

ASP.NET 下动态产生控件实现多文件随意上传^①

白金牛 (包头医学院网络与信息中心 内蒙古包头 014010)

摘要:介绍了在 ASP.NET 环境下通过浏览器上传多文件的一种方案,该方法利用动态 HTML 语言(DHTML) 结合客户端 JavaScript 脚本动态产生控件,并用 C#语言编写后台代码真正实现了基于 Web 的无组件多文件上传,对上传文件的个数和类型都没有什么限制,并将上传后的文件名保存到数据库中,本方法具有灵活、简单、易用的特点,是 WEB 应用中值得借鉴的一种方法。

关键词:ASP.NET DHTML C# JavaScript 数据库 多文件上传

1 问题的提出

Web 开发中,文件上传技术是一种非常实用的技术,在 ASP.NET 的前一个版本 ASP 里实现这个功能,就必须使用第三方的组件或者自己开发组件了。在 ASP.NET 下有什么更好的上传方法呢?

D 就笔者看到上传文件的例子而言,一般涉及的上传文件数是固定的,有的表单一次只能上传一个文件,但在有的场合要求上传的文件个数很可能是随机的,对于上传文件数固定的情况,一般都是在表单上预先设定好固定数目的上传文件组件,但这种情形对上传文件的数目有了具体的限制,程序功能过于死板,如果上传的文件个数突破了这个数目限制,这样的程序就不实用了。

鉴于上面的情况,笔者考虑如何能够实现多个文件的随意上传,即能够随用户自己来决定要上传几个文件,上传一个文件,就添加一个,上传两个就添加两个……依此类推,这样在上传表单中就不用限定放几个上传组件了。

2 解决办法

由于要上传的文件数是随机的,在界面设计时不能首先预知上传文件的个数,这就需要在网页界面设计时设计成灵活的按钮,笔者想到利用动态 HTML 语言(DHTML) 动态产生控件的技术以及动态删除控件技术来解决此问题。思路是在上传文件表单上设计了

一个“添加文件”按钮如图 1 所示,每点击一次此按钮,



就添加一个“File Field”上传文件选择框,用户可以在客户端点击“浏览…”按钮选择要上传的文件,如果在没有上传前还想删除其中的某些要上传的文件,可以点击删除按钮 Delete,这样使添加文件和删除文件的功能更加灵活。

上传文件的保存采用现在常用的只将上传的文件名保存到数据库的方法,这是因为如果把上传的文件也保存到数据库中会造成数据库的过于庞大而不利于在网上访问。而且在数据库中只用了一个字段来保存上传的多个文件名,节省了存储空间。

在正式解决问题前,我们先介绍在 DHTML 中要用到的几个方法。

3 DHTML 中的几个方法介绍

DHTML 中内嵌的 JavaScript 经常用到下面的方法:

(1) 创建一个新元素

格式: `createElement(String type)`

功能: 创建指定类型的新元素,并返回该元素的一个引用。

(2) 在文档中插入 HTML 语句

格式: `obj.innerHTML = "……"`

功能: 在 obj 对象内动态插入“……”包含的 HTML 语句。

^① 国家计委西部大学校园计算机网络建设工程项目(计投资 20012437)资助。

(3) 添加子结点

格式: `obj.appendChild("para")`

功能: 把 `para` 指定的结点插入到 `obj` 中作为 `obj` 的子结点。

(4) 移除子结点

格式: `obj.removeChild("para")`

功能: 移除现有的结点 `obj` 的子结点, 子结点由参数 `para` 指定。

多文件随意上传

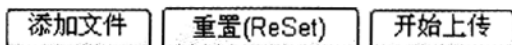


图 1 多文件上传界面

4 上传界面及 HTML 代码

图 1 是上传文件的界面, 用户点击界面上的“添加文件”按钮, 便会产生一个类似下图 2 的 File Field 的 HTML 控件, 以便在客户端选择要上传的文件, 图 3 示出了四次点击界面上的“添加文件”按钮的效果。

