Денис Колисниченко

СЕКРЕТЫ, НАСТРОЙКА И ОПТИМИЗАЦИЯ РЕЕСТРА

Windows 7

- Аудит и мониторинг
- Резервное копирование
- Восстановление системы
- Тонкая настройка системы и приложений
- Групповые политики
- Твикеры
- Программы для оптимизации реестра

Денис Колисниченко

СЕКРЕТЫ, НАСТРОЙКА И ОПТИМИЗАЦИЯ РЕЕСТРА Windows 7

Санкт-Петербург
«БХВ-Петербург»
2010

УДК 681.3.06
ББК 32.973.26-018.2
К60

Колисниченко Д. Н.

К60 Секреты, настройка и оптимизация реестра Windows 7. — СПб.: БХВ-Петербург, 2010. — 320 с.: ил.

ISBN 978-5-9775-0488-1

Рассмотрено устройство, настройка и оптимизация реестра, секреты и трюки при работе с ним, параметры популярных Windows-приложений. Описаны программы для мониторинга, чистки и быстрой настройки реестра, которые пригодятся каждому пользователю. Для администраторов систем даны приемы управления реестром (политики, списки доступа), использования Windows Installer, тонкая настройка системы и приложений, примеры действий в различных нештатных ситуациях. Некоторые настройки реестра, приведенные в этой книге, будут работать не только в Windows 7, но и в Windows Vista и Windows XP.

Для широкого круга пользователей Windows

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Ольга Кокорева*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Дизайн обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

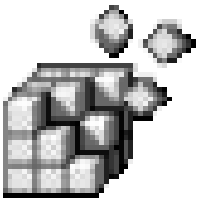
Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.10.09.
Формат 70×100$^{1}/_{16}$. Печать офсетная. Усл. печ. л. 25,8.
Тираж 1500 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0488-1

Оглавление



ГЛАВА 1

Основы реестра

1.1. Что такое реестр и для чего он используется?

Все версии Windows, начиная с Windows 95, хранят как свои собственные настройки, так и настройки большинства приложений в реестре. Реестр можно рассматривать как конфигурационную базу данных Windows.

Многие пользователи считают, что реестр — далеко не самая важная часть системы, поскольку она им не видна. Однако это не так. Да, на первый взгляд роль реестра по отношению к пользователям пассивна: они не замечают его работы и поэтому не осознают его важности.

Действительно, редактируя документы или бороздя просторы Интернета, пользователь непосредственно не сталкивается с реестром. Зато операционная система с ним работает непрерывно. Если запустить программу мониторинга реестра (в этой книге мы рассмотрим такие программы), то вы увидите, что практически при любом действии — будь то запуск программы или переход в другую папку в окне Проводника (Windows Explorer) — происходит обращение к реестру.

Опытные пользователи, знакомые со структурой реестра, могут очень тонко настраивать свою систему, потому что путем редактирования реестра можно выполнить многие настройки, недоступные через графический интерфейс пользователя (Graphical User Interface, GUI). Например, через Панель управления (Control Panel) вы никак не сможете скрыть те или иные вкладки окна параметров Internet Explorer, не сможете отключить дефрагментацию загрузочных файлов, которая выполняется при каждой загрузке компьютера, тормозя запуск системы, и т. д.

Вы можете спросить: а зачем обычному пользователю вообще нужно знать о реестре? Ведь не зря разработчики Windows "убрали" его подальше от глаз пользователей. Действительно, в Windows можно работать, не обращая внимания на реестр, а при настройке системы довольствоваться Панелью управления (Control Panel). Но в один не очень прекрасный момент Windows может дать сбой из-за повреждения реестра: записи в него некорректной информации или удаления необходимых данных (например, вирусом). Что делать? Можно переустановить Windows и все приложения, потратив на это целый день, а можно просто восстановить реестр, что займет не более получаса (разумеется, если у вас есть под рукой все, что для этого необходимо). Выходит, не только программистам и системным администраторам, но и обычным пользователям нужно знать, как минимум, что такое реестр и как выполнять его резервное копирование и восстановление в случае сбоя. Но если мы знаем, что такое реестр, то можно не останавливаться на полпути, а освоить хотя бы минимальные навыки работы с ним. Мне, например, намного удобнее запустить программу regedit.exe, найти раздел `Run`, отвечающий за автозапуск программ, и удалить из него все ненужное, чем использовать для этой цели какую-то специальную программу, будь то встроенная программа Windows Msconfig.exe или, например, какая-нибудь сторонняя утилита наподобие Starter (**http://codestuff.tripod.com/products_starter.html**). При этом вашей любимой программы от стороннего производителя может просто не оказаться под рукой, так же, как и доступа в Интернет, откуда можно было бы ее скачать. А вот редактор реестра regedit.exe, который мы рассмотрим в *главе 2*, входит в состав операционной системы, предоставляет более широкие возможности, нежели встроенные графические утилиты, и в умелых руках может творить чудеса.

