



Блог GunSmoker-a

...when altering one's mind becomes as easy as programming a computer, what does it mean to be human?..

Главная	Переводы	Вело	Лучшие публикации	Ресурсы Delphi	Обо мне	
-------------------------	--------------------------	----------------------	-----------------------------------	--------------------------------	-------------------------	--

10 сентября 2011 г.

Сериализация - общие сведения о файлах

[Оглавление серии.](#)

Для начала вспомним, что такое файл. Файл - это устройство с последовательным доступом, к которому можно обратиться по имени. Самый типичный файл - это файл на диске. Но в более широком смысле файлом являются и файловые устройства - вроде сетевого соединения, потоков данных, каналов pipe и т.п. Соответственно, операцию записи или загрузки данных из файлов на диске можно рассматривать в более широком смысле, применительно к любым файлам.

Файлы на внешних устройствах (дисках) часто называют физическими файлами. Они имеют имена, составленные согласно правилам именования файлов операционной системы. К примеру, на Windows это:

- myfile.txt
- MyFile.txt
- C:\MyFile.txt
- C:\MyFolder\MyFile.txt
- .\MyFile.txt
- ..\MyFile.txt

Именование файлов

Все файловые системы следуют одной и той же общей системе именования отдельных файлов: **базовое имя файла** (MyFile) и дополнительное **расширение файла** (txt), разделенные точкой. Базовое имя файла вместе с расширением файла называется **именем файла**: (MyFile.txt). Тем не менее, каждая файловая система (вроде NTFS, CDFS, ExFAT, UDF, FAT и FAT32) может иметь конкретные и иные правила формирования отдельных компонентов в пути к каталогу или файлу. Обратите внимание, что **каталог** (также называемый директорией), предназначенный для упорядочивания файлов путём группировки, - это просто файл со специальным атрибутом, отмечающим его как каталог, но в остальном каталоги должны следовать всё тем же правилам именования, как и обычные файлы. Поскольку термин "каталог" просто ссылается на специальный тип файла, то некоторые справочные материалы используют общий термин "файл", чтобы охватить как понятия каталога, так и понятие файла данных как такового. Из-за этого, если не указано иное, любые имена и правила использования или примеры для файла применимы также и к каталогам. Каталог не следует путать с **папкой**. Папка - это **более общее понятие**. Каталог всегда физически представлен на диске, а папка может как быть каталогом, так и представлять виртуальное (логическое) размещение - к примеру, папка "Сетевое окружение" или "Мой компьютер". Каталог самого верхнего уровня на диске называется **корневым**. Корневой каталог всегда единственен, но у каждого диска он свой.

В каждом каталоге, кроме корневого, существуют псевдо-каталоги со специальными зарезервированными именами **.** (точка) и **..** (две точки). Каталог **.** ссылается на этот же каталог, а **..** - на предыдущий (родительский каталог, каталог верхнего уровня).

Термин "**путь**" ссылается на один или несколько каталогов (или папок), **разделённых обратной косой чертой** (\ - обратный слэш, бэкслэш, back-slash), и, возможно, на имя тома (C:) или имя сервера (\\server, \\?)

\UNC\server или \\?\C:). Примечание: в некоторых дальневосточных версиях Windows для [разделителя пути используется иной символ](#), но надо понимать, что это ровно тот же символ (с тем же ANSI-кодом), [просто он выглядит иначе](#).

Всего в Windows используется три типа путей:

1. LFS (Local File System) - имена в локальной файловой системе, например: C:\MyFolder\MyFile.txt
2. UNC (Uniform Naming Convention) - сетевые UNC-имена, например: \\server\MyFolder\MyFile.txt
3. Long UNC или UNCW - длинные имена, например: \\?\UNC\server\MyFolder\MyFile.txt или \\?\C:\MyFolder\MyFile.txt

Путь файла вместе с именем файла называется **полным именем файла** (C:\MyFolder\MyFile.txt). Каждый каталог, имя тома и имя файла в пути называются **компонентами** пути.

