

9. 파일: 여러 파일을 동시에 업로드 하기

문제점:

여러 파일을 동시에 업로드하고 싶다.

해결책:

`HttpContext` 포함된 `Request` 속성의 `Files` 컬렉션을 이용하여 여러 파일을 동시에 업로드할 수가 있다.

방법:

인터넷 상에서 자료실 게시판은 보편화되어 있는 웹 서비스 중에 하나이다. 자료실 게시판은 우리에게 서로 필요로 하는 자료를 업로드하고, 다운로드를 하여 자료를 교환하는 중요한 역할로서 사용이 된다. 대부분 자료실 게시판은 하나의 파일을 업로드하고, 다운로드 하는 것이 보편화되어 있다. 여기서 더 나아가, 여러 파일을 다루는 것을 논의하려고 한다. 여러 파일들을 다운로드 하기 위해서는 업로드한 자료에 여러 링크를 만들면 되는 쉬운 작업이기에, 여기서는 여러 파일을 어떻게 업로드하는지에 대해서만 다루어 보고자 한다.

여러 파일을 업로드하는 방법을 설명하기 위하여, 적당히 3개 정도의 파일을 업로드할 수 있는 예제 프로그램으로 설명을 하겠다. 그리고, 이 예제에 간단한 기능을 추가하여, 모든 파일이 아닌 웹에서 보여줄 수 있는 jpg, gif 이미지 파일만 업로드할 수 있게 하였다. 더 많은 파일과 다른 종류의 파일들도 업로드하고 싶다면, 이 예제를 조금만 더 고치면 기능이 추가된 업로드 프로그램이 될 것이라고 믿는다.

