

9 Установка и удаление компонентов операционной системы

Одной из задач администратора является удаление ненужных или установка дополнительных компонентов операционной системы. Именно о дополнительных компонентах операционной системы Windows Vista и работе с ними пойдет речь в этой главе.

Особенно актуально вопрос удаления ненужных компонентов стоит перед пользователями Windows Vista Ultimate, которым приходится жертвовать почти 10 Гбайт объема жесткого диска на нужды операционной системы.

9.1. Компоненты Windows

Расположение: %systemroot%\system32\optionalfeatures.exe.

Самым простым способом установки и удаления компонентов операционной системы является использование окна Компоненты Windows (рис. 9.1). Это окно можно вызвать либо из папки Панель управления (значок Программы и компоненты, затем ссылка Включение или отключение компонентов Windows), либо с помощью исполняемого файла optionalfeatures.exe.

Надо заметить, что при работе с данным окном операционная система никогда не просит вставить загрузочный диск для копирования с него файлов устанавливаемого компонента. Это говорит о том, что все компоненты операционной системы устанавливаются вместе с ней и расположены на диске в сжатом виде (в папке %systemroot%\winsxs).

Компоненты операционной системы

В Windows Vista Ultimate окно Компоненты Windows содержит различные компоненты операционной системы, которые можно установить или удалить. Заметьте, что для каждого элемента окна Компоненты Windows приводится название компонента операционной системы, который устанавливается или удаляется с его помощью. Эти сведения могут понадобиться в будущем.

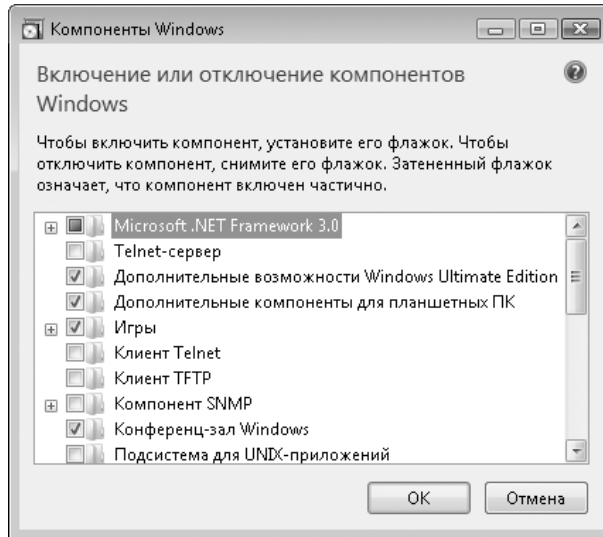


Рис. 9.1. Окно Компоненты Windows

В табл. 9.1. представлен перечень компонентов операционной системы, их внутренние имена, а также версии Windows Vista, в которых они установлены по умолчанию. При описании версий Windows Vista, в которых компонент установлен по умолчанию, используются следующие сокращения:

- Да — по умолчанию компонент установлен;
- Нет — по умолчанию компонент не установлен;
- - — в этой версии Windows данный компонент недоступен;
- <название компонента> — по умолчанию установлен именно данный дочерний компонент, остальные же дочерние компоненты не установлены.

Таблица 9.1. Компоненты Windows Vista

Название	Имя	Ultimate	Enterprise	Home Premium
Microsoft .NET Framework 3.0	NetFx3	XPS Viewer	XPS Viewer	XPS Viewer
Служба установщика ActiveX	AxInstallService	Нет	Нет	-
Служба репликации DFS	DFSR-Infrastructure-ClientEdition	Да	Да	Да
Факсы и сканирование Windows	FaxServicesUltimate	Да	Нет	-
Службы IIS	IIS-WebServerRole	Нет	Нет	Нет
Служба индексирования	Indexing-Service-Package	Нет	Нет	Нет
Сервер очереди сообщений (MSMQ).	MSMQ-Container	Нет	Нет	Нет
Службы для NFS	ServicesForNFS-ClientOnly	Нет	Нет	-
Конференц-зал Windows	Adhoc-Meetings	Да	Да	Да

Название	Имя	Ultimate	Enterprise	Home Premium
Службы печати	Printing-Foundation-Features	Клиент интернет-печати	Клиент интернет-печати	Клиент интернет-печати
Прослушиватель RIP	RasRip	Нет	Нет	Нет
Удаленное разностное сжатие	MSRDC-Infrastructure	Да	Да	Да
Управление съемными носителями	Microsoft-Windows-RemovableStorageManagement	Нет	Нет	Нет
Игры	InboxGames	Да	Нет	Да
Простые службы TCP/IP (такие как echo, daytime и т. п.)	SimpleTCP	Нет	Нет	Нет
Компонент SNMP	SNMP	Нет	Нет	Нет
Подсистема для UNIX-приложений	SUA	Нет	Нет	-
Дополнительные компоненты для планшетных ПК	TabletPCOC	Нет	Нет	Нет
Клиент Telnet	TelnetClient	Нет	Нет	Нет
Telnet-сервер	TelnetServer	Нет	Нет	Нет
Клиент TFTP	TFTP	Нет	Нет	Нет
Дополнительные возможности Windows Ultimate Edition	Windows-Ultimate-Extras	Да	-	-
Служба активации Windows	-	Нет	Нет	Нет

Теперь подробнее рассмотрим некоторые компоненты операционной системы.

Компонент Microsoft .NET Framework 3.0

Microsoft .NET Framework 3.0, также называемый WinFX, представляет собой набор API, который включает в себя подсистему презентаций Windows Presentation Foundation (бывшую Avalon), платформу для веб-сервисов (бывшую Indigo), а также Windows Workflow Foundation и Windows CardSpace (вспомните одноименный значок папки **Панель управления**). Все это является фундаментом для приложений следующего поколения, разрабатываемых специально для Windows Vista.

Данный компонент состоит, в свою очередь, из трех компонентов: **Windows Communication Foundation HTTP Activation** (название компонента — WCF-HTTP-Activation), **Windows Communication Foundation Non-HTTP Activation** (название компонента — WCF-NonHTTP-Activation) и **XPS Viewer** (название компонента — XPS-Viewer).

Первые два по умолчанию не установлены, а третий установлен. Ранее при описании работы с принтерами в операционной системе Windows Vista обсуждались основы нового формата XPS и стандартного принтера операционной системы Microsoft XPS Document Writer. Данный принтер позволял создавать файлы в XPS-формате, которые потом можно было просмотреть с помощью компонента XPS Viewer. Этот компонент является частью браузера Internet Explorer 7.0.

Если вы удалите компонент XPS Viewer, то не сможете просматривать файлы в XPS-формате, хотя по-прежнему сможете создавать их с помощью стандартного принтера Microsoft XPS Document Writer.

Компонент Telnet-сервер

Он создает службу **Telnet** (файл `tlntsvr.exe`), которая позволяет удаленному пользователю подключаться к данному компьютеру и управлять им с помощью программы `telnet.exe`. Служба расположена в подразделе реестра `TlntSvr`, запускается от имени локальной службы, но с дополнительными привилегиями `SeAssignPrimaryTokenPrivilege`, `SeAuditPrivilege`, `SeChangeNotifyPrivilege`, `SeCreateGlobalPrivilege`, `SeImpersonatePrivilege` и `SeIncreaseQuotaPrivilege`.

ПРИМЕЧАНИЕ

Для сетевого взаимодействия службы при установке данного компонента также создается несколько правил стандартного брандмауэра.

