.0/19 exact
  accept;
         }
      }
      term catch-rest {
          then reject;
      }
   }
```



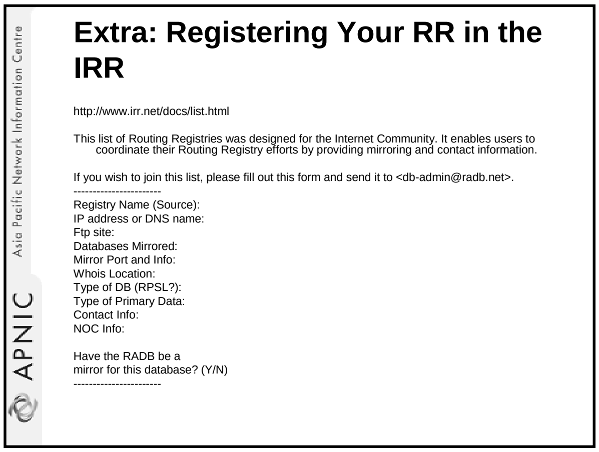
Extra: BGP Path Selection Algorithm

- Ignore the path if there is no route to the next hop
- Prefer the route with the highest weight
  - Local to the router; Cisco specific
- Prefer the route with the highest local preference
  - AS-wide
- Prefer locally originated route
- Prefer the route with the shortest AS path
- Prefer the route with the lowest origin type/code
  - BGP < EGP < incomplete
- Prefer the route with the lowest MED
  - Default: only if the paths are from the same AS
    - Can change with “bgp-always-compare-med”
- Prefer eBGP over iBGP
- Prefer lowest IGP metric to next hop
- Prefer the route with the lowest router ID (if multipath not enabled)
  - Lowest neighbour address



Extra: BGP Route Attributes Summary

- AS-path (mandatory, transitive)
- Next hop (mandatory, non-transitive)
- Local preference (optional, non-transitive)
- Multi-exit Discriminator (mandatory, non-transitive)
- Community (optional, transitive)
  - Well-known: no-export, no-advertise, local-AS

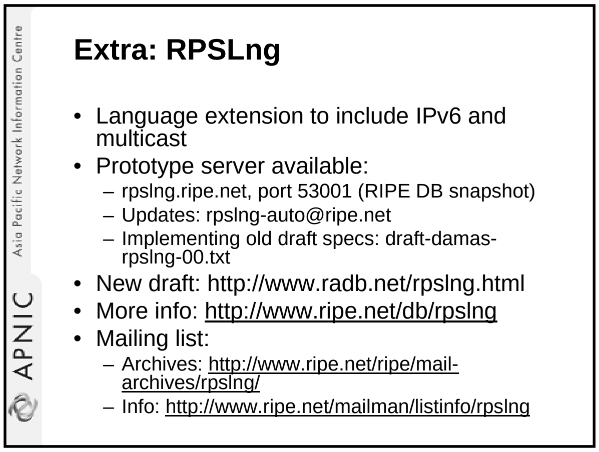


Extra: Registering Your RR in the IRR

http://www.irr.net/docs/list.html

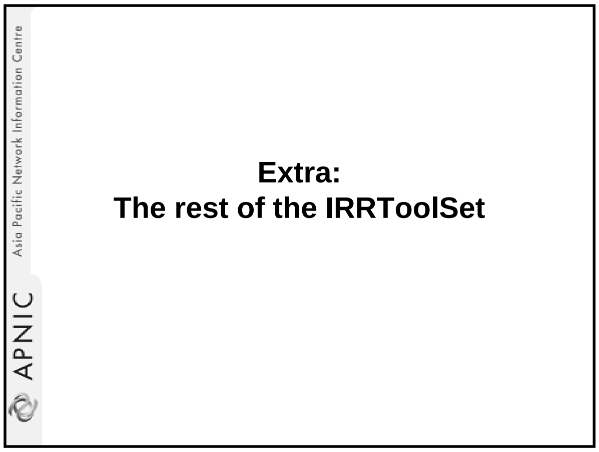
This list of Routing Registries was designed for the Internet Community. It enables users to coordinate their Routing Registry efforts by providing mirroring and contact information.

```
If you wish to join this list, please fill out this form and send it to <db-admin@radb.net>.
