```
**********************************************************
   18816 Thu May 22 21:03:26 2014
new/usr/src/man/man3socket/getaddrinfo.3socket
4775 Fixed formatting in getaddinfo(3) man page
**********************************************************
______unchanged_portion_omitted_
```

```
.fi
.in -2

.sp
.LP
In this \fIhints\fR structure, all members other than \fBai_flags\fR,
\fBai_family\fR, \fBai_socktype\fR, and \fBai_protocol\fR must be 0 or a null
pointer. A value of \fBPF_UNSPEC\fR for \fBai_family\fR indicates that the
caller will accept any protocol family. A value of 0 for \fBai_socktype\fR
indicates that the caller will accept any socket type.  A value of 0 for
\fBai_protocol\fR indicates that the caller will accept any protocol. For
example, if the caller handles only TCP and not UDP, then the \fBai_socktype\fR
member of the \fIhints\fR structure should be set to \fBSOCK_STREAM\fR when
\fBgetaddrinfo()\fR is called. If the caller handles only IPv4 and not IPv6,
then the \fBai_family\fR member of the \fIhints\fR structure should be set to
\fBPF_INET\fR when \fBgetaddrinfo()\fR is called. If the third argument to
\fBgetaddrinfo()\fR is a null pointer, it is as if the caller had filled in an
\fBaddrinfo\fR structure initialized to 0 with \fBai_family\fR set to
\fBPF_UNSPEC\fR.
.sp
.LP
Upon success, a pointer to a linked list of one or more \fBaddrinfo\fR
structures is returned through the final argument.  The caller can process each
\fBaddrinfo\fR structure in this list by following the \fBai_next\fR pointer,
until a null pointer is encountered. In each returned \fBaddrinfo\fR structure
the three members \fBai_family\fR, \fBai_socktype\fR, and \fBai_protocol\fR are
the corresponding arguments for a call to the \fBsocket\fR(3SOCKET) function.
In each \fBaddrinfo\fR structure the \fBai_addr\fR member points to a filled-in
socket address structure whose length is specified by the \fBai_addrlen\fR
member.
.sp
.LP
If the \fBAI_PASSIVE\fR bit is set in the \fBai_flags\fR member of the
\fIhints\fR structure, the caller plans to use the returned socket address
structure in a call to \fBbind\fR(3SOCKET). In this case, if the \fInodename\fR
argument is a null pointer, the IP address portion of the socket address
structure will be set to \fBINADDR_ANY\fR for an IPv4 address or
\fBIN6ADDR_ANY_INIT\fR for an IPv6 address.
.sp
.LP
If the \fBAI_PASSIVE\fR bit is not set in the \fBai_flags\fR member of the
\fIhints\fR structure, then the returned socket address structure will be ready
for a call to \fBconnect\fR(3SOCKET) (for a connection-oriented protocol) or
either \fBconnect\fR(3SOCKET), \fBsendto\fR(3SOCKET), or \fBsendmsg\fR(3SOCKET)
(for a connectionless protocol). If the \fInodename\fR argument is a null
pointer, the IP address portion of the socket address structure will be set to
the loopback address.
.sp
.LP
If the \fBAI_CANONNAME\fR bit is set in the \fBai_flags\fR member of the
\fIhints\fR structure, then upon successful return the \fBai_canonname\fR
member of the first \fBaddrinfo\fR structure in the linked list will point to a
null-terminated string containing the canonical name of the specified
\fInodename\fR. A numeric host address string is not a name, and thus does not
have a canonical name form; no address to host name translation is performed.
.sp
.LP
If the \fBAI_NUMERICHOST\fR bit is set in the \fBai_flags\fR member of the