Один из каталогов на диске является активным для работающей программы. Он называется **текущим каталогом**. Текущий каталог всегда один, он задаётся при запуске программы и может меняться в процессе её работы (путём вызова функции смены каталога). Текущий каталог является активным, рабочим - он используется при разрешении имён (см. ниже). Кроме текущего каталога программы система также отдельно отслеживает текущий каталог каждого диска. Диск, указанный в текущем каталоге, называется **текущим диском**.

Путь, начинающийся с имени тома (C:\MyFolder\MyFile.txt), имени сервера (\\server\MyFolder\MyFile.txt) или корневого каталога (\MyFolder\MyFile.txt) называется **абсолютным** - потому что такое имя всегда однозначно указывает на один и тот же файл, вне зависимости от внешнего окружения. В противном случае путь называется **относительным** (вроде MyFile.txt, .\MyFile.txt, .\MyFolder\MyFile.txt или ..\..\MyFolder\MyFile.txt). Относительные пути трактуются в зависимости от текущего каталога. Поэтому один и тот же относительный путь может ссылаться на разные файлы. К примеру, путь MyFile.txt и .\MyFile.txt ссылаются на C:\MyFolder\MyFile.txt, если текущий каталог (или каталог, относительно которого происходит разрешение имени) равен C:\MyFolder\, но эти же имена будут ссылаться на D:\Program Files\MyFolder\MyFile.txt, если текущий каталог - D:\Program Files\MyFolder\. Не следует путать полное имя файла с абсолютным. Это немного разные понятия, хотя часто их рассматривают как синонимы. Под полным именем файла понимается имя файла с путём - имя, по которому можно найти файл. Но оно не обязано быть абсолютным. С другой стороны, любое абсолютное имя всегда является полным именем. В английском языке используется термин "fully-qualified path" ("полностью указанный путь") - это синоним абсолютного пути файла.

Ограничения на количество символов также могут быть различны и меняться в зависимости от файловой системы и способа именования файла. Это осложняется ещё и поддержкой обратной совместимости. Например, старые файловые системы MS-DOS поддерживают максимум 8 символов для базового имени файла и 3 символа для расширения - в общей сложности 12 символов, включая точку-сепаратор. Кроме того, эти имена не могли включать в себя многие символы - к примеру, пробел. Этот формат имени файла широко известен как "формат файла 8.3" или **короткое имя файла**. Файловые системы Windows не имеют подобного ограничения, и хотя они поддерживают имена формата 8.3 для обратной совместимости, в основном они работают с **длинными именами файлов**.

Даже если ваша программа - "современная" и не использует короткие имена файлов MS-DOS, вам всё равно нужно про них знать, поскольку Windows содержит множество не очевидных обходных путей для обратной совместимости со старыми программами, например:

- [Почему система конвертирует TEMP в короткое имя?](#)
- [Почему FindFirstFile находит короткие имена файлов?](#)
- [Апокрифичная история туннелинга файловой системы;](#)
- ["Красивые" имена файлов в Windows.](#)

Соглашения по именованию

Следующие основные правила позволяют приложениям создавать и обрабатывать допустимые имена для

файлов и каталогов, вне зависимости от файловой системы:

- Используйте точку для отделения базового имени файла от расширения в имени файла или каталога. Каталоги могут иметь расширение, хотя обычно оно не используется.
- Используйте обратную косую черту (\) для разделения компонентов пути. Обратная косая черта разделяет имя файла от пути к нему, и имя одного каталога от другого каталога в пути. Вы не можете использовать обратную косую черту как часть имени реального файла или каталога, потому что это зарезервированный символ, который делит полное имя файла на компоненты.
- Используйте обратную косую черту в соответствии с требованиями как часть имени тома, например, C:\ в C:\path\file или \\server\share в \\server\share\path\file.
- Имена файлов не чувствительны к регистру. Например, имена OSCAR, Oscar и oscar ссылаются на один и тот же файл. Примечание: в целях совместимости с POSIX стандартом вы можете включить чувствительность к регистру для файловых имён, но это нестандартное поведение и оно не рекомендуется к использованию в общих сценариях.
- Имена томов (буквы дисков) также не чувствительны к регистру. Например, D: и d: относятся к одному и тому же тому.
- Вы можете использовать любой символ для имени файла, включая Unicode символы, за исключением следующих специальных символов:
 - < (меньше)
 - > (больше)
 - : (двоеточие)
 - " (двойные кавычки)
 - / (косая черта, слэш)
 - \ (обратная косая черта, обратный слэш)
 - | (вертикальная черта, труба)
 - ? (знак вопроса)
 - * (звёздочка)
 - Ноль (NUL-символ)
 - Символы, чьи коды лежат в диапазоне от 1 до 31 (за исключением альтернативных потоков данных, где эти символы допускаются)
 - Любые другие символы, который не поддерживает нижележащая файловая система
- Используйте точку как компонент пути для представления текущего каталога, например: .\Temp.txt.
- Используйте две последовательные точки (..) в качестве компонента пути для представления родительского каталога, например: ..\Temp.txt.
- Не используйте следующие зарезервированные имена устройств для файловых имён: CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8 и LPT9. [Также избегайте этих имен в базовых именах файлов - например, NUL.txt.](#)
- Не заканчивайте имя файла или каталога точкой. Хотя нижележащая файловая система может поддерживать такие имена, оболочка Windows, пользовательский интерфейс и прикладные программы - нет. Тем не менее, вполне возможно указывать точку первым символом имени файла, например: .temp.

Пути

Путь к указанному файлу состоит из одного или нескольких компонентов, разделенных специальным символом (обратный слэш), при этом каждый компонент обычно является именем каталога или именем файла, но с некоторыми исключениями, обсуждаемыми ниже. Очень часто решающее значение для интерпретации пути в системе имеет начало пути - так называемый **префикс** пути. Этот префикс определяет пространство имён для использования с этим путём, и, кроме того, какие специальные символы могут использоваться в пути - включая последний символ.

Если какой-то компонент пути является именем файла, то он должен быть последним компонентом в пути.

Каждый компонент пути также имеет ограничение на максимальную длину имени, зависящее от конкретной файловой системы. Чаще всего, эти ограничения сводятся к двум основным группам: короткие и длинные имена файлов. Обратите внимание, что имена каталогов хранятся в файловой системе как особый тип файлов, так что правила именования файлов распространяются также на названия каталогов. Подводя итог: путь - это просто строковое представление иерархии между всеми каталогами, которые существуют для определённого файла или каталога.

Абсолютные и относительные пути

Для функций, работающих с файлами, имя файла может задаваться относительно текущего каталога (либо какого-то иного, явно указанного каталога), либо полностью, абсолютно. Имя файла относительно текущего каталога, если оно не начинается с одной из следующих вещей:

- UNC-имя любого формата, которое всегда начинается с двух бэк-слешей (\\).
- Обозначение диска с бэк-слешем, например: C:\ или D:\.
- Один обратный бэк-слеш, представляющий корневой каталог - например, \folder или \file.txt.

Если имя файла начинается только с обозначения диска, но без бэк-слеша после двоеточия, то оно интерпретируется как относительный путь - относительно текущего каталога на заданном диске. Например:

- C:tmp.txt ссылается на файл с именем tmp.txt в текущем каталоге на диске C.
- C:Temp\tmp.txt ссылается на файл tmp.txt в подпапке Temp текущего каталога диска C.