-----------------------
Registry Name (Source):
IP address or DNS name:
Ftp site:
Databases Mirrored:
Mirror Port and Info:
Whois Location:
Type of DB (RPSL?):
Type of Primary Data:
Contact Info:
NOC Info:

Have the RADB be a
mirror for this database? (Y/N)
-----------------------
```

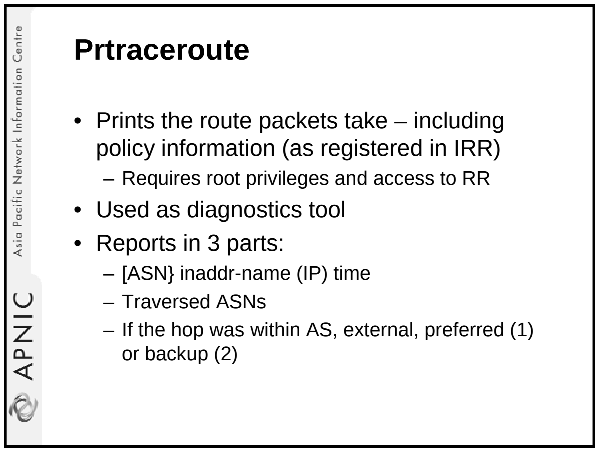


Extra: RPSLng

- Language extension to include IPv6 and multicast
- Prototype server available:
  - rpslng.ripe.net, port 53001 (RIPE DB snapshot)
  - Updates: rpslng-auto@ripe.net
  - Implementing old draft specs: draft-damas-rpslng-00.txt
- New draft: http://www.radb.net/rpslng.html
- More info: http://www.ripe.net/db/rpslng
- Mailing list:
  - Archives: http://www.ripe.net/ripe/mail-archives/rpslng/
  - Info: http://www.ripe.net/mailman/listinfo/rpslng

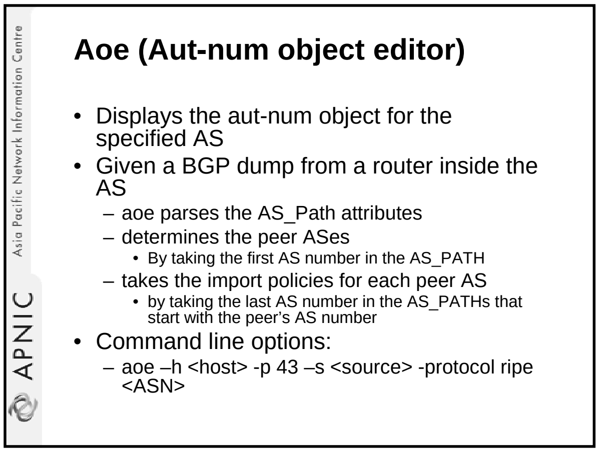


Extra:
The rest of the IRRToolSet



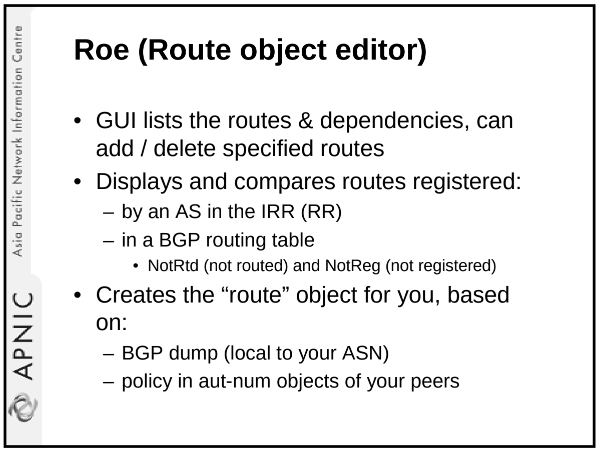
Prtraceroute

- Prints the route packets take – including policy information (as registered in IRR)
  - Requires root privileges and access to RR
- Used as diagnostics tool
- Reports in 3 parts:
  - [ASN} inaddr-name (IP) time
  - Traversed ASNs
  - If the hop was within AS, external, preferred (1) or backup (2)



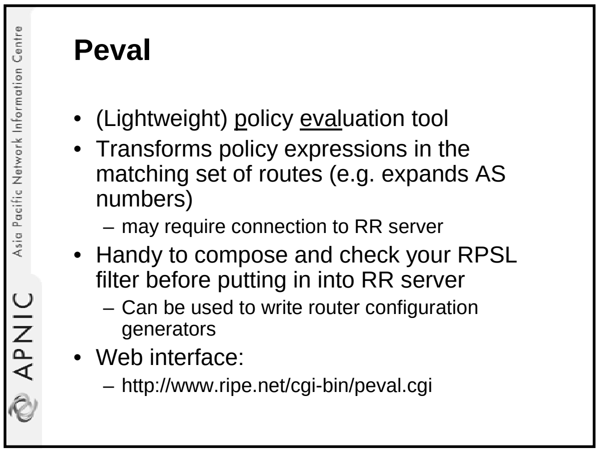
Aoe (Aut-num object editor)

- Displays the aut-num object for the specified AS
- Given a BGP dump from a router inside the AS
  - aoe parses the AS_Path attributes
  - determines the peer ASes
    - By taking the first AS number in the AS_PATH
  - takes the import policies for each peer AS
