

XFrameBrowser MainScreen 가

Last update: 2023/05/11

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

XFrameBrowser MainScreen 가 1
..... 1
..... 1

XFrameBrowser MainScreen 가

XPlusTcpSvr.ocx

XPlusTcpSvr.ocx

TCP

: /XPLUS/xplus_tcpsvr

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

```
/**
 * XPlusTcpSvr
 * @param objInst XPlusTcpSvr
 * @param pRemoteIpAddr          IP
 * @param nRemotePortNo         TCP
 * @param nSocketKey
 * @param nLength
 * @param pDataKey              Key , GetData
 */
function
objXPlusTcpSvr_OnReceive(objInst, pRemoteIpAddr, nRemotePortNo, nSocketKey, nLength, pDataKey)
{
    factory.consoleprint("pRemoteIpAddr = " + pRemoteIpAddr);
    factory.consoleprint("nRemotePortNo = " + nRemotePortNo);
    factory.consoleprint("nSocketKey = " + nSocketKey);
    factory.consoleprint("nLength = " + nLength);
    factory.consoleprint("pDataKey = " + pDataKey);

    fldRecvIpAddr.setText(pRemoteIpAddr);
    fldRecvPortNo.setText(nRemotePortNo);
    fldRecvDataLength.setText(nLength);

    /**
 * @return strRecvData
 */
```

```
var strRecvData = objXPlusTcpSvr.innerctrl.GetData(pDataKey);
factory.consoleprint("RecvData = " + strRecvData);
fldRecvData.settext(strRecvData);

return;
}

/**
 * TCP
 * @param objInst XPlusTcpSvr
 * @param pRemoteIpAddr      IP
 * @param nRemotePortNo     TCP
 * @param nSocketKey
 */
function
objXPlusTcpSvr_OnConnect(objInst, pRemoteIpAddr, nRemotePortNo, nSocketKey)
{
    factory.consoleprint("SessionConnected> pRemoteIpAddr = " +
pRemoteIpAddr);
    factory.consoleprint("SessionConnected> nRemotePortNo = " +
nRemotePortNo);
    factory.consoleprint("SessionConnected> nSocketKey = " + nSocketKey);

    var nRow = grdSessionList.additem();
    grdSessionList.setitemtext(nRow, 0, pRemoteIpAddr);
    grdSessionList.setitemtext(nRow, 1, nRemotePortNo);
}

/**
 * TCP
 * @param objInst XPlusTcpSvr
 * @param pRemoteIpAddr      IP
 * @param nRemotePortNo     TCP
 * @param nSocketKey
 */
function
objXPlusTcpSvr_OnClose(objInst, pRemoteIpAddr, nRemotePortNo, nSocketKey)
{
    factory.consoleprint("SessionClosed> pRemoteIpAddr = " + pRemoteIpAddr);
    factory.consoleprint("SessionClosed> nRemotePortNo = " + nRemotePortNo);
    factory.consoleprint("SessionClosed> nSocketKey = " + nSocketKey);

    var nRowCount = grdSessionList.getrowcount();
    var i;

    for(i = 0; i < nRowCount; i++) {
        if(grdSessionList.getitemtext(i, 0) == pRemoteIpAddr) {
            if(grdSessionList.getitemtext(i, 1) == nRemotePortNo) {
                grdSessionList.deleterow(i);
                break;
            }
        }
    }
}
```

```

    }
}

return;
}

// XTcpSvrComm
function btnInitXTcpSvrComm_on_mouseup(objInst)
{
    var strBindPortNo = fldBindPortNo.gettext();
    var strLogDir = "C:\\\\xFrame\\log";
    var nRet;

    /**
     * XPlusTcpSvr
     * @param pLogDir
     * @param pLogFilePrefix
     * @param pLogLevel ("DEBUG", "INFO", "WARN", "ERROR", "FATAL")
     ("DEBUG"
     )
     * @param nLogEncFlag (0 1)
     * @param nBindPortNo Listen TCP
     * @param nLengthFieldLength
     * @param nLenIncFlag (0 1)
     * @returns
     * 0: Success
     * 1: Already Initialized
     * 2: Invalid Port Number
     * 3: Invalid Length Field Length
     */
    nRet = objXPlusTcpSvr.innerctrl.InitXPlusTcpSvr(strLogDir,
    "XPlusTcpSvr", "DEBUG", 0, strBindPortNo, 8, 0);
    factory.consoleprint("InitXPlusTcpSvr Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("InitXPlusTcpSvr() Fail, ErrorCode = " + nRet);
    }

    /**
     * TCP UTF8
     * @param nSocketKey
     * @returns
     * 0: Success
     * 1: Invalid Parameter
     */
    nRet = objXPlusTcpSvr.innerctrl.SetUtf8Flag(1);
    factory.consoleprint("SetUtf8Flag Return Value = " + nRet);

    /**
     * TCP ( : )
     * @param nSocketKey
     * @returns
     * 0: Success

