

Программирование для Интернета в Visual C++

Создание браузера

- Написать собственный браузер на Visual C++ достаточно просто. Microsoft Internet Explorer может использоваться в программах как управляющий элемент.
- Чтобы вывести страницу Microsoft в Web-браузере, достаточно включить в программу элемент Microsoft Web Browser, создать для него соответствующий объект и вызвать метод Navigate().
Чтобы сделать это, создайте новую программу на базе диалогового окна с именем browser:
- 1. Выполните команду Project > Add to Project ► Components and Controls — открывается окно с коллекцией компонентов и элементов Visual C++, Components and Controls Gallery.
- 2. Дважды щелкните на строке Registered ActiveX Controls, чтобы открыть список элементов ActiveX, зарегистрированных в вашей системе .
- 3. В списке компонентов выберите элемент Microsoft Web Browser Control, затем нажмите кнопку Insert.

- 4. Visual C++ спрашивает, какой класс следует создать для нового элемента. Подтвердите имя по умолчанию CWebBrowser (Visual C++ может предложить например имя CWebBrowser2). Элемент браузера появится в редакторе диалоговых окон.
- 5. Откройте главное диалоговое окно. Элемент браузера находится в нижней части палитры.
- 6. Перетащите элемент в диалоговое окно и Отрегулируйте его размеры. Новому элементу присваивается идентификатор IDC_EXPLORER1.
- 7. С помощью Class Wizard свяжите с этим элементом переменную и присвойте ей имя m_browser.
- 8. Кроме того, разместите в диалоговом окне кнопку Browse и свяжите с ней обработчик OnButton1().
- 9. При нажатии кнопки Browse() наш браузер должен обратиться по адресу:
<http://www.microsoft.com>. Для этого мы воспользуемся методом Navigate() (последние четыре параметра — флаги, которым в нашем примере присваивается значение 0):
- ```
void CBrowserDlg::OnButton1()
{
m_browser.Navigate("http://www.microsoft.com". 0, 0, 0, 0);
}
```
- 10. Вот и все, что от нас требовалось, — браузер готов. Запустите программу и нажмите кнопку Browse.

# Использование протокола HTTP

- В следующем примере мы воспользуемся протоколом HTTP (основным протоколом World Wide Web) для того, чтобы загрузить HTML-код страницы.

Программа будет основана на встроенной поддержке WinInet в Visual C++.

В программе будет присутствовать кнопка **Получить Web-страницу**.

HTTP

Получить Web-страницу

OK

Cancel

Текст Web-страницы:

При нажатии этой кнопки программа загружает фрагмент HTML-кода страницы (скажем, первые 1000 байт) и отображает его в текстовом поле. С помощью AppWizard создайте программу на базе диалогового окна и назовите ее HTTP. Разместите в диалоговом окне текстовое поле и кнопку с надписью Получить Web-страницу (рис. выше).

- Свяжите с кнопкой обработчик `OnButton1()`. В этом методе мы создадим новый сеанс Интернета (Internet session). Сеанс Интернета представляет собой объект класса `CInternetSession`, на котором основана вся поддержка Интернета в Visual C++ (не забудьте добавить строку `«include <afxinet. h>»` для работы с Интернет-компонентами):
- ```
void CHTTPIg::OnButton1()  
(  
• CInternetSession* pInternetSession;  
• }  
• Класс CInternetSession поддерживает протоколы HTTP, FTP и Gopher, а его методы перечислены в таблице (след. слайд).
```

Методы класса CInternetSession

Метод	Назначение
CInternetSession	Создает объект класса CInternetSession
Close	Разрывает соединение с Интернетом при завершении сеанса
EnableStatusCallback	Задаёт функцию косвенного вызова для отображения статуса
GetContext	Получает идентификатор контекста для сеанса Интернета или приложения
GetFtpConnection	Открывает сеанс FTP с сервером. Производит регистрацию пользователя
GetGopherConnection	Открывает сервер Gopher для приложения, которое пытается установить соединение
GetHttpConnection	Открывает сервер HTTP для приложения, которое пытается установить соединение
OnStatusCallback	Обновляет информацию о статусе операции при заданной функции косвенного вызова
OpenURL	Анализирует и открывает URL (адрес)
Оператор HINTERNET	Логический номер текущего сеанса Интернета
QueryOption	Предоставляет ряд возможностей для проверки ошибок
ServiceTypeFromHandle	Получает информацию о типе сервиса по логическому номеру запроса
SetOption	Задаёт параметры сеанса

