

**NAME**

radvd.conf – configuration file of the router advertisement daemon **radvd**

**DESCRIPTION**

This file describes the information which is included in the router advertisement (RA) of a specific interface.

The file contains one or more interface definitions of the form:

```
interface name {
    list of interface specific options
    list of prefix definitions
    list of clients (IPv6 addresses) to advertise to
    list of route definitions
    list of RDNSS definitions
};
```

All the possible interface specific options are detailed below. Each option has to be terminated by a semi-colon.

Prefix definitions are of the form:

```
prefix prefix/length {
    list of prefix specific options
};
```

Prefix can be network prefix or the address of the interface. The address of interface should be used when using Mobile IPv6 extensions.

Special prefix "::/64" is also supported on systems that implement getifaddrs() (on other systems, configuration activation fails and radvd exits). When configured, radvd picks one non-link-local prefix assigned to the interface and starts advertising it. This may be applicable in non-6to4 scenarios where the upstream prefix might change. This option is incompatible with Base6to4Interface option. AdvRouterAddr option is always enabled when this configuration is used.

All the possible prefix specific options are described below. Each option has to be terminated by a semi-colon.

Decimal values are allowed only for MinDelayBetweenRAs, MaxRtrAdvInterval and MinRtrAdvInterval. Decimal values should be used only when using Mobile IPv6 extensions.

Route definitions are of the form:

```
route prefix/length {
    list of route specific options
};
```

The prefix of a route definition should be network prefix; it can be used to advertise more specific routes to the hosts.

RDNSS (Recursive DNS server) definitions are of the form:

```
RDNSS ip [ip] [ip] {
    list of rdns specific options
};
```

By default radvd will send route advertisements so that every node on the link can use them. The list of clients (IPv6 address) to advertise to, and accept route solicitations from can be configured. If done, radvd does not send messages to the multicast addresses but to the configured unicast addresses only. Solicitations from other addresses are refused. This is similar to UnicastOnly but includes periodic messages and incoming client access configuration. See examples section for a use case of this.

The definitions are of the form:

```
clients {  
    list of IPv6 addresses  
};
```

## INTERFACE SPECIFIC OPTIONS

### **IgnoreIfMissing on|off**

A flag indicating whether or not the interface is ignored if it does not exist at start-up. By default, radvd exits.

This is useful for dynamic interfaces which are not active when radvd starts or which are dynamically disabled and re-enabled during the time radvd runs.

Current versions of radvd automatically try to re-enable interfaces.

Enabling IgnoreIfMissing also quenches certain warnings in log messages relating to missing interfaces.

Default: off

### **AdvSendAdvert on|off**

A flag indicating whether or not the router sends periodic router advertisements and responds to router solicitations.

This option no longer has to be specified first, but it needs to be **on** to enable advertisement on this interface.

Default: off

### **UnicastOnly on|off**

Indicates that the interface link type only supports unicast. This will prevent unsolicited advertisements from being sent, and will cause solicited advertisements to be unicast to the soliciting node. This option is necessary for non-broadcast, multiple-access links, such as ISATAP.

Default: off

### **MaxRtrAdvInterval seconds**

The maximum time allowed between sending unsolicited multicast router advertisements from the interface, in seconds.

Must be no less than 4 seconds and no greater than 1800 seconds.

Minimum when using Mobile IPv6 extensions: 0.07.

For values less than 0.2 seconds, 0.02 seconds is added to account for scheduling granularities as specified in RFC3775.

Default: 600 seconds

**MinRtrAdvInterval** seconds

The minimum time allowed between sending unsolicited multicast router advertisements from the interface, in seconds.

Must be no less than 3 seconds and no greater than  $0.75 * \text{MaxRtrAdvInterval}$ .

Minimum when using Mobile IPv6 extensions: 0.03.

Default:  $0.33 * \text{MaxRtrAdvInterval}$

**MinDelayBetweenRAs** seconds

The minimum time allowed between sending multicast router advertisements from the interface, in seconds.

