

XPlusUdpComm 가

Last update: 2023/05/11

www.softbase.co.kr
Copyright © SOFTBase Inc. All rights reserved.

XPlusUdpComm 가 1
..... 1
..... 1

XPlusUdpComm 가

XPlusUdpComm.ocx

XPlusUdpComm.ocx

UDP /

: /XPLUS/xplus_udpcomm

- [xplus_udpcomm.xml](#)
- [xplus_udpcomm.js](#)
- [xplus_udpcomm.ocx](#)

```
function btnInitXPlusUdpSvrComm_on_mouseup(objInst)
{
    var nRet;
    var strLogDir = "C:\\xFrame\\log";

    /**
     * XPlusUdpComm
     * @param pLogDir
     * @param pLogFilePrefix
     * @param pLogLevel ("DEBUG"<"INFO"<"WARN"<"ERROR")
     * ("DEBUG" 가 )
     * @param nLogEncFlag
     * @return
     * 0 : OK
     * 9 : Already Initialized
     */
    nRet = objXPlusUdpSvrComm.innerctrl.InitXPlusUdpComm(strLogDir,
    "XPlusUdpSvrComm", "DEBUG", 0);
    factory.consoleprint("InitXPlusUdpComm Return Value = " + nRet);
    if (nRet != 0) {
        screen.alert("InitXPlusUdpComm Fail, Error Code = " + nRet);
    }

    /**
     * UTF8
     * @param nUtf8Flag UTF8
     * @return
     */
}
```

```
*      0 : OK
*      9 : Not Initialized
*/
objXPlusUdpSvrComm.innerctrl.SetUtf8Flag(1);
}

function btnInitXPlusUdpCltComm_on_mouseup(objInst)
{
    var nRet;
    var strLogDir = "C:\\xFrame\\log";

    nRet = objXPlusUdpCltComm.innerctrl.InitXPlusUdpComm(strLogDir,
"XPlusUdpCltComm", "DEBUG", 0);
    factory.consoleprint("InitXPlusUdpComm Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("InitXPlusUdpComm Fail, Error Code = " + nRet);
    }

    objXPlusUdpCltComm.innerctrl.SetUtf8Flag(1);
}

function btnOpenUdpSvrSocket_on_mouseup(objInst)
{
    var nRet;

    /**
     * XPlusUdpComm      UDP
     * @param nBindPortNo      가 0      , 0
     * @return
     *      0 : OK
     *      1 : Can't Create UDP Socket
     *      2 : Can't Attach Socket
     *      3 : Can't Bind Socket
     *      4 : Can't Get Local UDP Port No
     *      9 : Not Initialized
     */
    nRet =
objXPlusUdpSvrComm.innerctrl.OpenSocket(fldUdpSvrPortNo.gettext());
    factory.consoleprint("OpenSocket Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("OpenSocket Fail, Error Code = " + nRet);
    }
}

function btnOpenUdpCltSocket_on_mouseup(objInst)
{
    var nRet;

    nRet =
objXPlusUdpCltComm.innerctrl.OpenSocket(fldUdpCltPortNo.gettext());
```

```
factory.consoleprint("OpenSocket Return Value = " + nRet);
if(nRet != 0) {
    screen.alert("OpenSocket Fail, Error Code = " + nRet);
    return;
}

/**
 * XPlusUdpComm      UDP
 * OpenSocket API    nBindPortNo      0
 * @return
 * 0 :
 * >0 :      UDP
 */
fldUdpCltPortNo.setText(objXPlusUdpCltComm.innerctrl.GetPortNo());

fldReceiverPortNo1.setText(fldUdpCltPortNo.getText());
}

function btnCloseUdpSvrSocket_on_mouseup(objInst)
{
    /**
     * XPlusUdpComm      UDP
     */
    objXPlusUdpCltComm.innerctrl.CloseSocket();
}

function btnCloseUdpCltSocket_on_mouseup(objInst)
{
    objXPlusUdpCltComm.innerctrl.CloseSocket();
}

function btnSendData1_on_mouseup(objInst)
{
    var nRet;

    /**
     * XPlusUdpComm      UDP
     * @param pRemoteIpAddr      IP
     * @param nRemotePortNo      UDP
     * @param pData
     * @return
     * 0 : Success
     * 1 : Invalid Socket
     * 2 : Invalid Port No
     * 3 : Invalid Address
     * 5 : Data Length is 0
     * 5 : Send Fail
     * 9 : Not Initialized
     */
    nRet =
objXPlusUdpSvrComm.innerctrl.SendData(fldReceiverIpAddr1.getText(),
```

```
fldReceiverPortNo1.getText(), fldSendData1.getText());
    if (nRet != 0) {
        screen.alert("SendData Fail, Error Code = " + nRet);
        return;
    }
}

function btnSendData2_on_mouseup(objInst)
{
    var nRet;

    nRet =
objXPlusUdpCltComm.innerctrl.SendData(fldReceiverIpAddr2.getText(),
fldReceiverPortNo2.getText(), fldSendData2.getText());
    if (nRet != 0) {
        screen.alert("SendData Fail, Error Code = " + nRet);
        return;
    }
}

/**
 *
 * @param objInst XPlusUdpComm
 * @param nLength
 * @param pDataKey          Key , GetData
 * @param pRemoteIpAddr    IP
 * @param nRemotePortNo    UDP
 */
function
objXPlusUdpSvrComm_OnReceive(objInst, nLength, pDataKey, pRemoteIpAddr, nRemoteP
ortNo)
{
    factory.consoleprint("nLength = " + nLength);
    factory.consoleprint("pDataKey = " + pDataKey);

    fldRecvDataLength1.setText(nLength);
    fldSenderIdAddr1.setText(pRemoteIpAddr);
    fldSenderIdPortNo1.setText(nRemotePortNo);

    /**
     *
     * @return strRecvData
     */
    var strRecvData = objXPlusUdpSvrComm.innerctrl.GetData(pDataKey);
    factory.consoleprint("RecvData = " + strRecvData);
    fldRecvData1.setText(strRecvData);
}

function
objXPlusUdpCltComm_OnReceive(objInst, nLength, pDataKey, pRemoteIpAddr, nRemoteP
ortNo)
```

```
{
    factory.consoleprint("nLength = " + nLength);
    factory.consoleprint("pDataKey = " + pDataKey);

    fldRecvDataLength2.setText(nLength);
    fldSenderIdAddr2.setText(pRemoteIpAddr);
    fldSenderPortNo2.setText(nRemotePortNo);

    var strRecvData = objXPlusUdpCltComm.innerctrl.GetData(pDataKey);
    factory.consoleprint("RecvData = " + strRecvData);
    fldRecvData2.setText(strRecvData);
}

function screen_on_load()
{
    fldSendData1.setText("1234ABCD가      1");
    fldSendData2.setText("1234ABCD가      2");
}

function btnClearData1_on_mouseup(objInst)
{
    fldRecvDataLength1.setText("");
    fldSenderIdAddr1.setText("");
    fldSenderPortNo1.setText("");
    fldRecvData1.setText("");
}

function btnClearData2_on_mouseup(objInst)
{
    fldRecvDataLength2.setText("");
    fldSenderIdAddr2.setText("");
    fldSenderPortNo2.setText("");
    fldRecvData2.setText("");
}
```

From:

<https://technet.softbase.co.kr/wiki/> - **xFrame5 TechNet**

Permanent link:

https://technet.softbase.co.kr/wiki/guide/xplus/xplus_udpcomm



Last update: **2023/05/11 16:21**