Параметры настройки службы **Telnet** хранятся в ветви реестра `HKLM\SOFTWARE\Microsoft\TelnetServer\Defaults`. Например, данная ветвь реестра содержит следующие параметры `REG_DWORD`-типа.

- `DefaultShell` — оболочка, к которой будет выполняться подключение удаленного пользователя (строковый тип). По умолчанию значение этого параметра равно `%systemroot%\system32\cmd.exe`.
- `DisconnectKillAllApps` — определяет, будут ли автоматически завершаться все запущенные пользователем задачи при завершении его сеанса. По умолчанию значение параметра равно 1.
- `LogFile` — путь к файлу, в который будут записываться сведения о работе службы **Telnet** (строковый тип). По умолчанию файл журнала не используется.
- `LoginScript` — путь к файлу сценария, который будет выполняться для всех пользователей, которые подключаются к компьютеру (строковый тип). По умолчанию параметр равен `%WINDIR%\system32\login.cmd`.
- `MaxConnections` — максимальное количество разрешенных одновременных подключений к данному компьютеру. По умолчанию параметр равен 2.
- `TelnetPort` — номер порта, используемый службой **Telnet** для работы с удаленными пользователями. По умолчанию значение параметра равно 17.
- `MaxFailedLogins` — количество неудачных подключений к компьютеру, после достижения которого попытки подключения удаленного пользователя будут блокироваться. По умолчанию значение параметра равно 3.

Кроме того, данный компонент создает такие файлы: `tlntsvr.exe`, `tlntsess.exe`, `tlntadm.exe`, `termcap` и `login.cmd`.

Файл `tlntadm.exe` представляет собой программу командной строки, с помощью которой можно настроить работу сервера **Telnet** либо управлять ею (запускать, останавливать и т. д.). Есть несколько основных синтаксисов данной программы. Во всех них для аутентификации пользователя можно также указывать следующие параметры: `<имя компьютера> -u <имя пользователя> -p <пароль>`.

- `tlntadm.exe -s|-k|-m <идентификатор сессии>` — отображает список активных сессий, прекращает одну из них или посылает сообщение пользователю, инициировавшему сессию, соответственно.
- `tlntadm.exe start|stop|pause|continue` — управляет работой сервера Telnet.
- `tlntadm.exe <параметры настройки>` — настраивает параметры работы сервера Telnet. Описание данных параметров можно просмотреть, введя команду `tlntadm.exe /?`.

Файл `termcap` представляет собой обычный текстовый файл, содержащий набор команд, заканчивающих сессию.

Файл `login.cmd` представляет собой обычный текстовый файл, в котором содержится только одна основная команда: `cd /d %HOMEDRIVE%\%HOMEPATH%`.

Дополнительные компоненты для планшетных ПК

Данный компонент представляет собой набор файлов для работы с пером и другими элементами планшетных компьютеров. Например, такими файлами являются программы `StickyNot.exe`, `SnippingTool.exe`, а также большинство программ из раздела `%ProgramFiles%\Common Files\microsoft shared\ink`. Общий размер данных файлов составляет примерно 170 Мбайт.

Игры

Данный компонент представляет собой набор игр из папки `%programfiles%\Microsoft Games`, а также библиотеку `CardGames.dll`. Он занимает примерно 111 Мбайт свободного места на жестком диске.

При этом можно удалить как все стандартные игры операционной системы, так и только некоторые из них. Игры также считаются компонентами операционной системы и имеют следующие названия: `Solitaire`, `SpiderSolitaire`, `Hearts`, `FreeCell`, `Minesweeper`, `PurplePlace`, `CardGames`, `Chess`, `Shanghai` и `Inkball`.

Клиент Telnet

Данный компонент устанавливает файл `telnet.exe` и библиотеку `telnetcr.dll`.

Программа `telnet.exe` позволяет подключиться к службам удаленного компьютера, на котором установлен сервер Telnet, для управления их работой. Основной синтаксис программы следующий: `telnet.exe <IP-адрес компьютера> <порт>`. При подключении можно также использовать дополнительные параметры:

- `-a -l <пользователь>` — при подключении к службе автоматически на ней регистрироваться;
- `-f <путь к файлу журнала и его имя>` — имя файла, в который будут заноситься сведения о работе с сервером Telnet.

Клиент TFTP

Он представляет собой единственный исполняемый файл — `tftp.exe`. Этот файл должен быть вам знаком по операционной системе Windows XP. Если в ней он присутствовал по умолчанию, то в Windows Vista его нужно устанавливать отдельно.

Данная программа предназначена для передачи файлов на удаленный компьютер (или наоборот). Ее основной синтаксис следующий: `tftp <IP-адрес удаленного компьютера> GET|PUT <имя передаваемого файла>`. Если используется параметр `GET`, то файл передается на удаленный компьютер. Если же используется параметр `PUT`, то файл передается на локальный компьютер.

Компонент SNMP

С помощью данного компонента можно установить файлы, предназначенные для работы со службой SNMP и агентами SNMP.

Добавляемые службы

Установка данного компонента приводит к созданию служб **SNMP Service** (файл `snmp.exe`) и **SNMP Trap Service** (файл `snmptrap.exe`). Первая из этих служб предназначена для обработки запросов SNMP на локальном компьютере. Вторая же позволяет собирать сведения от агентов SNMP и передавать их программам, которые ожидают данные сведения.

Служба **SNMP Service** определяется в подразделе `snmp`, запускается автоматически с правами локальной системы и без дополнительных привилегий.

ПРИМЕЧАНИЕ

При установке компонента создаются несколько правил стандартного брандмауэра Windows, разрешающих доступ к сети службе **SNMP Service**.

Вы можете настроить работу данной службы довольно экстравагантным способом — с помощью консоли `services.msc`. Как вы уже знаете, данная консоль отображает список всех служб, зарегистрированных в операционной системе. При этом если в контекстном меню определенной службы вы выберете команду **Свойства**, то откроется окно, описывающее настройки службы. Это окно содержит четыре вкладки для всех служб: **Общие**, **Вход в систему**, **Восстановление** и **Зависимости**, но только не для службы **SNMP Service**. Окно данной службы содержит дополнительные вкладки, описанные ниже. Эти вкладки добавляются с помощью расширения, устанавливаемого вместе с компонентом **Компонент SNMP**.

- **Agent** — хранит сведения об агентах SNMP.
- **Traps** — содержит сведения об элементах, собирающих данные от агентов SNMP.

Данная вкладка редактирует параметры подразделов ветви реестра `HKLM\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\TrapConfiguration`.

- **Безопасность** — с помощью данной вкладки можно добавить новые сообщества, указав их права доступа, а также указать компьютеры, от которых можно получать SNMP-пакеты.

Новые сообщества описываются в виде строковых параметров ветви реестра `HKLM\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ValidCommunities`.

Имена компьютеров, от которых можно получать SNMP-пакеты, хранятся как параметры строкового типа ветви реестра `HKLM\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\PermittedManagers`.

Другим способом настройки работы SNMP является применение групповых политик. Все они описаны в файле `Snmp.admx` и расположены в подразделе **Конфигурация компьютера** ▶ **Административные шаблоны** ▶ **Сеть** ▶ **SNMP**.

