# ezConfig Program Library for JAVA

2009-10-08

Sollay Systems. Co., Ltd.    http://www.sollae.co.kr

# Agenda

# 1 Introduction

This library is exported by Eclipse 3.2.1, jre 1.6.0 and tested by java version 1.6.0.

## WARNING:

- All "reserved" or "not used" members of structure are NOT allowed use.
  Please don’t use “reserved” or “not used” members.
- Please check ezTCP’s MAC Address or IP Address before using “write” function.
  If you write wrong information to ezTCP, then it may does not work correctly.
- We DO NOT guarantee any damage occurred by illegal use of this library.

# 2 Data Structures

## 2.1 ezcfglib.df.Spe_env

### 2.1.1 Overview

Basic data structure for ezConfig Program Library Functions.

```
public class Spe_Env {

public byte[]       getEth_addr();                     /* Ethernet address. 6bytes */
public void         setEth_addr(byte[] eth_addr);      /* Never modify. */

public boolean      getDhcp();
public void         setDhcp(boolean tf);

public boolean      getArp();
public void         setArp(boolean tf);

public boolean      getEzcfg();
public void         setEzcfg(boolean tf);

public boolean      getPppoe();
public void         setPppoe(boolean tf);

public boolean      getPwrDown();
public void         setPwrDown(boolean tf);

public boolean      getRemoteConfig();
public void         setRemoteConfig(boolean tf);

public boolean      getMacId() ;
public void         setMacId(boolean tf);

public boolean      getSioNoIdle();
public void         setSioNoIdle(boolean tf);

public boolean      getSioRxDrop();
public void         setSioRxDrop(boolean tf);

public int          getLocal_ip();
public void         setLocal_ip(int local_ip);

public int          getNet_mask();
public void         setNet_mask(int net_mask);

public int          getGate_ip();
public void         setGate_ip(int gate_ip);

public int          getParity();
public void         setParity(int parity);

public int          getDatabit();
public void         setDatabit(int databit);
public boolean   getRtsCts();
public void      setRtsCts(boolean tf);

public boolean   getXonOff();
public void      setXonOff(boolean tf);

public boolean   getTelnet();
public void       setTelnet(boolean tf);
public boolean   getWlanCfg();
public void      setWlanCfg(boolean tf); /* READ ONLY. Never modify */

public int       getSerialType();
public void      setSerialType(int stype);

public int       getStopbit();
public void      setStopbit(int stopbit);

public boolean   getMultipleConnection();
public void      setMultipleConnection(boolean tf);

public int       getParity2();
public void      setParity2(int parity2);

public boolean   getTxDelay();
public void      setTxDelay(Boolean tf);

public int       getSio_baud();
public void      setSio_baud(int sio_baud);

public short     getTimeout();
public void      setTimeout(short timeout);

public byte      getMux_type();
public void      setMux_type(byte mux_type);

public short     getWater_mark();
public void      setWater_mark(short water_mark);

public int       getRemote_ip();
public void      setRemote_ip(int remote_ip);

public int       getRemote_port();
public void      setRemote_port(int remote_port);

public int       getLocal_port();
public void      setLocal_port(int local_port);

public byte[]    getPoe_pwd();                /* PPPoE password. Maximum 8bytes */
public void      setPoe_pwd(byte[] poe_pwd);

public byte[]    getPoe_uid();                /* PPPoE ID. Maximum 8bytes */
public void      setPoe_uid(byte[] poe_uid);

public byte[]    getPasswd();                 /* ezTCP password. Maximum 8bytes*/
public void      setPasswd(byte[] passwd);
public void       fromBytes(byte[] buff);        /* NEVER USE THIS FUNCTION */
public byte[]     toBytes();                      /* NEVER USE THIS FUNCTION */

}
```

### 2.1.2 Member Functions.

- Ethernet address

```
public byte[] getEth_addr();
public void setEth_addr(byte[] eth_addr);
```

[get] Ethernet address (6bytes.)
**NOT allowed write (READ-ONLY) by using setEth_addr function.**
**We DO NOT guarantee any damage occurred by using illegal directions and also we DO NOT support any individual use.**
eg. The MAC address 0030f9123456 is stored in eth_addr array like this.
eth_addr[0] = 0x00, eth_addr[1] = 0x30, eth_addr[2] = 0xf9,
eth_addr[3] = 0x12, eth_addr[4] = 0x34, eth_addr[5] = 0x56,

- DHCP

```
public boolean getDhcp();
public void setDhcp(boolean tf);
```

[set/get] Enable or disable DHCP.
If return value is true, dhcp protocol is enabled.
Also set it true for enabling dhcp protocol.

