

Common Networking Type Definitions v1.0

Copyright © Riverbed Technology Inc. 2024

Created Jan 16, 2024 at 02:01 PM

Type: ipv4address

IPv4 address (x.y.z.k): A 32-bit IPv4 address in dotted-decimal format. It is described in IETF publication RFC791.

JSON

string

Property Name	Type	Description	Notes
<i>ipv4address</i>	<i><string></i>	IPv4 address (x.y.z.k): A 32-bit IPv4 address in dotted-decimal format. It is described in IETF publication RFC791.	Pattern: '^(([0-9] [1-9][0-9] 1[0-9]{2} 2[0-4][0-9] 25[0-5])\.){3}([0-9] [1-9][0-9] 1[0-9]{2} 2[0-4][0-9] 25[0-5])\$';

Type: ipv4prefix

IPv4 prefix/subnet (x.y.z.k/<0-32>): A CIDR notion of IPv4 prefix/subnet representation, which consists of an IPv4 address along with its routing prefix separated by a forward slash character("/").

JSON

string

Property Name	Type	Description	Notes
<i>ipv4prefix</i>	<i><string></i>	IPv4 prefix/subnet (x.y.z.k/<0-32>): A CIDR notion of IPv4 prefix/subnet representation, which consists of an IPv4 address along with its routing prefix separated by a forward slash character("/").	Pattern: '^(([0-9] [1-9][0-9] 1[0-9]{2} 2[0-4][0-9] 25[0-5])\.){3}([0-9] [1-9][0-9] 1[0-9]{2} 2[0-4][0-9] 25[0-5])(\d{1,2} \d{3} 0)\$';

Type: ipv6address

IPv6 address (x:y:z::k): A 128-bit IPv6 address described in IETF publication RFC2460, RFC4291, RFC5952, RFC6052, RFC6145

JSON

string

Property Name	Type	Description	Notes
<i>ipv6address</i>	<i><string></i>	IPv6 address (x:y:z::k): A 128-bit IPv6 address described in IETF publication RFC2460, RFC4291, RFC5952, RFC6052, RFC6145	Pattern: '^s*(([0-9A-Fa-f]{1,4}){7}([0-9A-Fa-f]{1,4}) ((([0-9A-Fa-f]{1,4}){6}(:[0-9A-Fa-f]{1,4}) ((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.){25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)}{3}) ((([0-9A-Fa-f]{1,4}){5}(:[0-9A-Fa-f]{1,4}){1,2}) ((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.){25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)}{3}) ((([0-9A-Fa-f]{1,4}){4}(:[0-9A-Fa-f]{1,4}){1,3}) ((:[0-9A-Fa-f]{1,4})?:((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.){25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)}{3}) ((([0-9A-Fa-f]{1,4}){3}(:[0-9A-Fa-f]{1,4}){1,4}) ((:[0-9A-Fa-f]{1,4}){0,2}:((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.){25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)}{3}) ((([0-9A-Fa-f]{1,4}){2}(:[0-9A-Fa-f]{1,4}){1,5}) ((:[0-9A-Fa-f]{1,4}){0,3}:((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.){25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)}{3}) ((([0-9A-Fa-f]{1,4}){1,6}) ((:[0-9A-Fa-f]{1,4}){0,4}:((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.){25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)}{3}) ((([0-9A-Fa-f]{1,4}){1,7}) ((:[0-9A-Fa-f]{1,4}){0,5}:((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.){25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)}{3}) ((%.)?s*\$)';

Type: ipv6prefix

IPv6 prefix/subnet (x:y:z::k/<0-128>): A CIDR notion of IPv6 prefix/subnet representation, which consists of an IPv6 address along with its routing prefix separated by a forward slash character("/").