Но редактирование раздела `Run` — это лишь самое тривиальное действие, которое можно выполнить с помощью приложения regedit.exe. Пользователи, по долгу службы занимающиеся администрированием компьютерных систем или желающие стать администраторами, наверняка оценят политики безопасности, о которых мы тоже поговорим в этой книге.

1.2. Краткая история реестра

Как мы помним, первой операционной системой для персональных компьютеров от Microsoft была MS-DOS. В этой операционной системе было два основных конфигурационных файла: config.sys и autoexec.bat. Первый из этих файлов содержал инструкции по загрузке драйверов и резидентных программ. В autoexec.bat указывались команды, которые выполнялись при загрузке MS-DOS, например, устанавливались переменные окружения, а также вызывались оболочки наподобие Norton Commander.

Кроме config.sys и autoexec.bat в MS-DOS не было ни других общесистемных конфигурационных файлов, ни реестра. Каждое приложение хранило свои настройки в отдельном файле, формат и местонахождение которого был известен только самому этому приложению. У одних приложений конфигурационные файлы были текстовыми (их можно было редактировать вручную в любом текстовом редакторе), у других — двоичными (такие файлы можно было редактировать только с помощью самого приложения, которое "знало" формат файла).

MS-DOS не устраивала пользователей своей однозадачностью и отсутствием дружественного пользовательского интерфейса. Многие сторонние разработчики выпускали свои *оболочки* для MS-DOS, облегчающие для пользователя процесс работы с операционной системой. Microsoft тоже не осталась в стороне, разработав собственную оболочку, которая получила название Windows. Первые версии Windows, по мнению многих довольно авторитетных пользователей, вообще не заслуживали внимания. Более или менее удачной стала только третья версия Windows — Windows 3.0. В этой версии для хранения настроек системы использовались INI-файлы, которые, однако, имели массу недостатков. Главным среди них была так называемая "плоская" структура — в INI-файлах не допускалось создание вложенных разделов (в отличие от современного реестра Windows, имеющего иерархическую древовидную структуру). Во-вторых, INI-файлы были текстовыми, что затрудняло хранение в них двоичной информации. С другой стороны, это позволяло редактировать INI-файлы в любом текстовом редакторе, чего нельзя сделать с современным реестром. Нужно отметить также, что INI-файлы имели единый формат для хранения настроек Windows-приложений. Ведь намного проще использовать уже известный формат и готовые API-функции для работы с ним, чем заново "изобретать велосипед", придумывая собственный формат конфигурационных файлов. Некоторые программы и до сих пор используют не реестр, а INI-файлы.

В Windows 3.1 впервые появилось некое подобие реестра, но он использовался только для хранения настроек механизма OLE (Object Linking and Embedding), а все остальные настройки системы по-прежнему хранились в INI-файлах.

С появлением Widows 95 появился и реестр в сегодняшнем понимании этого слова. Конечно, в последующих версиях Windows (Windows 2000/XP/Vista) структура реестра была изменена. Тем не менее, реестр Windows 95 уже был максимально похож на современный, несмотря на то, что многие приложения по-прежнему использовали INI-файлы для хранения своих настроек.

Реестры современных версий Windows (2000, XP, Vista, Windows 7) в значительной степени схожи, но все же у каждого есть свои отличия. Данная книга

ориентирована на новейшие версии Windows — Vista и Windows 7, поэтому об отличиях в Windows 2000 мы говорить не будем. Далее будет указываться, к какой из версий — Windows Vista или Windows 7 — относится сказанное, если же версия не уточняется, то сказанное справедливо для обеих систем.

1.3. Что нужно знать для работы с реестром?