- ARP

```
public boolean getArp();
public void setArp(boolean tf);
```

[set/get] Enable or disable ARP function.
If return value is true, arp protocol is enabled.
Also set it true for enabling arp protocol.
**Please see product user's manual about this function.**

- ezCFG

```
public boolean getEzcfg();
public void setEzcfg(boolean tf);
```

[set/get]
If return value is true, ezCFG probe is enabled.
When this flag is not set, you can probe only in ISP mode.

- PPPoE

```
public boolean getPppoe();
public void setPppoe(boolean tf);
```

[set/get] Enable or disable PPPoE.
If return value is true, pppoe protocol is enabled.
Also set it true for enabling pppoe protocol.

- Power Down Mode

```
public boolean getPwrDown();
public void setPwrDown(boolean tf);
```

[set/get]
This parameter is for Wireless LAN product only.
If return value is true, power down mode is enabled.

- Remote ezCFG

```
public boolean getRemoteConfig();
public void setRemoteConfig(boolean tf);
```

[set/get]
If return value is true, remote configuration is enabled.

This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-200L, EZL-220, EZL-200F |
| Wireless LAN | EZL-300L, EZL-300S,<br>EZL-80 / 80c / 90 / 300W Lite (Firmware v1.3i or higher) |

- Send MAC Address

```
public boolean    getMacId() ;
public void       setMacId(boolean tf);
```

[set/get]
If this value is non zero, a ezTCP sends its MAC address to the remote host right after the connection is established.

This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-70, EZL-200L (Firmware v1.2c or higher) |
| Wireless LAN | |

- Disable TCP Transmission Delay

```
public boolean    getSioNoIdle();
public void       setSioNoIdle(boolean tf);
```

[set/get]
If this value is non zero, a ezTCP sends data from serial port to the network immediately. Because of this, it may cause inefficiency with each TCP header when the data comes frequently.

This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-70, EZL-200L<br>(Firmware v1.2c or higher) |
| Wireless LAN | |

- Drop SIO RX Data

```
public boolean    getSioRxDrop();
public void       setSioRxDrop(boolean tf);
```

[set/get]
If this value is non zero, a ezTCP drops the data which are received from SIO before TCP/IP connection is established. This flag is only considered when a ezTCP running as a TCP client.

This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-70, EZL-200L<br>(Firmware v1.2c or higher) |
| Wireless LAN | |

- Local IP Address

```
public int getLocal_ip();
public void setLocal_ip(int local_ip);
```

[set/get] Local IP address. Network byte order. (big endian)

- Subnet Mask

```
public int getNet_mask();
public void setNet_mask(int net_mask);
```

[set/get] Subnet mask. Network byte order. (big endian)

- Gateway IP Address

```
public int getGate_ip();
public void setGate_ip(int gate_ip);
```

[set/get] Gateway IP address. Network byte order. (big endian)

- Password

```
public byte[] getPasswd();
public void setPasswd(byte[] passwd);
```

[set/get]
If you set a password to a ezTCP then "*password*" parameter need for "write" function. The maximum length is 8bytes.

- Parity Bit

```
public int getParity();
public void setParity(int parity);
```

[set/get] Serial parity bit.

| Parity | Description |
|---|---|
| 0 | None |
| 1 | Even |
| 2 | Odd |
| 3 | Use parity2 parameter. |

The value 3 is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-220, EZL-200F |
| Wireless LAN | EZL-300S |

- Data Bit

```
public int getDatabit();
public void setDatabit(int databit);
```

[set/get] Serial data bit.

- RTS / CTS Flow Control

```
public boolean getRtsCts();
public void setRtsCts(boolean tf);
```

[set/get] If return value is true, flow control is enabled.

- Xon / Xoff Flow Control

```
public boolean getXonOff();
public void setXonOff(boolean tf);
```

[set/get] If return value is true, xonxoff is enabled.

- Telnet

```
public boolean getTelnet();
public void setTelnet(boolean tf);
```

[set/get] If return value is true, telnet connection is enabled for device setting.

This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-100, EZL-110, EZL-200A, EZL-220, EZL-200F |
| Wireless LAN | EZL-300S |

- WLAN Configuration

```
public boolean getWlanCfg();
public void setWlanCfg(boolean tf); /* READ ONLY. Never modify */
```

[get] If return value is true, this product is wireless LAN product.