JSON

string

Property Name	Type	Description	Notes
<i>ipv6prefix</i>	<string>	IPv6 prefix/subnet (x:y:z:k/<0-128>): A CIDR notion of IPv6 prefix/subnet representation, which consists of an IPv6 address along with its routing prefix separated by a forward slash character("/").	Pattern: '^\\s*((([0-9A-Fa-f]{1,4}:){7}([0-9A-Fa-f]{1,4}:) ((([0-9A-Fa-f]{1,4}:){6}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d 1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){5}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){4}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){3}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){2}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){1,6}) ((([0-9A-Fa-f]{1,4}:){0,4}:((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){1,7}) ((([0-9A-Fa-f]{1,4}:){0,5}:((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3})))) (%+)?\\s*(\\(\\d \\d 1[0-1] 12[0-8]))\$';

Type: ipv4ipv6address

IPv4 or IPv6 address (x.y.z.k|x:y:z::k)

JSON

```
{  
  "ipv4address": ipv4address,  
  "ipv6address": ipv6address  
}
```

Property Name	Type	Description	Notes
<i>ipv4ipv6address</i>	<object>	IPv4 or IPv6 address (x.y.z.k x:y:z::k)	
<i>ipv4ipv6address.ipv4address</i>	<ipv4address>	IPv4 address (x.y.z.k): A 32-bit IPv4 address in dotted-decimal format. It is described in IETF publication RFC791.	Pattern: '^((([0-9] 1-9 10-99 100-999 2[0-4][0-9] 25[0-5])\\.){3}([0-9] 1-9 10-99 100-999 2[0-4][0-9] 25[0-5]))\$';
<i>ipv4ipv6address.ipv6address</i>	<ipv6address>	IPv6 address (x:y:z:k): A 128-bit IPv6 address described in IETF publication RFC2460, RFC4291, RFC5952, RFC6052, RFC6145	Pattern: '^\\s*((([0-9A-Fa-f]{1,4}:){7}([0-9A-Fa-f]{1,4}:) ((([0-9A-Fa-f]{1,4}:){6}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){5}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){4}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){3}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){2}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){1,6}) ((([0-9A-Fa-f]{1,4}:){0,4}:((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3}) ((([0-9A-Fa-f]{1,4}:){1,7}) ((([0-9A-Fa-f]{1,4}:){0,5}:((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d)\\.((25[0-5] 2[0-4] 1\\d 1\\d [1-9]?\\d))){3})))) (%+)?\\s*\$';
<i>ipv4ipv6address.oneOf[0]</i>	<object>		Required properties: [ipv4address];
<i>ipv4ipv6address.oneOf[0].<prop></i>	<any>		Optional;
<i>ipv4ipv6address.oneOf[1]</i>	<object>		Required properties: [ipv6address];
<i>ipv4ipv6address.oneOf[1].<prop></i>	<any>		Optional;

Type: ipv4ipv6prefix

IPv4 or IPv6 prefix/subnet (x.y.z.k/<0-32>|x:y:z::k/<0-128>)

