

Вы находитесь здесь : [scriptcoding.ru](#) » [Объектная модель Windows Script Host](#) » Объект FileSystemObject метод CreateTextFile - Как создать текстовый файл

Объект FileSystemObject метод CreateTextFile - Как создать текстовый файл

2014-12-18

Рубрика: [Объектная модель Windows Script Host](#)

Всем привет, с вами автор блога scriptcoding.ru. В этой статье мы подробно рассмотрим метод объекта FileSystemObject (fso) CreateTextFile, который показывает, как создавать текстовый файл. Метод CreateTextFile fso объекта позволяет получить доступ к другому объекту – [TextStream](#), который в свою очередь отвечает за чтение и запись данных в созданный текстовый файл.



Прежде всего нужно сделать небольшое лирическое отступление... Смотрите, как создать текстовый файл рядовыми методами без программирования? В голову сразу приходит текстовый редактор Блокнот. Как и с любым каталогом, создать текстовый файл можно всего в два этапа: *щелкаем правой кнопкой мыши по пустому месту на диске или в директории, далее из контекстного меню выбираем пункт "Создать/Текстовый документ"*, далее просто задаем имя файла и все. Но, в большинстве случаев достаточно открыть программу Блокнот и уже оттуда сохранить текстовый файл в нужном месте.

Хорошо, теперь можно приступить к программированию, но сначала, давайте рассмотрим параметры метода CreateTextFile, который и показывает как создать текстовый файл:

CreateTextFile(Filename [,Overwrite[, Unicode]]) – Видим, что метод принимает три параметра, только первый параметр является обязательным.

Рубрики

[Объектная модель Windows Script Host](#)

[Объектная модель Microsoft Word](#)

[Объектная модель Windows Management Instrumentation](#)

[Примеры скриптов и макросов для Word и WSH](#)

[Полезный софт, сервисы и познавательные статьи](#)

[Тонкости и трудности ведения блога](#)

Получить статьи по почте



VBSCRIPT

JSCRIPT

VBA

WMI

CMD



- **Filename** – Данный параметр содержит строку, которая определяет путь и имя для создаваемого текстового файла. Если указать только имя файла, то он будет создан в текущем каталоге (каталог, из которого произведен запуск сценария)
- **Overwrite** – Параметр является необязательным. Может принимать логическое true – перезаписать файл, если он уже существует или false – возможность перезаписи отсутствует. По умолчанию, стоит значение false.
- **Unicode** – Данный параметр не является обязательным, он определяет кодировку создаваемого текстового файла. Может принимать логическое значение true – формат [Unicode](#) или false – формат [ASCII](#).

Хорошо, теперь можно приступить к программированию, но сначала небольшое пояснение.

Используя метод `fso CreateTextFile`, мы создаем не просто текстовый файл с данными, а произведем запись в поток. **Поток** – данное выражение присуще только файловой системе NTFS. Суть заключается в том, что к любому документу или даже диску можно привязать один или несколько потоков. Как создать новый текстовый файл. Например, идёт имя файла, двоеточие, имя потока. Поток, содержит имя и расширение.

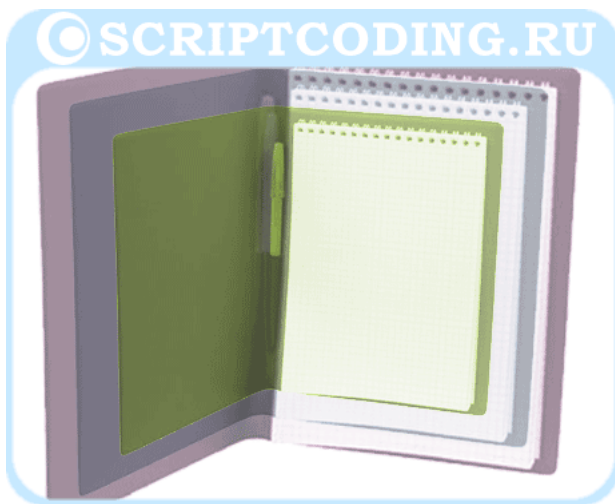
Для эксперимента, откройте командную строку и пропишите следующий код:

```
echo "Hello, word">new.txt:potok.txt
```

В данном случае, к файлу new.txt будет привязан поток potok.txt и в него будет записана строка **"Hello, word"**. Вы увидите, что документ содержит нулевой размер, и при попытке открыть его с помощью простого блокнота вы ничего не увидите. Для этого в окне **"Пуск/Выполнить"** нужно прописать:

```
notepad new.txt:potok.txt
```

Что то подобное мы сделаем с помощью сценариев.