- Serial Type

```
public int getSerialType();
public void setSerialType(int stype);
```

[set/get] Serial port type.

| Stype | Description |
|---|---|
| 0 | RS-232 |
| 1 | RS-422 |
| 2 | RS-485 |

This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|

| | |
|---|---|
| Wired LAN | EZL-220, EZL-200F |
| Wireless LAN | EZL-300S |

- Stop Bit

```
public int getStopbit();
public void setStopbit(int stopbit);
```

[set/get] Stop bit.

| Stop Bit | Description |
|---|---|
| 0 | 1 Stop Bit |
| 1 | 2 Stop Bit |

This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-220, EZL-200F |
| Wireless LAN | EZL-300S |

- TCP/IP Multiple Connection

```
public boolean getMultipleConnection();
public void setMultipleConnection(boolean tf);
```

[set/get] If telnet flag is true and this flag is true then telnet multiple connection is enabled for device setting. This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-200F |
| Wireless LAN | EZL-300S |

- Parity Bit 2

```
public int getParity2();
public void setParity2(int parity2);
```

[set/get] Serial Parity Bit

| Parity | Description |
|---|---|

| | |
|---|---|
| 0 | Mark |
| 1 | Space |

This parameter is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-220, EZL-200F |
| Wireless LAN | EZL-300S |

- UART Slow Transmission

```
public boolean getSlowTX();
public void setSlowTX(boolean tf);
```

[set/get] UART Slow Transmission
Please refer to EZL-50L user’s manual for more information.

This flag is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-200L, EZL-50M<br>(Firmware v1.1k or higher) |
| Wireless LAN | |

- Serial Baud Rate

```
public int getSio_baud();
public void setSio_baud(int sio_baud);
```

[set/get] Serial baud rate.
ex) sio_baud = 19200

- Timeout

```
public short getTimeout();
public void setTimeout(short timeout);
```

[set/get] Time out value.
In T2S, COD and ATC mode, after time-out value seconds without communication, ezTCP will disconnect the connection automatically. If this value is zero, ezTCP will not disconnect automatically. In the other mode, this value can be set for customization.

- MUX Type

```
public byte getMux_type();
public void setMux_type(byte mux_type);
```

[set/get] ezTCP operating mode.

| MUX Type | Mode | Product Name |
|---|---|---|
| 0 | T2S | Server mode. ezTCP will wait for connection. |
| 1 | ATC | AT command mode. ezTCP can sever or client mode by using AT commands. |
| 2 | COD | Client mode. ezTCP will connect to specified peer IP address and peer port, when amount of water mark data on serial port is arrived. |
| 3 | U2S | UDP mode. ezTCP will use UDP. |

In case of EZL-50L, EZL-50M, EZL-200, EZL-200L, EZL-220, EZL-200F, EZL-300S, EZL-300L, EZL-80, EZL-80c and EZL-90 assigning this value is allowed. However EZL-50 and EZL-60 are set this value by using a HOTFLASH.

- Water Mark

```
public short getWater_mark();
public void setWater_mark(short water_mark);
```

[set/get] Amount of data that can allow to start connection.
This value is only considered in COD or U2S mode.

- Remote IP Address

```
public int getRemote_ip();
public void setRemote_ip(int remote_ip);
```

[set/get] Target IP address. Network byte order. (big endian)
This value is only considered in COD or U2S mode.

- Remote Port Number

```
public int getRemote_port();
public void setRemote_port(int remote_port);
```

[set/get] Target port number.
This value is only considered in COD or U2S mode.

- Local Port Number

```
public int getLocal_port();
public void setLocal_port(int local_port);
```

[set/get] Local IP address.
This value is only considered in T2S mode.

- PPPoE User ID

```
public byte[] getPoe_uid();
public void setPoe_uid(byte[] poe_uid);
```

[set/get] PPPoE log-in ID.
The maximum length is 8bytes.

- PPPoE Password

```
public byte[] getPoe_pwd(); /* PPPoE password. Maximum 8bytes */
public void setPoe_pwd(byte[] poe_pwd);
```

[set/get] PPPoE log-in password.
The maximum length is 8bytes.