JSON

```
{
  "ipv4prefix": ipv4prefix,
  "ipv6prefix": ipv6prefix
}
```

Property Name	Type	Description	Notes
<i>ipv4ipv6prefix</i>	<object>	IPv4 or IPv6 prefix/subnet (x.y.z.k/<0-32> x:y:z::k/<0-128>)	
<i>ipv4ipv6prefix.ipv4prefix</i>	<ipv4prefix>	IPv4 prefix/subnet (x.y.z.k/<0-32>): A CIDR notion of IPv4 prefix/subnet representation, which consists of an IPv4 address along with its routing prefix separated by a forward slash character("/").	Pattern: '^((([0-9] [1-9][0-9])1[0-9]{2} 2[0-4][0-9] 25[0-5])\.)\{3\}([0-9] [1-9][0-9])1[0-9]{2} 2[0-4][0-9] 25[0-5])(V(\d [1-2]\d 3[0-2]))\$';
<i>ipv4ipv6prefix.ipv6prefix</i>	<ipv6prefix>	IPv6 prefix/subnet (x:y:z::k/<0-128>): A CIDR notion of IPv6 prefix/subnet representation, which consists of an IPv6 address along with its routing prefix separated by a forward slash character("/").	Pattern: '^s*((([0-9A-Fa-f]{1,4}:){7}([0-9A-Fa-f]{1,4}:) ((([0-9A-Fa-f]{1,4}:){6}(:[0-9A-Fa-f]{1,4} ((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d))\{3\}:) ([0-9A-Fa-f]{1,4}:){5}((:[0-9A-Fa-f]{1,4}){1,2}) ((25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d))\{3\}:) ((([0-9A-Fa-f]{1,4}:){4}((:[0-9A-Fa-f]{1,4}){1,3}) ((:[0-9A-Fa-f]{1,4})?:)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d))\{3\}:) ((([0-9A-Fa-f]{1,4}:){3}((:[0-9A-Fa-f]{1,4}){1,4}) ((:[0-9A-Fa-f]{1,4}){0,2}:)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d))\{3\}:) ([0-9A-Fa-f]{1,4}:){2}((:[0-9A-Fa-f]{1,4}){1,5}) ((:[0-9A-Fa-f]{1,4}){0,3}:)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d))\{3\}:) ((([0-9A-Fa-f]{1,4}:){1}((:[0-9A-Fa-f]{1,4}){1,6}) ((:[0-9A-Fa-f]{1,4}){0,4}:)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d))\{3\}:) ((:[0-9A-Fa-f]{1,4}){0,5}:)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d)\.)(25[0-5] 2[0-4]\d 1\d\d [1-9]?\d))\{3\}:) ((%.)?s*(V(\d [1-2]\d 3[0-8]))\$';
<i>ipv4ipv6prefix.oneOf[0]</i>	<object>		Required properties: [ipv4prefix];
<i>ipv4ipv6prefix.oneOf[0].<prop></i>	<any>		Optional;
<i>ipv4ipv6prefix.oneOf[1]</i>	<object>		Required properties: [ipv6prefix];
<i>ipv4ipv6prefix.oneOf[1].<prop></i>	<any>		Optional;

Type: port

Valid port number

JSON

```
integer
```

Property Name	Type	Description	Notes
<i>port</i>	<integer>	Valid port number	Range: 0 to 65535;

Type: portlabel

A textual name assigned to a range of ports, for example the predefined portlabel "RBT-Proto" refers to ports in range of 7744, 7800-7801, 7810, 7820, 7850, 7860, 7870.

JSON

```
string
```

Property Name	Type	Description	Notes
<i>portlabel</i>	<string>	A textual name assigned to a range of ports, for example the predefined portlabel "RBT-Proto" refers to ports in range of 7744, 7800-7801, 7810, 7820, 7850, 7860, 7870.	

Type: port_portlabel

Port number or portlabel

JSON

```
{
  "port": port,
  "portlabel": portlabel
}
```

Property Name	Type	Description	Notes
<i>port_portlabel</i>	<object>	Port number or portlabel	
<i>port_portlabel</i> .port	<port>	Valid port number	
<i>port_portlabel</i> .portlabel	<portlabel>	A textual name assigned to a range of ports, for example the predefined portlabel "RBT-Proto" refers to ports in range of 7744, 7800-7801, 7810, 7820, 7850, 7860, 7870.	
<i>port_portlabel</i> .oneOf[0]	<object>		Required properties: [port];
<i>port_portlabel</i> .oneOf[0].<prop>	<any>		Optional;
<i>port_portlabel</i> .oneOf[1]	<object>		Required properties: [portlabel];
<i>port_portlabel</i> .oneOf[1].<prop>	<any>		Optional;

Type: vlan_tag

Vlan tag (all, untagged, tagged - [1-4094])

JSON

```
{
  "state": string,
  "tag": number
}
```

Property Name	Type	Description	Notes
<i>vlan_tag</i>	<object>	Vlan tag (all, untagged, tagged - [1-4094])	
<i>vlan_tag</i> .state	<string>	The state of a vlan tag.	Optional; Values: all, untagged, tagged;
<i>vlan_tag</i> .tag	<number>	The "tag" is valid if "state" == "tagged".	Optional; Range: 1 to 4094;
<i>vlan_tag</i> .oneOf[0]	<object>		Required properties: [state];
<i>vlan_tag</i> .oneOf[0].state	<string>		Values: all, untagged;
<i>vlan_tag</i> .oneOf[1]	<object>		Required properties: [state, tag];
<i>vlan_tag</i> .oneOf[1].state	<string>		Values: tagged;
<i>vlan_tag</i> .oneOf[1].tag	<number>		