- **Сообщества** — эта групповая политика позволяет определить сообщества SNMP. Все сообщества, определяемые ей, заносятся в ветвь реестра `HKLM\SOFTWARE\Policies\SNMP\Parameters\ValidCommunities`.
- **Разрешенные диспетчеры** — дает возможность указать компьютеры, от которых можно получать SNMP-пакеты. Все компьютеры, определяемые ей, заносятся в ветвь реестра `HKLM\SOFTWARE\Policies\SNMP\Parameters\PermittedManagers`.
- **Ловушки только для публичного сообщества** — позволяет определить ловушки, собирающие данные от агентов SNMP для публичного сообщества. Все ловушки, определяемые ей, заносятся в ветвь реестра `HKLM\SOFTWARE\Policies\SNMP\Parameters\TrapConfiguration\public`.

Служба **SNMP Trap Service** определяется в подразделе `SNMPTRAP` и запускается вручную с правами локальной службы, а также с дополнительной привилегией `SeChangeNotifyPrivilege`.

Добавляемые файлы

Данный компонент также устанавливает следующие библиотеки:

- `evntagnt.dll` — определяется в параметре строкового типа `Pathname` ветви `HKLM\SOFTWARE\Microsoft\SNMP_EVENTS\CurrentVersion`;
- `hostmib.dll` — указывается в параметре строкового типа `Pathname` ветви реестра `HKLM\SOFTWARE\Microsoft\HostMIB\CurrentVersion`;
- `lmmib2.dll` — определяется в параметре строкового типа `Pathname` ветви `HKLM\SOFTWARE\Microsoft\LANManagerMIB2Agent\CurrentVersion`;
- `snmpmib.dll` — указывается в параметре строкового типа `Pathname` ветви реестра `HKLM\SOFTWARE\Microsoft\SNMPMIB\CurrentVersion`;
- `inetmib1.dll` — определяется в параметре строкового типа `Pathname` ветви `HKLM\SOFTWARE\Microsoft\RFC1156Agent\CurrentVersion`.

Все библиотеки представляют собой агенты SNMP. При этом настройки этих агентов хранятся в отдельных файлах, которые называются `hostmib.mib`, `lmmib2.mib`, `snmpmib.mib` и т. д. соответственно.

Кроме того, компонент устанавливает следующие исполняемые файлы: `snmp.exe`, `evntwin.exe` и `evntcmd.exe`.

Программа `evntcmd.exe` позволяет выполнить конфигурацию работы службы **SNMP Trap Service** с помощью командной строки. Чтобы просмотреть параметры данной программы, нужно воспользоваться командой `evntcmd.exe /?`.

Программа `evntwin.exe` является аналогом `evntcmd.exe`, но, в отличие от нее, предоставляет графический интерфейс настройки службы **SNMP Trap Service**.

Программа `snmp.exe` представляет собой службу **SNMP Service**. При этом данная служба поддерживает следующие параметры запуска: `/Wow64TrapEvent:`, `/Wow64Event64:`, `/Wow64Event32:`, `/Wow64TrapQMutex:`, `/Wow64Mutex:`, `/TrapQSharedMemory:`, `/SharedMemory:`, `/Debug`, `/logtype:` и `/loglevel:`.

Подсистема для UNIX-приложений

С помощью данного компонента можно устанавливать и запускать программы, написанные для операционных систем семейства UNIX. Он устанавливает подсистему POSIX (переносимый интерфейс операционной системы на основе UNIX) на ваш компьютер (в предыдущих версиях операционной системы Windows данная подсистема устанавливалась автоматически).

Нововведением подсистемы POSIX в операционной системе Windows Vista является поддержка 64-разрядных вычислений, возможность подключения к базам данных Oracle и Microsoft SQL Server с помощью интерфейсов Oracle Call Interface (OCI) и Open Database Connectivity (ODBC), а также поддержка двух сред операционной системы UNIX: SVR-5 и BSD.

Но кроме подсистемы POSIX, в SUA входит большинство стандартных программ операционных систем семейства UNIX. Фактически SUA представляет собой дополнительную операционную систему, в которой без перекомпиляции (или с незначительными изменениями) смогут исполняться многие программы, написанные для UNIX.

ПРИМЕЧАНИЕ

Настроить интерфейс ODBC можно с помощью стандартных программ операционной системы `odbcad32.exe` и `odbcconf.exe`.

При своей установке данный компонент создает следующие библиотеки: `posixsscom.dll`, `psxdll.dll`, `psxdllsvr.dll`, `suares.dll`. Кроме того, создается исполняемый файл `psxss.exe` (исполняемый файл подсистемы POSIX), а также папка SUA (в подразделе `%systemroot%`). Данная папка является корневым разделом для файловой системы UNIX и содержит основную структуру данной файловой системы (подразделы и конфигурационные файлы).

ПРИМЕЧАНИЕ

Список всех подсистем (исполняемых файлов, которые их представляют), поддерживающихся операционной системой, содержится в ветви реестра `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\SubSystems`.

При установке данного компонента также регистрируется драйвер `psxdrv.sys`, описываемый в подразделе реестра `PsxDrv`.

Но и это еще не все. Если вы впервые установили подсистему POSIX и никогда раньше ей не пользовались, то обязательно загляните в меню **Пуск** ▶ **Подсистема для UNIX-приложений**. В нем вы найдете ссылку на сайт, с которого можно загрузить программы, работающие в подсистеме POSIX, а также ссылку на файл помощи, подробно описывающий все возможности данной подсистемы и ее назначение.

Прослушиватель RIP

Данный компонент устанавливает библиотеку `iprip.dll`, которая реализует работу новой службы **RIP Listener**. Она предназначена для поддержки обновления таблиц маршрутизации (между маршрутизаторами сети) на основе протокола **RIPv1**.

Данная служба запускается автоматически с правами локальной службы и дополнительной привилегией **SeChangeNotifyPrivilege**.

При установке данного компонента также создается новое правило брандмауэра, которое описывается в параметре строкового типа Rip-Listener-1 ветви HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\RestrictedServices\Configurable\System.

Простые службы TCP/IP (такие как echo, daytime и т. п.)

Данный компонент устанавливает службу Simple TCP/IP Services (файл tcpsvcs.exe), позволяющую компьютеру работать со следующими протоколами:

- Character Generator — генерирует ответный непрерывный поток символов при получении пакета на порт 19;
- Daytime — возвращает дату и время при получении запроса на порт 13;
- Discard — отвергает любой пакет, полученный на порт 9;
- Echo — посылает ответы на все TCP или UDP-запросы, полученные на порт 7;
- Quote of the Day — посылает строку текста в ответ на запрос, полученный на порт 17.

Данная служба описывается в подразделе реестра simptcp, запускается автоматически с правами локальной службы и дополнительными привилегиями SeCreateGlobalPrivilege и SeAuditPrivilege.

Настройка компонента. Вы можете самостоятельно определить те из протоколов, которые будут поддерживаться службой Simple TCP/IP Services. Для этого нужно отредактировать следующие параметры REG_DWORD-типа ветви реестра HKLM\SYSTEM\CurrentControlSet\Services\simptcp\Parameters: EnableUdpDaytime, EnableTcpDaytime, EnableUdpQotd, EnableTcpQotd, EnableUdpChargen, EnableUdpDiscard, EnableTcpChargen, EnableTcpDiscard, EnableUdpEcho и EnableTcpEcho. Названия этих параметров говорят сами за себя.

Можно также воспользоваться параметром строкового типа QotdFileName данной ветви реестра, чтобы указать путь к файлу, который содержит цитаты, отправляемые службой Quote of the Day при получении запроса. По умолчанию значение равно %SystemRoot%\system32\drivers\etc\quotes.