### 2.1.3 Remarks

- **DO NOT use not allowed setXXX_XXXX functions.**

## 2.2 ezcfglig.df.Wlan_Env

### 2.2.1 Overview

Wlan_Env structure is for wireless LAN products.

| LAN type | Product Name |
|---|---|
| Wireless LAN | EZL-80 / 80c / 90 / 300W Lite / 300L |

Basic class for function probe, write and status that are used in ezCfg library.

```
public class Wlan_Env {

public int        getCCType();
public void       setCCType(int cctype);

public int        getChannel();
public void       setChannel(int channel);

public int        getWep();
public void       setWep(int wep);

public int        getWepId();
public void       setWepId(int wep_id);

public boolean    getStartAdhoc();
public void       setStartAdhoc(boolean tf);

public boolean    getStartIfs();
public void       setStartIfs(boolean tf);
public int        getAuth();
public void       setAuth(int auth);

public byte[]     getNew_ssid();                      /* SSID. Maximum 32bytes*/
public void       setNew_ssid(byte[] new_ssid);

public byte[]     getTarget_ssid();                   /* SSID. Maximum 32bytes*/
public void       setTarget_ssid(byte[] target_ssid);

public byte[][]   getKey128();                        /* 4-sets of 128bits key. key128[4][14] */
public void       setKey128(byte[][] key128);

public byte[][]   getKey64();                         /* 4-sets of 64bits key. key64[4][6] */
public void       setKey64(byte[][] key64);
public void       fromBytes(byte[] buff);             /* NEVER USE THIS FUNCTION */
public byte[]     toBytes();                          /* NEVER USE THIS FUNCTION */

}
```

### 2.2.2 Member Functions.

- Connection Control Type

```
public int getCCType();
public void setCCType(int cctype);
```

[set/get] Connection Control Type (0 - adhoc, 1 - infrastructure)

- Channel

```
public int getChannel();
public void setChannel(int channel);
```

[set/get] Wireless LAN channel

- WEP

```
public int getWep();
public void setWep(int wep);
```

[set/get] WEP(Wired Equivalent Privacy) setting (0-no WEP, 1 - 64bit, 2 - 128 bit)

- WEP Key ID Index

```
public int getWepId();
public void setWepId(int wep_id);
```

[set/get] WEP index number (0, 1, 2, 3)

- AD-HOC Start-Up Option

```
public boolean getStartAdhoc();
public void setStartAdhoc(boolean tf);
```

[set/get] if this flag is set, ezTCP connects to default adhoc network during several seconds after booting. See below figure.

- Infrastructure Start-Up Option

```
public boolean getStartIfs();
public void setStartIfs(boolean tf);
```

[set/get] if this flag is set, ezTCP connects to default infrastructure network during several seconds after booting. See below figure.

- Authentication Mode

```
public int getAuth();
public void setAuth(int auth);
```

[set/get] Authentication mode for infrastructure network.
( 0 - Open system, 1 - Shared key )

- Infrastructure SSID

```
public byte[] getTarget_ssid();
public void setTarget_ssid(byte[] target_ssid);
```

[set/get] SSID for infrastructure network.
The maximum length is 32bytes.

- AD-HOC SSID

```
public byte[] getNew_ssid();
public void setNew_ssid(byte[] new_ssid);
```

[set/get] SSID for adhoc network.
The maximum length is 32bytes.

- WEP Key 64bits

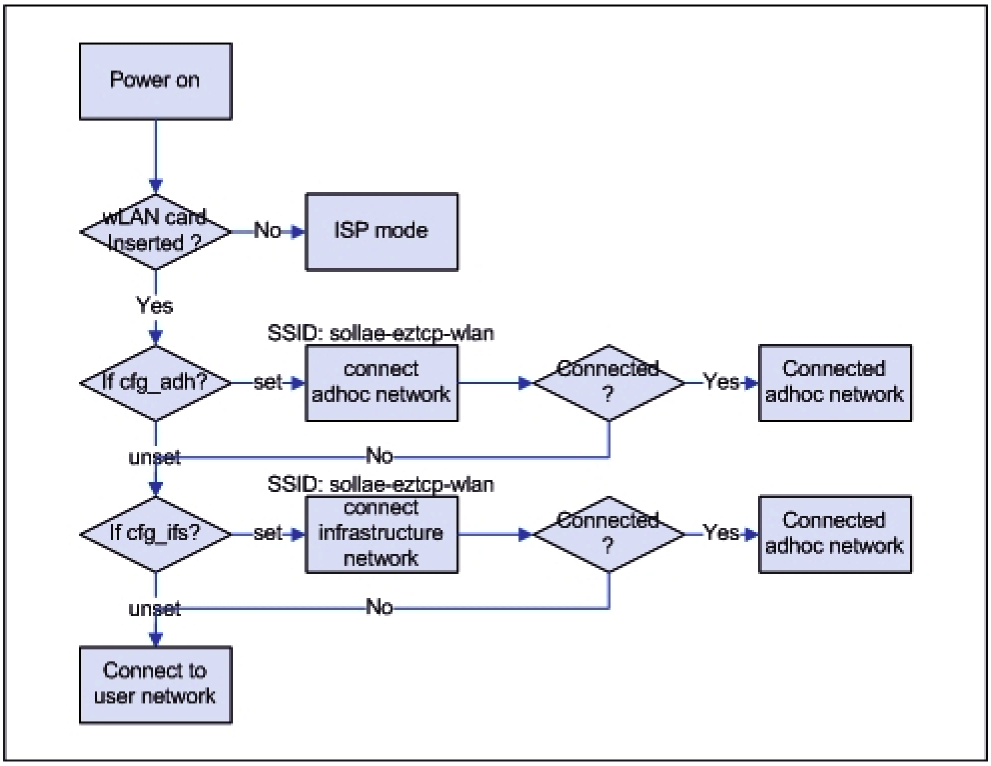
```
public byte[][] getKey64();
public void setKey64(byte[][] key64);
```