Иногда путь также называют относительным, если он содержит каталог .. в качестве одного из своих компонентов. К примеру:

- ..\tmp.txt указывает на файл с именем tmp.txt, расположенный в родительском каталоге текущего каталога.
- ..\..\tmp.txt указывает на файл, находящийся на два каталога выше текущего каталога.
- ..\Temp\tmp.txt указывает на файл с именем tmp.txt, находящийся в каталоге Temp, который в свою очередь находится в родительском каталоге текущего каталога.
- C:..\Temp\tmp.txt указывает на файл с именем tmp.txt, находящийся в каталоге Temp, который в свою очередь находится в родительском каталоге текущего каталога диска C.
- C:\Temp\..\Temp\tmp.txt и C:\Temp\.\tmp.txt - эти два пути ссылаются на файл C:\Temp\tmp.txt. Хотя никто не будет задавать путь в таком виде, но подобные пути могут получаться после склейки полного пути из нескольких компонентов из разных источников. Хотя путь такого вида является абсолютным (не относительным) в смысле исходного определения, иногда его всё же называют относительным, подчёркивая наличие компонента .. в пути.

Максимальное ограничение длины пути

В Windows максимальная длина пути равна MAX_PATH символов, где MAX_PATH определена как константа, равная 260 - за некоторыми исключениями, обсуждаемыми ниже. Локальный путь состоит из следующей последовательности: буква диска, двоеточие, бэк-слеш, компоненты имени, разделённые бэк-слешами. Например, максимальный путь на диске D имеет вид D:\какие-то-256-символов-пути (и ещё один символ, до 260, занимает терминирующий ноль).

Обратите внимание, что [файловые функции Windows автоматически преобразуют слеш \(/\) в бэк-слеш \(\\) в файловых путях.](#)

В Windows также имеются функции, которые позволяют использовать расширенные пути файлов. Для таких путей ограничение на максимальную длину имени равно 32'767 символов. А каждый компонент в пути ограничен значением, зависящим от файловой системы - как правило, 255 символов. Подобные пути задаются (и трактуются) специальным образом. Для задания такого пути нужно использовать префикс \\?\, например: \\?\D:\очень-длинный-путь или \\?\UNC\server\очень-длинный-путь.

Подобные имена можно использовать только в Unicode-функциях Windows. К ним (именам) следует относиться с осторожностью по двум причинам. Во-первых, обычные программы не смогут получить доступ к файлам и каталогам, имена которых превысят типичное ограничение в MAX_PATH. Во-вторых, UNCW-имена передаются нижележащей файловой системе "как есть", минуя обычный слой нормализации путей. К примеру, / не будет заменён на \, имена .. (две точки) и . (одна точка) не будут являться специальными и не будут разворачиваться в реальные имена каталогов. Вот почему и появляется возможность задавать имена более 260 символов в пути (а также имена с именами, иначе считающимися недопустимыми - скажем, с точкой на конце) - потому что имена передаются файловой системе без обработки, так что слой нормализации не накладывает ограничение в 260 символов (и другие правила файловых имён Windows).

Пространства имён

Префикс имени файла определяет пространство имён, к которому принадлежит путь. Существуют две основные категории пространств имён, используемые в Windows API: **пространства имён NT** и **пространств имён Win32**. Пространство имён NT было разработано как пространство имён низкого уровня, корневым пространством имён, поверх которого могли бы существовать другие пространства имён - включая подсистему Win32 и, как следствие, пространство имён Win32. POSIX является еще одним примером подсистемы в Windows, которая построена поверх NT.

Для исследования пространства имён вы можете использовать [утилиту WinObj](#) от SysInternals.

Данное понятие не следует путать с [пространством имён Оболочки \(Shell\)](#).

Файловые пространства имён Win32

К ним относятся имена, начинающиеся с \\?\ - мы уже разобрали их выше.

Префиксы вида C:\ являются псевдонимами.

Пространства имён устройств Win32

Для доступа к устройствам вместо физических файлов используется пространство имён устройств. Для указания пути при этом используется префикс \\.\ (два бэк-слеша, точка, бэк-слеш). К примеру, так вы можете получить доступ к диску как физическому устройству, без обращения к файловой системе. Но, конечно же, "устройства" не ограничиваются только дисками.

