

Commit 36f23c9



RaimoNiskanen committed on Mar 17 · ✓ 1 / 1 · Verified

Merge branch 'raimo/kernel/inet_res-id-27/OTP-20037' into raimo/kernel/inet_res-id-28/OTP-20037

* raimo/kernel/inet_res-id-27/OTP-20037:
Update after feedback

master (#10864) · OTP-29.0-rc3 OTP-28.4.2

2 parents [3041c0f](#) + [dd15e8e](#) commit 36f23c9

7 files changed +58 -25 lines changed

↑ Top

Filter files...

lib/kernel/src

- gen_udp.erl
- inet.erl
- inet6_sctp.erl
- inet_db.erl
- inet_dns.erl
- inet_res.erl
- inet_sctp.erl

7 files changed +58 -25 lines changed

Search within code

```

lib/kernel/src/gen_udp.erl
  ↑  @@ -185,7 +185,7 @@ The 3-tuple form _isn't_ supported on all platforms.
185  185
186  186  -doc("#{equiv => open(Port, [])}).

```

187	187	-spec open(Port) -> {ok, Socket} {error, Reason} when
188	-	Port :: inet:port_number() -1,
188	+	Port :: inet:port_number(),
189	189	Socket :: socket(),
190	190	Reason :: system_limit inet:posix().
191	191	
		@@ -299,7 +299,7 @@ can be truncated without warning.
299	299	The default value for the receive buffer option is `{recbuf, 9216}`.
300	300	"".
301	301	-spec open(Port, Opts) -> {ok, Socket} {error, Reason} when
302	-	Port :: inet:port_number() -1,
302	+	Port :: inet:port_number(),
303	303	Opts :: [inet:inet_backend() open_option()],
304	304	Socket :: socket(),
305	305	Reason :: system_limit inet:posix().

lib/kernel/src/inet.erl		
		@@ -3960,7 +3960,7 @@ gethostbyaddr_tm_native(Addr, Timer, Opts) ->
3960	3960	{ip6_address() 'any' 'loopback',
3961	3961	port_number()}}
3962	3962	undefined, % Internal - no bind()
3963	-	BPort :: port_number() -1,
3963	+	BPort :: port_number(),
3964	3964	Opts :: [socket_setopt()],
3965	3965	Protocol :: socket_protocol() 'mptcp',
3966	3966	Family :: address_family(),
		@@ -4016,7 +4016,7 @@ open(Fd_or_OpenOpts, BAddr, BPort, Opts, Protocol,
		Family, Type, Module)
4016	4016	{ip6_address() 'any' 'loopback',
4017	4017	port_number()}}
4018	4018	undefined, % Internal - translated to 'any'
4019	-	BPort :: port_number() -1,
4019	+	BPort :: port_number(),
4020	4020	Opts :: [socket_setopt()],
4021	4021	Protocol :: socket_protocol() 'mptcp',
4022	4022	Family :: address_family(),
		@@ -4119,24 +4119,38 @@ open_setopts(S, BAddr, BPort, Opts, Module) ->

```

4119 4119
4120 4120
4121 4121
4122 - bind(S, Addr, Port) when is_list(Addr), ?port(Port) ->
4122 + bind(S, Addr, Port) when is_list(Addr), is_integer(Port) ->
4123 4123     bindx(S, Addr, Port);
4124 4124     bind(S, Addr, -1) ->
4125 -     bind_random(S, Addr, 5);
4126 - bind(S, Addr, Port) when ?port(Port) ->
4127 -     %% ?DBG([s, S], {addr, Addr}, {port, Port}),
4128 -     prim_inet:bind(S, Addr, Port).
4129 -
4130 - bind_random(S, Addr, Cnt) when is_integer(Cnt) ->
4125 +     bind_random(S, Addr);
4126 + bind(S, Addr, Port) when is_integer(Port) ->
4127 +     do_bind(S, Addr, Port).
4128 +
4129 + do_bind(S, Addr, Port) ->
4130 +     Result = prim_inet:bind(S, Addr, Port),
4131 +     %% ?DBG([s, S], {addr, Addr}, {port, Port}, Result)),
4132 +     Result.
4133 +
4134 + bind_random(S, Addr) ->
4135 +     Cnt = 3,
4136 +     bind_random(S, Addr, Cnt).
4137 + %%
4138 + bind_random(S, Addr, 0 = _Cnt) ->
4139 +     Port = 0,
4140 +     do_bind(S, Addr, Port);
4141 + bind_random(S, Addr, Cnt) when is_integer(Cnt), 0 < Cnt ->
4131 4142     Port = inet_db:res_option(random_port),
4132 -     %% ?DBG([s, S], {addr, Addr}, {port, Port}),
4143 +     bind_random(S, Addr, Cnt, Port).
4144 + %%
4145 + bind_random(S, Addr, _Cnt, 0 = Port) ->
4146 +     do_bind(S, Addr, Port);
4147 + bind_random(S, Addr, Cnt, Port) when ?port(Port) ->
4133 4148     case prim_inet:bind(S, Addr, Port) of
4134 -         {ok, _} = OK           -> OK;
4135 -         {error, _} = Error ->

```

```

4136 -             if Port == 0             -> Error;
4137 -                 0 < Cnt               -> bind_random(S, Addr, Cnt - 1);
4138 -                 true                   -> Error
4139 -             end
4149 +             {ok, _} = OK ->
4150 +                 %% ?DBG([s, S], {addr, Addr}, {port, Port}, OK]),
4151 +                 OK;
4152 +             {error, _} ->
4153 +                 bind_random(S, Addr, Cnt - 1)
4140 4154         end.
4141 4155
4142 4156     bindx(S, [Addr], Port0) ->