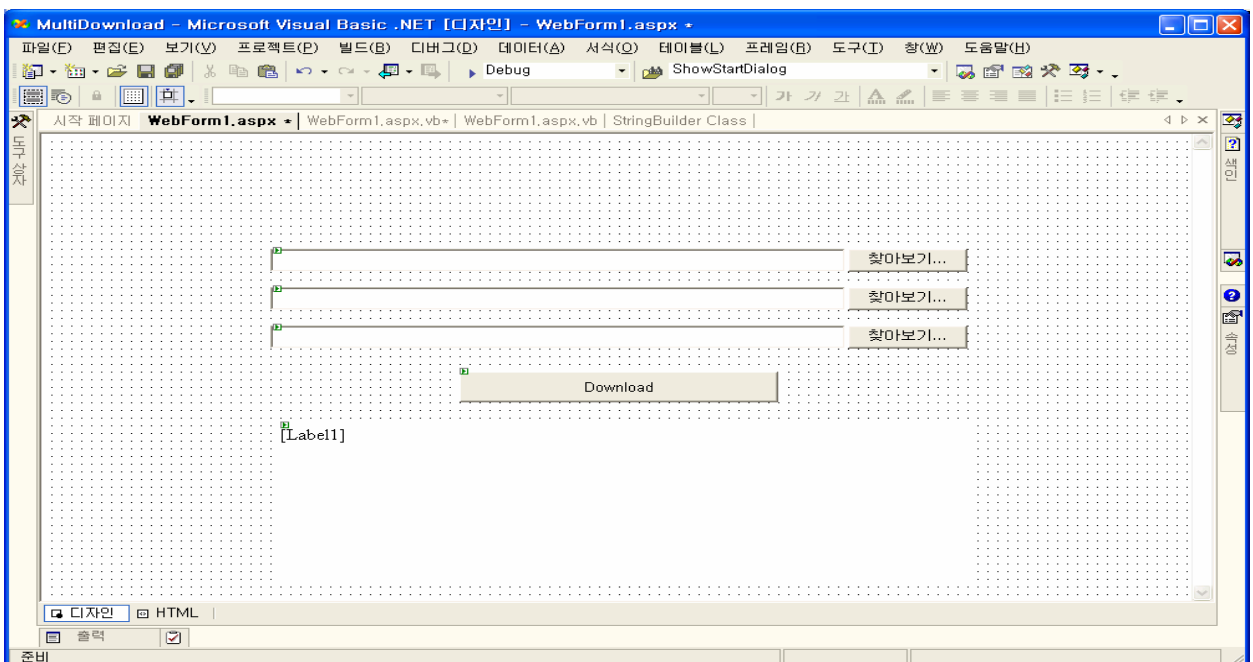


그림 1. 업로드 페이지

업로드를 위하여 디자인 창에 몇몇 컨트롤을 가져다 놓아 보자. 그림1.과 같이 파일 업로드 할 수 있도록 `HtmlInputFile` 컨트롤을 가져다 놓는다. `HtmlInputFile` 컨트롤은 `WebControls`이 아니라 `HtmlControls`에 포함되어 있으므로 도구상자의 HTML에서 발견할 수 있다. 직접 HTML을 코딩하려면 `<Input>` HTML 태그에서 `Type`을 `file`로 설정하면 될 것이다. 업로드 동작을 위한 버튼과 간단한 결과 표현을 위한 레이블 또한 가져다 놓아두자. 이렇게 하여 업로드를 위한 디자인 작업은 끝났다. 이제 업로드 동작을 위한 코딩을 살펴보자.

우리가 업로드하려는 파일들을 위한 모든 정보는 웹 페이지의 `HttpContext`에 포함이 되어 있다. 특히, `HttpRequest`를 위한 `HttpContext`의 `Request` 속성은 우리가 업로드한 파일을 찾을 수 있게 한다. 이들 업로드 파일들은 `Files` 컬렉션을 이용하여 추출할 수 있다. 따라서, 우리에게 필요한 정보들을 `System.Web.HttpContext.Current.Request.Files`에서 `_files` 파일 컬렉션에 할당을 한다. 물론 `_files`는 `Files`과 같은 타입인 `HttpFileCollection`으로 선언을 하여야 할 것이다.

반복문을 통하여 업로드된 각각의 `HttpPostedFile`을 `_files`로부터 추출할 수 있다. 이미 이야기 한 것처럼 모든 파일이 업로드가 아니라, `jpg`, `gif` 이미지 파일만을 업로드하기로 하였기 때문에 파일 하나씩 확장자를 비교한다. 편리하게 `System.IO.Path.GetExtension()`을 통하여 쉽게 확장자를 분리하여, 확장자에 따라 해당디렉토리에 업로드된 파일들을 저장한다. 여기서는 이미지 파일 종류에 따라 `gif` 파일은 이 예제가 동작하는 디렉토리에 있는 `gifs` 디렉토리에, `jpg`파일은 `jpps` 디렉토리에 저장하였다. 이를 위해서 `MapPath`를 이용하여 필요한 디렉토리를 알 수 있다. `gifs`, `jpps` 디렉토리는 스스로 만들고 그 디렉토리에 대한 권한도 파일 쓰기 가능하게 설정하기 바란다. 만약, 업로드한 파일이 이미지 파일이 아닌 경우는 적당한 메시지를 레이블에 출력을 한다. 모든 파일을 업로드하지 않아도 `_files`에는 3개의 파일이 전달이 되므로, 파일이름을 비교하여 파일이름이 없으면 업로드되지 않은 것으로 간주하고 업로드에 관련된 작업을 하지 않게 하자. 혹 발생할 수 있는 에러를 처리하기 위하여 `Try-End Try`로 묶어 두는 것도 좋은 생각이다.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Dim _files As System.Web.HttpFileCollection
    _files = System.Web.HttpContext.Current.Request.Files

    Dim _message As String
    Try
        Dim _iFile As Integer

        For _iFile = 0 To _files.Count - 1
            Dim _postedFile As System.Web.HttpPostedFile
            _postedFile = _files(_iFile)

            Dim _fileName, _fileExtension As String
            _fileName = System.IO.Path.GetFileName(_postedFile.FileName)
            If _fileName.Trim <> "" Then
                _fileExtension = System.IO.Path.GetExtension(_fileName)

                If _fileExtension = ".gif" Then
                    _postedFile.SaveAs(System.Web.HttpContext.Current.Request.MapPath("gifs\") & _fileName)
                    _message &= _fileName & "<br>"
                ElseIf _fileExtension = ".jpg" Then

```

