

# 用 WinInet 类编写基于 HTTP 的 Internet 客户应用程序

常新功 (太原山西财经大学信息系 030006)

**摘要:** 本文介绍了 Visual C++6.0 中 WinInet 类的组成和工作原理, 并在此基础上进一步阐述了用 WinInet 编写基于 HTTP 的 Internet 客户应用程序的主要步骤。文章最后还给出了一个简单的实例, 以期收理论联系实际之效。

**关键词:** WinInet HTTP Internet 客户应用程序

Internet 网络应用大多采取客户机/服务器 (client/server) 体系结构。其中, 服务器作为 Internet 网络资源的管理者和提供者, 运行于提供服务的网络主机之上, 向众多的 Internet 客户应用提供某种特定的服务; 客户机即 Internet 客户应用程序, 作为用户访问 Internet 的接口, 运行于用户的本地计算机之上, 采用 HTTP、FTP、gopher 等多种 Internet 协议与服务器交互, 为用户提供对 Internet 网络资源的访问。

WinInet (Win32 Internet Extensions) 是 Microsoft 在 VC++6.0 MFC (Microsoft Foundation Classes) 中提供的用于 Internet 客户应用程序开发的一个类集。该类集提供了访问 HTTP、FTP、gopher 等 Internet 协议的简便方法, 使您的 Internet 客户应用程序对 HTTP、FTP、gopher 服务器的访问, 就像从您的本地硬盘上读取文件一样容易。利用 WinInet, 你可以在一个较高的层次编写 Internet 客户应用程序而无需了解 WinSock、TCP/IP 或某些特定的 Internet 协议的底层细节。具体地说, 就是利用 WinInet, 你可以很方便地创建 Internet 客户应用程序以下载超文

本页面, 发送 FTP 请求上传 (upload) 或下载 (download) 文件, 用 gopher 的菜单系统访问 Internet 资源……。而这一切都无需顾及底层细节, 例如, 连接到一个远程 HTTP 服务器在底层需要许多步骤才能完成, 但使用 WinInet 类, 你只需调用 `CInternetSession::GetHttpConnection` 即可创建连接。WinInet 的使用, 大大简化了开发人员的工作, 它使 Internet 客户应用程序的开发不再是那么枯燥乏味, 只需几行代码, 或是在 AppWizard 对话框中选择一个检查框, 就能给你的应用程序增加许多令人惊奇的功能。以下我们就 WinInet 的组成、工作原理以及基于 HTTP 的 Internet 客户应用程序的开发步骤等问题分几部分加以说明。

## 1 WinInet 类的组成

WinInet 是一个类集, 专门用于 Internet 客户应用程序的编写, 它由一些类和全局函数组成。表 1、2 给出了该类集的组成、类的层次关系及每个类的简要用途说明。其中, WinInet 的组成类见表 1, WinInet 的全局函数见表 2。

表 1

类	父类	用途
CInternetSession	CObject	用于创建和初始化一个或多个同时进行的 Internet 会话
CInternetConnection	CObject	用于管理本地计算机与 Internet 服务器建立的连接
CFtpConnection	CInternetConnection	在管理 FTP 连接的同时还可直接操纵该服务器上的目录和文件
CGopherConnection	CInternetConnection	管理一个 gopher 连接
CHttpConnection	CInternetConnection	管理一个 HTTP 连接
CInternetFile	CStdioFile	利用 Internet 协议访问远程系统上的文件
CGopherFile	CInternetFile	在一个 gopher 服务器上查寻和读取文件
CHttpFile	CInternetFile	在一个 HTTP 服务器上请求和读取文件
CFileFind	CObject	执行本地文件查寻
CFtpFileFind	CFileFind	在 FTP 服务器进行文件查寻
CGopherFileFind	CFileFind	在 gopher 服务器进行文件查寻
CGopherLocator	CObject	从 gopher 服务器获取一个 gopher 定位器用于 CgopherFileFind
CInternetException	CException	用于 Internet 操作异常处理

以上这些类和全局函数除 CFileFind 在 AFX.H 中声明外,其他均在 AFXINET.H 中声明。因此在编写 Internet 客户应用程序时应包含以上头文件。

## 2 Wininet 工作原理

Internet 网络应用是基于客户机/服务器体系结构的,它通过客户程序和服务器程序的相互交互来完成各种网络应用任务。我们将客户程序和服务器程序间的交互称为 Internet 会话。因此,每个 Internet 应用包括 Internet 客户应用程序在工作之前必须首先建立一个 Internet 会话,这一步是通过对类 CInternetSession 的实例化来完成的。

Internet 客户应用程序是用户访问 Internet 网络资源的接口。而服务器是网络资源的管理者和提供者。Internet 客户应用程序在访问 Internet 网络资源之前,必须首先和特定的服务器建立连接,这可以通过创建 CInternetConnection 对象来实现。也可以通过调用 CInternetSession::GetFtpConnection, CInternetSession::GetHttpConnection, CInternetSession::GetGopherConnection 分别建立基于 FTP、HTTP、gopher 协议的连接。

在创立了会话和连接后,为了能够访问网络资源,即在服务器上读写信息,就必须创建 Internet 文件对象。这可以通过以下任何一种函数调用来完成:①调用 CInternetSession::OpenURL,该调用返回指向 CInternetFile 或 CStdioFile (此时,URL 指向本地文件)的指针;②在分别建立了基于 HTTP、FTP、gopher 协议的连接后,调用 CHttpConnection::OpenRequest,

