

【学术研究】

ASP 无组件上传技术分析

於肇鹏

(辽阳职业技术学院, 辽宁 辽阳 111004)

摘要: 阐述 ASP 技术中通过无组件方式实现文件上传管理的原理及方法, 并进一步阐述在文件上传过程中一些特殊要求的处理方法和原则.

关键词: ASP 技术; 无组件上传; 分析; RFC1867

中图分类号: TP393 093

文献标识码: A

文章编号: 1008-5688(2011)02-0042-03

0 前言

目前, ASP 技术中主要通过两类方法实现文件的上传功能: 第一类, 利用第三方组件实现上传, 但这类方法需要在服务器上注册; 第二类, 无组件上传, 当使用别人的服务器时, 对方可能不允许安装第三方组件, 这时就必须使用无组件上传.

所谓无组件上传, 实际上就是利用表单将所有文件和其他说明信息等当作二进制数据提交上去, 然后自己手工从中分析出什么是普通表单内容, 什么是文件内容. 也就是不使用第三方组件, 直接上传文件.

了解无组件上传原理, 有助于大家理解 AspUpload 等组件到底是怎么工作的, 甚至可以动手开发属于自己的文件上传组件. 要掌握无组件上传, 最为重要的就是分析客户端传上来的数据. 不管是用什么组件, 上传表单文件都是一样的. 也就是说, 用户将文件提交到服务器端的时候都是一样的, 区别就是服务器端得到数据后, 用什么方法来分析得到其中的文件名称、文件大小、文件内容、文件说明等信息.

本文将通过分析“化境 ASP 无组件上传类”原理及代码来阐述无组件上传的基本原理. 一般, 要实现 ASP 无组件上传, 主要采用的都是 RFC1867 标准.

1 规定 HTML 网页表单内容格式

要在 HTML 网页中实现上传, 必须要在 `<form>` 标签中加入如下代码, 以保证上传文件以二进制形式传输^[1].

`ENCTYPE="multipart/form-data"`

2 服务器对浏览器发送的二进制数据进行处理

如图 1 所示, 用户通过表单的方式将文件上传到服务器, 服务器首先建立一个缓冲池, 将所有表单内容以二进制形式保存到缓冲池中, 再建立第二个缓冲池, 以集合的形式分类保存普通表单项目和各个文件内容. 最后, 以二进制流数据保存到文件中.

对于缓冲池中的数据, 可以通过 Stream 对象来实现数据保存, 至于为什么不选择 Fso 对象则是因为在缓冲池中的数据是以二进制流的形式存在, 并不适用于 Fso 对象的使用.

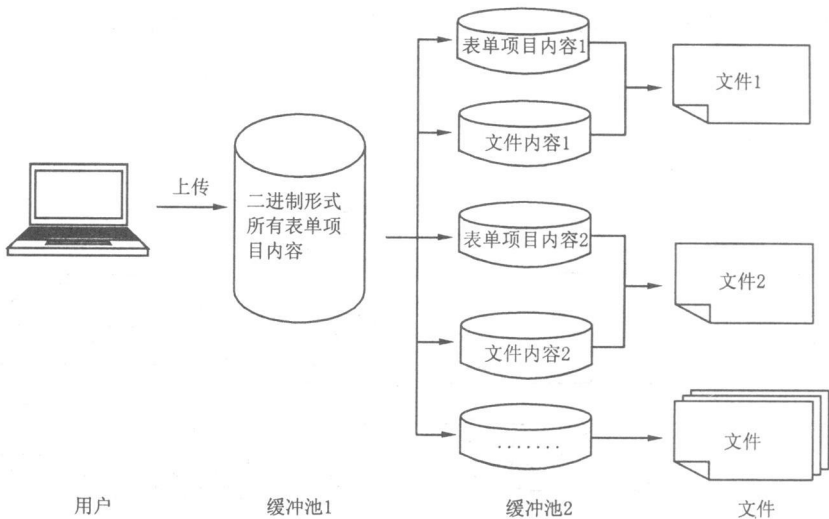


图1 文件上传及处理过程示意图

收稿日期: 2011-04-10

作者简介: 於肇鹏 (1979-), 男, 辽宁辽阳市人, 讲师, 主要从事计算机软件开发与应用方面研究.