```

_postedFile.SaveAs(System.Web.HttpContext.Current.Request.MapPath("~/jogsW") & _fileName)
    _message &= _fileName & "<br>"
Else
    _message &= _fileName & "<font color=red> Failed!! You can only
download .gif and .jpg images allowed ! </font><br>"
End If
End If
Next
Label1.Text = _message
Catch ex As Exception
Label1.Text = ex.Message
End Try End Sub

```

표1. 업로드 프로시저

이렇게 하여 업로드 작업을 위한 코딩을 마쳤다. 이제 예제 프로그램을 실행하여 파일을 업로드하여 보자. 그림2.와 같이 업로드를 위한 웹 페이지가 나타나는가? 그러면 각각의 “찾아보기” 버튼을 클릭하면 나타나는 업로드 창에 업로드하려는 파일을 선택을 하자. 업로드 기능이 잘 동작하는지 확인하기 위하여 확장자가 jpg, gif인 파일을 선택하고 아닌 파일도 선택하여 적절히 테스트하여 보자. 그림 2.의 예제는 jpg, gif 이미지 파일과 이미지 파일이 아닌 doc 워드파일을 업로드 파일로 선택하여 테스트를 하였다.

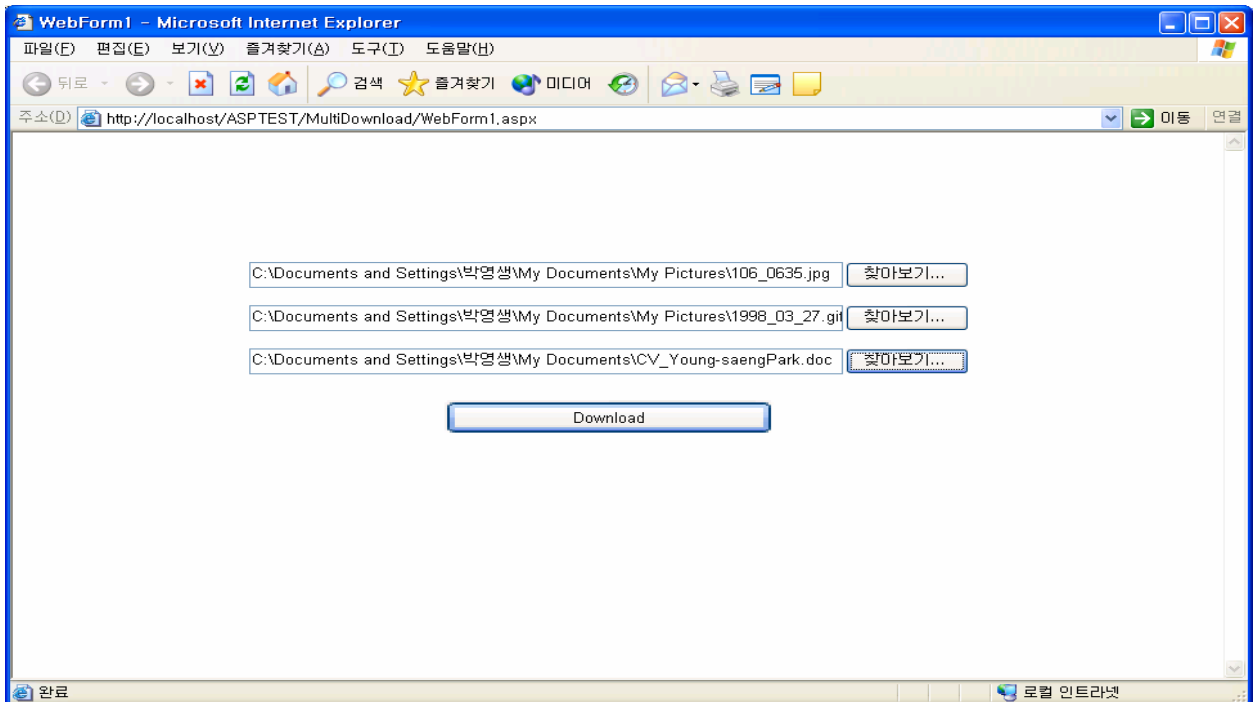


그림2. 파일 업로드

업로드한 결과로 우리가 허락한 jpg, gif 이미지 파일들에 대해서는 파일이름만이 출력되고, 아닌 파일들에 대해서는 간단히 에러 메세지가 그림3.처럼 나타날 것이다. 여기서는 파일 중복 업로드에 관한 기능은 추가하지 않았기에 스스로 추가하기 바란다.

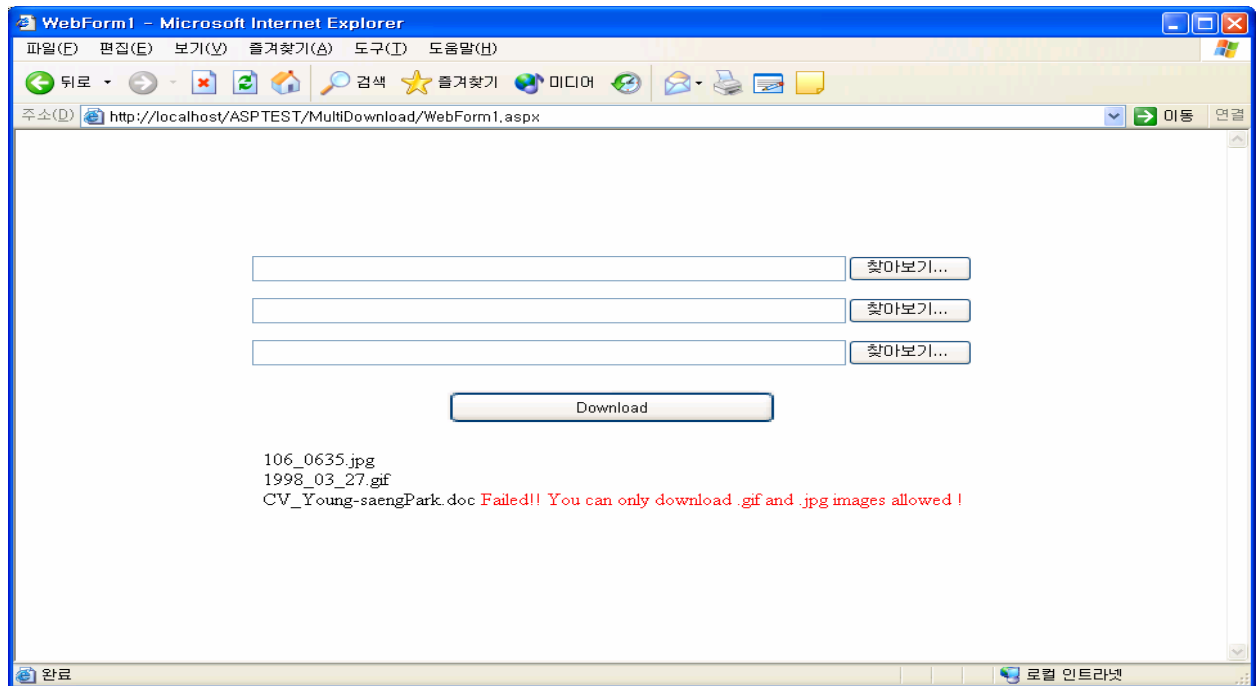


그림1. 업로드 결과

강산이 한번 바뀌기 전의 이야기지만 유닉스 시스템 상에서 C 언어로 웹 업로드 프로그램을 개발한 기억이 있다. 그때는 업로드를 위하여 Multipart MIME 포맷의 데이터 스트림을 덤프받아 직접 분석하여 제작하였기에 시간도 많이 걸리고 코딩도 많이 소요되었던 거 같다. 하지만, 이제는 그런 작업이 필요 없이 간단히 업로드를 할 수 있는 프로그램을 제작 할 수 있을 것이다. 위의 프로그램은 단지 예제이다. 더 좋은 기능을 위해서는 스스로 더 많은 기능들을 추가하기 바란다.