- После объявления указателя `pInternetSession` на объект нового сеанса, можно приступить к созданию самого сеанса:

- ```
void CHTTPIg::OnButton1()
{
```
- ```
CInternetSession* pInternetSession;  
pInternetSession = new CInternetSession();
```
- ```
}
```

- В результате начинается новый сеанс работы с Интернетом, причем для этого используется способ, заранее определенный пользователем — то есть, если компьютер не подключен к Интернету, программа выведет окно соединения и установит связь. Если же попытка окажется неудачной, программу следует завершить:
- ```
void CHTTPIg::OnButton1()  
{  
• CInternetSession* pInternetSession;  
  pInternetSession = new CInternetSession();  
• if (!pInternetSession)  
  {  
• AfxMessageBox("Не удалось начать сеанс Internet", MB_OK);  
  return;  
• }  
• }  
• }
```
- В случае удачного подключения к Интернету приложение должно загрузить код HTML для Web-страницы.

Пересылка файлов через HTTP

- Метод `OpenURL()` класса `CInternetSession` открывает Web-страницу для передачи данных через HTTP. Он возвращает указатель на файловый объект класса `CStdioFile` (этот класс является производным от `CFile` и представляет не физический файл на диске, а поток C++, который мы сохраняем в переменной `pFile`):
- ```
void CHTTPIg::OnButton1()
{
 CInternetSession* pInternetSession;
 pInternetSession = new CInternetSession();
 if (!pInternetSession)
 {
 AfxMessageBox("Не удалось начать сеанс Internet", MB_OK);
 return;
 }
 CStdioFile* pFile = NULL;
 pFile = pInternetSession->OpenURL(CString("http://www.microsoft.com"));
}
```

У нас есть указатель на файловый объект, представляющий необходимую нам Web-страницу. Мы можем работать с ним, как с обычным файлом. Предположим требуется прочитать первые 1000 байт файла. Для этого нужно создать буфер для хранения данных и вызвать метод `Read()`:

- void CHTTPIg::OnButton1()  
{
- CInternetSession\* pInternetSession;  
pInternetSession = new CInternetSession();
- if (!pInternetSession)  
{
- AfxMessageBox("Не удалось начать сеанс Internet", MB\_OK);  
return;
- >
- CStdioFile\* pFile = NULL;
- char\* buffer;
- buffer = new char[1000];
- pFile = pInternetSession->OpenURL(CString("<http://www.microsoft.com>"));
- pFile->Read(buffer,1000);
- }

- Остается лишь отобразить полученные данные. Для этого мы свяжем с текстовым полем переменную `m_text` и занесем в него информацию. Кроме того, следует закрыть открытый файл вместе с сеансом Интернета:

- ```
void CHTTPIg::OnButton1()
{
```

- ```
 CInternetSession* pInternetSession;
 pInternetSession = new CInternetSession();
```

- ```
    if (!pInternetSession)
    {
```

- ```
 AfxMessageBox(" Не удалось начать сеанс Internet", MB_OK);
 return;
```

- ```
    }
```

- ```
 CStdioFile* pFile = NULL;
```

- ```
    char* buffer;
```

- ```
 buffer = new char[1000];
```

- ```
    pFile = pInternetSession->OpenURL(CString("http://www.microsoft.com"));
```

- ```
 pFile->Read(buffer,1000);
```

- ```
    m_text = CString(buffer, 1000);
    UpdateData(false);
```

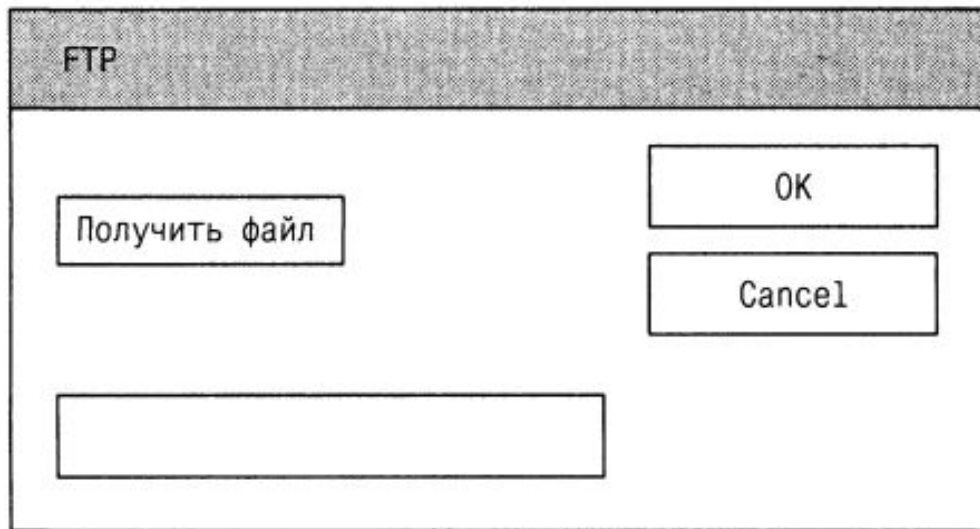
- ```
 pFile->Close();
 pInternetSession->Close();
```

- ```
    }
```