[set/get] 4-sets of 64 bits key value. This is 2-dimension array.
The length is 4x6. The maximum length of 64 bits key is 5bytes.

- WEP Key 128bits

```
public byte[][] getKey128(); /* 4-sets of 128bits key. key128[4][14] */
public void setKey128(byte[][] key128);
```

[set/get] 4-sets of 128 bits key value. This is 2-dimension array.
The length is 4x14. The maximum length of 128 bits key is 13bytes.



## 2.3 ezcfglib.df.Etc_Opt

### 2.3.1 Overview

ezcfglib.df.Etc_Opt class is for ezTCP products, which are supports etcetera options. Currently, only use comment option.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-200L, EZL-220, EZL-200F |
| Wireless LAN | EZL-300L, EZL-300S,<br>EZL-80 / 80c / 90 / 300W Lite (Firmware v1.3i or higher) |

```
public class Etc_Opt {

public byte[]       getComment();
public void         setComment(byte[] comment);

public void         fromBytes(byte[] buff);           /* NEVER USE THIS FUNCTION */
public byte[]       toBytes();                        /* NEVER USE THIS FUNCTION */

}
```

### 2.3.2 Member Functions.

- Comments

```
public byte[] getComment();
public void setComment(byte[] comment);
```

[set/get] When use multiple ezTCP, the comments option helps you distinguish each ezTCP.

## 2.4 ezcfglib.df.Stat_Env

### 2.4.1 Overview

Basic status class has status value from function status that are used in EzCfg library.

```
public class Stat_Env{

public long         getEth_rx();
public long         getEth_tx();
public String       getFreq();
public String       getGate_addr();
public String       getIp_addr();
public String       getMac_addr();
public long         getSio_rx();
public long         getSio_tx();
public String       getSub_mask();
public String       getText();
public String       getUptime();
public String       getVer();
public void         fromBytes(byte []buf);          /* NEVER USE THIS FUNCTION */

}
```

### 2.4.2 Member Functions.

- Firmware Version

```
public String getVer();
```

[get] Firmware version.

- Frequency

```
public String getFreq();
```

[get] Frequency of ezTCP.

- Ethernet Address

```
public String getMac_addr();
```

[get] MAC address.

- Local IP Address

```
public String getIp_addr();
```

[get] Local IP address.

- Subnet Mask

```
public String getSub_mask();
```

[get] Subnet mask.

- Gateway IP Address

```
public String getGate_addr();
```

[get] Gateway IP address.

- Up-Time

```
public String getUptime();
```

[get] ezTCP's alive days and time since the last boot up of ezCFG.
*WriteEzTCP* function will make EzTCP reboot.

- Serial Receive Bytes

```
public long getSio_rx();
```

[get] Received bytes from serial connection.

- Serial Transmission Bytes

```
public long getSio_tx();
```

[get] Transmitted bytes to serial connection.

- Ethernet Receive Bytes

```
public long getEth_rx();
```

[get] Received packets from ethernet connection.

- Ethernet Transmission Bytes

```
public long getEth_tx();
```

[get] Transmitted packets to ethernet connection.

- Additional Information

```
public String getText();
```

[get] Additional information about ezTCP.