(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

```
‘ 程序代码片段                                     ( Request. TotalBytes)
建立第一个缓冲池, 用于接收用户传来的数据           建立第二个缓冲池, 用于单独保存文件
set Data_ 5xsoft = Server. CreateObject ( " adodb.      set tStream = Server. CreateObject ( " adodb.
stream ")                                              stream ")
.....                                              .....
Data_ 5xSoft_ Stream. write Request. BinaryRead      tStream. SaveToFile FullPath, 2
RFC1867 中已经规定, 在 Request 对象中所有数据通过特定的边界进行分隔. 通过对上传数据的解
读, 可以发现, 在所有的表单项目进行提交的时候, 分隔的方式是固定的, 或者说, RFC 建议采用相同
的分隔字符对不同的表单项目进行区分. 在数据的实际传输过程中, 采用“填充字符+换行符+回车符”
的方式对表单项目的内容进行分隔[2]. 也就是说, 只要找到二进制数据中相应的换行回车, 就可以找到数
据的分隔符. 通常可以通过如下代码找到分隔符, 并实现对数据的开始及终止位置的寻址.
iFormEnd = LenB ( RequestData)                      iFormStart= iFormStart+ iStart+ 1
vbCrLf = chrB ( 13) & chrB ( 10)                   .....
sStart = MidB ( RequestData, 1, InStrB ( iFormStart,      iInfoEnd = InStrB ( iFormStart, RequestData, vbCrLf
RequestData, vbCrLf) - 1)                             & vbCrLf) + 3
iStart = LenB ( sStart)
```

需要注意的是, 在这个寻址过程中所分割的数据, 并不是所有的数据都是上传的文件, 还有其他的表
单中的数据, 同样也以类似的方式呈现. 用来区分二者的方式同样可以用代码来实现, 分析了 Request 对
象中的二进制数据之后, 服务器的主要工作就是把 Form 普通项目数据与文件数据分类保存, 在大多数的
无组件上传类中总有类似的代码来实现相关的功能. 相关代码部分:

```
.....                                              .....
sInfo = tStream. ReadText                          objFile. add sFormName, theFile `将文件数据保存到
tStream. Close                                     objFile 集合中
如果是文件                                         else
if InStr ( 45, sInfo, " filename= """, 1) > 0 then   如果是表单项目, 则将表单项目信息添加到表单数据集
set theFile= new FileInfo                           合中.
建立一个新文件数据, 用于保存文件信息
```

对于以上功能的实现, 可以通过建立一个封装上述所有功能的类来实现, 在 ASP 中可以通过建立一个
独立的文件来定义这个类, 其主要功能包括:

- (1) 读取用户发送的二进制数据并将其保存到缓冲区, 这个过程一般在类的初始化过程中实现.
- (2) 执行对二进制数据流的处理, 其中包括对把各个文件以集合形式读取到缓冲区.
- (3) 将各个文件利用 A dodb. stream 对象的 SaveAs() 功能逐个保存.
- (4) 将各个表单项目数据读取到以集合形式存储的缓冲区.

要使用这个类的时候, 只需要通过 ASP 中的< ! -- #include file= "" --> 来调用. 建立一个类
的对象, 通过该类实现对上传文件及表单项目的处理.

```
upload= new upload_ 5xsoft 建立上传对象           的文件
? .....                                           file. SaveAs Server. mappath (FolderPath) 保存文件
for each formName in upload. objFile `列出所有上传了
```

3 对上传文件的特殊处理

3.1 限制上传文件的大小

如上文, 我们已经获得了上传的文件, 一般来说, 我们对上传文件的大小进行限制以保证服务器不会
被占用过多空间, 在这里可以采用两种方法, 第一种方法是通过对 Request 对象的 TotalBytes 属性来进行
限制, 如果 Request. TotalBytes 的值超过预先设定的上限, 服务器可以向用户反馈文件规模过大的信
息. 这种方法可以对上传文件的总的大小进行限制. 如果想对每一个上传的文件都进行限制, 则可以使用
第二种方法, 利用在类中定义的保存文件的集合, 通过逐个提取集合中的文件进行字节数量判断的方法,
在保存文件的同时对文件的大小进行限制, 如果文件大小大于事先规定的字节数量, 则取消上传.

3.2 限制上传文件的类型

如果我们把各种类型的文件不加限制地上传, 服务器可能被一些恶意程序所破坏, 为了保证服务器的安
全, 必须对上传文件的类型进行规定, 这一点, 可以通过接收到的文件的信息来进行判别, 利用如下代码:

split (File FileName, ".")

