

Commit dd15e8e



RaimoNiskanen committed last month Verified

Merge branch 'raimo/kernel/inet_res-id-26/OTP-20037' into raimo/kernel/inet_res-id-27/OTP-20037

* raimo/kernel/inet_res-id-26/OTP-20037:
Update after feedback

master (#10864) · OTP-29.0-rc3 ... OTP-27.3.4.10

2 parents [c24a165](#) + [b057a9d](#) commit dd15e8e

7 files changed +58 -26 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 -26 lines changed

Search within code

```

lib/kernel/src/gen_udp.erl
@@ -183,7 +183,7 @@ The 3-tuple form _isn't_ supported on all platforms.
183 183
184 184 -doc({equiv => open(Port, [])}).

```

185	185	-spec open(Port) -> {ok, Socket} {error, Reason} when
186	-	Port :: inet:port_number() -1,
186	+	Port :: inet:port_number(),
187	187	Socket :: socket(),
188	188	Reason :: system_limit inet:posix().
189	189	
		@@ -297,7 +297,7 @@ can be truncated without warning.
297	297	The default value for the receive buffer option is `{recbuf, 8192}`.
298	298	"".
299	299	-spec open(Port, Opts) -> {ok, Socket} {error, Reason} when
300	-	Port :: inet:port_number() -1,
300	+	Port :: inet:port_number(),
301	301	Opts :: [inet:inet_backend() open_option()],
302	302	Socket :: socket(),
303	303	Reason :: system_limit inet:posix().

lib/kernel/src/inet.erl		
		@@ -3910,7 +3910,7 @@ gethostbyaddr_tm_native(Addr, Timer, Opts) ->
3910	3910	{ip6_address() 'any' 'loopback',
3911	3911	port_number()}}
3912	3912	undefined, % Internal - no bind()
3913	-	BPort :: port_number() -1,
3913	+	BPort :: port_number(),
3914	3914	Opts :: [socket_setopt()],
3915	3915	Protocol :: socket_protocol(),
3916	3916	Family :: address_family(),
		@@ -3966,7 +3966,7 @@ open(Fd_or_OpenOpts, BAddr, BPort, Opts, Protocol,
		Family, Type, Module)
3966	3966	{ip6_address() 'any' 'loopback',
3967	3967	port_number()}}
3968	3968	undefined, % Internal - translated to 'any'
3969	-	BPort :: port_number() -1,
3969	+	BPort :: port_number(),
3970	3970	Opts :: [socket_setopt()],
3971	3971	Protocol :: socket_protocol(),
3972	3972	Family :: address_family(),
		@@ -4069,24 +4069,38 @@ open_setopts(S, BAddr, BPort, Opts, Module) ->

```

4069 4069
4070 4070
4071 4071
4072 - bind(S, Addr, Port) when is_list(Addr), ?port(Port) ->
4072 + bind(S, Addr, Port) when is_list(Addr), is_integer(Port) ->
4073 4073     bindx(S, Addr, Port);
4074 4074     bind(S, Addr, -1) ->
4075 -     bind_random(S, Addr, 5);
4076 - bind(S, Addr, Port) when ?port(Port) ->
4077 -     %% ?DBG([s, S], {addr, Addr}, {port, Port}),
4078 -     prim_inet:bind(S, Addr, Port).
4079 -
4080 - bind_random(S, Addr, Cnt) when is_integer(Cnt) ->
4075 +     bind_random(S, Addr);
4076 + bind(S, Addr, Port) when is_integer(Port) ->
4077 +     do_bind(S, Addr, Port).
4078 +
4079 + do_bind(S, Addr, Port) ->
4080 +     Result = prim_inet:bind(S, Addr, Port),
4081 +     %% ?DBG([s, S], {addr, Addr}, {port, Port}, Result]),
4082 +     Result.
4083 +
4084 + bind_random(S, Addr) ->
4085 +     Cnt = 3,
4086 +     bind_random(S, Addr, Cnt).
4087 + %%
4088 + bind_random(S, Addr, 0 = _Cnt) ->
4089 +     Port = 0,
4090 +     do_bind(S, Addr, Port);
4091 + bind_random(S, Addr, Cnt) when is_integer(Cnt), 0 < Cnt ->
4081 4092     Port = inet_db:res_option(random_port),
4082 -     %% ?DBG([s, S], {addr, Addr}, {port, Port}),
4093 +     bind_random(S, Addr, Cnt, Port).
4094 + %%
4095 + bind_random(S, Addr, _Cnt, 0 = Port) ->
4096 +     do_bind(S, Addr, Port);
4097 + bind_random(S, Addr, Cnt, Port) when ?port(Port) ->
4083 4098     case prim_inet:bind(S, Addr, Port) of
4084 -         {ok, _} = OK           -> OK;
4085 -         {error, _} = Error ->

```

```

4086 -             if Port == 0             -> Error;
4087 -                 0 < Cnt                 -> bind_random(S, Addr, Cnt - 1);
4088 -                 true                     -> Error
4089 -             end
4099 +             {ok, _} = OK ->
4100 +                 %% ?DBG([s, S], {addr, Addr}, {port, Port}, OK]),
4101 +                 OK;
4102 +             {error, _} ->
4103 +                 bind_random(S, Addr, Cnt - 1)
4090 4104         end.
4091 4105
4092 4106     bindx(S, [Addr], Port0) ->

