

Network Working Group
Internet-Draft
Updates: [4012](#) (if approved)
Intended status: Standards Track
Expires: November 24, 2014

J. Snijders
Independent
May 23, 2014

**The "import-via" and "export-via" attributes in RPSL Policy
Specifications
draft-ietf-grow-rpsl-via-00**

Abstract

This document defines two attributes in the aut-num Class which can be used in RPSL policy specifications to publish desired routing policy regarding non-adjacent networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 24, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Notational Conventions	2
3.	Background	3
4.	Import and Export Via Syntax and Semantics	3
5.	Example Usage	4
6.	Ambiguity Resolution	5
7.	Security Considerations	6
8.	IANA Considerations	6
9.	Acknowledgments	6
10.	References	6
10.1.	Normative References	6
10.2.	Informative References	7
Appendix A.	Grammar Rules	7
Appendix B.	Document Change Log	9
	Author's Address	9

[1.](#) Introduction

The Routing Policy Specification Language [[RFC4012](#)] allows operators to specify routing policies regarding directly adjacent networks through various import and export attributes. These attributes only apply to directly adjacent networks.

This document proposes to extend RPSL according to the following goals and requirements:

- o Provide a way for network (A) to describe what an adjacent network (B) could use as routing policy towards its adjacent networks (C, D, E .. N).
- o The extension should be backward compatible with minimal impact on existing tools and processes, following [Section 10.2 of \[RFC2622\]](#).

The addition of the "import-via" and "export-via" attributes in the aut-num Class will especially help participants of Multi-Lateral Peering services to inform the intermediate autonomous system what routing policy should be applied towards other participants.

[2.](#) Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

3. Background

The via keyword specifically benefits operators who were assigned a 32 bit AS Number and transit providers when participating in Multi-Lateral Peering Agreements facilitated by a Route Server.

Often Route Server operators overload BGP Communities [[RFC1997](#)] to facilitate signaling of desired routing policy between the participants and the Route Server. Because BGP Communities have a length of 32 bit, it is not possible to signal a 32 bit AS Number coupled with an action. In practise this means Route Server participants who use a 32 bit AS Number cannot specifically be included or excluded during path distribution calculations on the Route Server unless a mapping system is applied.

Transit providers often have a routing policy which states that the transit provider does not want to exchange paths with its downstream customers through Route Servers via public Internet Exchanges. The import-via and export-via attributes allow transit providers to participate in Multi-Lateral Peering services while instructing Route Server operators through a simple routing policy specification that paths should not be distributed to downstream customers and reducing the likelihood of Path Hiding on the Route Server.

4. Import and Export Via Syntax and Semantics

The two attributes can be used within the aut-num class.

import-via:

export-via:

The syntax for these attributes is as follows:

Attribute	Value	Type
import-via	[protocol <protocol-1>] [into <protocol-2>] [afi <afi-list>] <mp-peering-1> from <mp-peering-2> [action <action-1>; ... <action-N>;] . . . <mp-peering-3> from <mp-peering-M> [action <action-1>; ... <action-N>;] accept <mp-filter> [;]	optional, multi-valued
export-via	[protocol <protocol-1>] [into <protocol-2>] [afi <afi-list>] <mp-peering-1> to <mp-peering-2> [action <action-1>; ... <action-N>;] . . . <mp-peering-3> to <mp-peering-M> [action <action-1>; ... <action-N>;] announce <mp-filter> [;]	optional, multi-valued

Figure 1

The import-via and export-via attributes are optional, and should be ignored by implementations which do not support interpretation of those attributes. The syntax closely mimics the mp-import and mp-export attributes described in [Section 2.5 of \[RFC4012\]](#), with the exception that before the "from" and "to" keywords an <mp-peering> is defined to indicate the common AS between two non-adjacent networks.

In the above example <peering-1> and <peering-3> are directly adjacent networks, for instance a Multi-Lateral Peering service. <peering-2> is a non-adjacent network.

5. Example Usage

Putting it all together:


```
aut-num: AS15562
import-via: AS6777
    from AS15562
    action pref = 2;
    accept AS-SNIJDERS
export-via: AS6777
    to AS15562 action community.={15562:40};
    announce AS-SNIJDERS
import-via: AS15562:AS-ROUTESERVERS
    from AS15562:AS-CUSTOMERS
    accept NOT ANY
export-via: AS15562:AS-ROUTESERVERS
    to AS15562:AS-CUSTOMERS announce NOT ANY
import-via: AS6777
    from AS4247483647
    accept AS4247483647
export-via: AS6777
    to AS4247483647 action community.={5580:40};
    announce AS-SNIJDERS
```

Figure 2

In the above examples AS15562 and AS15562 are Route Server participants. AS4247483647 is a participant who has been assigned a 32 bit AS Number. AS6777 functions as a Route Server [[I-D.ietf-idr-ix-bgp-route-server](#)] and AS-SET AS15562:AS-ROUTESERVERS contains a list of Route Server AS Numbers. AS-SET AS15562:AS-CUSTOMERS contains a list of downstream transit customers from AS15562.

The intention of the above policy would be to enable the exchange of NLRI's through AS6777 with two parties: AS15562 and AS4247483647, yet prevent the Route Server from distributing NLRI's announced by AS15562 towards customers of said network. Publishing the policy that AS15562 will not accept customer routes through the Route Server can help counteract the "path hiding" phenomenon as described in Section 2.3.1 of [[I-D.ietf-idr-ix-bgp-route-server](#)], as the Route Server now is informed which NLRI's should not be considered in the best path selection process.