This applies to solicited multicast RAs. This is defined as the protocol constant MIN\_DELAY\_BETWEEN\_RAS in RFC4861. MIPv6 redefines this parameter to have a minimum of 0.03 seconds.

Minimum when using Mobile IPv6 extensions: 0.03.

Default: 3

**AdvManagedFlag** on|off

When set, hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. The use of this flag is described in RFC 4862.

Default: off

**AdvOtherConfigFlag** on|off

When set, hosts use the administered (stateful) protocol for autoconfiguration of other (non-address) information. The use of this flag is described in RFC 4862.

Default: off

**AdvLinkMTU** integer

The MTU option is used in router advertisement messages to insure that all nodes on a link use the same MTU value in those cases where the link MTU is not well known.

If specified, i.e. not 0, must not be smaller than 1280 and not greater than the maximum MTU

allowed for this link (e.g. ethernet has a maximum MTU of 1500. See RFC 4864).

Default: 0

**AdvReachableTime** milliseconds

The time, in milliseconds, that a node assumes a neighbor is reachable after having received a reachability confirmation. Used by the Neighbor Unreachability Detection algorithm (see Section 7.3 of RFC 4861). A value of zero means unspecified (by this router).

Must be no greater than 3,600,000 milliseconds (1 hour).

Default: 0

**AdvRetransTimer** milliseconds

The time, in milliseconds, between retransmitted Neighbor Solicitation messages. Used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 4861). A value of zero means unspecified (by this router).

Default: 0

**AdvCurHopLimit** integer

The default value that should be placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The value should be set to the current diameter of the Internet. The value zero means unspecified (by this router).

Default: 64

**AdvDefaultLifetime** seconds

The lifetime associated with the default router in units of seconds. The maximum value corresponds to 18.2 hours. A lifetime of 0 indicates that the router is not a default router and should not appear on the default router list. The router lifetime applies only to the router's usefulness as a default router; it does not apply to information contained in other message fields or options. Options that need time limits for their information include their own lifetime fields.

Must be either zero or between MaxRtrAdvInterval and 9000 seconds.

Default: 3 \* MaxRtrAdvInterval (Minimum 1 second).

**AdvDefaultPreference** low|medium|high

The preference associated with the default router, as either "low", "medium", or "high".

Default: medium

**AdvSourceLLAddress** on|off

When set, the link-layer address of the outgoing interface is included in the RA.

Default: on

**AdvHomeAgentFlag on|off**

When set, indicates that sending router is able to serve as Mobile IPv6 Home Agent. When set, minimum limits specified by Mobile IPv6 are used for MinRtrAdvInterval and MaxRtrAdvInterval.

Default: off

**AdvHomeAgentInfo on|off**

When set, Home Agent Information Option (specified by Mobile IPv6) is included in Router Advertisements. AdvHomeAgentFlag must also be set when using this option.

Default: off

**HomeAgentLifetime seconds**

The length of time in seconds (relative to the time the packet is sent) that the router is offering Mobile IPv6 Home Agent services. A value 0 must not be used. The maximum lifetime is 65520 seconds (18.2 hours). This option is ignored, if AdvHomeAgentInfo is not set.

If both HomeAgentLifetime and HomeAgentPreference are set to their default values, Home Agent Information Option will not be sent.

Default: AdvDefaultLifetime

**HomeAgentPreference integer**

The preference for the Home Agent sending this Router Advertisement. Values greater than 0 indicate more preferable Home Agent, values less than 0 indicate less preferable Home Agent. This option is ignored, if AdvHomeAgentInfo is not set.

If both HomeAgentLifetime and HomeAgentPreference are set to their default values, Home Agent Information Option will not be sent.

Default: 0

**AdvMobRtrSupportFlag on|off**

When set, the Home Agent signals it supports Mobile Router registrations (specified by NEMO Basic). AdvHomeAgentInfo must also be set when using this option.