```

lib/kernel/src/inet6_sctp.erl

```

... @@ -1,7 +1,7 @@
1 1  %%
2 2  %% %CopyrightBegin%
3 3  %%
4 - %% Copyright Ericsson AB 2007-2024. All Rights Reserved.
4 + %% Copyright Ericsson AB 2007-2026. All Rights Reserved.
5 5  %%
6 6  %% Licensed under the Apache License, Version 2.0 (the "License");
7 7  %% you may not use this file except in compliance with the License.
... @@ -55,11 +55,12 @@ open(Opts) ->
55 55             ifaddr = Addr,
56 56             port  = Port,
57 57             type  = Type,
58 -             opts  = S0s}} ->
58 +             opts  = S0s}}
59 +             when ?port(Port) ->
59 60                 inet:open_bind(
60 61                     Fd, Addr, Port, S0s, ?PROTO, ?FAMILY, Type, ?MODULE);
61 -             Error ->
62 -             Error
62 +             {ok, _} -> {error, badarg};
63 +             Error -> Error
63 64         end.
64 65

```

```
65 66 close(S) ->
```



lib/kernel/src/inet_db.erl



```
@@ -2110,6 +2110,10 @@ generate_random_port() ->
```

```
2110 2110         undefined                                -> 0
```

```
2111 2111         end.
```

```
2112 2112
```

```
2113 + %% We use `crypto:rand_uniform/2` here, which is the simplest to use,
2114 + %% but it is deprecated, for outdated reasons. In really old, now obsolete,
2115 + %% libcrypto versions the function was not cryptographically strong,
2116 + %% but since OpenSSL 1.1.0 that is no longer the case.
```

```
2113 2117         -compile({nowarn_deprecated_function, {crypto,rand_uniform,2}}).
```

```
2114 2118
```

```
2115 2119         crypto_rand_range(Range) when is_integer(Range), 0 < Range ->
```



lib/kernel/src/inet_dns.erl



```
@@ -29,6 +29,7 @@
```

```
29 29     %% RFC 2181: Clarifications to the DNS Specification
```

```
30 30     %% RFC 2782: A DNS RR for specifying the location of services (DNS SRV)
```

```
31 31     %% RFC 2915: The Naming Authority Pointer (NAPTR) DNS Resource Rec
```

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

```
32 33     %% RFC 5936: DNS Zone Transfer Protocol (AXFR)
```

```
33 34     %% RFC 6488: DNS Certification Authority Authorization (CAA) Resource Record
```

```
34 35     %% RFC 6762: Multicast DNS
```



lib/kernel/src/inet_res.erl



```
@@ -1073,6 +1073,11 @@ udp_connect(#sock{inet=I}, {A,B,C,D}=IP, Port,
Verbose)
```

```
1073 1073         when ?ip(A,B,C,D), ?port(Port), is_boolean(Verbose) ->
```

```
1074 1074             udp_connect_socket(I, IP, Port, Verbose).
```

```
1075 1075
```

```
1076 + %% Port number -1 is an internal feature that randomizes
1077 + %% the bind port more than an ephemeral port, but it is not allowed
1078 + %% by the type spec for gen_udp:open/2, so we have to suppress
1079 + %% the Dialyzer warnings that that causes here...
1080 + -dialyzer([no_return, no_fail_call], udp_connect_fam/4).
```

```

1076 1081 udp_connect_fam(Fam, IP, Port, Verbose) ->
1077 1082     case gen_udp:open(-1, [{active,false},binary,Fam]) of
1078 1083         {ok, Socket} = OK ->
@@ -1395,7 +1400,12 @@ query_ns(S0, {Msg, Buffer}, IP, Port, Timer,
Retry, I,
1395 1400         end
1396 1401     end.
1397 1402
1398 -
1403 + %% See udp_connect_fam/4 above. The use of port number -1 causes
1404 + %% a secondary Dialyzer warning here since Dialyzer concludes
1405 + %% that S cannot be 'undefined' because that would cause udp_connect/4
1406 + %% to crash here, so `S := undefined` can never be `true`.
1407 + %% This is a suppression for that.
1408 + -dialyzer({no_match, query_udp/7}).
1399 1409 query_udp(_S, _Msg0, _Buffer, IP, Port, 0, Verbose) ->
1400 1410     ?verbose(Verbose, "No try UDP server : -p:-p (overdue)\n",
1401 1411         [IP,Port]),

```

lib/kernel/src/inet_sctp.erl

```

... @@ -1,7 +1,7 @@
1 1 %%
2 2 %% %CopyrightBegin%
3 3 %%
4 - %% Copyright Ericsson AB 2007-2024. All Rights Reserved.
4 + %% Copyright Ericsson AB 2007-2026. All Rights Reserved.
5 5 %%
6 6 %% Licensed under the Apache License, Version 2.0 (the "License");
7 7 %% you may not use this file except in compliance with the License.
@@ -55,10 +55,12 @@ open(Opts) ->
55 55         ifaddr = Addr,
56 56         port   = Port,
57 57         type   = Type,
58 -         opts  = S0s}} ->
58 +         opts  = S0s}}
59 +         when ?port(Port) ->
59 60         inet:open_bind(
60 61             Fd, Addr, Port, S0s, ?PROTO, ?FAMILY, Type, ?MODULE);

```

61	-	Error -> Error
62	+	{ok, _} -> {error, badarg};
63	+	Error -> Error
62	64	end.
63	65	
64	66	close(S) ->
↓		

Comments 0



Please [sign in](#) to comment.