**Работа с текстовыми файлами в Visual Basic.**

Совсем недавно в непомню какой коференции я видел запись какого-то человека, который просил, чтобы хоть кто-нибудь объяснил ему элементарную работу с файлами... Казалось бы такое простое... Но это простое для тех, кто уже хоть немного продвинулся в программировании, а кто это только начал? Это же, действительно, непроходимый лес! И я сильно сомневаюсь, чтобы кто-нибудь из других наших Web-мастеров когда-нибудь уделит этому хоть какое-то внимание. Поэтому этим займусь я и попробую в этой статье показать только основы.

**1. Открытие файла для чтения информации из него.**

Итак, чтобы безпроблемно работать с текстовыми файлами сначала их нужно открыть. Делается это с помощью оператора Open, например:

Open "c:\text.txt" For Input As #1

Обратите внимание на режим открытия файла - Input (для чтения). Здесь файл text.txt открывается для считывания под номером 1. Внимание!!! Номер файла - обязательный параметр, и вводить его нужно всегда! Если ваша программа будет иметь дело не с одним десятком открытых файлов, то уследить за их нумерацией будет сложновато. Используйте для этого функцию FreeFile. Она возвращает ещё не занятый номер (от 1 до 511) для файла:

Dim FileNum As Integer

FileNum = FreeFile

Open "c:\text.txt" For Input As FileNum

Так, файл открыли! Теперь из него нужно что-нибудь считать. Оператор Line Input считывает из открытого файла по одной строке при каждом вызове. Поэтому для считывания из файла нужно построить цикл, который бы вызывал этот оператор до тех пор, пока не дошёл бы до конца файла:

цикл выполняется, пока конец файла не достигнут

Do Until EOF(FileNum)

при каждом вызове оператора Line Input он записывает в

переменную новою строку

Line Input #FileNum, переменная

Loop

Закрыть открытый файл можно оператором Close #FileNum. А теперь вот как это должно выглядеть в практике:

Dim FileNum As Integer

Dim txt As String

Dim alltxt As String

FileNum = FreeFile

открываем файл в режиме чтения

Open "c:\text.txt" For Input As FileNum

запускаем вышеупомянутый цикл

Do Until EOF(FileNum)

считываем строку из файла с номером FileNum и кладём её в

переменную txt

Line Input #FileNum, txt

теперь к переменной alltxt добавляем содержимое переменной txt -

строки и vb-константу vbCrLf (Chr(13)+Chr(10)), т.к. каждая строка

должна быть с новой строки :-)

alltxt = alltxt + txt + vbCrLf

Loop

закрываем файл с номером FileNum

Close #FileNum

выводим содержимое переменной alltext

MsgBox alltxt

**2. Открытие файла для записи информации в него.**

Записать информацию в файл ещё проще!!! Эта операция отличается от предыдущей только тем, что открыть файл надо в режиме записи:

Open "c:\text.txt" For Output As FileNum

и вместо цикла нужно вставить оператор Print #, который записывает в файл содержимое переменной:

Print #FileNum, "Азбука Visual Basic"

или

Print #FileNum, alltext

Пример:

Dim FileNum As Integer

Dim txt As String

FileNum = FreeFile

txt = "http://members.tripod.de/cmvb"

открываем файл в режиме записи

Open "c:\text.txt" For Output As FileNum

Print #FileNum, txt

закрываем файл с номером FileNum

Close #FileNum

Проверьте содержимое файла.

И ещё хочу предложить второй способ записи информации в файл - с помощью API-функции WritePrivateProfileStringByKeyName. Считывать же информацию нужно с помощью API-функции GetPrivateProfileString. Теперь расскажу - как всё это работает?

1.Создайте стандартный проект.

2.Введите модуль.

3.Удалите форму и кликните два раза по модулю.

4.В раздел General Declarations следующее объявление двух, вышеназванных функций:

Declare Function GetPrivateProfileString& Lib \_

"kernel32" Alias "GetPrivateProfileStringA" (ByVal \_

lpszSection$, ByVal lpszKey$, ByVal lpszDefault$, \_

ByVal lpszReturnBuffer$, ByVal cchReturnBuffer&, \_

ByVal lpszFile$)

lpszSection -имя секции

lpszKey - имя ключа

lpszDefault - значение, возвращаемое по умолчанию, при строке ""

lpszReturnBuffer - место для строки

cchReturnBuffer - размер (длина) места для строки

lpszFile - имя файла

Declare Function WritePrivateProfileStringByKeyName% Lib \_

"kernel32" Alias "WritePrivateProfileStringA" (ByVal \_

lpApplicationName As String, ByVal lpKeyName As String, \_

ByVal lpString As String, ByVal lplFileName As String)

lpApplicationName - имя секции

lpKeyName - имя ключа

lpString - строка

lplFileName - имя файла

5.Создайте подпрограмму Main - просто введите Sub Main.

6.В подпрограмму Main введите:

Sub Main()

создаём файл test.ini, в нём заголовок "Main", ключ "URL" \_

и строку "http://members.tripod.de/cmvb"

Call WritePrivateProfileStringByKeyName("Main", "URL", \_

"http://members.tripod.de/cmvb", "c:\test.ini")

создаём файл test.ini, в нём заголовок "Main", ключ "eMail" \_

и строку "v1500mk@t-online.de"

Call WritePrivateProfileStringByKeyName("Main", "eMail", \_

"v1500mk@t-online.de", "c:\test.ini")

End Sub

При запуске проекта на диске С: тут же будет создан файл Test.ini со следующим содержимым:

[Main]

URL=http://members.tripod.de/cmvb

eMail=v1500mk@t-online.de

8:-O!!! Не правда ли, просто? А считать это чуть-чуть сложнее. Итак, файл создан, теперь поробуем считать информацию оттуда. Удалите всё, что находится между строками "Sub Main()" и "End Sub". Впишите следующее:

Sub Main()

создаём переменную, в которую будет передоваться строка

Dim MyString As String

"набиваем" её пробелами

MyString = Space(100)

открывается файл файл test.ini, и в переменную MyString \_

записывается строка из заголовка "Main", ключа "URL"

Call GetPrivateProfileString("Main", "URL", "", MyString, \_

Len(MyString), "C:\test.ini")

выводится на экран содержимое переменной MyString

MsgBox MyString

открывается файл файл test.ini, и в переменную MyString \_

записывается строка из заголовка "Main", ключа "eMail"

Call GetPrivateProfileString("Main", "eMail", "", MyString, \_

Len(MyString), "C:\test.ini")

выводится на экран содержимое переменной MyString

MsgBox MyString

End Sub