Работа с реестром заключается в редактировании значений параметров реестра, которые чаще всего представлены в виде текстовых строк, а также чисел в десятичной и других системах счисления. Кроме того, вам пригодятся знания идентификаторов безопасности (Security IDs, SIDs), глобальных идентификаторов (Globally Unique IDs, GUIDs) и некоторых других объектов реестра, которые будут рассмотрены в этом разделе.

1.3.1. Системы счисления

Помимо известной нам со школы десятичной системы счисления существует множество других систем счисления. В первую очередь нас будут интересовать те из них, которые получили широкое распространение в компьютерных технологиях: двоичная (binary), использующая только две цифры — `0` и `1`, восьмеричная (octal), использующая цифры от `0` до `7`, и шестнадцатеричная (hex), где применяются цифры от `0` до `9` и буквы латинского алфавита от `A` до `F`. В реестре Windows активно используются только две: десятичная и шестнадцатеричная. С первой системой мы все знакомы, тогда как вторая, вероятно, нуждается в некоторых пояснениях.

В десятичной системе используются десять цифр: от 0 до 9, поэтому она и называется десятичной. Если вы не прогуливали уроки математики, то должны знать, что любое N-значное десятичное число можно представить следующим образом:

$$A = A_1 \times 10^{N-1} + A_2 \times 10^{N-2} + ... + A_N \times 10^0$$

Исходя из этой формулы, можно написать более общее выражение, подходящее для любой системы счисления:

$$A = A_1 \times B^{N-1} + A_2 \times B^{N-2} + ... + A_N \times B^0,$$

где B (от base) — это основание системы счисления. В случае с десятичной системой B = 10.

Например, число 453 можно представить так:

$$453 = 4 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 = 4 \times 100 + 5 \times 10 + 3 \times 1 = 400 + 50 + 3 = 453$$

Теперь поговорим о шестнадцатеричной системе. В этой системе шестнадцать цифр:

`0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F`

Цифры `A`, `B`, `C`, `D`, `E` и `F` соответствуют числам 10, 11, 12, 13, 14 и 15 десятичной системы.

Вернемся к только что рассмотренной формуле, позволяющей представить число в любой системе счисления. Используя ее, вы можете с легкостью преобразовывать шестнадцатеричные числа в десятичные. Рассмотрим, например, преобразование в десятичную систему числа `AF`:

$A \times 16^1 + F \times 16^0 = 10 \times 16 + 15 = 175$

Проверку можно выполнить при помощи обычного калькулятора Windows 7. Запустите приложение Калькулятор (Calculator) — кстати, обратите внимание, что даже это простейшее приложение в Windows 7 оказалось дополненным целым рядом приятных мелочей — а затем из меню **Вид** (View) выберите команду **Программирование** (Programmer). Установите переключатель системы счисления в положение **Hex** (шестнадцатеричная), с помощью кнопок калькулятора или клавиш клавиатуры введите число `AF`, после чего установите переключатель системы в положение **Dec** (десятичная). В результате выполненных действий получаем 175 (рис. 1.1).

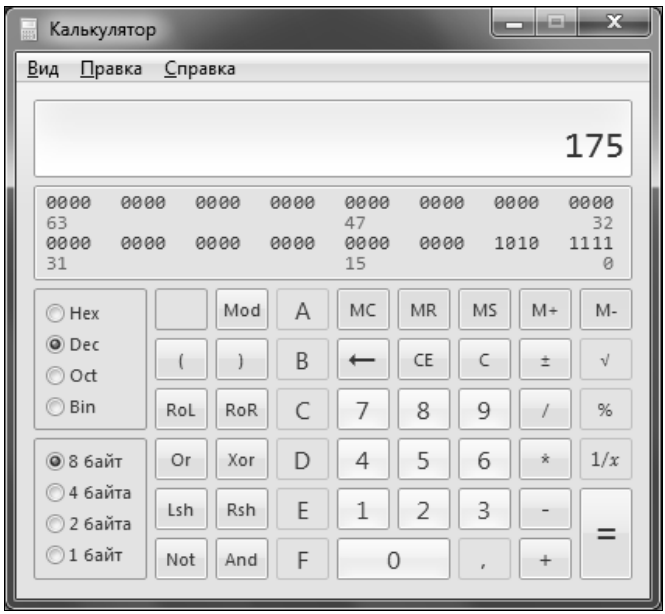


Рис. 1.1. Использование калькулятора для выполнения вычислений в шестнадцатеричной системе

Шестнадцатеричные числа часто записываются так: `0x`*число*. Например, запись `0x77` означает, что число 77 записано в шестнадцатеричной системе. Очевидно, что оно не равно числу 77 в десятичной системе: после преобразования `0x77` в десятичную систему мы получим число 119.