### 2.4.3 Remarks

## 2.5 ezcfglib.df.Res_Env

### 2.5.1 Overview

Basic class of ezCfg library which the number of founded ezTCP and error number will be stored in.

```
public class Res_Env {

public int      getErr();
public void     setErr(int err);                    /* NEVER USE THIS FUNCTION */

public int      getResultCount();
public void     setResultCount(int resultCount);    /* NEVER USE THIS FUNCTION */

}
```

### 2.5.2 Member Functions.

- Error Number

```
public int getErr();
```

[get] Error number.

| nErrNum | Description |
|---|---|
| EZTCP_ERR_PWD | If you seted a password to ezTCP, then you should set a password by using Spe_Env.setPassword(byte[] password) function. |
| EZTCP_ERR_RES | When error occurred during execute library functions, EZTCP_ERR_RES is stored in nErrNum. |
| EZTCP_NO_ERR | No error occurred. |

- Result Count

```
public int getResultCount();
```

[get] Founded ezTCP number.

# 3 ezcfglib.lib.EzcfgLib

## 3.1 Overview

Main class of ezConfig library. It has some functions help you make a utility like ezConfig program.

```
public class EzcfgLib {
public String GetLibVerEzTCP();

public void ProbeEzTCP( Spe_Env eztcpenv[],
                        Wlan_Env eztcpwenv[],
                        Etc_Opt eztcpopt[],
                        Res_Env res_env) throws Exception

public void RemoteReadEzTCP(InetAddress addr,
                            Spe_Env eztcpenv[],
                            Wlan_Env eztcpwenv[],
                            Etc_Opt eztcpopt[],
                            Res_Env res_env) throws Exception

public void StatusEzTCP( Spe_Env eztcpenv,
                         Stat_Env eztcpstat,
                         Res_Env res_env) throws Exception

public void RemoteStatusEzTCP( InetAddress addr,
                               Spe_Env eztcpenv,
                               Stat_Env eztcpstat,
                               Res_Env res_env) throws Exception

public void WriteEzTCP( Spe_Env eztcpenv,
                        Wlan_Env eztcpwenv,
                        Etc_Opt eztcpopt,
                        Res_Env res_env) throws Exception

public void RemoteWriteEzTCP( InetAddress addr,
                              Spe_Env eztcpenv,
                              Wlan_Env eztcpwenv,
                              Etc_Opt eztcpopt,
                              Res_Env res_env) throws Exception

public void ChangePwdEzTCP( Spe_Env eztcpenv,
                            String ChangePwd,
                            Res_Env res_env) throws Exception

public void RemoteChangePwdEzTCP( InetAddress addr,
                                  Spe_Env eztcpenv,
                                  String ChangePwd,
                                  Res_Env res_env) throws Exception
}
```

## 3.2 Member Functions.

### 3.2.1 ProbeEzTCP

- Overview
  The ProbeEzTCP function can find ezTCP in local network. Each ezTCP can discriminate by MAC address.
- Prototype

```
public void ProbeEzTCP( Spe_Env eztcpenv[],
                        Wlan_Env eztcpwenv[],
                        Etc_Opt eztcpopt[],
                        Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *eztcpenv[]* | [out] Array of Spe_Env class which ezTCP environment values will be stored in. |
| *eztcpwenv[]* | [out] Array of Wlan_Env class which ezTCP wireless environment values will be stored in. |
| *eztcpopt[]* | [out] Array of Etc_Opt class which ezTCP etcetera option values will be stored in. |
| *res_env* | [out] Instance of Res_Env class which the number of founded ezTCP and error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks
  Before using this function, you have to reserve enough space for *eztcpenv, eztcpwenv, eztcpopt* structure.
  ***eg. Spe_Env env_base[]       = new Spe_Env[256];***
  ***Wlan_Env wenv_base[]   = new Wlan_Env[256];***
  ***Etc_Opt etc_base[]         = new Etc_Opt[256];***
  Above example shows that env_base, wenv_base and etc_base can store 256 ezTCP environment. It means the maximum count of probe result is 256.

  **This function take at least 2 seconds to complete running and may take longer if the local network has a lot of ezTCP.**

### 3.2.2 WriteEzTCP

- Overview
  Write environment value to ezTCP specified MAC address in *eztcpenv*.

- Prototype

```
public void WriteEzTCP( Spe_Env eztcpenv,
                        Wlan_Env eztcpwenv,
                        Etc_Opt eztcpopt,
                        Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *eztcpenv* | [in] Instance of Spe_Env class containing environment values to be written. |
| *eztcpwenv* | [in] Instance of Wlan_Env class containing wireless environment values to be written. |
| *eztcpopt* | [in] Instance of Etc_Opt class containing etcetera option values to be written. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks
  Before using this function, you have to call *ProbeEzTCP* function for reliable data transmitting.

