

Fasit httpftp:

```
#include "stdafx.h"
#include "httpftp.h"
#include "httpftpDlg.h"
#include "afxdialogex.h"
#include "CRobotInternet.cpp"
#include <fstream>
using namespace std;
#ifdef _DEBUG
#define new DEBUG_NEW
#endif

CRobotInternet Robot; //Lagar eit objekt av klassen CRobotInternet
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

    // Dialog Data
    enum { IDD = IDD_ABOUTBOX };

    protected:
        virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support

    // Implementation
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(CAboutDlg::IDD)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// ChttpftpDlg dialog

ChttpftpDlg::ChttpftpDlg(CWnd* pParent /*=NULL*/)
    : CDialogEx(ChttpftpDlg::IDD, pParent)
    , m_Tid(0)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    m_Bruker = _T("");
    m_URL = _T("");
    m_Bruker2 = _T("");
}
```

```

        m_Filnamn = _T("");
        m_Lokaltfilnamn = _T("");
        m_Passord = _T("");
        m_Passord2 = _T("");
        m_URL2 = _T("");
    }

void ChttpftpDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Text(pDX, IDC_Tid, m_Tid);
    DDX_Text(pDX, IDC_Bruker, m_Bruker);
    DDX_Text(pDX, IDC_URL, m_URL);
    DDX_Text(pDX, IDC_Bruker2, m_Bruker2);
    DDX_Text(pDX, IDC_Filnamn, m_Filnamn);
    DDX_Check(pDX, IDC_ftp, m_ftp);
    DDX_Text(pDX, IDC_Lokaltfilnamn, m_Lokaltfilnamn);
    DDX_Text(pDX, IDC_Passord, m_Passord);
    DDX_Text(pDX, IDC_Passord2, m_Passord2);
    DDX_Text(pDX, IDC_URL2, m_URL2);
}

BEGIN_MESSAGE_MAP(ChttpftpDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    // ON_BN_CLICKED(IDC_BUTTON1, &ChttpftpDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON1, &ChttpftpDlg::OnClickedButton1)
    ON_WM_DESTROY()
    ON_WM_TIMER()
END_MESSAGE_MAP()

// ChttpftpDlg message handlers

BOOL ChttpftpDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }
}

```

```
// Set the icon for this dialog. The framework does this automatically
// when the application's main window is not a dialog
SetIcon(m_hIcon, TRUE);           // Set big icon
SetIcon(m_hIcon, FALSE);        // Set small icon

// TODO: Add extra initialization here
int installResult;
installResult=SetTimer(1,60000,NULL); //60000 millisekund
if(installResult==0)MessageBox("Kan ikkje installere timer");
m_Tid=1; //Eitt minutt
m_URL="SKSK.NO";
m_Filnamn="index.htm";
m_Lokaltfilnamn="dittnamn.htm";
m_URL2="192.168.20.29"; //L  rarPC: "Elevmaskin"
m_Bruker2="Test4\ftp";//Lagt inn som USER p   mappe //Inetpub/ftproot/Opplasting
m_Passord2="Password1";//Lagt inn som passord for bruker ftp

UpdateData(FALSE); //Oppdaterer skjemafelt

return TRUE; // return TRUE unless you set the focus to a control
}

void ChttpftpDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void ChttpftpDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
}
```

```

    }
    else
    {
        CDialogEx::OnPaint();
    }
}

// The system calls this function to obtain the cursor to display while the user
drags
// the minimized window.
HCURSOR ChttpftpDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

```

```

void ChttpftpDlg::OnClickedButton1()
{
    // TODO: Add your control notification handler code here
    int GamalTid=m_Tid;
    UpdateData(TRUE); //Data frå skjema til variablar
    if(GamalTid!=m_Tid){
        KillTimer(1);
        int millisekund=m_Tid*60000;
        int installResult=SetTimer(1,millisekund,NULL);
        if(installResult==0)MessageBox("Kan ikkje installere timer");
    }
    ofstream Utfil; //Lagar opplastingsskript-fila
    Utfil.open("ftpopp1.scr");
    Utfil<<"open "<<m_URL2<<endl;
    Utfil<<m_Bruker2<<endl;
    Utfil<<m_Passord2<<endl;
    Utfil<<"put "<<m_Lokaltfilnamn<<endl;
    Utfil<<"quit"<<endl;
    Utfil.close();
}

```

```

void ChttpftpDlg::OnDestroy()
{
    CDialogEx::OnDestroy();
    KillTimer(1);

    // TODO: Add your message handler code here
}

```

```

void ChttpftpDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    UpdateData(TRUE); //Data frå skjema til variablar
    int Resultat=0;
    bool OK=true;
    //Dersom vi har valgt å bruke FTP på nedlasting:
}

```

```

if(m_ftp){
CString URL=m_Bruker+": "+m_Passord+"@"+m_URL;NULL;
CString Dir=""; //Katalog på server. Tom fordi brukernamn gir direkte tilgang til katalogen
CString Feilmelding="";

//Brukar objektet Robot til nedlastinga:
Robot.ftpGetFile(URL,Dir,m_Filnamn,m_Lokaltfilnamn,Resultat,Feilmelding);
if(Resultat!=0)MessageBox(Feilmelding);
}
//Dersom vi vil bruke http-nedlasting:
else
{
    CString URL=m_URL+"/"+m_Filnamn;
    CString Feilmelding="";
    //Brukar objektet Robot til nedlastinga:
    Robot.httpGetFile(URL,m_Lokaltfilnamn,Resultat,Feilmelding);
    if(Resultat!=0)MessageBox(Feilmelding);
}
if(Resultat==0){ //Køyrer opplasting dersom nedlastinga gjekk fint
//system("ftp -s:ftppoppl.scr"); //Brukar skript til opplasting
//Her kunne vi brukt objektet Robot til opplastinga:
CString sUrl=m_Bruker2+": "+m_Passord2+"@"+m_URL2;NULL;
CString sDir=""; //Katalog på server. Tom fordi brukernamn gir direkte tilgang til katalogen
int nResult=0;
CString Feil="";
OK=Robot.ftpPutFile( sUrl, sDir,m_Lokaltfilnamn,m_Lokaltfilnamn,nResult, Feil);
if(nResult==0)MessageBox(Feil);
}

CDialogEx::OnTimer(nIDEvent);
}

```