**Specification**

USPD: US.ECO.00005-01 90

Component: IEcoSocketP02

UGUID:

Status: Draft

Date: November 8, 2021

Version: 1.0

|  |  |
| --- | --- |
| **Authors** | **Company** |
| Vladimir Bashev | PEERF |
|  |  |
|  |  |

Content

[**1.** **Overview** 4](#_Toc179660763)

[**1.1.** **Introduction** 4](#_Toc179660764)

[**1.2.** **Note** 4](#_Toc179660765)

[**1.3.** **Links** 4](#_Toc179660766)

[**2.** **Eco.** **SocketP02 Component** 5](#_Toc179660767)

[**3.** **Interface IEcoSocketP02** 6](#_Toc179660768)

[**3.1.** **IEcoSocketP02 IDL** 6](#_Toc179660769)

[**3.1.1.** **accept function** 8](#_Toc179660770)

[**3.1.2.** **bind function** 8](#_Toc179660771)

[**3.1.3.** **connect function** 8](#_Toc179660772)

[**3.1.4.** **getpeername function** 8](#_Toc179660773)

[**3.1.5.** **getsockname function** 8](#_Toc179660774)

[**3.1.6.** **getsockopt function** 8](#_Toc179660775)

[**3.1.7.** **listen function** 8](#_Toc179660776)

[**3.1.8.** **recv function** 8](#_Toc179660777)

[**3.1.9.** **recvfrom function** 8](#_Toc179660778)

[**3.1.10.** **recvmsg function** 9](#_Toc179660779)

[**3.1.11.** **recvmmsg function** 9](#_Toc179660780)

[**3.1.12.** **send function** 9](#_Toc179660781)

[**3.1.13.** **sendto function** 9](#_Toc179660782)

[**3.1.14.** **sendmsg function** 9](#_Toc179660783)

[**3.1.15.** **sendmmsg function** 9](#_Toc179660784)

[**3.1.16.** **setsockopt function** 9](#_Toc179660785)

[**3.1.17.** **shutdown function** 9](#_Toc179660786)

[**3.1.18.** **sockatmark function** 9](#_Toc179660787)

[**3.1.19.** **socket function** 10](#_Toc179660788)

[**3.1.20.** **socketpair function** 10](#_Toc179660789)

[**4.1.** **IEcoINetP02 IDL** 11](#_Toc179660790)

[**4.1.1.** **htonl function** 12](#_Toc179660791)

[**4.1.2.** **htons function** 12](#_Toc179660792)

[**4.1.3.** **ntohl function** 12](#_Toc179660793)

[**4.1.4.** **ntohs function** 12](#_Toc179660794)

[**4.1.5.** **inet\_addr function** 12](#_Toc179660795)

[**4.1.6.** **inet*\_*ntoa function** 12](#_Toc179660796)

[**4.1.7.** **inet\_ntop function** 12](#_Toc179660797)

[**4.1.8.** **inet\_pton function** 12](#_Toc179660798)

[**5.** **Error codes** 13](#_Toc179660799)

[**Appendix A: Training programs** 14](#_Toc179660800)

1. **Overview**

This document describes the requirements for the implementation of the Eco.SocketP02 component.

* 1. **Introduction**

Description.

* 1. **Note**
* Keywords
  1. **Links**

This paragraph contains links to information to help you understand this document:

[] – name of the link

Available by: http://address

1. **Eco.** **SocketP02 Component**

The Eco.SocketP02 component

The component has the following description:

1. **IEcoSocketP02 Interface**
   1. **IEcoSocketP02 IDL**

|  |
| --- |
| **ECO IDL** |
| import "IEcoBase1.h" | | | |
| [  object,  uguid(8EDA7C36-A3D0-422A-9C10-189A2BB86F5F),  ] | | | |
| interface IEcoSocketP02 : IEcoUnknown { | | | |
|  | | | |
| int | | ***accept*** | ([in] int,  [in] struct sockaddr \*,  [in, out] socklen\_t \*); |
|  | | | |
| int | | ***bind*** | ([in] int,  const struct sockaddr \*,  [in] socklen\_t); |
|  | | | |
| int | | ***Connect*** | ([in] int,  [in] const struct sockaddr \*,  [in] socklen\_t); |
|  | | | |
| int | | ***getpeername*** | ([in] int,  [out] struct sockaddr \*,  [in, out] socklen\_t \*); |
|  | | | |
| int | | ***getsockname*** | ([in] int,  [out] struct sockaddr \*,  [in, out] socklen\_t \*); |
|  | | | |
| int | | ***getsockopt*** | ([in] int,  [in] int,  [in] int,  [out] void \*,  [in,out] socklen\_t \*); |
|  | | | |
| int | | ***listen*** | ([in] int,  [in] int); |
|  | | | |
| ssize\_t | | ***recv*** | ([in] int,  [out] void \*,  [in] size\_t,  [in] int); |
|  | |  |  |
| ssize\_t | | ***recvfrom*** | ([in] int,  [out] void \*,  [in] size\_t,  [in] int,  [out] stuct sockaddr \*,  [in, out, optional] socklen\_t \*); |
|  | |  |  |
| ssize\_t | | ***recvmsg*** | (int,  struct msghdr \*,  int); |
|  | |  |  |
| int | | ***recvmmsg*** | (int,  struct mmsghdr \*,  unsigned int,  int,  struct timespec \*); |
|  | |  |  |
| ssize\_t | | ***send*** | ([in] int,  [in] const void \*,  [in] size\_t,  [in] int); |
|  | |  |  |
| ssize\_t | | ***sendto*** | ([in] int,  [in] const void \*,  [in] size\_t,  [in] int,  [in] const struct sockaddr \*,  [in] socklen\_t); |
|  | |  |  |
| ssize\_t | | ***sendmsg*** | (int,  const struct msghdr \*,  int); |
|  | |  |  |
| int | | ***sendmmsg*** | (int,  struct mmsghdr \*,  unsigned int,  int); |
|  | |  |  |
| int | | ***setsockopt*** | ([in] int,  [in] int,  [in] int,  [in] const void \*,  [in] socklen\_t); |
|  | |  |  |
| int | | ***shutdown*** | ([in] int,  [in] int); |
|  | |  |  |
| int | | ***sockatmark*** | ([in] int); |
|  | |  |  |
| int | | ***socket*** | ([in] int,  [in] int,  [in] int); |
|  | |  |  |
| int | | ***socketpair*** | (int,  int,  int,  int \*); |
|  | |  |  |
| } | |  |  |