И еще несколько параметров REG_DWORD-типа, которые могут находиться в данной ветви реестра:

- IoBufferSize — определяет размер используемого службой буфера;
- MaxTcpClients — указывает максимальное количество клиентов, которые могут подключиться к службам с помощью протокола TCP. По умолчанию разрешено до 16 клиентов.

Этот компонент также устанавливает библиотеку simptcp.dll.

Сервер очереди сообщений (MSMQ)

Он содержит набор вложенных компонентов, которые представляют собой сервер Microsoft Message Queue (MSMQ) и файлы для работы с ним.

Microsoft Message Queue позволяет клиенту обратиться к службе из серверного приложения COM+, даже если серверное приложение временно недоступно. В этом случае запрос клиента помещается в очередь MSMQ и обрабатывается,

когда серверное приложение станет доступным. Иначе говоря, компонент MSMQ представляет собой некий инструмент промежуточного хранения (стек) самих запросов и адресов, по которым эти запросы нужно доставить в случае обнаружения доступности серверного приложения, а также последовательности, в которой эти запросы были получены и в которой их нужно передать серверному приложению.

MSMQ для создания приложений, работающих с очередями, содержит как специальные API-функции, так и набор ActiveX-объектов, дублирующих возможности API-функций и позволяющих работать с очередями MSMQ с помощью Microsoft Visual Basic, Microsoft Visual Java, Borland Delphi.

Возможность работы с очередями установлена в операционной системе Windows Vista по умолчанию, а с помощью данного компонента можно управлять работой очередей MSMQ.

Данный компонент содержит вложенные дочерние компоненты.

Основные компоненты сервера очереди сообщений (MSMQ)

Компонент содержит в себе следующие дочерние компоненты.

- **MSMQ Active Directory Integration** (название компонента — MSMQ-ADIntegration) — позволяет выполнить интеграцию очереди MSMQ в домен Active Directory.

Он добавляет следующие библиотеки: mqqm.dll, mqrt.dll, mqsnap.dll, mqsec.dll, mqad.dll, mqcertui.dll, mqutil.dll, mqcmplugin.dll, mqoa.dll, mqlogmgr.dll и mqmigplugin.dll.

Кроме того, добавляются исполняемые файлы mqsvc.exe и mqbkup.exe и несколько MOF-файлов (msmqpub.mof, msmqtrc.mof и msmqtrcRemove.mof) для работы с MSMQ с помощью инструментария управления Windows.

Кроме того, в разделе %systemroot%\system32 создается папка **msmq**, хранящая файлы, необходимые для работы MSMQ. Однако на этом установка не заканчивается. После этого в ветвь реестра, предназначенную для автоматического запуска программ при входе пользователей в систему, заносится команда `regsvr32 /s mqrt.dll`.

- **MSMQ HTTP Support** (название компонента — MSMQ-HTTP) — позволяет работать с очередями MSMQ через протокол HTTP. При установке данного компонента также устанавливаются следующие компоненты сервера IIS: IIS-ManagementConsole, IIS-Metabase, IIS-CommonHttpFeatures, IIS-NetFxExtensibility, IIS-ISAPIExtensions и многие другие. Кроме множества устанавливаемых библиотек и MOF-файлов, также устанавливаются несколько исполняемых файлов: mqsvc.exe, iisreset.exe и mqbkup.exe. Кроме того, в операционной системе регистрируется оснастка **Internet Information Services**.
- **MSMQ Triggers**, название компонента: MSMQ-Triggers — устанавливает триггеры для работы с очередью MSMQ.

Прокси MSMQ DCOM

Данный компонент устанавливает четыре файла с описаниями новых свойств и методов для работы с DCOM: mqoa.tlb, mqoa10.tlb, mqoa20.tlb и mqoa30.tlb.

Среда .NET Environment (подкомпонент компонента Служба активации Windows)

Название компонента: WAS-NetFxEnvironment.

Данный подкомпонент устанавливается вместе с компонентом Process Model, а также некоторыми компонентами службы IIS. Компонент состоит из библиотек `admwprox.dll`, `iisrstap.dll`, `iisRtl.dll`, `wamregps.dll` и файла `iisreset.exe`.

После установки данного компонента будет создана служба Windows Activation Service, которая управляет процессом активации приложений и компонентов операционной системы.

Служба индексирования

Данный компонент состоит из следующих библиотек: `webhits.dll`, `ciadmin.dll`, `ciodm.dll`, `ixsso.dll` и `idq.dll`. В его состав также входят исполняемые файлы `cisvc.exe` и `cidaemon.exe` и оснастка `ciadv.msc`.

Пользователь, знакомый с операционной системой Windows XP, уже, наверное, понял, что представляет собой данный компонент операционной системы. Он содержит службу индексации и оснастку Служба индексирования, которые входили в стандартную поставку операционной системы Windows XP Professional. С их помощью можно было выполнять индексирование содержимого файлов для ускорения поиска по нему. Поиск проводился с помощью оснастки Служба индексирования.

Нельзя однозначно сказать, нужен ли данный компонент операционной системы Windows Vista, ведь она по умолчанию содержит службу, которая выполняет индексирование содержимого файлов для ускоренного поиска. Хотя нужно признать, что работа данной службы очень сильно влияет на производительность операционной системы.

Служба репликации DFS

Данный компонент состоит из исполняемого файла `dfsrmgr.exe`, а также библиотек `dfsres.dll` и `dfsperf.dll`.

Эти файлы необходимы для работы службы Репликация DFS (соответственно первый файл является службой, второй содержит ресурсы службы, а третий описывает счетчики производительности службы), предназначенной для равноправной репликации измененных файлов среди всех компьютеров файловой системы DFS. При этом для повышения производительности работы DFS используется алгоритм сжатия Remote Differential Compression, который позволяет передавать по сети лишь измененные части файлов.

Данная служба расположена в подразделе реестра DFSR, запускается вручную с правами локальной системы и множеством дополнительных привилегий.

Служба Репликация DFS позволяет объединить расположенные на разных компьютерах файлы и папки в одно общее пространство (файловую систему).

Служба установщика ActiveX

По умолчанию установка ActiveX-объектов разрешается только администраторам компьютера, однако в некоторых случаях может понадобиться определить набор

ActiveX-объектов, которые будет разрешено устанавливать не только администраторам, но и обычным пользователям. Сделать это можно для ActiveX-объектов, устанавливаемых через Интернет.

Одной из возможностей ActiveX-объектов является их автоматическая загрузка и установка с помощью тега с атрибутом CODEBASE, если необходимый для корректного отображения страницы ActiveX-объект отсутствует у пользователя. В этом случае отсутствующий ActiveX-объект будет автоматически загружен и перед пользователем отобразится панель информации с предложением установить его. По умолчанию установка ActiveX-объектов разрешена только администраторам, однако вы можете определить сайты, ActiveX-объекты от которых смогут устанавливаться и обычные пользователи.

Сделать это позволяет стандартный компонент операционной системы Windows Vista — **Служба установщика ActiveX**. Данный компонент по умолчанию не устанавливается, поэтому вам придется самостоятельно установить этот компонент с помощью окна **Компоненты Windows** (файл `optionalfeatures.exe`).

ПРИМЕЧАНИЕ

Данный компонент также можно установить с помощью команды `pkgmgr.exe /iu:AxInstallService`.