Default: off

**AdvIntervalOpt on|off**

When set, Advertisement Interval Option (specified by Mobile IPv6) is included in Router Advertisements. When set, minimum limits specified by Mobile IPv6 are used for MinRtrAdvInterval and MaxRtrAdvInterval.

The advertisement interval is based on the configured `MaxRtrAdvInterval` parameter except where this is less than 200ms. In this case, the advertised interval is ( `MaxRtrAdvInterval` + 20ms ).

Default: off

## **PREFIX SPECIFIC OPTIONS**

### **AdvOnLink on|off**

When set, indicates that this prefix can be used for on-link determination. When not set the advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some of the addresses belonging to the prefix being on-link and others being off-link.

Default: on

### **AdvAutonomous on|off**

When set, indicates that this prefix can be used for autonomous address configuration as specified in RFC 4862.

Default: on

### **AdvRouterAddr on|off**

When set, indicates that the address of interface is sent instead of network prefix, as is required by Mobile IPv6. When set, minimum limits specified by Mobile IPv6 are used for `MinRtrAdvInterval` and `MaxRtrAdvInterval`.

Default: off

### **AdvValidLifetime seconds|infinity**

The length of time in seconds (relative to the time the packet is sent) that the prefix is valid for the purpose of on-link determination. The symbolic value **infinity** represents infinity (i.e. a value of all one bits (0xffffffff)). The valid lifetime is also used by RFC 4862.

Note that clients will ignore `AdvValidLifetime` of an existing prefix if the lifetime is below two hours, as required in RFC 4862 Section 5.5.3 point e).

Note: RFC4861's suggested default value is significantly longer: 30 days.

Default: 86400 seconds (1 day)

### **AdvPreferredLifetime seconds|infinity**

The length of time in seconds (relative to the time the packet is sent) that addresses generated from the prefix via stateless address autoconfiguration remain preferred. The symbolic value **infinity** represents infinity (i.e. a value of all one bits (0xffffffff)). See RFC 4862.

Note: RFC4861's suggested default value is significantly longer: 7 days.

Default: 14400 seconds (4 hours)

**Base6to4Interface name**

If this option is specified, this prefix will be combined with the IPv4 address of interface **name** to produce a valid 6to4 prefix. The first 16 bits of this prefix will be replaced by **2002** and the next 32 bits of this prefix will be replaced by the IPv4 address assigned to interface **name** at configuration time. The remaining 80 bits of the prefix (including the SLA ID) will be advertised as specified in the configuration file. See the next section for an example.

If interface **name** is not available at configuration time, a warning will be written to the log and this prefix will be disabled until radvd is reconfigured.

This option enables systems with dynamic IPv4 addresses to update their advertised 6to4 prefixes simply by restarting radvd or sending a SIGHUP signal to cause radvd to reconfigure itself.

Note that 6to4 prefixes derived from dynamically-assigned IPv4 addresses should be advertised with a significantly shorter lifetime (see the **AdvValidLifetime** and **AdvPreferredLifetime** options).

For more information on 6to4, see RFC 3056.

Default: 6to4 is not used

**ROUTE SPECIFIC OPTIONS****AdvRouteLifetime** seconds|infinity

The lifetime associated with the route in units of seconds. The symbolic value **infinity** represents infinity (i.e. a value of all one bits (0xffffffff)).

Default: 3 \* MaxRtrAdvInterval

**AdvRoutePreference** low|medium|high

The preference associated with the default router, as either "low", "medium", or "high".

Default: medium

**RDNSS SPECIFIC OPTIONS****AdvRDNSSPreference** integer;

The preference of the DNS server, compared to other DNS servers advertised and used. 0 to 7 means less important than manually configured nameservers in resolv.conf, while 12 to 15 means more important.

NOTE: This feature was removed from the final RFC but can still be used for experimental purposes.

Default: 8

**AdvRDNSSOpen** on|off;

"Service Open" flag. When set, indicates that RDNSS continues to be available to hosts even if they moved to a different subnet.

NOTE: This feature was removed from the final RFC but can still be used for experimental purposes.

