

XCItComm 가

Last update: 2023/05/11

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

XCitComm 가 1
..... 1
..... 1

XCltComm 가

XPlusTcpComm.ocx

XPlusTcpComm.ocx

TCP

: /XPLUS/xplus_tcpcomm

- [xplus_tcpcomm.xml](#)
- [xplus_tcpcomm.js](#)
- [xplus_tcpcomm.js](#)

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

    /**
     * XCltComm
     * @param nLengthFieldLength
     * @param nLenIncFlag (0 1)
     * @param pLogDir
     * @param pLogFilePrefix
     * @param pLogLevel ("DEBUG"<"INFO"<"WARN"<"ERROR")
     * ("DEBUG" 가 )
     * @param nLogEncFlag
     * @return
     * 0 : Success
     * 1 : Invalid Length Field Length
     * 2 : Fail to Allocate Memory For Length Field
     */
    nRet = objXPlusTcpComm.innerctrl.InitXPlusTcpComm(strLogDir,
    "XPlusTcpComm", "DEBUG", 0, "127.0.0.1", fldTcpSvrPortNo.GetText(), 8, 0);
    factory.consoleprint("objXCltComm Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("objXCltComm Fail, Error Code = " + nRet);
    }
}
}
```

```
//
function btnConnectSession_on_mouseup(objInst)
{
    var strTcpSvrIpAddr = fldTcpSvrIpAddr.gettext();
    var nTcpSvrPortNo = fldTcpSvrPortNo.gettext();
    var nConnectTimeout = fldConnectTimeout.gettext();

    /**
     * TCP
     * @param strTcpSvrIpAddr TCP IP
     * @param nTcpSvrPortNo TCP
     * @param nConnectTimeout ( : ), 0

     * @return
     * 0 : OK
     * 1 : Fail To Connect
     * 2 : Fail to Select
     * 3 : Connect Timeout
     * 9 : Not Initialized
     */
    var nRet = objXPlusTcpComm.innerctrl.ConnectSession();
    factory.consoleprint("ConnectSession() Return = " + nRet);
    if(nRet != 0) {
        screen.alert("ConnectSession() Fail, Error Code = " + nRet);
    }
}

// TCP
function btnCloseSession_on_mouseup(objInst)
{
    /**
     * TCP
     * @return
     * 0 : OK
     * 9 : Not Initialized
     */
    var nRet = objXPlusTcpComm.innerctrl.CloseSession();
    factory.consoleprint("CloseSession() Return = " + nRet);
    if(nRet != 0) {
        screen.alert("CloseSession() Fail, Error Code = " + nRet);
    }
}

//
function btnSendData_on_mouseup(objInst)
{
    var strSendData = fldSendData.gettext();

    /**
     * TCP
     * @param strSendData
```

```
* @return
*   0 : Success
*   1 : Invalid Parameter
*   2 : Session is Not Connected
*   3 : Fail to Send Length Field
*   4 : Fail to Send Data Field
*   9 : Not Initialized
*/
var nRet = objXPlusTcpComm.innerctrl.SendData(strSendData);
factory.consoleprint("SendData() Return = " + nRet);
if(nRet != 0) {
    screen.alert("SendData Fail, Error Code = " + nRet);
    var strErrorMsg = objXPlusTcpComm.innerctrl.GetErrorMsg();
    factory.consoleprint("SendData Fail, Error Msg = " + strErrorMsg);
}
}

function btnSendDataAndClose_on_mouseup(objInst)
{
    var strSendData = fldSendData.gettext();

    /**
    *   TCP
    * @param strSendData
    * @return
    *   0 : Success
    *   1 : Invalid Parameter
    *   2 : Session is Not Connected
    *   3 : Fail to Send Length Field
    *   4 : Fail to Send Data Field
    *   9 : Not Initialized
    */
    var nRet = objXPlusTcpComm.innerctrl.SendData(strSendData);
    factory.consoleprint("SendData() Return = " + nRet);
    if(nRet != 0) {
        screen.alert("SendData Fail, Error Code = " + nRet);
        var strErrorMsg = objXPlusTcpComm.innerctrl.GetErrorMsg();
        factory.consoleprint("SendData Fail, Error Msg = " + strErrorMsg);
    }

    objXPlusTcpComm.innerctrl.CloseSession();
}

//
function btnClearData_on_mouseup(objInst)
{
    fldRecvDataLength.settext("");
    fldRecvData.settext("");
}
}
```

```
//
function btnSetSendTimeout_on_mouseup(objInst)
{
    var nSendTimeout = fldSendTimeout.gettext();

    /**
     * TCP
     * ( 10 )
     * @param nSendTimeout ( : )
     * @return
     * 0 : OK
     * 1 : Invalid Timeout
     * 9 : Not Initialized
     */
    var nRet = objXPlusTcpComm.innerctrl.SetSendTimeout(nSendTimeout);
    factory.consoleprint("SetSendTimeout Return = " + nRet);
    if(nRet != 0) {
        screen.alert("SetSendTimeout Fail, Error Code = " + nRet);
    }
}

//
function btnSetLogFlag_on_mouseup(objInst)
{
    var nLogFlag = fldLogFlag.gettext();

    /**
     * @param nLogFlag (1: , 0: )
     * @return
     * 0 : OK
     * 9 : Not Initialized
     */
    var nRet = objXCltComm.innerctrl.SetLogFlag(nLogFlag);
    factory.consoleprint("SetLogFlag Return = " + nRet);
    if(nRet != 0) {
        screen.alert("SetLogFlag Fail, Error Code = " + nRet);
    }
}

//
function btnIsConnected_on_mouseup(objInst)
{
    /**
     * @return
     * 0 :
     * 1 :
     */
    var nRet = objXPlusTcpComm.innerctrl.IsConnected();
    fldIsConnected.settext(nRet);
}
}
```

```
/**
 * TCP
 * CloseSession 가
 * @param objInst XPlusTcpComm
 * @param nEvent
 * @param pMsg
 */
function objXPlusTcpComm_OnClose(objInst,nEvent,pMsg)
{
    screen.alert("objXPlusTcpComm_OnClose> nEvent = " + nEvent + ", " +
pMsg);
}

/**
 *
 * @param objInst XPlusTcpComm
 * @param nLength
 * @param pDataKey Key , GetData
 */
function objXPlusTcpComm_OnReceive(objInst,nLength,pDataKey)
{
    factory.consoleprint("nLength = " + nLength);
    factory.consoleprint("pDataKey = " + pDataKey);

    fldRecvDataLength.setText(nLength);

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

    return;
}
```

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

Permanent link:
https://technet.softbase.co.kr/wiki/guide/xplus/xplus_tcpcomm



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