파일전송 모듈

## 0. 구현 목적

-미완성

## 1. 파일전송 구현 스펙

-단일 DLL 파일 업로드 구현

-전송은 WCF와 WPF 통신을 이용할 것

-WPF File Open Dialog, TextBox, Button 등을 이용하여 파일 선택 UI 구현

-전송하였을 때 파일명을 원본 파일명 그대로 전송되게 설정

-전송할 때 프로그레스바로 현재 상황 표시

-수정, 삭제 기능 구현

-업로드 File Path를 App.config에서 수정가능하고 폴더가 없을 시 강제로 생성

2. 파일전송 구현 내용을 위해 조사 및 한일들

-WCF 파일업로드 조사

-기존 DataServiceForManager 안에서 작동여부 확인

-파일전송만을 위한 작은 WPF 프로그램 제작

-같은 파일이 업로드 되었을 때 경고창 띄우는 로직 생성

-같은 파일명이 있는지 검사하기 위해서 디렉토리 안에 파일이 존재하는지 체크하는 메소드 작성

-삭제시 db파일명을 읽어서 db에서만 지우는게 아니라 파일도 삭제

-수정시 기존파일을 지우고 업로드

-WCF 서비스가 아닌 MS-SQL Service일 때 기능 막기

-파일이 업로드 되었을 때 업로드 된 파일 이름을 DB에 저장되도록 구현

-Progress bar UI 구현방법 검색(Progress bar 초기화 포함)

-파일을 선택하면 바로 업로드 될 것인지 아니면 파일은 선택만 되어있고 파일전송은 모든 자료 입력이 끝나고 OK 버튼을 눌렀을 때 업로드가 될 것인지 고민

-최대 업로드 사이즈 및 파일경로 app.config에서 설정

-DataGrid에서 삭제 했을시 파일도 같이 삭제

-각종 경고 메시지 출력

-영문, 한글 메시지들 리소스를 분리

-wsHttpBiing, basicHttpBinding 간의 자유롭게 교체 가능

-파일 업로드 사이즈 제한 해결(IIS문제 해결 및 테스트)

-수정 에서 OK 버튼을 눌렀을 때 파일 업로드를 하지 않고 단순히 값만 수정했을때는 파일 업로드 기능을 사용하지 않게 구현

3. 기타 조사

-폴더를 통째로 올리기 위해 WPF File Open Dialog에서 폴더 선택을 지원하는지.

 (조사한 결과 폴더만을 선택할 수 있는 Dialog는 있으나 파일과 폴더를 같이 선택할 수는 없음)

4. 문제점

-대용량 파일 업로드 시 파일을 로딩 시간이 오래 걸림

-폴더를 생성하면 기본적으로 속성이 읽기전용으로 되어있는데(vista일경우) 수동으로 풀어줘야함

-dll을 수정하거나 업로드 할려면 먼저 서비스부터 OFF 시켜야하는데 아직 서비스를 OFF 하는 프로세스가 없다.

5. 해결한것들

6. 해결할때 힘이 들었던점

-파일을 업로드 하는데 고용량 파일(60mb이상) 은 파일업로드가 안됨

 -해결법

IIS 연결 타임 아웃 설정 교체 기본설정은 2분이기 때문에 업로드 시간이 2분이 넘어가는 큰 파일들은 중간에 오류를 일으키게 됨, 인터넷 속도를 고려해서 바꿀 것.

IIS사이트 선택 -> 웹사이트 관리- 고급설정 – 연결시간 제한 초를 늘린다.