После установки компонента **Служба установщика ActiveX** будет создана новая служба **Установщик ActiveX (AxInstSV)**, подразделом которой является `AxInstSV`.

Групповая политика для управления службой установщика ActiveX. После установки компонента **Служба установщика ActiveX** также будет создана групповая политика **Веб-узлы, разрешенные для установки элементов управления ActiveX**, расположенная в разделе **Конфигурация компьютера** ▶ **Административные шаблоны** ▶ **Компоненты Windows** ▶ **Служба установки ActiveX**. Именно с помощью данной групповой политики определяются сайты Интернета, установка ActiveX-объектов из которых будет разрешена не только администраторам, но и обычным пользователям.

Чтобы указать новый сайт, необходимо ввести название этого сайта (в поле **Введите имя добавляемого элемента**), а также определить действия для установки ActiveX-объектов с этого сайта (в поле **Введите значение добавляемого элемента**).

Для определения действия для установки ActiveX-объектов с сайта необходимо указать четыре числа (через запятую), каждое из которых определяет поведение при загрузке ActiveX-объектов.

Первые два числа определяют действие при установке подписанного элемента управления ActiveX. Они могут принимать следующие значения:

- 0 — запретить установку элемента управления с данного сайта;
- 1 — отобразить запрос на установку элемента управления с данного сайта;
- 2 — автоматически устанавливать элементы управления с данного сайта.

Третье число определяет действие при установке неподписанного элемента управления ActiveX. Оно может принимать следующие значения:

- 0 — запретить установку элемента управления с данного сайта;
- 1 — отобразить запрос на установку элемента управления с данного сайта.

Последнее же число определяет действие при установке на основе содержимого сертификата элемента управления ActiveX (по умолчанию равно 0). Оно

представляет собой битовую маску, в которой могут быть установлены следующие биты:

- 0x00001000 — не выполнять проверку сертификатов на неверные канонические имена (CN);
- 0x00000100 — не выполнять проверку сертификатов на неизвестные центры сертификации;
- 0x00002000 — не выполнять проверку сертификатов на неверную дату сертификата;
- 0x00000200 — не выполнять проверку сертификатов на факт неправильного использования.

Событие 4097. Еще одной интересной особенностью службы Служба установщика ActiveX является то, что после ее установки в журнале событий приложений можно будет найти событие с кодом 4097 и источником AxInstallService.

Данное событие возникает в том случае, если для отображения какой-либо страницы требовался ActiveX-объект, установка которого доступна только от имени администратора (то есть возможность установки от имени пользователя с указанного сайта назначена не была).

Это значит, что с помощью события 4097 можно определить сайты, для работы которых необходима установка ActiveX-объектов, после чего легко назначить возможность пользовательской установки ActiveX-объектов с этих сайтов (если, конечно, эта возможность вам необходима).

Службы для NFS

Данный компонент позволяет вам создать однородную среду из компьютеров, на которых установлены операционные системы семейства Windows или UNIX. После установки данного компонента пользователи этих операционных систем смогут обращаться к общим папкам компьютеров с любой операционной системой с помощью протокола NFS.

Компонент Службы для NFS содержит несколько дочерних компонентов, предназначенных для работы со службой NFS (Network File System). Описание данных компонентов приведено далее.

Администрирование

Название компонента: NFS-Administration.

Данный компонент устанавливает файлы для работы со службой NFS. К этим файлам в первую очередь относятся консоль `nfsmgmt.msc` (Services for Network File System) и исполняемые файлы `nfsadmin.exe`, `rpcinfo.exe` и `showmount.exe`. Кроме того, устанавливаются следующие библиотеки: `nfsclicgrps.dll`, `nfscliclocks.dll`, `nfsccommgmt.dll`, `nfsrc.dll`.

Оснастка Services for Network File System. Консоль `nfsmgmt.msc` предоставляет интерфейс для управления работой таких служб, как Клиент и сервер для сетей NFS.

С помощью данной консоли вы не сможете примонтировать (или размонтировать) папки к файловой системе NFS — это выполняется с помощью программ, входящих в компонент Клиент для NFS. Однако с ее помощью вы сможете настроить работу служб NFS.

Для этого достаточно запустить данную оснастку, выбрать нужную службу из списка и в ее контекстном меню выбрать команду **Свойства**. Например, с помощью окна **Свойства** службы **Клиент для NFS** можно настроить следующие параметры работы файловой системы NFS (все эти настройки изменяют параметры REG_DWORD-типа ветви реестра HKLM\SOFTWARE\Microsoft\Client for NFS\CurrentVersion\Default).

- Используемые для соединения протоколы — можно использовать либо протоколы TCP и UDP, либо один из них. Эта настройка изменяет значение параметра `Protocols`.
- Используемый по умолчанию тип монтирования папки к файловой системе NFS — эта настройка изменяет значение параметра `MountType`.
- Интервал поиска общих папок файловой системы NFS — эта настройка изменяет значение параметра `Timeout`.
- Используемые по умолчанию разрешения доступа к папкам файловой системы NFS.

Благодаря данной оснастке можно определить имя домена Active Directory, а также имя сервера User Name Mapping. Для этого достаточно воспользоваться командой **Свойства** контекстного меню корневого раздела оснастки.

Исполняемые файлы. Кроме оснастки **Services for Network File System**, данный компонент добавляет набор программ командной строки, которые позволяют управлять работой файловой системы NFS.

Программа `nfsadmin.exe` позволяет управлять клиентом, сервером или картой файловой системы NFS. Для этого применяются специальные разновидности данной программы (также с каждой командой можно указывать параметр `\\<IP-адрес>`, чтобы удаленно управлять компонентами файловой системы NFS).

- `server <параметры>` — позволяет управлять службой `NfsSvc`. Параметры данной команды позволяют изменить те же настройки, что и окно **Свойства** раздела **Server for NFS** оснастки **Services for Network File System**.
- `client <параметры>` — позволяет управлять службой `NfsClnt`. Параметры данной команды позволяют изменить те же настройки, что и окно **Свойства** раздела **Client for NFS** оснастки **Services for Network File System**.
- `mapping <параметры>` — параметры данной команды позволяют изменить те же настройки, что и окно **Свойства** корневого раздела оснастки **Services for Network File System**.

Программа `showmount.exe` позволяет просматривать общие папки, примонтированные к файловой системе NFS. Список параметров данной программы можно посмотреть с помощью команды `showmount.exe /?`.

Программа `rpcinfo.exe` позволяет просматривать информацию о протоколе RPC и использующих его программах. Список параметров данной программы можно посмотреть с помощью команды `rpcinfo.exe /?`.

Клиент для NFS

Название компонента: `ClientForNFS-Infrastructure`.

Добавляемые службы. Данный компонент устанавливает службу **Client for NFS** (файл `nfsclnt.exe`), позволяющую данному компьютеру получать доступ к общим папкам файловой системы NFS. Служба запускается автоматически с правами сетевой

службы и дополнительными привилегиями `SeAuditPrivilege`, `SeChangeNotifyPrivilege` и `SeImpersonatePrivilege`.

Настройки службы `Client for NFS` хранятся в параметрах ветви реестра `HKLM\SYSTEM\CurrentControlSet\Services\NfsClnt\NFS LANs\Default LAN`.