  **This function take at least 3 seconds to complete running and may take longer if the local network has a lot of ezTCP.**

### 3.2.3 StatusEzTCP

- Overview
  Read status value from ezTCP specified MAC address in *eztcpenv*.
- Prototype

```
public void StatusEzTCP( Spe_Env eztcpenv,
                         Stat_Env eztcpstat,
                         Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *eztcpenv* | [in] Instance of Spe_Env class which has target MAC address. |

| | |
|---|---|
| *eztcpstat* | [out] Instance of Stat_Env class which will store ezTCP's status values. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks
  Before using this function, you have to call *ProbeEzTCP* function for reliable data transmitting.

  **This function take at least 2 seconds to complete running and may take longer if the local network has a lot of ezTCP.**

### 3.2.4 ChangePwdEzTCP

- Overview
  Change or erase password of ezTCP specified MAC address in *eztcpenv*.
- Prototype

```
public void ChangePwdEzTCP( Spe_Env eztcpenv,
                           String ChangePwd,
                          Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *eztcpenv* | [in] Instance of Spe_Env class which has target MAC address. |
| *ChangePwd* | [in] String value of password. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks
  Before using this function, you have to call *ProbeEzTCP* function for reliable data transmitting.

  **This function take at least 3 seconds to complete running and may take longer if**

**the local network has a lot of ezTCP.**

### 3.2.5 RemoteReadEzTCP

- Overview

  The RemoteReadEzTCP function can find ezTCP in local or remote network. Each ezTCP can discriminate by IP address.
  This function is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-70, EZL-200L, EZL-220, EZL-200F |
| Wireless LAN | EZL-300L, EZL-300S,<br>EZL-80 / 80c / 90 / 300W Lite (Firmware v1.3i or higher) |

- Prototype

```
public void RemoteReadEzTCP(InetAddress addr,
                            Spe_Env eztcpenv[],
                            Wlan_Env eztcpwenv[],
                            Etc_Opt eztcpopt[],
                           Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *addr* | [in] The ezTCP's local ip address to read environment values. |
| *eztcpenv[]* | [out] Array of Spe_Env class which ezTCP environment values will be stored in. |
| *eztcpwenv[]* | [out] Array of Wlan_Env class which ezTCP wireless environment values will be stored in. |
| *eztcpopt[]* | [out] Array of Etc_Opt class which ezTCP etcetera option values will be stored in. |
| *res_env* | [out] Instance of Res_Env class which the number of founded ezTCP and error number will be stored in. |

- Return Value
- Throws

  Exception - if may have any exception.

- Remarks

  Before using this function, you have to reserve enough space for *eztcpenv*, *eztcpwenv*, *eztcpopt* structure.
  ***eg. Spe_Env env_base[]  = new Spe_Env[256];***

```
Wlan_Env wenv_base[]  = new Wlan_Env[256];
Etc_Opt etc_base[]    = new Etc_Opt[256];
```

Above example shows that env_base, wenv_base and etc_base can store 256 ezTCP environment. It means the maximum count of probe result is 256.

**This function take at least 2 seconds to complete running and may take longer if the local network has a lot of ezTCP.**

### 3.2.6 RemoteWriteEzTCP

- Overview

  Write environment value to ezTCP specified IP address. This function is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-70, EZL-200L, EZL-220, EZL-200F |
| Wireless LAN | EZL-300L, EZL-300S,<br>EZL-80 / 80c / 90 / 300W Lite (Firmware v1.3i or higher) |

- Prototype

```
public void RemoteWriteEzTCP( InetAddress addr,
                              Spe_Env eztcpenv,
                              Wlan_Env eztcpwenv,
                              Etc_Opt eztcpopt,
                              Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *addr* | [in] The ezTCP's local ip address to read environment values. |
| *eztcpenv* | [in] Instance of Spe_Env class containing environment values to be written. |
| *eztcpwenv* | [in] Instance of Wlan_Env class containing wireless environment values to be written. |
| *eztcpopt* | [in] Instance of Etc_Opt class containing etcetera option values to be written. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws

Exception - if may have any exception.

- Remarks
  Before using this function, you have to call *ProbeEzTCP* function for reliable data transmitting.