 해당 WCF 서비스에서

<httpRuntime maxRequestLength="51200" /> 라인을

**<httpRuntime executionTimeout="999999" maxRequestLength="51200" />**

**로 바꾼다.**

참고로 여기서 maxRequestLength 의 51200 은 50\* 1024 즉 50 MB를 뜻하는 것임. 이 값을 50MB 로 놔둬도 아래의 나머지 세팅을 해 주면 문서 라이브러리에 대용량 파일 업로드는 문제가 없지만 공지사항이나 토론 같은 리스트의 첨부는 여전히 50MB 을 넘을 수 없음. 따라서 리스트 첨부로 큰 파일을 올리고자 할 때에는 이 값을 늘려 주어야 함. 예를 들어 리스트 첨부를 300MB 까지 허용한다면 300 \* 1024 = 307200 을 세팅해 주면 됩.

만약 Windows 2008 /IIS7 환경의 경우 추가 요소

WCF config 파일에서 아래와 같은 세팅해야함

<system.webServer>

  <security>

    <requestFiltering>

      <requestLimits maxAllowedContentLength="52428800"/>

    </requestFiltering>

  </security>

</system.webServer>

52428800 은 50 MB 이고 이 값은 업로드를 허용할 최대 크기보다 약간 더 크게 설정되어야 함. 예를 들어 한도가 300MB 로 하고 싶다면 301\*1024\*1024 = 315621376 정도로 설정하시면 됨

7. MTOM 이란?

MTOM은 Message Transmission Optimization Mechanism의 약자로 SOAP 메시지 통신 최적화 메커니즘이다. 웹 서비스를 통해 바이너리 데이터를 효과적으로 편리하게 전송할 수 있도록 하는 표준이다.

쉽게 설명하면 Soap 메시지는 일반적으로 XML 텍스트로 전달되는데 이때 텍스트가 아닌 데이터들도 텍스트 형태로 변환되어 전달되어 짐, 이 변환 과정에서 불필요한 자원의 낭비가 발생할뿐 아니라 변환된 데이터는 원본보다 커지게 됨 WCF에서는 기본적으로 텍스트 변환을 위해 Base64bit 인코딩을 사용.

텍스트가 아닌 데이터를 텍스트로 변환/전달하는 과정의 손실을 줄이기 위해 WCF는 MTOM을 사용함.

결론: 지금까지 MTOM 에 대해서 잘 몰랐을 때에 는 MTOM 이 파일 용량사이즈에 영향을 받는 부분인지 알았지만 결국 대용량 파일전송을 할 때 성능을 좀더 좋게 하기 위해서 쓰는것이다. 그래서 현재 DataService에서는 성능보다는 호환성과 안전성을 더 중요시하기 때문에 성능문제는 파일전송 서비스를 따로 빼서 하는것이 바람직하다고 생각한다.

MTOM 설정 방법

binding에서 MessageEncoding = “Mtom” 으로 변경하면 됨, 보통은 Text로 설정

8. 파일 업로드를 구현하기 위해 만든 메소드

-WCF Server

void UploadFile(RemoteFileInfo request)

실질적인 파일업로드 메소드로써 파일경로, 파일명, 파일길이를 받아서 스트림을 읽어 바이트에 넣어 쓴다.

bool CheckFile(string FileDirectoryName, string FileName)

파일이 업로드 디렉토리에 존재하는지 체크하는 메소드로써 파일을 업로드 할 때 같은 파일이 있을 경우 덮어 씌울건지 경고창을 띄울 때 쓴다.

bool DeleteFile(string FileDirectoryName, string FileName)

파일을 삭제할 때 쓰는 메소드로써 파일이 존재하는지 체크해서 파일이 있을경우에 삭제한다.

-iCONTROL Manager

SettingDeviceInterface.xaml.cs

void ShowUpLoadFileWindow()

파일 업로드 Window을 띄우는 역할을 한다. 띄울때 업로드를 위해 FileOpenDialog에서 선택한 파일명과 폴더를 전달한다.