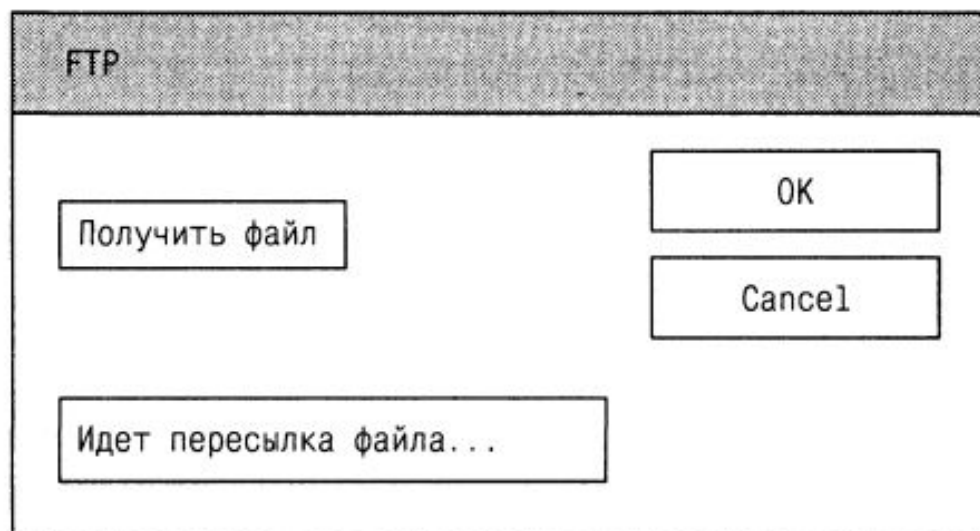
- Как мы видим из программы, метод `OpenURL()` предоставляет простую, но достаточно мощную возможность — использовать протокол HTTP для того, чтобы открыть находящийся в Web объект как обычный файл, чтобы в дальнейшем с ним можно было работать, применяя стандартные файловые методы. Тем не менее, при некотором опыте работы с HTTP у вас может возникнуть желание использовать стандартные запросы HTTP — `Get`, `Put` или `Post`. В Visual C++ они реализуются столь же просто: следует лишь вызвать метод `GetHttpConnection` класса `CInternetSession`, чтобы создать объект класса MFC `CHttpConnection`. Затем метод `OpenRequest()` этого класса используется для выполнения HTTP-запросов, при этом ему передаются такие «командные» константы, как `HTTP_VERB_GET`, `HTTP_VERB_PUT` или `HTTP_VERB_POST`.
- Класс `CHttpConnection` содержит всего два метода: конструктор и метод `OpenRequest()` для выполнения HTTP-запросов.
- Следовательно, помимо использования упрощенных методов типа `OpenURL()` вы также можете выполнять в своих программах общие HTTP-запросы; благодаря такой поддержке классы MFC заметно упрощают работу с протоколом HTTP.

Использование протокола FTP

- В следующей программе рассматривается пример работы с протоколом FTP. Он предназначен для пересылки файлов в Интернет, поэтому мы загрузим файл с FTP-узла Microsoft. Файлы на этом FTP-узле то появляются, то исчезают, но один остается постоянно (во всяком случае, он находится там уже несколько лет) — это файл disclaimer.txt. В нем содержится сообщение Microsoft о том, что файлы на FTP-узле предоставляются «как есть», без всяких гарантий. Мы загрузим этот файл в новом приложении.
- В нашей программе имеется кнопка с надписью **Получить файл** и текстовое поле.



Когда пользователь нажимает кнопку и начинает загрузку файла на диск, в текстовом поле появляется надпись «Идет пересылка файла...».



- При помощи AppWizard создайте программу на базе диалогового окна и присвойте ей имя FTP. Затем разместите в главном диалоговом окне кнопку с надписью Получить файл и текстовое поле. Свяжите с содержимым текстового поля переменную m_text и создайте обработчик OnButton1() для кнопки. Далее необходимо подключиться к Интернету.