多文件随意上传

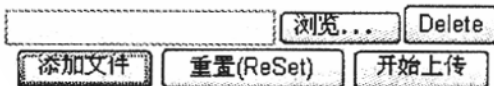


图 2 点击添加文件按钮后的界面

多文件随意上传

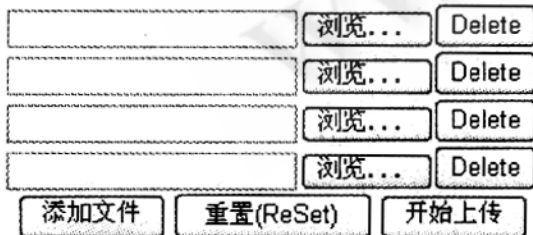


图 3 四次点击添加文件按钮后的界面

下面是本界面的 HTML 代码和内嵌的 JavaScript

脚本

```
<HTML >
<HEAD >
<Title > 多文件随意上传例 </Title >
<script language = " JavaScript" >
function addFile( )
{
//下面语句定义字符串变量 str,其中存放的是能产生
File Field 控件的 HTML 代码;
var str = `<INPUT type = " file" size = " 20" NAME
= " File" > <input type = button value = Delete onclick
= " remove( this)" > <br >`;
var o = document. all[ " files" ]; //在文档中找到 ID
为 files 的元素,保存到变量 o 中以便引用
var div = document. createElement( " div" ); //在文
档中创建元素 div,并把返回值赋值给变量 div,以便在
随后的代码中引用
div. innerHTML = str; //在新创建的 div 位置插入 HT-
ML 代码,HTML 代码内容放在变量 str 中
o. appendChild( div ); 调用对象 o 的 appendChilded
方法在文档中 files 位置添加 div 节点
}
function remove( obj)
{
//以下代码实现 File Field 控件的删除
obj. parentElement. parentElement. removeChild( obj.
parentElement );
}
</script >
</HEAD >
<body >
<form id = " form1" method = " post" enctype = " mul-
tipart/form - data" runat = " server" >
<div align = " center" >
<h3 >
<asp:Label id = " Label1" runat = " server" >Label </
asp:Label > 多文件上传 </h3 >
<P id = " MyFile" > </P >
<div id = " files" > </div >
<input onclick = " addFile( )" type = " button" value
```

```

=" 添加文件" > <input onclick = " this. form. reset
( )" type = " button" value = " 重置( ReSet)" >
<asp:button id = " UploadButton" Text = " 开始上传"
Runat = " server" > </asp:button >
<P id = " P1" > <FONT face = " 宋体" > </FONT >
&nbsp; &nbsp; &nbsp; </P> </div >
</form >
</body >
</HTML >

```

5 “开始上传”按钮的 click 事件过程

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.OleDb;
namespace uploadfile
{
    /// <summary>
    /// upload 的摘要说明。
    /// </summary>
    public class upload : System.Web.UI.Page
    {
        protected OleDbConnection myConnection = new
        OleDbConnection();
        protected System.Web.UI.WebControls.Button
        UploadButton;
        private void Page_Load ( object sender, System.
        EventArgs e )
        {
            // 在此处放置用户代码以初始化页面
            myConnection. ConnectionString = " Provider = Mi-
            crosoft. Jet. OleDb. 4. 0;" + " Data Source = " + Server.
            MapPath ( " database/database. mdb" );//以 Access

```

数据库为例连接数据库

```

}
private void UploadButton_Click ( object sender,
System. EventArgs e )
{
    HttpFileCollection files = HttpContext. Current. Re-
    quest. Files;
    string fileName, fileNameTemp, strFileName;
    string UpFullFileName = " ";
    for ( int iFile = 0; iFile < files. Count; iFile + + )
    {
        //检查文件扩展名字
        HttpPostedFile postedFile = files[ iFile ];
        string fileExtension;
        fileName = System. IO. Path. GetFileName
        ( postedFile. FileName ). ToString ( ); //获取要上传的
        文件名
        fileExtension = System. IO. Path. GetExtension
        ( fileName ). ToLower ( );
        if ( ( fileName ! = " " ) )
        {
            //以下两句生成以年月日时分秒格式的文件名
            以防上传到服务器的文件名重名
            fileNameTemp = System. DateTime. Now. ToS-
            tring ( " yyyyMMddHHmmss" );
            //在文件名后再加两位数字以避免同一次上传
            的几个文件重名
            fileName = fileNameTemp + iFile. ToString ( "
            00" ) + fileExtension;
            strFileName = " img" + iFile. ToString ( " 00" ) +
            fileName + " |||";
            UpFullFileName = UpFullFileName + strFileName;
            //将几个上传的文件名连在一起中间用|||分隔,
            postedFile. SaveAs ( System. Web. HttpContext.
            Current. Request. MapPath ( " upimg/" ) + fileName );
        }
    }
    string ConnStr, SQL;
    ConnStr = " Provider = Microsoft. Jet. OLEDB. 4.
    0;" + " Data Source = " + Server. MapPath ( " database/

```

database.mdb");//数据库链接

```
SQL = "Insert Into news1 (news_img) Values ("
+ UpFullFileName + ")"; //向数据库插入数据,假设库
中只有一个 news_img 字段,将连接后的文件名保存
到此字段里。
```

```
OleDbConnection Conn = new OleDbConnection
(ConnStr);
Conn.Open();
OleDbCommand Cmd = new OleDbCommand
(SQL, Conn);
Cmd.ExecuteNonQuery();
Conn.Close();
}
}
}
```

6 程序调试中应注意的问题

(1) 上传文件时,为保证多个文件能上传,要对 Web.config 配置文件进行如下配置,在 <system.web> 配置节中增加下列配置语句,以保证能够上传多个文件。

```
<httpRuntime maxRequestLength = "10240" >
</httpRuntime >
```

如果要上传更大的文件,还可以将 maxRequestLength 的值设置得更大,这里取 10240,单位是 KB,意思是最大能上传 10MB。

(2) 上传时文件的保存。文件上传后,为保证上传到服务器的文件名不重复,本方法利用上传时系统的日期和时间对文件名进行命名,并精确到秒,并在秒后补两位数,补的两位数是当时上传的文件序号,保证了上传文件名是唯一的而不发生重名。此外还将上传的文件名保存到一个字段中,各个文件名之间用“||”符号隔开,以后提取时只要对字段的内容稍做处理即可取得各个文件名。

7 总结

通过上面的方法利用非常简短的一段程序便可实现多文件的随意上传,本方法简单实用,灵活方便,实现起来也不复杂,很好地解决了 WEB 应用中上传文件的问题。以上程序在 Microsoft Visual Studio .NET 2003 环境 ©《计算机系统应用》编辑部 <http://www.c-s-a.org.cn>