К примеру, если вы хотите открыть порт последовательной связи номер 1, то вы можете использовать имя COM1 в вызове [функции CreateFile](#). Это работает, потому что COM1-COM9 являются частью зарезервированных имён в пространстве имён NT. Это работает как псевдоним на устройство, хотя вы можете и явно указывать префикс \\.\. Для сравнения: если вдруг у вас есть сто COM-портов и вам надо обратиться к 56-му COM-порту, то вы не сможете открыть его по имени COM56 - потому что для него нет никакого предопределённого псевдонима или резервирования. Вам нужно будет открыть его по имени \\.\COM56.

Пространства имён NT

Существуют также API функции, которые позволяют использовать именование в стиле NT, но в большинстве случаев это не нужно. Для наиболее востребованных объектов создаются ссылки (псевдонимы), чтобы к ним можно было получить доступ, используя обычные функции. К примеру, к пространству имён NT относятся такие вещи как Serial0 и Serial1, HarddiskVolume1 и Harddisk0, но обычно с ними работают через пространство имён Win32, используя такие имена как C: и \\.\PhysicalDrive0.

Как уже было сказано, другие пространства имён реализуются поверх пространства имён NT. К примеру, для реестра в корне создаётся элемент REGISTRY, объекты ядра находятся в KernelObjects, про устройства и файлы Win32 я уже говорил, тут же находятся и сессии и, скажем, глобальные и локальные имена объектов IPC и так далее.

Напоминаю, что вы можете использовать [утилиту WinObj](#) для просмотра пространств имён.

На этом я заканчиваю рассказ про файлы и перехожу к собственно сериализации данных.

Теги [Delphi](#), [начинающим](#), [Статья](#)

3 комментария:



Fr0sT 28 декабря 2011 г., 10:41

Про длинные пути и пространства имен устройств слышал краем уха, было интересно почитать глубже, спасибо!

[Ответить](#)



Александр Алексеев  28 декабря 2011 г., 10:48

Кстати, спасибо за исправления (если это ты их делаешь) :)

[Ответить](#)



Fr0sT 28 декабря 2011 г., 14:45

Пожалуйста :)

[Ответить](#)

Введите комментарий...

Подпись комментария: ▼

[Публикация](#) [Просмотр](#)

Можно использовать некоторые HTML-теги, например:

`Жирный`

`<i>Курсив</i>`

`Ссылка`

Вам необязательно регистрироваться для комментирования - для этого просто выберите из списка "Анонимный" (для анонимного комментария) или "Имя/URL" (для указания вашего имени и (опционально) ссылки на сайт). Все прочие варианты потребуют от вас входа в вашу учётку (поддерживается OpenID).

Пожалуйста, по возможности используйте "Имя/URL" вместо "Анонимный". URL можно просто не указывать.

Ваше сообщение может быть помечено как спам спам-фильтром - не волнуйтесь, оно появится после проверки администратором.

Ссылки

[Создать ссылку](#)

[Следующее](#) • • • • • [Главная страница](#) • • • • • [Предыдущее](#)

Подписаться на: [Комментарии к сообщению \(Atom\)](#)

Поиск по блогу

Google™ Пользовательский поиск

Поиск

Архив

Архив ▼

Тэги

Delphi (185) [Статья \(72\)](#)
[задачи \(38\)](#) [ты можешь это сделать \(30\)](#) [начинающим \(26\)](#)
[обработка ошибок \(26\)](#) [случайные мысли \(26\)](#) [EurekaLog \(22\)](#) [блог \(17\)](#) [прочее \(16\)](#) [Windows \(14\)](#) [не делай так \(11\)](#) [роботы/киберпанк \(9\)](#) [журнал \(6\)](#) [7 \(4\)](#) [Tiburon \(4\)](#) [Vista \(4\)](#) [Королевство Delphi \(4\)](#) [TasksEx \(3\)](#) [x64 \(2\)](#) [Коты \(2\)](#) [кроссплатформенность \(1\)](#) [работа \(1\)](#)

Подписка



Шаблон "Simple". Технологии [Blogger](#).