Подключение к Интернету через FTP

- Прежде всего мы создадим в методе `OnButton1()` указатель на сеанс Интернета `pInternetSession`, как это было сделано в предыдущем примере (как и прежде, включите строку `«include <afxinet. h>»` для работы с Интернет-компонентами):
- `void CFTPDlg::OnButton1()`
`{`
- `CInternetSession* pInternetSession;`
`pInternetSession = new CInternetSession();`
- `if ((pInternetSession)`
`{`
- `AfxMessageBox("Не удалось начать сеанс Internet", MB_OK);`
`return;`
- `}`

- Затем мы создаем `pFTPConnection` — указатель на объект класса `CFtpConnection`, предназначенного для работы с FTP в программах на Visual C++. Чтобы создать этот объект, мы вызываем метод `GetFtpConnection()` класса `CInternetSession`, который устанавливает анонимное FTP-подключение к узлу `Microsoft`. В качестве параметра методу передается имя FTP-узла, к которому необходимо подключиться, — в данном случае <ftp.microsoft.com>:

- void CFTPDlg::OnButton1()
{
- CInternetSession* pInternetSession;
CFTPConnection* pFTPConnection;
- pInternetSession = new CInternetSession();
- if ((pInternetSession)
{
- AfxMessageBox("Не удалось начать сеанс Internet",
MB_OK);
return;
- }
- pFTPConnection = pInternetSession->
- GetFtpConnection(CString("<ftp.microsoft.com>"));
- }

Класс CftpConnection позволяет выполнять стандартные для протокола FTP действия — получать, отправлять, удалять файлы и т. д. Его методы перечислены в таблице

Метод	Назначение
CftpConnection	Создает объект класса CftpConnection
Close	Закрывает подключение к серверу
CreateDirectory	Создает каталог на сервере
GetCurrentDirectory	Получает текущий каталог для подключения
GetCurrentDirectoryAsURL	Получает текущий каталог для подключения в виде URL
GetFile	Получает файл с сервера
OpenFile	Открывает файл на сервере
PutFile	Помещает файл на сервер
Remove	Удаляет файл с сервера
RemoveDirectory	Удаляет заданный каталог с сервера
Rename	Переименовывает файл на сервере
SetCurrentDirectory	Задает текущий каталог для FTP

Если вы не можете или не хотите ограничиваться анонимным подключением, в методе `GetFtpConnection()` можно задать имя пользователя и пароль. Если попытка установления связи с FTP-узлом оказалась неудачной, следует вывести сообщение об ошибке и завершить работу программы

```
void CFTPDlg::OnButton1()
{
    CInternetSession* pInternetSession;
    CFtpConnection* pFTPConnection;

    pInternetSession = new CInternetSession();

    if (!pInternetSession)
    {
        → AfxMessageBox("Не удалось начать сеанс Internet", MB_OK);
        → return;
    }

    pFTPConnection = pInternetSession->
        GetFtpConnection(CString("ftp.microsoft.com"));

    if(!pFTPConnection){
        AfxMessageBox("Не удалось установить FTP-подключение", MB_OK);
        return;
    }
    .
    .
    .
}
```


В противном случае мы сообщаем о начале процесса загрузки, помещая в текстовое поле сообщение «Идет пересылка файла...»:

```
void CFTPDlg::OnButton1()
{
    CInternetSession* pInternetSession;
    CFTPConnection* pFTPConnection;

    pInternetSession = new CInternetSession();

    if (!pInternetSession)
    {
        AfxMessageBox("Не удалось начать сеанс Internet", MB_OK);
        return;
    }

    pFTPConnection = pInternetSession->
        GetFtpConnection(CString("ftp.microsoft.com"));

    if(!pFTPConnection){
        AfxMessageBox("Не удалось установить FTP-подключение", MB_OK);
        return;
    }
    → else{
    →     m_text = "Идет пересылка файла...";
    →     UpdateData(false);
    }
    .
    .
    .
}
```

Получить файл очень просто — достаточно вызвать метод `GetFile()` класса `CFTPConnection`:

```
void CFTPDlg::OnButton1()
{
    CInternetSession* pInternetSession;
    .
    .
    .
    → pFTPConnection->GetFile(CString("disclaimer.txt"),
        CString("disclaimer.txt"));

    pFTPConnection->Close();
    pInternetSession->Close();
}
```