UpLoadFileWindow.xaml.cs

프로그레스 바 제공 및 실질적인 파일 업로드 프로세서

void UPLoadFile(SettingDeviceInterface parentPage, string filepath, string fileName)

FileOpenDialog에서 얻은 파일 정보를 스트림으로 읽어서 WCF Service의 UploadFile 메소드에 보내는 역할을 함.

void UpdateProgress(double progress)

File Upload Progressbar에 Value값을 Update, Set

void uploadStreamWithProgress\_ProgressChanged(object sender, StreamUtil.ProgressChangedEventArgs e)

프로그레스 이벤트가 바뀔때마다 호출해서 변경내용을 UpdateProgress() 메소드에 전달

9. 현재 시스템 사용법

-현재 시스템의 Binding은 wsHttpBinding이다. 이 wsHttpBinding을 이용하면

파일전송 용량 제한은 ? 이며 보안이 더 좋다?

<endpoint address="" binding="wsHttpBinding" bindingConfiguration="wsHttpBindingConfig" contract="IService">

이시스템을 바꾸는 법은

<endpoint address="" binding="basicHttpBinding" bindingConfiguration="basicHttpBindingConfig" contract="IService">

이렇게 binding과 bindingConfiguration만 바꾸면 된다.

WPF에서는

<endpoint address="http://172.16.10.99:16100/Service.svc" binding="wsHttpBinding"

 bindingConfiguration="WSHttpBinding\_IService" contract="DataServiceForManager.IService"

 name="WSHttpBinding\_IService">

이렇게 되어있는 것을

<endpoint address="http://172.16.10.99:16100/Service.svc" binding="basicHttpBinding"

 bindingConfiguration="BasicHttpBinding\_IService" contract="DataServiceForManager.IService"

 name="WSHttpBinding\_IService">

이렇게 바인딩 방식을 basic으로 바꾼다.

BasicHttpBiding방식과 wsHttpBinding방식의 차이점

# 사용 설명서



수정이나 ADD 버튼을 누른다.



Device Lib File Upload 항목 오른쪽 맨끝에있는 OPEN 버튼을 클릭한다.



UpLoad 파일을 선택하고 열기 버튼을 클릭한다.



만약 파일이 존재한다면 “파일이 이미 존재합니다. 삭제하시고 다시 업로드 하시겠습니까?” 라는 메시지가 나오고 OK 버튼을 클릭하면 현재 서버에 저장된 파일은 삭제하고 다시 업로드 한다.

만약 수정모드라면 기존파일은 삭제 된다.(파일이 많아지면 용량문제가 있을수 있으므로)



파일이 성공적으로 upload가 되었다면 “File Upload Success” 라는 메시지가 출력되며 OK 버튼을 누르면 Device Interface Setting 창이 닫긴다.



파일이 정상적으로 업로드 되었는지 DataGrid에서 확인한다.

만약 del버튼으로 DB내용을 삭제시 파일도 같이 삭제된다.

# WCF의 transferMode 속성의 "Streamed" 설명

WCF(Windows Communication Foundation)에서는 버퍼링된 전송 또는 스트리밍 전송을 사용하여 메시지를 보낼 수 있다.

-기본 설정인 버퍼링된 전송 모드에서는 메시지가 완전히 전달되어야 수신자가 읽을 수 있다.

-스트리밍 전송 모드에서는 메시지가 완전히 전달되기 전에 수신자가 메시지 처리를 시작할 수 있다.

-전달되는 정보가 길고 연속적으로 처리 가능한 경우 스트리밍 모드가 유용.

-전체를 버퍼링하기에는 메시지가 너무 큰 경우에도 스트리밍 모드가 효과적임.

-HttpBinding 중에서는 유일하게 BasicHttpBinding에서만 설정 가능(NetTcpBinding, NetNamePipieBinding은 기본적으로 제공)

TrasferMode 속성 설정

  **-Buffered**.  기본값

 -**Streamed**. 양 방향으로 스트리 및 통신이 가능

 -**StreamedRequest**. 요청만 스트리밍이 가능

-**StreamedResponse**. 응답만 스트리밍이 가능

참고 사이트: <http://msdn.microsoft.com/en-us/library/ms731913.aspx>