* + 1. **accept function**

The function allows an incoming socket connection attempt.

* + 1. **bind function**

The function associates a local address with a socket.

* + 1. **connect function**

The function establishes a connection to the specified socket.

* + 1. **getpeername function**

The function retrieves the address of the peer node to which the socket is connected.

* + 1. **getsockname function**

The function retrieves the local name of the socket.

* + 1. **getsockopt function**

The function retrieves the socket parameter.

* + 1. **listen function**

The function places the socket in a state in which it is listening for an incoming connection.

* + 1. **recv function**

The function receives data from a connected socket or a linked socket without establishing a connection.

* + 1. **recvfrom function**

The function receives the datagram and stores the source address.

* + 1. **recvmsg function**

The function receives messages on a socket with a socket descriptor and stores them in an array of message headers.

* + 1. **recvmmsg function**

Функция

* + 1. **send function**

The function sends data to the connected socket.

* + 1. **sendto function**

The function sends data to a specific destination.

* + 1. **sendmsg function**

The function sends messages to the socket.

* + 1. **sendmmsg function**

The function sends multiple messages to a socket.

* + 1. **setsockopt function**

The function sets the socket parameter.

* + 1. **shutdown function**

The function disables sending or receiving data on the socket.

* + 1. **sockatmark function**

The function determines if the socket has an out-of-band data marker.

* + 1. **socket function**

The function creates an endpoint for communication and returns a file descriptor for the socket.

* + 1. **socketpair function**

The function gets a pair of sockets of the specified type, which have no name and are connected in the specified domain and use the specified protocol.

1. **IEcoINetP02 Interface**
   1. **IEcoINetP02 IDL**

|  |
| --- |
| **ECO IDL** |
| import "IEcoBase1.h" | | | |
| [  object,  uguid(2C313FFA-19C2-4D63-82F0-62E83B1DC103),  ] | | | |
| interface IEcoINetP02 : IEcoUnknown { | | | |
|  | | | |
| uint32\_t | | ***htonl*** | ([in] uint32\_t); |
|  | | | |
| uint16\_t | | ***htons*** | ([in] uint16\_t); |
|  | | | |
| uint32\_t | | ***ntohl*** | ([in] uint32\_t); |
|  | | | |
| uint16\_t | | ***ntohs*** | ([in] uint16\_t); |
|  | | | |
| in\_addr\_t | | ***inet\_addr*** | (const char \*); |
|  | | | |
| char\* | | ***inet\_ntoa*** | (struct in\_addr); |
|  | | | |
| const char\* | | ***inet\_ntop*** | ([in] int,  [in] const void\* [restrict],  [out] char\* [restrict],  [in] socklen\_t); |
|  | | | |
| int | | ***inet\_pton*** | ([in] int,  [in] const char\* [restrict],  [out] void\* [restrict]); |
| } | |  |  |

* + 1. **htonl function**

The function returns the value in TCP/IP's network byte order.

* + 1. **htons function**

The function returns the value in TCP/IP network byte order.

* + 1. **ntohl function**

The function converts u\_long from TCP/IP network byte order to host byte order.

* + 1. **ntohs function**

The function converts u\_short from TCP/IP network byte order to host byte order.

* + 1. **inet\_addr function**

The function converts a string containing a decimal IPv4 address with dots into a valid address for the IN\_ADDR structure.

* + 1. **inet*\_*ntoa function**

The function converts an Internet address (Ipv4) to an ASCII string in standard Internet decimal format.

* + 1. **inet\_ntop function**

The function converts an IPv4 or IPv6 Internet address to a string in standard Internet format.

* + 1. **inet\_pton function**

The function converts an IPv4 or IPv6 network address in the standard text representation form to a numeric binary form.

1. **Error codes**

The following table contains the error codes.

|  |  |  |
| --- | --- | --- |
| **Error code** | **Value** | **Description** |
| ERR\_ECO\_SUCCESES | 0x0000 | Operation successful. |
| ERR\_ECO\_UNEXPECTED | 0xFFFF | Unexpected condition. |
| ERR\_ECO\_POINTER | 0xFFEE | NULL was passed incorrectly for a pointer value. |
| ERR\_ECO\_NOINTERFACE | 0xFFED | No such interface supported. |
| ERR\_ECO\_COMPONENT\_NOTFOUND | 0xFFE9 | The component was not found. |
|  |  |  |
|  |  |  |

# **Appendix A: Training programs**