Компонент также устанавливает драйвер `Server for NFS Open RPC (ONCRPC)`, который представлен файлом `rpcxdr.sys` и запускается вручную. Именно он и является клиентом службы `NFS` и выполняет взаимодействие с драйвером `nfsrdr.sys`, который является сервером службы `NFS`. Драйвер `nfsrdr.sys` получает запросы от драйверов `rpcxdr.sys` и передает их драйверу локальной файловой системы `ntfs.sys` на обработку.

Поведение данного драйвера также можно настроить. Для этого применяются параметры `REG_DWORD`-типа ветви реестра `HKLM\SYSTEM\CurrentControlSet\Services\RpcXdr\Parameters`.

- `DefaultNumberOfWorkerThreads` — количество рабочих потоков, используемых по умолчанию входящими запросами `NFS`. Значение данного параметра не может превышать 64 потока.
- `DefaultWorkerThreadPriority` — используемый по умолчанию приоритет рабочих потоков.
- `MaxWorkItems` — максимальное количество рабочих элементов, которые могут использоваться драйвером для обработки запросов.

Добавляемые файлы. Данный компонент устанавливает следующие исполняемые файлы: `mount.exe`, `nfsadmin.exe`, `nfsclnt.exe`, `rpcinfo.exe`, `showmount.exe` и `umount.exe`. Большинство из них устанавливаются также и компонентом `Administrative Tools`. Новыми же являются только два файла: `mount.exe` и `umount.exe`. С их помощью можно добавить (удалить) общую папку к файловой системе `NFS`.

Устанавливаются также следующие библиотеки: `nfscligrps.dll`, `nfsclilocks.dll`, `nfsccprop.dll`, `nfsnp.dll` и `nfsrsrc.dll`.

Службы печати

Содержит вложенные компоненты, позволяющие выполнять печать по сети или на компьютерах с операционными системами `UNIX`. Среди содержимого данного раздела можно встретить следующие компоненты.

Клиент интернет-печати

Название компонента: `Printing-Foundation-InternetPrinting-Client`.

По умолчанию данный компонент установлен. Он позволяет использовать принтер, подключаясь к нему по протоколу `TCP/IP`. Данный компонент состоит из исполняемого файла `wppinst.exe` и библиотек `inetppui.dll` и `inetpp.dll`.

Монитор LPR-портов

Название компонента: `Printing-Foundation-LPRPortMonitor`.

По умолчанию не установлен. Данный компонент позволяет работать с серверами печати `UNIX` (или `VAX`), для чего устанавливаются библиотеки `lprhelp.dll`, `lprmon.dll`, `lprmonui.dll` и `SetupLpr.dll`.

Библиотека `lprmon.dll` представляет собой порт `LPR`, который добавляется к списку портов ветви системного реестра `HKLM\SYSTEM\CurrentControlSet\Control\`

Print\Monitors. Иначе говоря, после установки данного компонента в списке **Порты принтера** (отображается после нажатия кнопки **Добавить порт**, расположенной на вкладке **Порты** окна **Свойства: Сервер печати**) появляется новый элемент **LPR Port**.

Этот компонент также устанавливает два исполняемых файла командной строки: `lprq.exe` и `lpr.exe`.

С помощью программы `lprq.exe` можно просмотреть очередь удаленного принтера, установленного на компьютере с операционной системой семейства UNIX. Для этого достаточно воспользоваться данной программой, применив следующий синтаксис: `lprq.exe -s<имя сервера> -r<имя принтера>`.

Если же вам нужно распечатать файл на удаленном принтере, то достаточно воспользоваться программой `lpr.exe`. Она имеет следующий основной синтаксис: `lpr.exe -s<имя сервера> -r<имя принтера> <путь к файлу, который нужно распечатать, и его имя>`. Вы также можете использовать дополнительные параметры:

- `-c <класс>` — класс нового задания;
- `-j <имя>` — название задания;
- `-o <тип>` — определяет тип распечатываемого файла (по умолчанию считается, что распечатываемый файл текстовый);
- `-d` — говорит о том, что перед печатью распечатываемый файл нужно скопировать на сервер печати.

Служба печати LPD

Название компонента: `Printing-Foundation-LPDPrintService`.

По умолчанию компонент не установлен. Он устанавливает библиотеку `lpdsvc.dll`, необходимую для работы с LPD (Line Printer Daemon) и Remote Line Printer в качестве клиента.

После установки данного компонента в операционной системе будет зарегистрирована новая служба **TCP/IP Print Server**, подразделом которой (в реестре) является подраздел `LPD SVC`. Служба запускается автоматически, работает от имени операционной системы, но никаких дополнительных системных привилегий не использует.

Настройки данной службы хранятся в параметрах ветви реестра `HKLM\SYSTEM\CurrentControlSet\Services\LPD SVC\Parameters`, имеющих тип `REG_DWORD`.

- `AllowJobRemoval` — определяет, разрешено ли принтеру выполнять удаленные задания. По умолчанию разрешено.
- `AllowPrinterResume` — определяет, разрешено ли пользователям возобновлять работу принтера. По умолчанию разрешено.
- `MaxConcurrentUsers` — максимальное количество пользователей, которые могут пользоваться данным принтером одновременно. По умолчанию значение параметра равно 100.

Удаленное разностное сжатие

Данный компонент представлен библиотекой `msrdc.dll`. Именно этот способ сжатия используется некоторыми службами операционной системы Windows Vista для повышения скорости работы в сети.

Управление съемными носителями

Данный компонент устанавливает следующие библиотеки: `mll_hp.dll`, `mll_mtf.dll`, `mll_qic.dll`, `ntmsapi.dll`, `ntmsdba.dll`, `ntmsev.dll`, `ntmsmgr.dll`, `ntmssvc.dll` и `rsmps.dll`. Кроме того, устанавливаются следующие исполняемые файлы: `rsm.exe`, `rsmmllsv.exe`, `rsmsink.exe` и `rsmui.exe`.

Файл `rsm.exe` представляет собой программу командной строки, позволяющую управлять съемными устройствами. Чтобы просмотреть возможности данной программы, достаточно воспользоваться командой `rsm.exe /?`.

В процессе установки данного компонента также устанавливаются консоли `ntmsmgr.msc` (**Съемные ЗУ**) и `ntmsoprq.msc` (**Запросы операторов съемных ЗУ**).

Библиотеки и исполняемые файлы, которые устанавливаются данным компонентом, необходимы не только для работы устанавливаемых консолей, но и для работы новой службы **Съемные ЗУ** (для своей работы она использует библиотеку `ntmssvc.dll`). Данная служба запускается вручную, работает от имени локальной системы, а также использует множество дополнительных привилегий.

Фактически данный компонент не является нововведением операционной системы Windows Vista, так как он присутствовал и в операционной системе Windows XP. Он позволяет управлять и каталогизировать подключенные к компьютеру съемные устройства, что может повысить скорость доступа к ним.

Факсы и сканирование Windows

Данный компонент состоит из исполняемых файлов `wfs.exe`, `fxsunatd.exe`, `fxssvc.exe` и `fxscover.exe`, а также библиотек `wifax.dll`, `wfsr.dll`, `fxsxp32.dll`, `fxsutility.dll`, `fxstiff.dll`, `fxst30.dll`, `fxsst.dll`, `fxsroute.dll`, `fxsresm.dll`, `fxsmon.dll`, `fxsxt32.dll`, `fxsevent.dll`, `fxscomposer.es.dll`, `fxscompose.dll`, `fxscomex.dll`, `fxscom.dll` и `fxsapi.dll`.