  **This function take at least 3 seconds to complete running and may take longer if the local network has a lot of ezTCP.**

### 3.2.7 RemoteStatusEzTCP

- Overview
  Read status value from ezTCP using specified IP address. This function is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-70, EZL-200L, EZL-220, EZL-200F |
| Wireless LAN | EZL-300L, EZL-300S,<br>EZL-80 / 80c / 90 / 300W Lite (Firmware v1.3i or higher) |

- Prototype

```
public void RemoteStatusEzTCP( InetAddress addr,
                               Spe_Env eztcpenv,
                               Stat_Env eztcpstat,
                               Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *addr* | [in] The ezTCP's local ip address to read environment values. |
| *eztcpenv* | [in] Instance of Spe_Env class which has target MAC address. |
| *eztcpstat* | [out] Instance of Stat_Env class which will store ezTCP's status values. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks
  Before using this function, you have to call *ProbeEzTCP* function for reliable data

transmitting.

**This function take at least 2 seconds to complete running and may take longer if the local network has a lot of ezTCP.**

### 3.2.8 RemoteChangePwdEzTCP

- Overview
  Change or erase password of ezTCP using specified IP address. This function is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-70, EZL-200L, EZL-220, EZL-200F |
| Wireless LAN | EZL-300L, EZL-300S,<br>EZL-80 / 80c / 90 / 300W Lite (Firmware v1.3i or higher) |

- Prototype

```
public void RemoteChangePwdEzTCP( InetAddress addr,
                                  Spe_Env eztcpenv,
                                  String ChangePwd,
                                  Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *addr* | [in] The ezTCP's local ip address to read environment values. |
| *eztcpenv* | [in] Instance of Spe_Env class which has target MAC address. |
| *ChangePwd* | [in] String value of password. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks
  Before using this function, you have to call *ProbeEzTCP* function for reliable data transmitting.

**This function take at least 3 seconds to complete running and may take longer if**

**the local network has a lot of ezTCP.**

### 3.2.9 GetLibVerEzTCP

- Overview
  Get library version information.
- Prototype

```
public String GetLibVerEzTCP();
```

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks

### 3.2.10 CloseTCP

- Overview
  The CloseTCp function can terminate a TCP/IP connection of specific ezTCP in network.
  Before using this function you have to set a password to ezTCP.

  This function is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-200L, EZL-70<br>(Firmware v1.1k or higher) |
| Wireless LAN | |

- Prototype

```
public void CloseTCP(Spe_Env eztcpenv,
                     String Pwd,
                     Res_Env res_env
                     ) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *eztcpenv* | [in] Instance of Spe_Env class which has target MAC address. |
| *Pwd* | [in] String value of password. |

| | |
|---|---|
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks

### 3.2.11 ResetEzTCP

- Overview
  The ResetEzTCP function can reset specific ezTCP in network.
  Before using this function you have to set a password to ezTCP.

  This function is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-200L, EZL-70<br>(Firmware v1.1k or higher) |
| Wireless LAN | |

- Prototype

```
public void ResetEzTCP(Spe_Env eztcpenv,
                        String Pwd,
                        Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *eztcpenv* | [in] Instance of Spe_Env class which has target MAC address. |
| *Pwd* | [in] String value of password. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks

### 3.2.12 RemoteCloseTCP

- Overview

  The CloseTCp function can terminate a TCP/IP connection of specific ezTCP in network.
  Before using this function you have to set a password to ezTCP.

  This function is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-200L, EZL-70<br>(Firmware v1.1k or higher) |
| Wireless LAN | |

- Prototype

```
public void RemoteCloseTCP(InetAddress addr,
                           Spe_Env eztcpenv,
                           String Pwd,
                           Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *addr* | [in] The ezTCP’s local ip address to read environment values. |
| *eztcpenv* | [in] Instance of Spe_Env class which has target MAC address. |
| *Pwd* | [in] String value of password. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws

  Exception - if may have any exception.
- Remarks

### 3.2.13 RemoteResetEzTCP

- Overview

  The ResetEzTCP function can reset specific ezTCP in network.
  Before using this function you have to set a password to ezTCP.

  This function is currently considered in below ezTCP products.

| LAN type | Product Name |
|---|---|
| Wired LAN | EZL-50L, EZL-50M, EZL-200L, EZL-70<br>(Firmware v1.1k or higher) |
| Wireless LAN | |

- Prototype

```
public void RemoteResetEzTCP(InetAddress addr,
                             Spe_Env eztcpenv,
                             String Pwd,
                             Res_Env res_env) throws Exception
```

- Parameters

| Parameter | Description |
|---|---|
| *addr* | [in] The ezTCP's local ip address to read environment values. |
| *eztcpenv* | [in] Instance of Spe_Env class which has target MAC address. |
| *Pwd* | [in] String value of password. |
| *res_env* | [out] Instance of Res_Env class which an error number will be stored in. |

- Return Value
- Throws
  Exception - if may have any exception.
- Remarks