Default: off

**AdvRDNSSLifetime** seconds|infinity;

The maximum duration how long the RDNSS entries are used for name resolution. A value of 0 means the nameserver should no longer be used. The maximum duration how long the RDNSS entries are used for name resolution. A value of 0 means the nameserver should no longer be used. The value, if not 0, must be at least MaxRtrAdvInterval. To ensure stale RDNSS info gets removed in a timely fashion, this should not be greater than 2\*MaxRtrAdvInterval.

Default: 2\*MaxRtrAdvInterval

## EXAMPLES

```
interface eth0
{
    AdvSendAdvert on;
    prefix 2001:db8:0:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

It says that router advertisement daemon should advertise (AdvSendAdvert on;) the prefix 2001:db8:0:1:: which has a length of 64 on the interface eth0. Also the prefix should be marked as autonomous (AdvAutonomous on;) and as on-link (AdvOnLink on;). All the other options are left on their default values.

To support movement detection of Mobile IPv6 Mobile Nodes, the address of interface should be used instead of network prefix:

```
interface eth0
{
    AdvSendAdvert on;
    prefix 2001:db8:0:1::4/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

For 6to4 support, include the **Base6to4Interface** option in each prefix section. When using a dynamic IPv4 address, set small prefix lifetimes to prevent hosts from retaining unreachable prefixes after a new IPv4 address has been assigned. When advertising to on a dynamic interface (e.g., Bluetooth), skip the interface if it is not active yet.

```
interface bnep0
{
    IgnoreIfMissing on;
    AdvSendAdvert on;

    # Advertise at least every 30 seconds
```



```

MaxRtrAdvInterval 30;

prefix 0:0:0:5678::/64
{
    AdvOnLink on;
    AdvAutonomous on;
    Base6to4Interface ppp0;

    # Very short lifetimes for dynamic addresses
    AdvValidLifetime 300;
    AdvPreferredLifetime 120;
};

```

Since 6to4 is enabled, the prefix will be advertised as 2002:WWXX:YYZZ:5678::/64, where WW.XX.YY.ZZ is the IPv4 address of ppp0 at configuration time. (IPv6 addresses are written in hexadecimal whereas IPv4 addresses are written in decimal, so the IPv4 address WW.XX.YY.ZZ in the 6to4 prefix will be represented in hex.)

In this specific case, the configuration scripts may send HUP signal to radvd when taking bnep0 up or down to notify about the status; in the current radvd releases, sending HUP is no longer mandatory when the link comes back up.

```

interface eth0
{
    AdvSendAdvert on;
    prefix 2001:db8:0:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
    clients
    {
        fe80::21f:16ff:fe06:3aab;
        fe80::21d:72ff:fe96:aaff;
    };
};

```

This configuration would only announce the prefix to fe80::21f:16ff:fe06:3aab and fe80::21d:72ff:fe96:aaff. Furthermore, all RA requests of other clients are denied.

This may come in handy if you want to roll out IPv6 only partially because some clients are broken or untested.

## FILES

```

/usr/sbin/radvd
/etc/radvd.conf
/var/run/radvd/radvd.pid
/var/log/radvd.log

```

## CREDIT

The description of the different flags and variables is in large parts taken from RFC 4861.

**RFCS**

Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 4861, September 2007.

Thomson, S., Narten, T., T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

Deering, S., and R. Hinden, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

Conta, A., Deering, S., and M. Gupta "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)", RFC 4443, March 2006.

Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

Carpenter B., K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001. (6to4 specification)

Draves, R., D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005.

J. Jeong, L. Beloeil, and S. Madanapalli, "IPv6 Router Advertisement Option for DNS Configuration", RFC 5006, September 2007.

**SEE ALSO**

**radvd(8), radvdump(8)**

**BUGS**

radvd does not support splitting up RAs to multiple packets (RFC4861 6.2.3 last paragraph). In practise this limits advertising to ~45 prefixes on a link, but there is no reason to be able to so.