\fIhints\fR structure, then a non-null \fInodename\fR string must be a numeric
host address string. Otherwise an error of \fBEAI_NONAME\fR is returned. This
```



```
flag prevents any type of name resolution service (such as DNS) from being
called.
.sp
.LP
If the \fBAI_NUMERICSERV\fR flag is specified, then a non-null servname string
supplied will be a numeric port string.  Otherwise, an [\fBEAI_NONAME\fR] error
is returned. This flag prevents any type of name resolution service (for
example, NIS+) from being invoked.
.sp
.LP
If the \fBAI_V4MAPPED\fR flag is specified along with an \fBai_family\fR of
\fBAF_INET6\fR, then \fBgetaddrinfo()\fR returns IPv4-mapped IPv6 addresses on
finding no matching IPv6 addresses (\fBai_addrlen\fR shall be 16). For example,
if no AAAA records are found when using DNS, a query is made for A records. Any
found records are returned as IPv4-mapped IPv6 addresses.
.sp
.LP
The \fBAI_V4MAPPED\fR flag is ignored unless \fBai_family\fR equals
\fBAF_INET6\fR.
.sp
.LP
If the \fBAI_ALL\fR flag is used with the AI_V4MAPPED flag, then
\fBgetaddrinfo()\fR returns all matching IPv6 and IPv4 addresses. For example,
when using the DNS, queries are made for both AAAA records and A records, and
\fBgetaddrinfo()\fR returns the combined results of both queries. Any IPv4
addresses found are returned as IPv4-mapped IPv6 addresses.
.sp
.LP
The \fBAI_ALL\fR flag without the \fBAI_V4MAPPED\fR flag is ignored.
.sp
.LP
When \fBai_family\fR is not specified (\fBAF_UNSPEC\fR), \fBAI_V4MAPPED\fR and
\fBAI_ALL\fR flags are used only if \fBAF_INET6\fR is supported.
.sp
.LP
If the \fBAI_ADDRCONFIG\fR flag is specified, IPv4 addresses are returned only
if an IPv4 address is configured on the local system, and IPv6 addresses are
returned only if an IPv6 address is configured on the local system. For this
case, the loopback address is not considered to be as valid as a configured
address. For example, when using the DNS, a query for AAAA records should occur
only if the node has at least one IPv6 address configured (other than IPv6
loopback) and a query for A records should occur only if the node has at least
one IPv4 address configured (other than the IPv4 loopback).
.sp
.LP
All of the information returned by \fBgetaddrinfo()\fR is dynamically
allocated: the \fBaddrinfo\fR structures as well as the socket address
structures and canonical node name strings pointed to by the \fBaddrinfo\fR
structures. The \fBfreeaddrinfo()\fR function is called to return this
information to the system. For \fBfreeaddrinfo()\fR, the \fBaddrinfo\fR
structure pointed to by the \fIai\fR argument is freed, along with any dynamic
storage pointed to by the structure. This operation is repeated until a null
\fBai_next\fR pointer is encountered.
.sp
.LP
To aid applications in printing error messages based on the \fBEAI_\fR* codes
returned by \fBgetaddrinfo()\fR, the \fBgai_strerror()\fR is defined. The
argument is one of the \fBEAI_\fR* values defined below and the return value
points to a string describing the error. If the argument is not one of the
\fBEAI_\fR* values, the function still returns a pointer to a string whose
contents indicate an unknown error.
.sp
.LP
The \fBgetnameinfo()\fR function looks up an IP address and port number
provided by the caller in the name service database and system-specific
database, and returns text strings for both in buffers provided by the caller.
```