```

lib/kernel/src/inet6_sctp.erl

```

@@ -3,7 +3,7 @@
3 3  %%
4 4  %% SPDX-License-Identifier: Apache-2.0
5 5  %%
6 - %% Copyright Ericsson AB 2007-2025. All Rights Reserved.
6 + %% Copyright Ericsson AB 2007-2026. All Rights Reserved.
7 7  %%
8 8  %% Licensed under the Apache License, Version 2.0 (the "License");
9 9  %% you may not use this file except in compliance with the License.
@@ -57,11 +57,12 @@ open(Opts) ->
57 57             ifaddr = Addr,
58 58             port   = Port,
59 59             type   = Type,
60 -             opts   = S0s}} ->
60 +             opts   = S0s}}
61 +             when ?port(Port) ->
61 62                 inet:open_bind(
62 63                     Fd, Addr, Port, S0s, ?PROTO, ?FAMILY, Type, ?MODULE);
63 -             Error ->
64 -             Error
64 +             {ok, _} -> {error, badarg};
65 +             Error -> Error
65 66         end.
66 67

```

```
67 68 close(S) ->
```



lib/kernel/src/inet_db.erl



```
@@ -2122,6 +2122,10 @@ generate_random_port() ->
```

```
2122 2122         undefined                                -> 0
```

```
2123 2123         end.
```

```
2124 2124
```

```
2125 + %% We use `crypto:rand_uniform/2` here, which is the simplest to use,
2126 + %% but it is deprecated, for outdated reasons. In really old, now obsolete,
2127 + %% libcrypto versions the function was not cryptographically strong,
2128 + %% but since OpenSSL 1.1.0 that is no longer the case.
```

```
2125 2129         -compile({nowarn_deprecated_function, {crypto,rand_uniform,2}}).
```

```
2126 2130
```

```
2127 2131         crypto_rand_range(Range) when is_integer(Range), 0 < Range ->
```



lib/kernel/src/inet_dns.erl



```
@@ -31,6 +31,7 @@
```

```
31 31     %% RFC 2181: Clarifications to the DNS Specification
```

```
32 32     %% RFC 2782: A DNS RR for specifying the location of services (DNS SRV)
```

```
33 33     %% RFC 2915: The Naming Authority Pointer (NAPTR) DNS Resource Rec
```

```
34 + %% RFC 5452: Measures for Making DNS More Resilient against Forged Answers
```

```
34 35     %% RFC 5936: DNS Zone Transfer Protocol (AXFR)
```

```
35 36     %% RFC 6488: DNS Certification Authority Authorization (CAA) Resource Record
```

```
36 37     %% RFC 6762: Multicast DNS
```



lib/kernel/src/inet_res.erl



```
@@ -1137,6 +1137,11 @@ udp_connect(#sock{inet=I}, {A,B,C,D}=IP, Port,
Verbose)
```

```
1137 1137         when ?ip(A,B,C,D), ?port(Port), is_boolean(Verbose) ->
```

```
1138 1138             udp_connect_socket(I, IP, Port, Verbose).
```

```
1139 1139
```

```
1140 + %% Port number -1 is an internal feature that randomizes
1141 + %% the bind port more than an ephemeral port, but it is not allowed
1142 + %% by the type spec for gen_udp:open/2, so we have to suppress
1143 + %% the Dialyzer warnings that that causes here...
1144 + -dialyzer({[no_return, no_fail_call], udp_connect_fam/4}).
```

```

1140 1145 udp_connect_fam(Fam, IP, Port, Verbose) ->
1141 1146     case gen_udp:open(-1, [{active,false},binary,Fam]) of
1142 1147         {ok, Socket} = OK ->
@@ -1460,6 +1465,12 @@ query_ns(S0, {Msg, Buffer}, IP, Port, Timer,
Retry, I,
1460 1465     end.
1461 1466
1462 1467
1468 + %% See udp_connect_fam/4 above. The use of port number -1 causes
1469 + %% a secondary Dialyzer warning here since Dialyzer concludes
1470 + %% that S cannot be 'undefined' because that would cause udp_connect/4
1471 + %% to crash here, so `S :=: undefined` can never be `true`.
1472 + %% This is a suppression for that.
1473 + -dialyzer({no_match, query_udp/7}).
1463 1474 query_udp(_S, _Msg, _Buffer, IP, Port, 0, Verbose) ->
1464 1475     ?verbose(Verbose, "No try UDP server : -p:-p (overdue)\n",
1465 1476         [IP,Port]),
@@ -

```

lib/kernel/src/inet_sctp.erl

```

@@ -3,7 +3,7 @@
3 3 %%
4 4 %% SPDX-License-Identifier: Apache-2.0
5 5 %%
6 - %% Copyright Ericsson AB 2007-2025. All Rights Reserved.
6 + %% Copyright Ericsson AB 2007-2026. All Rights Reserved.
7 7 %%
8 8 %% Licensed under the Apache License, Version 2.0 (the "License");
9 9 %% you may not use this file except in compliance with the License.
@@ -57,10 +57,12 @@ open(Opts) ->
57 57         ifaddr = Addr,
58 58         port = Port,
59 59         type = Type,
60 -         opts = S0s}} ->
60 +         opts = S0s}}
61 +         when ?port(Port) ->
61 62         inet:open_bind(
62 63             Fd, Addr, Port, S0s, ?PROTO, ?FAMILY, Type, ?MODULE);
63 -         Error -> Error

```

```
64 +      {ok, _} -> {error, badarg};
65 +      Error  -> Error
64 66      end.
65 67
66 68      close(S) ->
```

Comments 0



Please [sign in](#) to comment.