```

```

    *           1: Invalid Parameter
    */
    nRet = objXPlusTcpSvr.innerctrl.SetSendTimeout(5000);    // 5
    factory.consoleprint("SetSendTimeout Return Value = " + nRet);
}

// XTcpSvrComm
function btnStartXTcpSvrComm_on_mouseup(objInst)
{
    /**
     * XPlusTcpSvr           TCP
     * @param nBindPortNo Listen TCP           (0           ,
InitXPlusTcpSvr           )
     * @returns
     *     0: Success
     *     1: Invalid Port No
     *     2: Fail To Create Socket
     *     3: Fail To Socket Listen
     *     9: Not Initialized
     */
    nRet = objXPlusTcpSvr.innerctrl.StartXPlusTcpSvr(0);
    factory.consoleprint("StartXPlusTcpSvr Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("StartXPlusTcpSvr() Fail, ErrorCode = " + nRet);
    }
}

// XTcpSvrComm
function btnStopXTcpSvrComm_on_mouseup(objInst)
{
    /**
     * XPlusTcpSvr           ,           TCP
     * @param nCloseSessionFlag           TCP
     * @param nFireEventFlag
     * @returns
     *     0: Success
     *     9: Not Initialized
     */
    var nCloseSessionFlag = 1;
    var nFireEventFlag = 0;
    var nRet;

    nRet = objXPlusTcpSvr.innerctrl.StopXPlusTcpSvr(nCloseSessionFlag,
nFireEventFlag);
    factory.consoleprint("StopXPlusTcpSvr Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("StopXPlusTcpSvr() Fail, ErrorCode = " + nRet);
    }

    grdSessionList.deleteall();
}

```

```
//
function btnCloseSession_on_mouseup(objInst)
{
    var nCheckedRowCount = grdSessionList.getcheckedrowcount();
    if(nCheckedRowCount == 0) {
        screen.alert("                    .");
        return;
    }

    var nCheckedRow = grdSessionList.getcheckedrow(0);
    var strRemoteIpAddr = grdSessionList.getitemtext(nCheckedRow, 0);
    var strRemotePortNo = grdSessionList.getitemtext(nCheckedRow, 1);
    var nRet;

    /**
    *      TCP
    *      .
    *      IP
    * @param strRemoteIpAddr
    * @param strRemotePortNo
    * @param nFireEventFlag
    * @return
    * 0 : OK
    * 1 : Invalid Parameter
    * 2 : Fail To Find Session
    * 9 : Not Initialized
    */
    var nFireEventFlag = 1;
    nRet = objXPlusTcpSvr.innerctrl.CloseSession(strRemoteIpAddr,
strRemotePortNo, nFireEventFlag);
    // nRet = objXPlusTcpSvr.innerctrl.CloseSessionByKey(nSocketKey,
nFireEventFlag);
    factory.consoleprint("CloseSession Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("CloseSession() Fail, ErrorCode = " + nRet);
    }
}

//
function btnSendData_on_mouseup(objInst)
{
    var nCheckedRowCount = grdSessionList.getcheckedrowcount();
    if(nCheckedRowCount == 0) {
        screen.alert("                    .");
        return;
    }

    var nCheckedRow = grdSessionList.getcheckedrow(0);
    var strRemoteIpAddr = grdSessionList.getitemtext(nCheckedRow, 0);
    var nRemotePortNo = grdSessionList.getitemtext(nCheckedRow, 1);
    var strSendData = fldSendData.gettext();
    var nRet;
```

```
/**
 *
 * @param pRemoteIpAddr          IP
 * @param nRemotePortNo
 * @param pData
 * @returns
 *      0: OK
 *      1: Fail To Find Session
 *      2: Invalid Parameter
 *      3: Session is Not Connected
 *      4: Fail To Send Length Part
 *      5: Fail To Send Data Part
 *      9: Not Initialized
 */
nRet = objXPlusTcpSvr.innerctrl.SendData(strRemoteIpAddr, nRemotePortNo,
strSendData);
// nRet = objXPlusTcpSvr.innerctrl.SendDataByKey(nSocketKey,
strSendData);
factory.consoleprint("SendData Return Value = " + nRet);
if(nRet != 0) {
    screen.alert("SendData() Fail, ErrorCode = " + nRet);
}
}

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

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

Permanent link:
http://technet.softbase.co.kr/wiki/guide/xplus/xplus_tcpsvr



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