表 2

全局函数	用途
AfxParseURL	对 URL (统一资源定位器) 进行词法分析
AfxGetInternetHandleType	用于决定一个 Internet 句柄的类型
AfxThrowInternetException	用于抛出 Internet 操作异常

CFtpConnection::OpenFile, CGopherConnection::OpenFile, 它们分别返回指向 CHttpFile、CInternetFile、CGopherFile 的指针。

然后,可通过创建 CFileFind、CFtpFileFind 或 CGopherFileFind 对象分别用于在本地或网络服务器上进行文件查寻,用 OpenFile 打开找到的文件,用 CInternetFile::Read 或 CInternetFile::Write 向该文件读写信息。

在程序的最后,不要忘了关闭打开的文件,撤消不再使用的会话和连接。

## 3 用 Wininet 编写基于 HTTP 的 Internet 客户应用程序的步骤

HTTP (Hyper Text Transfer Protocol),即超文本传输协议,是目前 Internet 网上最著名、最流行的信息查询和信息浏览服务 WWW (World Wide Web) 的支撑协议。通过该协议,WWW 客户应用程序可以通过建立连接、发出请求、等待响应、关闭连接的过程和 Web 服务器交互,以访问 Web 资源。

本文的重点在于阐明用 Wininet 编写基于 HTTP 的 Internet 客户应用程序的基本原理和方法,编写基于 FTP 或 gopher 的 Internet 客户应用程序的原理和方法,请参考 visual C++ 帮助文件和有关书籍。

一个典型的基于 HTTP 的 Internet 客户应用程序可通过以下几个主要步骤来创建:

第1步：创建和初始化一个 Internet 会话。

这一步通过实例化一个 CInternetSession 对象来完成。

第2步：连接到一个 HTTP 服务器。

调用 CInternetSession::GetHttpConnection("服务器名", 端口号, "用户名", "口令") 建立与指定服务器指定端口的连接。该调用返回指向 CHttpConnection 对象的指针, 利用该对象可管理刚才与指定的 HTTP 服务器建立的连接。

第3步：建立并打开一个 HTTP 请求。

调用 CHttpConnection::OpenRequest(本次 HTTP 请求的类型, "本次请求操作的对象") 打开一个 HTTP 请求。其中, 操作对象通常为一个文件名、一个可执行模块或一个查寻标识; 请求类型分别代表 GET、PUT、POST、HEAD、LINK、UNLINK、DELETE 等不同的操作。该调用返回指向 CHttpFile 对象的指针, 利用该对象用户可从已连接的 HTTP 服务器上读取数据。

第4步：发送一个 HTTP 请求并在 HTTP 服务器上找到指定的文件。

调用 CHttpFile::AddRequestHeaders("请求头") 为 HTTP 请求加上一个或多个请求头以灵活而细致地控制请求的语义。调用 CHttpFile::SendRequest() 将建立好的请求发送到 HTTP 服务器。

第5步：用 CHttpFile 类从第4步打开的文件中读取指定字节数的数据到本地缓冲区。

第6步：异常处理。

用 CInternetException 类处理所有本次会话中的一切 Internet 异常。即将第二步到最后一步的所有操作均要放入 try 块中, 并用 catch 捕获其中的一切异常并做相应的处理。

第7步：结束本次 HTTP 会话。

清除 CInternetSession 对象以结束本次 HTTP 会话。此时程序会自动清除已打开的文件句柄和已建立的连接。

## 4 用 WinInet 编写 Internet 客户应用程序

### 示例

由于篇幅所限, 本文只举一例来演示用 WinInet 编写 Internet 客户应用程序的轮廓步骤, 更为详细的论述请参考有关书籍。

例：下载一个 Web 页面

以下代码演示了下载一个 Web 页面的全过程, 语句后的注释注明了该语句的功能和所处的步骤, 以便与3中所述的编程步骤相互参照。

```
#include <afxinet.h> // WinInet 类在该头文件中声明
CInternetSession session("My Session");// 第1步, 创
```

建会话对象

```
CHttpConnection* pServer = NULL;
CHttpFile* pFile = NULL;
try
{
    CString strServerName;
    INTERNET_PORT nPort;
    pServer = session.GetHttpConnection(strServerName,
nPort);// 第2步, 连接到指定服务器名的 HTTP 服务器的
指定的端口号
    pFile = pServer->OpenRequest(CHttpConnection::
HTTP_VERB_GET, strObject);// 第3步, 建立并打开一个
HTTP 请求, 返回一个 CHttpFile 文件指针 pFile
    pFile->AddRequestHeaders(szHeaders);
    pFile->SendRequest();// 第4步, 发送该请求
    pFile->QueryInfoStatusCode(dwRet);// 获取该请求的
状态号以判断该请求成功与否
    if (dwRet == HTTP_STATUS_OK)
    {
        UINT nRead = pFile->Read(szBuff, 1023);// 第5
步, 从 pFile 中读取 1023 字节数据到缓冲区 szBuff
        while (nRead > 0)
        {
            //read file...
        }
    }
    delete pFile;
    delete pServer;
}
catch (CInternetException* pEx)
{
    // 第6步, 捕获并处理 Internet 异常
}
session.Close();// 第7步, 关闭本次会话并撤消连接
```

### 参考文献

- 1 韩兆兵、李小进、方海英编著,《Visual C++6.0 网络开发技术》, 人民邮电出版社, 1999.9.
- 2 张永奎、白山编著,《INTERNET 与 JAVA 程序设计》, 山西大学计算机科学系讲义, 1998.3.
- 3 Visual C++ 6.0 帮助文件.
- 4 马尚才等编著,《信息网络概论》, 中国商业出版社, 2000.6.