    - by taking the last AS number in the AS_PATHs that start with the peer's AS number
- Command line options:
  - aoe –h <host> -p 43 –s <source> -protocol ripe <ASN>



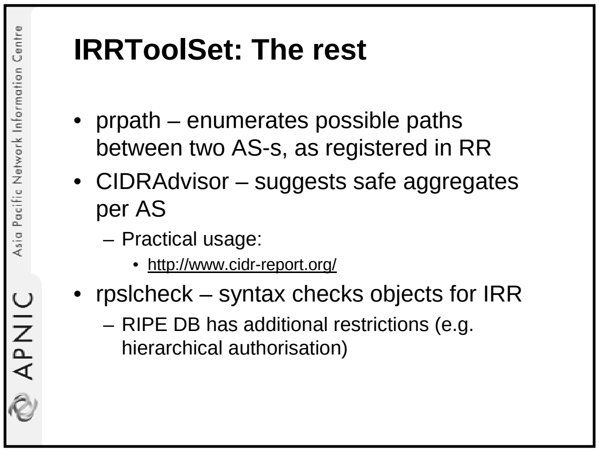
Roe (Route object editor)

- GUI lists the routes & dependencies, can add / delete specified routes
- Displays and compares routes registered:
  - by an AS in the IRR (RR)
  - in a BGP routing table
    - NotRtd (not routed) and NotReg (not registered)
- Creates the “route” object for you, based on:
  - BGP dump (local to your ASN)
  - policy in aut-num objects of your peers



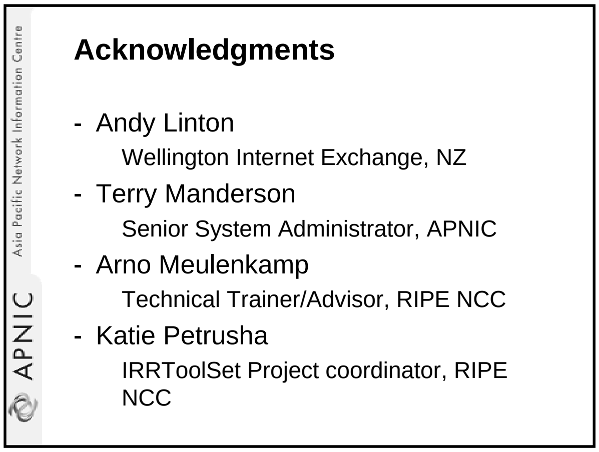
Peval

- (Lightweight) policy evaluation tool
- Transforms policy expressions in the matching set of routes (e.g. expands AS numbers)
  - may require connection to RR server
- Handy to compose and check your RPSL filter before putting in into RR server
  - Can be used to write router configuration generators
- Web interface:
  - http://www.ripe.net/cgi-bin/peval.cgi



IRRToolSet: The rest

- prpath – enumerates possible paths between two AS-s, as registered in RR
- CIDRAdvisor – suggests safe aggregates per AS
  - Practical usage:
    - http://www.cidr-report.org/
- rpslcheck – syntax checks objects for IRR
  - RIPE DB has additional restrictions (e.g. hierarchical authorisation)



Acknowledgments

- Andy Linton
  Wellington Internet Exchange, NZ
- Terry Manderson
  Senior System Administrator, APNIC
- Arno Meulenkamp
  Technical Trainer/Advisor, RIPE NCC
- Katie Petrusha
  IRRToolSet Project coordinator, RIPE NCC

Asia Pacific Network Information Centre
APNIC