Иногда для указания того, что число записано в шестнадцатеричной системе вместо префикса `0x` добавляют суффикс `h`: `77h`.

Рассмотрим теперь порядок следования байтов в шестнадцатеричном числе. Для числа `0xA1FF` `0xA1` — это старший байт, а `0xFF` — младший байт. Левый байт называется старшим, поскольку вы умножаете его значение на более высокую степень числа 16.

В зависимости от архитектуры микропроцессора, для которой они изначально разрабатывались, одни программы хранят числа в таком порядке следования байтов, когда младший байт сохраняется по младшему адресу, а старший — по старшему (в англоязычной литературе он называется Little-Endian, или формат "остроконечников"), в то время как другие — в порядке Big-Endian, или формат "тупоконечников", иными словами, в порядке "от старшего к младшему". Если используется формат Big-Endian, то первыми сохраняются старшие байты, а затем — младшие. Предположим, что нам нужно сохранить в памяти число `0x00010203`. Если используется порядок "от старшего к младшему", то число будет сохранено в памяти таким образом:

```
0x00 0x01 0x02 0x03
```

Однако процессоры фирмы Intel, например, работают с обратным порядком следования байтов, в котором сначала сохраняются младшие байты, а потом — старшие. Следовательно, наше число `0x00010203` будет сохранено в памяти так:

```
0x03 0x02 0x01 0x00
```

Об этом нужно помнить при работе с программами редактирования реестра, хотя в большинстве случаев они корректно работают как с прямым, так и с обратным порядком следования байтов.

1.3.2. Идентификаторы безопасности

Уникальное имя какого-нибудь объекта называется *идентификатором* (identifier, ID). С помощью идентификаторов можно однозначно выделить объект из множества ему подобных. Например, идентификатором может быть имя пользователя, под которым он регистрируется в системе. Зная имя пользователя, например, `Dennis` (в данном случае строка `Dennis` — идентификатор), вы сможете произвести операции именно с этим пользователем, выделив его из числа других пользователей системы.

В Windows имена пользователей, компьютеров, принимающих участие в работе сети, групп пользователей и других объектов, подчиняются правилам безопасности. Для однозначного определения этих правил используются идентификаторы безопасности — SID (Security Identifier).

Каждый раз, когда создается объект, подчиняющийся правилам безопасности, Windows генерирует SID. Локальные SID (локальные идентификаторы, относящиеся только к данному конкретному компьютеру) генерируются локальными средствами защиты (LSA, Local Security Authority) и хранятся в локальной базе данных.

Кроме локальных средств защиты, есть еще средства защиты домена (Domain Security Authority). DSA генерируют идентификаторы безопасности для домена и сохраняют их не в локальной базе данных, а в Active Directory (службе каталогов) на контроллере домена.

Понятно, что локальные SID уникальны в пределах компьютера (в пределах локальной базы данных), а SID домена уникальны в пределах домена (базы данных Active Directory). Очевидно также, что локальные SID на разных компьютерах сети могут совпадать, так же как в разных доменах могут существовать одинаковые доменные SID.

Локальные SID никогда не повторяются. Предположим, в системе зарегистрирован пользователь `Dennis`. Его учетной записи будет сопоставлен некий SID. Если вы удалите эту учетную запись, а затем создадите новую учетную запись с таким же именем, то SID у этой учетной записи будет другой.

К учетной записи в Windows можно обратиться как по ее имени, так и по SID, поскольку SID однозначно идентифицирует учетную запись. Но обращаться по SID к учетной записи крайне неудобно, поскольку выражения SID достаточно громоздки, например:

```
S-1-5-21-2052111302-436374069-1343024091-1003
```

Очевидно, намного проще запомнить имя `Dennis`, чем приведенный SID, однако формат SID все равно нужно знать. SID всегда начинается с буквы `S`, после которой следует номер версии SID, обычно `1`. Далее, как правило, стоит число `5`, что означает систему NT (NT authority). Все последующие числа (`21-2052111302-436374069-1343024091`) являются идентификатором домена, а последнее число (`1003`) — идентификатором группы, к которой принадлежит данный пользователь.