6. Ambiguity Resolution

The same peering can be covered by more than one "via" policy attribute or by a combination of multi-protocol policy attributes, or multi-protocol policy attributes (when specifying IPv4 unicast policy) and the previously defined IPv4 unicast policy attributes. In these cases, implementations should follow the specification-order rule as defined in [Section 6.4 of RFC 2622](#) [1]. Operators should

take note that in order to break ambiguity, the action corresponding to the first peering specification is used.

Consider the following example regarding ambiguity resolution.

```
aut-num: AS15562
export:    to AS6777 195.69.144.255 announce AS15562
export-via: AS6777 195.69.144.255 to AS-AMS-IX-RS announce AS-SNIJDERS
```

Figure 3

As both policy specifications cover the same peering with AS6777, specification-order rule is used and Route Server AS6777 195.69.144.255 should only accept AS15562, instead of AS-SNIJDERS, even though the export-via specification is more specific. If the intended policy was to announce all routes which can be resolved through AS-SNIJDERS on this particular peering, the operator should have specified:

```
aut-num: AS15562
export-via: AS6777 195.69.144.255 to AS-AMS-IX-RS announce AS-SNIJDERS
export:    to AS6777 195.69.144.255 announce AS15562
```

Figure 4

7. Security Considerations

There are no security considerations for this specification.

8. IANA Considerations

This document has no IANA actions.

9. Acknowledgments

The author would like to thank Remko van Mook for confirming that "via" is a better keyword than 'through' or 'thru', Nick Hilliard for his unparalleled support, Jeffrey Haas for providing historic perspective, David Croft and Martin Pels for nitpicking.

10. References

10.1. Normative References

[RFC1997] Chandrasekeran, R., Traina, P., and T. Li, "BGP Communities Attribute", [RFC 1997](#), August 1996.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2622] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., and M. Terpstra, "Routing Policy Specification Language (RPSL)", [RFC 2622](#), June 1999.
- [RFC4012] Blunk, L., Damas, J., Parent, F., and A. Robachevsky, "Routing Policy Specification Language next generation (RPSLng)", [RFC 4012](#), March 2005.

[10.2.](#) Informative References

- [I-D.ietf-idr-ix-bgp-route-server]
Jasinska, E., Hilliard, N., Raszuk, R., and N. Bakker,
"Internet Exchange Route Server", [draft-ietf-idr-ix-bgp-route-server-02](#) (work in progress), February 2013.

[Appendix A.](#) Grammar Rules

Note that only 'via' specific grammar rules have been listed.
Currently two routing registry whois daemons have support for the
'via' attributes: IRRd 3.0.7 and RIPE Whois Server 1.71.

```
%type <string> line_autnum
%type <string> attr_autnum
%type <string> attr_import_via
%type <string> attr_export_via

%token T_IV_KEY // *** import-via: ***
%token T_EV_KEY // *** export-via: ***
%token T_AFI T_PROTOCOL T_WORD T_INTRO T_EXCEPT T_REFINE
%token T_ACCEPT T_ANNOUNCE T_TO T_FROM T_PRNGNAME

line_autnum: attr_autnum
    | attr_import_via
    | attr_export_via

attr_import_via: T_IV_KEY attr_import_syntax

attr_import_syntax: opt_protocol_from opt_protocol_into import_simple
    | opt_protocol_from opt_protocol_into afi_import_exp

attr_export_via: T_EV_KEY attr_export_syntax

attr_export_syntax: opt_protocol_from opt_protocol_into export_simple
```



```
| opt_protocol_from opt_protocol_into afi_export_exp

opt_afi_specification:
| T_AFI afi_list

afi_list: afi_token
| afi_list ',' afi_token

afi_token: afi_name

opt_protocol_from:
| T_PROTOCOL T_WORD

opt_protocol_into:
| T_INTRO T_WORD

import_simple: opt_afi_specification import_factor

export_simple: opt_afi_specification export_factor

afi_import_exp: opt_afi_specification import_exp

afi_export_exp: opt_afi_specification export_exp

import_exp: import_term
| import_term T_EXCEPT afi_import_exp
| import_term T_REFINE afi_import_exp

import_factor_list: import_factor ';'
| import_factor_list import_factor ';'

import_term: import_factor ';'
| '{' import_factor_list '}'

export_exp: export_term
| export_term T_EXCEPT afi_export_exp
| export_term T_REFINE afi_export_exp

export_factor_list: export_factor ';'
| export_factor_list export_factor ';'

export_term: export_factor ';'
| '{' export_factor_list '}'

import_factor: import_peering_action_list T_ACCEPT filter

export_factor: export_peering_action_list T_ANNOUNCE filter
```



```
peering: as_expression opt_router_expression opt_router_expression_with_at
| T_PRNGNAME

// Below are two grammar rules that actually differ
// from mp-import: + mp-export:

import_peering_action_list: peering T_FROM peering opt_action
| import_peering_action_list peering T_FROM peering opt_action

export_peering_action_list: peering T_TO peering opt_action
| export_peering_action_list peering T_TO peering opt_action
```

Figure 5

Appendix B. Document Change Log

(RFC Editor - this Appendix can be removed upon publication as RFC)

1. Initial document.
2. Changes to [draft-snijders-rpsl-via-01.txt](#)
 - A. Moved from adding a new RPSL keyword to a new RPSL attribute to improve backwards compatibility.
3. Changes to [draft-snijders-rpsl-via-02.txt](#)
 - A. Added grammar appendix.
 - B. Added section about Ambiguity Resolution.
4. Changes to [draft-snijders-rpsl-via-03.txt](#)
 - A. Updated current IRR implementations.

Author's Address

Job Snijders
Independent
Theodorus Majofskistraat 100
Amsterdam 1065 SZ
NL

Email: job@instituut.net