Для начала создадим сценарий createtextfile vbs и пропишем в нем следующие строки кода:

```
*****
' fso createtextfile
' как создать текстовый файл
' запись данных в поток
' createtextfile.vbs
*****

Option Explicit

dim fso, c_file, t_file, WshShell, data, data_file

' создаем ссылку на объект Wscript.Shell
set WshShell = CreateObject("Wscript.Shell")
' создаем ссылку на объект Scripting.FileSystemObject
set fso=CreateObject ("Scripting.FileSystemObject")

'
' открываем текущий сценарий и записываем
' его содержимое в переменную data
'
```



```

set c_file=fso.OpenTextFile(WScript.ScriptFullName, 1, false)
data = c_file.ReadAll
c_file.Close()

'-----
'
'-----
' создаем в текущей папке текстовый документ
' записываем в поток secret
' содержимое переменной data
'-----

set t_file=fso.CreateTextFile("my_file.txt:secret.txt", true)
t_file.WriteLine(data)
t_file.Close ()

'-----
' освобождаем ссылку на объект
set t_file = nothing

'-----
' открываем созданный документ и считываем
' данные из потока data
'-----

set t_file=fso.OpenTextFile(WshShell.CurrentDirectory & "\my_file.txt:secret.txt", 1, false)
data_file = t_file.ReadAll
t_file.Close()

'-----
' выводим содержимое потока
' если использовать для вывода функцию MsgBox
' будет показана только часть данных
WScript.Echo data_file

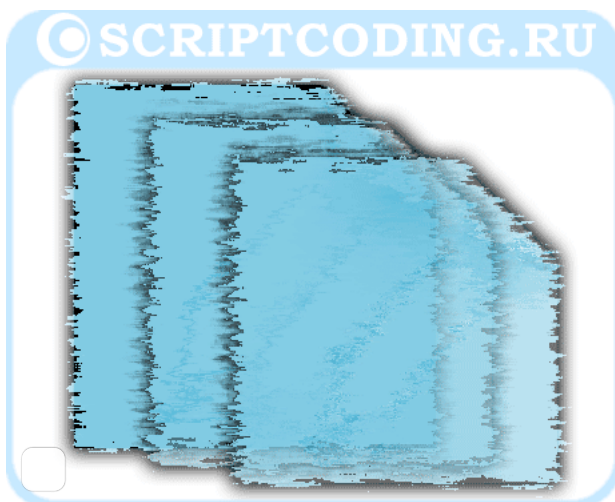
```

Тут стоит обратить внимание на следующую особенность, если мы попытаемся с помощью функции [OpenTextFile](#) открыть документ my_file.txt, не прописав после него через двоеточие имя потока, то получим ошибку:

Ввод данных за пределами файла

[Скачать архив с примерами](#)

Видимо, сервер [Windows Script Host](#) умеет определять, что в документе содержится скрытая информация. Также, вы заметили, что вместо функции [MsgBox](#) для вывода данных я использовал стандартную [WScript.Echo](#). Не знаю почему, но MsgBox выводит только часть данных. Также, в данном примере я использовал собственную функцию **CreateObject** языка VBScript, а для освобождения ссылки на объект применил ключевое слово [nothing](#).



А вот аналогичный сценарий createtextfile vbs, но уже на языке jscript:

```

//*****
// fso createtextfile
// как создать текстовый файл
// запись данных в поток
// createtextfile.js
//*****

```



```

var fso, c_file, t_file, read_file, WshShell, data, data_file;

// создаем ссылку на объект Wscript.Shell
WshShell = WScript.CreateObject("Wscript.Shell");
// создаем ссылку на объект Scripting.FileSystemObject
fso=WScript.CreateObject ("Scripting.FileSystemObject");

//-----
// открываем текущий сценарий и записываем
// его содержимое в переменную data
//-----

c_file=fso.OpenTextFile(WScript.ScriptFullName, 1, false);
data = c_file.ReadAll();
c_file.Close();

//-----

//-----
// создаем в текущей папке текстовый документ
// записываем в поток secret
// содержимое переменной data
//-----

t_file=fso.CreateTextFile("my_file.txt:secret_1.txt", true);
t_file.WriteLine(data);
t_file.Close ();

//-----

//-----
// открываем созданный документ и считываем
// данные из потока data
//-----

read_file=fso.OpenTextFile(WshShell.CurrentDirectory + "\\my_file.txt:secret_1.txt", 1, false);
data_file = read_file.ReadAll();
read_file.Close();

//-----

// выводим содержимое потока
WScript.Echo (data_file);

```

В этом примере, мы просто привязали еще один поток к файлу. Фактически, один поток будет содержать содержимое скрипта на языке [vbscript](#) (createtextfile vbs), а другой – содержимое на языке [javascript](#) (createtextfile js).

И так, мы рассмотрели как создать текстовый файл программным путем с помощью метода CreateTextFile и с помощью стандартных средств Windows.

Спасибо за внимание. Автор блога [Владимир Баталий](#)

« Объект WScript - методы sleep, quit и echo и 8 свойств

Объект FileSystemObject метод OpenTextFile - Как
открыть текстовый файл »

Советую ещё почитать:

[Виды поисковых запросов - просто о сложном](#)
[Класс TextBox - создание текстового поля VBA](#)
[Объект WScript - свойство Arguments](#)
[Объект Word FillFormat - Заливка фигур](#)
[Урок 9 по VBScript: Массивы](#)

>>>Подписаться на обновления блога!<<<

Комментарии закрыты.