The function indicates successful completion by a 0 return value; a non-zero return value indicates failure.

.sp
.LP
The first argument, \fIsa\fR, points to either a \fBsockaddr_in\fR structure (for IPv4) or a \fBsockaddr_in6\fR structure (for IPv6) that holds the IP address and port number. The \fIsalen\fR argument gives the length of the \fBsockaddr_in\fR or \fBsockaddr_in6\fR structure.

.sp
.LP
The function returns the node name associated with the IP address in the buffer pointed to by the \fIhost\fR argument.

.sp
.LP
The function can also return the IPv6 zone-id in the form:

.sp
.in +2
.nf
<address>%<zone-id>
.fi
.in -2

.sp
.LP
The caller provides the size of this buffer with the \fIhostlen\fR argument. The service name associated with the port number is returned in the buffer pointed to by \fIserv\fR, and the \fIservlen\fR argument gives the length of this buffer. The caller specifies not to return either string by providing a 0 value for the \fIhostlen\fR or \fIservlen\fR arguments. Otherwise, the caller must provide buffers large enough to hold the node name and the service name, including the terminating null characters.

.sp
.LP
To aid the application in allocating buffers for these two returned strings, the following constants are defined in <\fBnetdb.h\fR>:

.sp
.in +2
.nf
#define NI_MAXHOST 1025
#define NI_MAXSERV   32
.fi
.in -2

.sp
.LP
The final argument is a flag that changes the default actions of this function. By default, the fully-qualified domain name (\fBFQDN\fR) for the host is looked up in the name service database and returned. If the flag bit \fBNI_NOFQDN\fR is set, only the node name portion of the \fBFQDN\fR is returned for local hosts.

.sp
.LP
If the flag bit \fBNI_NUMERICHOST\fR is set, or if the host’s name cannot be located in the name service, the numeric form of the host’s address is returned instead of its name, for example, by calling \fBinet_ntop()\fR (see \fBinet\fR(3SOCKET)) instead of \fBgetipnodebyname\fR(3SOCKET). If the flag bit \fBNI_NAMEREQD\fR is set, an error is returned if the host’s name cannot be located in the name service database.

.sp
.LP
If the flag bit \fBNI_NUMERICSERV\fR is set, the numeric form of the service address is returned (for example, its port number) instead of its name. The two \fBNI_NUMERIC\fR* flags are required to support the \fB-n\fR flag that many commands provide.

.sp
.LP

A fifth flag bit, \fBNI_DGRAM\fR, specifies that the service is a datagram service, and causes \fBgetservbyport\fR(3SOCKET) to be called with a second argument of \fBudp\fR instead of the default \fBtcp\fR. This is required for the few ports (for example, 512-514) that have different services for UDP and TCP.

.sp
.LP
These \fBNI_\fR* flags are defined in <\fBnetdb.h\fR> along with the \fBAI_\fR* flags already defined for \fBgetaddrinfo()\fR.

.SH RETURN VALUES
.sp
.LP
For \fBgetaddrinfo()\fR, if the query is successful, a pointer to a linked list of one or more \fBaddrinfo\fR structures is returned by the fourth argument and **the function returns \fB0\fR. The order of the addresses returned in the fourth** *the function returns \fB0\fR. The order of the addresses returned i nthe fourth* argument is discussed in the ADDRESS ORDERING section. If the query fails, a non-zero error code will be returned. For \fBgetnameinfo()\fR, if successful, the strings hostname and service are copied into \fIhost\fR and \fIserv\fR, respectively. If unsuccessful, zero values for either \fIhostlen\fR or \fIservlen\fR will suppress the associated lookup; in this case no data is copied into the applicable buffer. If \fBgai_strerror()\fR is successful, a pointer to a string containing an error message appropriate for the \fBEAI_\fR* errors is returned. If \fIerrcode\fR is not one of the \fBEAI_\fR* values, a pointer to a string indicating an unknown error is returned.

.SS "Address Ordering"
.sp
.LP
AF_INET6 addresses returned by the fourth argument of \fBgetaddrinfo()\fR are ordered according to the algorithm described in \fIRFC 3484, Default Address Selection for Internet Protocol version 6 (IPv6)\fR. The addresses are ordered using a list of pair-wise comparison rules which are applied in order. If a rule determines that one address is better than another, the remaining rules are irrelevant to the comparison of those two addresses. If two addresses are equivalent according to one rule, the remaining rules act as a tie-breaker. The address ordering list of pair-wise comparison rules follow below:

.sp

.sp
.TS
box;
l | l
l | l .
Avoid unusable destinations.   T{
Prefer a destination that is reachable through the IP routing table.
T}
_
Prefer matching scope. T{
Prefer a destination whose scope is equal to the scope of its source address. Se
T}
_
Avoid link-local source.   T{
Avoid selecting a link-local source address when the destination address is not
T}
_
Avoid deprecated addresses.   T{
Prefer a destination that is not deprecated (\fBIFF_DEPRECATED\fR).
T}
_
T{
Prefer matching label. This rule uses labels that are obtained through the IPv6
T}   T{
Prefer a destination whose label is equal to the label of its source address.
T}
_
T{

```
353 Prefer higher precedence. This rule uses precedence values that are obtained thr
354 T}      T{
355 Prefer the destination whose precedence is higher than the other destination.
356 T}
357 _
358 Prefer native transport.        T{
359 Prefer a destination if the interface that is used for sending packets to that d
360 T}
361 _
362 T{
363 Prefer smaller scope. See \fBinet6\fR(7P) for the definition of this rule.
364 T}      T{
365 Prefer the destination whose scope is smaller than the other destination.
366 T}
367 _
368 Use longest matching prefix.    T{
369 When the two destinations belong to the same address family, prefer the destinat
370 T}
371 .TE

373 .SH ERRORS
374 .sp
375 .LP
376 The following names are the error values returned by \fBgetaddrinfo()\fR and
377 are defined in <\fBnetdb.h\fR>:
378 .sp
379 .ne 2
380 .na
381 \fB\fBEAI_ADDRFAMILY\fR\fR
382 .ad
383 .RS 18n
384 Address family for nodename is not supported.
385 .RE

387 .sp
388 .ne 2
389 .na
390 \fB\fBEAI_AGAIN\fR\fR
391 .ad
392 .RS 18n
393 Temporary failure in name resolution has occurred .
394 .RE

396 .sp
397 .ne 2
398 .na
399 \fB\fBEAI_BADFLAGS\fR\fR
400 .ad
401 .RS 18n
402 Invalid value specified for \fBai_flags\fR.
403 .RE

405 .sp
406 .ne 2
407 .na
408 \fB\fBEAI_FAIL\fR\fR
409 .ad
410 .RS 18n
411 Non-recoverable failure in name resolution has occurred.
412 .RE

414 .sp
415 .ne 2
416 .na
417 \fB\fBEAI_FAMILY\fR\fR
418 .ad
```



```
419 .RS 18n
420 The \fBai_family\fR is not supported.
421 .RE

423 .sp
424 .ne 2
425 .na
426 \fB\fBEAI_MEMORY\fR\fR
427 .ad
428 .RS 18n
429 Memory allocation failure has occurred.
430 .RE

432 .sp
433 .ne 2
434 .na
435 \fB\fBEAI_NODATA\fR\fR
436 .ad
437 .RS 18n
438 No address is associated with \fInodename\fR.
439 .RE

441 .sp
442 .ne 2
443 .na
444 \fB\fBEAI_NONAME\fR\fR
445 .ad
446 .RS 18n
447 Neither \fInodename\fR nor \fIservname\fR is provided or known.
448 .RE

450 .sp
451 .ne 2
452 .na
453 \fB\fBEAI_SERVICE\fR\fR
454 .ad
455 .RS 18n
456 The \fIservname\fR is not supported for \fBai_socktype\fR.
457 .RE

459 .sp
460 .ne 2
461 .na
462 \fB\fBEAI_SOCKTYPE\fR\fR
463 .ad
464 .RS 18n
465 The \fBai_socktype\fR is not supported.
466 .RE

468 .sp
469 .ne 2
470 .na
471 \fB\fBEAI_OVERFLOW\fR\fR
472 .ad
473 .RS 18n
474 Argument buffer has overflowed.
475 .RE

477 .sp
478 .ne 2
479 .na
480 \fB\fBEAI_SYSTEM\fR\fR
481 .ad
482 .RS 18n
483 System error was returned in \fIerrno\fR.
484 .RE
```

```
.SH FILES
.sp
.ne 2
.na
\fB\fB/etc/inet/hosts\fR\fR
.ad
.RS 22n
local database that associates names of nodes with IP addresses
.RE

.sp
.ne 2
.na
\fB\fB/etc/netconfig\fR\fR
.ad
.RS 22n
network configuration database
.RE

.sp
.ne 2
.na
\fB\fB/etc/nsswitch.conf\fR\fR
.ad
.RS 22n
configuration file for the name service switch
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for description of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Committed
_
MT-Level        MT-Safe
_
Standard        See \fBstandards\fR(5).
.TE

.SH SEE ALSO
.sp
.LP
\fBipaddrsel\fR(1M), \fBgethostbyname\fR(3NSL), \fBgetipnodebyname\fR(3SOCKET),
\fBhtonl\fR(3SOCKET), \fBinet\fR(3SOCKET), \fBnetdb.h\fR(3HEAD),
\fBsocket\fR(3SOCKET), \fBhosts\fR(4), \fBnsswitch.conf\fR(4),
\fBattributes\fR(5), \fBstandards\fR(5), \fBinet6\fR(7P)
.sp
.LP
Draves, R. \fIRFC 3484, Default Address Selection for Internet Protocol version
6 (IPv6)\fR. Network Working Group. February 2003.
.SH NOTES
.sp
.LP
IPv4-mapped addresses are not recommended.
```