Помимо персональных учетных записей пользователей в Windows есть постоянные или "короткие SID": они одинаковы на всех компьютерах. Знать эти SID просто необходимо, поскольку они часто встречаются в реестре. В табл 1.1 приведен список некоторых так называемых "широко известных" SID (well-known SIDs). Подробные списки "широко известных" SID и более

детальную информацию о них можно найти здесь: **http://support.microsoft.com/kb/243330**[1].

Таблица 1.1. *Некоторые постоянные SID*

SID	Пользователь или группа
S-1-0	Нет полномочий, "пустые" полномочия, соответствует имени пользователя "nobody" ("никто")
S-1-0-0	Тоже пустые полномочия, нет участника безопасности
S-1-1	Полномочия мира, так называемый "международный администратор"
S-1-1-0	Все. Группа, в которую входят все пользователи, даже анонимные пользователи и гости
S-1-2	Локальные полномочия, так называемый "локальный администратор"
S-1-3	Администратор-создатель (Creator)
S-1-3-0	Создатель/владелец (Creator/Owner)
S-1-3-1	Группа создателя
S-1-3-2, S-1-3-3	Создатель-владелец сервер и группа-создатель сервер соответственно. Используются в серверных версиях ОС Windows
S-1-3-4	Права владельца. Используется для управления правами владельца над объектом безопасности
S-1-4	Неуникальные полномочия
S-1-5	Администратор NT
S-1-5-1	Удаленный доступ. Группа, в которую входят все пользователи, вошедшие в систему с использованием удаленного доступа
S-1-5-2	Сеть. К этой группе относятся все пользователи, вошедшие в систему с использованием сетевого подключения
S-1-5-3	Партия. Группа, в которую входят все пользователи, вошедшие в систему с использованием средства пакетной очереди
S-1-5-4	Интерактивный. К этой группе относятся пользователи, вошедшие в систему с использованием интерактивного входа
S-1-5-5-*X*-*Z*	Сеанс входа в систему. Значения *X* и *Z* для этих идентификаторов SID меняются в каждом сеансе

[1] См. также следующие адреса, по которым можно найти подробную информацию для углубленного изучения:
http://www.registrycleanersreviews.info/list-of-well-know-registry-sids, **http://www.windowsconfiguration.com/2007/04/well_known_sids.html**, **http://www.winzero.ca/WEllKnownSIDs.htm**, **http://msdn.microsoft.com/en-us/library/aa379649(VS.85).aspx**. — *Прим. ред.*

Таблица 1.1 *(окончание)*

SID	Пользователь или группа
S-1-5-6	Служба (сервис)
S-1-5-7	Anonymous (анонимный пользователь)

1.3.3. Глобальные идентификаторы

Полное название глобальных идентификаторов — глобальные уникальные идентификаторы (Global Unique Identifier, GUID). GUID — это число, однозначно идентифицирующее какой-либо объект: компьютер, аппаратное устройство, программный компонент. GUID жестко привязывается к объекту: имя объекта можно изменить, а GUID — нет. GUID очень похожи на SID, но выполняют несколько иную роль: GUID никак не связаны с безопасностью и правами доступа.

Формат GUID, в отличие от формата SID, одинаков для всех объектов. GUID — это 16-байтное шестнадцатеричное число, разбитое на группы, состоящие из 8, 4, 4, 4 и 12 шестнадцатеричных цифр, соответственно. Группы в составе GUID отделяются друг от друга дефисами, а весь GUID заключен в фигурные скобки, например:

```
{645FF040-5081-101B-9F08-00AA002F954E}
```

Для создания GUID используется утилита guidgen.exe. Microsoft гарантирует, что сгенерированный GUID будет уникальным в пределах системы. Прочитать о том, как использовать guidgen.exe можно по следующему адресу:

http://msdn2.microsoft.com/en-us/library/ms241442(VS.80).aspx

1.3.4. Использование битовых масок

Как мы знаем, при использовании формата ASCII для представления одного символа используется один байт. Таким образом, слово "байт" занимает 4 байта (4 символа). В одном байте восемь битов, каждый из которых может принимать значение `0` или `1`.

Пойдем дальше. Возьмем любой символ, например, 1. В ASCII-таблице этому символу соответствует код 49. Переведем 49 в двоичную систему и получим вот такое число:

```
0011 0001
```

Зачем нам это все нужно знать? Дело в том, что некоторые простые настройки в реестре Windows хранятся в виде однобайтных значений.