Все эти файлы необходимы не только для работы программы **Факсы и сканирование Windows**, но и для работы службы **Факс**. Данная служба запускается вручную с правами сетевой службы и множеством дополнительных привилегий.

Работа с компонентами

После того как вы выберете устанавливаемый или удаляемый компонент операционной системы и нажмете кнопку **ОК**, начнется процесс установки или удаления выбранного компонента соответственно. Этот процесс выполняется автоматически и не требует от вас никаких действий. Если в процессе установки или удаления компонента программа установки обнаружит, что не может выполнить какое-либо действие, то будет выведено окно о необходимости перезагрузки компьютера и данное действие будет выполнено при следующем запуске. При этом сведения об отложенных действиях запишутся в файл `pending.xml`, расположенный в папке `%systemroot%\WinSxS`, а в параметр `REG_MULTI_SZ`-типа `SetupExecute`, расположенный в ветви реестра `HKLM\SYSTEM\CurrentControlSet\Control\SetControl\Set001\Control\Session Manager`, занесется значение `poqexec.exe \SystemRoot\WinSxS\pending.xml`.

Сведения о том, как прошел процесс удаления или установки компонентов, можно просмотреть в файле `CBS.log`, расположенном в папке `%systemroot%\logs`.

СОВЕТ

Перед установкой или удалением компонента операционной системы всегда выполняется создание точки восстановления, поэтому, если при выполнении операции произошел сбой, просто вернитесь к созданной точке восстановления с помощью Восстановления системы.

Этот совет вам может пригодиться, так как иногда процесс установки или удаления компонента может зависнуть (особенно если в это время работает множество других программ), и если вы перезагрузите компьютер, не дождавшись завершения установки или удаления компонента, то, скорее всего, при следующем запуске операционной системы окно Компоненты Windows окажется пустым. Это говорит о том, что данные, на основе которых строится это окно, повреждены или некорректны. В этом случае достаточно воспользоваться точкой восстановления, чтобы все вернулось на свои места.

Конечно, в окне Компоненты Windows показаны не все компоненты операционной системы Windows Vista, однако теперь отобразить скрытые компоненты не так-то просто. Если раньше сведения о тех компонентах, которые будут отображаться, хранились в файле `sysoc.inf` раздела `%systemroot%\inf`, то теперь данный файл отсутствует.

Сведения обо всех компонентах операционной системы Windows Vista хранятся в папке `%systemroot%\servicing\packages`. Каждый компонент представлен в нем двумя или более файлами. Первый из них имеет расширение `CAT` и содержит информацию о подписи компонента. Остальные же имеют расширение `MUM` и представляют собой текстовые файлы в XML-формате, которые описывают название компонента, а также его поведение в операционной системе или дополнительные языковые настройки.

Сведения о компонентах операционной системы, которые отображаются в окне Компоненты Windows, можно найти и в ветви реестра `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Component Based Servicing`. Эта ветвь включает в себя два вложенных подраздела `PackageDetect` и `Packages`. С помощью подраздела `PackageDetect` можно удалить из окна Компоненты Windows некоторые элементы. Для этого достаточно удалить или присвоить значение 1 параметру, названному в соответствии с названием компонента и расположенному в ветви реестра `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Component Based Servicing\PackageDetect\x86_microsoft-windows-foundation-package_31bf3856ad364e35_0.0.0.0_none_1bcadd73a6fabe3b`.

Сами же программы и компоненты, которые можно установить в операционной системе Windows Vista, хранятся в подразделах папки `%systemroot%\winsxs`. Именно поэтому при установке и удалении компонентов операционной системы больше не нужно использовать загрузочный диск Windows.

Установка необязательного компонента Windows

Расположение: `%systemroot%\system32\ocsetup.exe`.

Окно Компоненты Windows, с помощью которого можно удалить или установить тот или иной компонент операционной системы, уже было рассмотрено, однако это не единственный способ установки и удаления компонентов Windows. Для этого также можно воспользоваться программой командной строки `ocsetup.exe`. Основной синтаксис данной программы следующий: `ocsetup <название компонента> <параметры>`. Список названий компонентов операционной системы Windows Vista был приведен в подразд. «Компоненты операционной системы» этого раздела.

Параметры, которые можно использовать при работе с данной программой, следующие.

- /uninstall — удаляет данный компонент операционной системы. Если не указывать этот параметр, то соответствующий компонент операционной системы будет установлен.
- /passive — скрывает информацию об установке или удалении компонента операционной системы.
- /unattendfile:<путь к файлу> — указывает путь к файлу ответов.
- /quiet — подавляет все запросы к пользователю, которые по умолчанию необходимо выполнить для корректной установки. Данные для корректной установки должны быть взяты из файла ответов.
- /norestart — указывает на необходимость после установки или удаления компонента операционной системы не выполнять перезагрузку компьютера.
- /log:<путь к файлу> — указывает путь к файлу, в который будут заноситься сведения о процессе удаления или установки компонента.
- /x:<дополнительные параметры> — дополнительные параметры работы базового установщика данного компонента.

Диспетчер пакетов Windows

Расположение: %systemroot%\system32\PkgMgr.exe.

Для установки или удаления компонентов операционной системы можно также использовать программу командной строки PkgMgr.exe. Список параметров данной программы можно просмотреть, воспользовавшись командой PkgMgr.exe /?. В остальном же использование данной программы похоже на использование программы ossetup.exe, описанной выше (более того, программа ossetup.exe при своей работе вызывает программу PkgMgr.exe).

В листинге 9.1 представлена команда, с помощью которой выполняется установка всех компонентов, связанных с IIS.

Листинг 9.1. Команда для установки всех компонентов IIS

```
pkgmgr /l:log.etw /iu:IIS-WebServerRole;IIS-WebServer;IIS-CommonHttpFeatures;
IIS-StaticContent;IIS-DefaultDocument;IIS-DirectoryBrowsing;IIS-HttpErrors;
IIS-HttpRedirect;IIS-ApplicationDevelopment;IIS-ASPNET;IIS-NetFxExtensibility;
IIS-ASP;IIS-CGI;IIS-ISAPIExtensions;IIS-ISAPIFilter;IIS-ServerSideIncludes;
IIS-HealthAndDiagnostics;IIS-HttpLogging;IIS-LoggingLibraries;IIS-RequestMonitor;
IIS-HttpTracing;IIS-CustomLogging;IIS-ODBCLogging;IIS-Security;
IIS-BasicAuthentication;IIS-WindowsAuthentication;IIS-DigestAuthentication;
IIS-ClientCertificateMappingAuthentication;IIS-IISCertificateMappingAuthentication;
IIS-URLAuthorization;IIS-RequestFiltering;IIS-IPSecurity;IIS-Performance;
IIS-HttpCompressionStatic;IIS-HttpCompressionDynamic;IIS-WebServerManagementTools;
IIS-ManagementConsole;IIS-ManagementScriptingTools;IIS-ManagementService;
IIS-IIS6ManagementCompatibility;IIS-Metabase;IIS-WMICompatibility;
IIS-LegacyScripts;IIS-LegacySnapIn;IIS-FTPPublishingService;IIS-FTPServer;
IIS-FTPManagement;WAS-WindowsActivationService;WAS-ProcessModel;
WAS-NetFxEnvironment;WAS-ConfigurationAPI
```

Использование репозитория CIM

Работать с компонентами операционной системы можно и с помощью класса `Win32_OsBaseline`, принадлежащего пространству имен `\\root\cimv2`. Данный класс содержит методы для описания компонентов, из которых состоит операционная система. Описание этих методов можно найти среди дополнительного материала к книге.