可以提取上传的文件类型,利用限定词对上传的文件类型进行判断,如果上传文件的类型与限定词匹配,则满足上传条件,可以把文件保存,如果不匹配,则不满足上传条件,不对文件进行保存操作^[3]。

3.3 文件的重命名

有的时候由于上传的文件过多,难免会出现上传的文件与已经保存的文件重名的现象,为了避免这一情况的发生,可以采用对上传文件重新命名的方式,一般可以采用“上传的时间+随机数”的方式对上传的文件进行命名,以保证在同一时间上传的文件也不会发生冲突。

参考文献:

[1] 胡晓明. 基于 ASP 实现文件上传的研究 [J]. 西安文理学院学报(自然科学版), 2005, (1): 72—75.

[2] 周庆, 王续跃. 一种基于 ASP 实现无组件上传文件的新方法 [J]. 微机发展, 2004, (12): 83—85.

[3] 梁霄波. 基于 ASP 的图片上传技术的设计与实现 [J]. 现代计算机(专业版), 2010, (12): 107—109.

(审稿人 李树东, 责任编辑 朱成杰)

(上接 41 页)

3.2 系统各环节技术介绍

系统中采用的都是目前 J2EE 领域最热门的技术,同时也是被大量采用的成熟的技术。

客户层采用 HTML 浏览器和 RSS 阅读器。Web 表现层,主要采用业界成熟的 Struts 框架。Struts 是一个基于 SunJ2EE 平台的 MVC 框架,主要是采用 Servlet 和 JSP 技术来实现的。Struts 把 Servlet、JSP、自定义标签和信息资源整合到一个统一的框架中,开发人员利用其进行开发时不用再自己编码实现全套 MVC 模式,极大地节省了时间。业务逻辑层,主要采用业界成熟的 Spring 框架。框架的主要优势之一就是其分层架构,分层架构允许开发人员选择使用哪一个组件,同时为 J2EE 应用程序开发提供集成的框架。数据访问层,主要采用 Hibernate 框架。Hibernate 是一个开放源代码的对象关系映射框架,它对 JDBC 进行了非常轻量级的对象封装,使得 Java 程序员可以随心所欲地使用对象编程思维来操纵数据库。

3.3 纯 P2P 技术模式^[1]

为了实现同一区域内的开放性和共享性,结合分布式特点,提出采用 P2P 技术将资源的负载分布在客户端。对基于 P2P 技术的资源平台做了总体设计,包括服务器端、客户端以及数据库的设计。分析了资源共享的设计,包括 JXTA 底层网络建立的消息和管理。实现整个系统的过程中,关键的地方就是要解决多用户访问速度慢、资源共享、信息孤岛问题。具体方法如下:

第一,为了利用 P2P 技术带来的资源共享、传输等优势,同时又能吸收传统集中式系统的一些优点,如更有效的管理、低访问压力下的稳定性好等特点,教育资源共享网络采用两层体系结构:超节点层和用户层。在核心层采用集中管理为基础的 P2P 技术混合模式,在用户层采用纯 P2P 技术模式。

第二,异构资源库集成技术,运用资源门户网站,采用分布式资源网,各节点资源库除提供资源管理外,还提供对其他资源库的访问和服务,以提高资源的共享率,以实现资源的共享。

3.4 J2EE 分布式组件技术

J2EE 是 Java 2 Platform Enterprise Edition 的缩写,J2EE 核心是一组技术规范与指南,其中所包含的各类组件、服务架构及技术层次,均有共通的标准及规格。J2EE 在平台的开发中具备以下几点优势:独立于硬件配置和操作系统^[2];坚持面向对象的设计原则;灵活性、可移植性和互操作性;轻松的信息系统集成。

4 结语

构建一个开放、统一的教育资源共享平台,实现教育资源的有效共享是一项长期的、复杂的系统工程。本文介绍的 Eduray 教育资源门户网站设计及运用此技术开发的铁岭市教育资源平台的设计及实现,正进入应用阶段。在今后的研发中,将继续把资源的范围从基础教育扩充到高等教育、继续教育、学前教育等,把资源的组织形式从基础教育资源扩充到主题学习资源、学生自主学习资源等,实现使用优质教学资源提高教育教学质量的目的。

参考文献:

[1] 杨红飞. 分布式选课系统的研究与设计 [D]. 广州: 广东工业大学, 2008.

[2] 朱治军. J2EE 技术在电子文件管理系统开发中的应用研究 [J]. 兰台世界(下半月), 2010, (4): 14—15.

(审稿人 李树东, 责任编辑 朱成杰)