9.2. Установка и удаление сетевых компонентов

Все стандартные сетевые компоненты, доступные в операционной системе Windows Vista, устанавливаются по умолчанию. Единственным исключением из этого правила является протокол Надежный многоадресный протокол (RMP).

Установить данный протокол либо удалить установленные стандартные сетевые компоненты Windows Vista можно двумя способами: либо с помощью специального окна, либо с помощью программы командной строки `netcfg.exe`.

Окно Свойства сетевого интерфейса

Чтобы установить или удалить сетевой компонент, достаточно воспользоваться окном **Свойства** необходимого вам сетевого интерфейса, а точнее, кнопками **Удалить** или **Установить** этого окна.

После того как вы нажмете на кнопку **Установить**, перед вами отобразится окно **Выбор сетевых компонентов**, в котором нужно выбрать тип устанавливаемого компонента: клиент, служба или протокол. Как было сказано выше, по умолчанию в операционной системе Windows Vista не установлен только протокол RMP.

Работа с программой netcfg.exe

Расположение: `%systemroot%\system32\netcfg.exe`.

В поставку операционной системы Windows Vista входит специальная программа командной строки, позволяющая просмотреть, установить или удалить такие сетевые компоненты операционной системы, как протоколы NetBIOS, TCP/IPv6, IPX и т. д., сетевые службы QoS, Служба доступа к файлам и принтерам сетей Microsoft и др. Чтобы просмотреть описание работы программы `netcfg.exe`, введите команду `netcfg /?`.

Работать с программой `netcfg.exe` довольно просто, однако для этого нужно знать названия сетевых компонентов, которые с ее помощью можно установить или удалить.

Просмотр установленных сетевых компонентов

Как уже было сказано, данная программа позволяет просмотреть список установленных сетевых компонентов. Для этого применяется команда `netcfg -s n`. После

ее выполнения перед вами отобразится список установленных сетевых адаптеров, протоколов, служб и клиентов. Обратите внимание на левый столбец выводимых данных (в правом столбце отображается описание сетевого компонента). В нем содержится сокращенное имя компонента, которое применяется во всех командах программы `netcfg.exe`.

Можно также просмотреть список установленных адаптеров, для чего применяется команда `netcfg -s a`, или список привязок для определенного сетевого компонента, для чего применяется команда `netcfg -b <сокращенное имя сетевого компонента>`, например, `netcfg -b ms_tcpip`.

Кроме того, вы можете просмотреть состояние только определенного сетевого компонента. Для этого служит команда `netcfg -q <сокращенное имя сетевого компонента>`. Например, чтобы просмотреть, установлен ли компонент NetBIOS, нужно воспользоваться командой `netcfg -q ms_netbios`.

Установка сетевых компонентов

Для установки определенного сетевого компонента нужно знать его сокращенное имя или название INF-файла. Немного ниже приведен список всех возможных сокращенных имен.

Чтобы установить сетевой компонент, используя сокращенное имя, нужно воспользоваться командой `netcfg -c <класс компонента> -I <сокращенное имя компонента>`. Здесь класс компонента может принимать следующие значения: `p` для сетевого протокола, `s` для сетевой службы и `c` для сетевого клиента.

Чтобы установить сетевой компонент, используя его INF-файл, нужно воспользоваться командой, которая немного отличается от предыдущей: `netcfg -l <путь к INF-файлу> -c <класс компонента> -I <сокращенное имя компонента, которое будет применяться для его идентификации>`.

Можно также воспользоваться командой `netcfg -winpe`, которая при установке операционной системы используется для установки TCP/IP, NetBIOS и клиента для сетей Microsoft.

Сведения об установке сетевого компонента заносятся в файл журнала `setupapi.app.log`, расположенный в папке `%windows%\inf`.

Удаление сетевых компонентов

Для удаления сетевых компонентов используется команда `netcfg -u <сокращенное имя компонента>`. Например, чтобы удалить клиент для сетей Microsoft, нужно воспользоваться командой `netcfg -u MS_MSCLIENT`.

Стандартные сетевые компоненты

Теперь можно рассмотреть список стандартных сетевых компонентов, которые вы можете добавлять или удалять. Все INF-файлы, описанные ниже, расположены в папке `%systemroot%\INF`.

Для удобства список разбит на сетевые протоколы, службы и клиенты.

Сетевые протоколы

Сначала будут описаны сетевые протоколы (табл. 9.2).

Таблица 9.2. Сетевые протоколы

Название	Добавляемый сетевой протокол
ms_pppoe	Протокол точка-точка по Ethernet, который добавляется путем установки в систему драйвера Remote Access PPPOE. INF-файл netrast.inf, секция Ndi-PppoeProtocol. Протокол позволяет передавать кадры PPP через локальную сеть Ethernet, используя туннели. Как и протокол PPTP, он поддерживает авторизацию, шифрование и сжатие
ms_tcpip6	Протокол Интернета версии 6 (TCP/IPv6). INF-файл netip6.inf, секция MS_TCPIP6.Install. Протокол поддерживает расширенное до 128 бит адресное пространство, обеспечивает упрощенный механизм автоконфигурирования адресов, а также масштабируемость групповых адресов
ms_tcpip6_tunnel	Microsoft TCP/IP версии 6-туннели. INF-файл netip6.inf, секция MS_TCPIP6.Tunnel.Install
ms_ndiswan	NDIS-драйвер WAN удаленного доступа. INF-файл netrast.inf, секция Ndi-NdisWan
MS_wanarp	Драйвер удаленного доступа IP ARP. INF-файл netrast.inf, секция Ndi-Wanarp
ms_netbt_smb	Протокол сообщений TCP/IP (сеанс SMB). INF-файл nettcpip.inf, секция MS_NETBT_SMB.PrimaryInstall. Добавляемый протокол является протоколом прикладного уровня, предназначенным для совместного использования файлов. Работает поверх протоколов NBT или NetBEUI, но также может работать на основе IPX/SPX
ms_netbt	Протокол клиента WINS (TCP/IP). INF-файл nettcpip.inf, секция MS_WINS.PrimaryInstall. Основным назначением службы WINS является организация процесса разрешения имен NetBIOS в соответствующие IP-адреса. Для этого и используется протокол, определенный данным сетевым компонентом
MS_RSPNDR	Ответчик обнаружения топологии канального уровня. INF-файл rspndr.inf, секция Install. Данный протокол используется сетевым модулем системы для отображения графического представления сети (можно увидеть в мастере Центр управления сетями и общим доступом). Если система не будет поддерживать данный протокол, то соответствующий компьютер не будет отображаться в графическом представлении сети
MS_SMB	Microsoft NetbiosSmb. INF-файл nettcpip.inf, секция MS_SMB.Install. На его основе работают экземпляры NetBT_Tcpip
MS_TCPIP	Протокол Интернета версии 4 (TCP/IPv4). INF-файл nettcpip.inf, секция MS_TCPIP.PrimaryInstall. Данный протокол является стандартным протоколом, используемым для сетевого взаимодействия
ms_pptp	Туннельный протокол точка-точка. INF-файл netrast.inf, секция Ndi-PptpProtocol. Протокол позволяет создавать туннели, которые будут передавать кадры PPP через сети на базе протокола TCP/IP. При этом протокол поддерживает шифрование кадров с помощью механизма MPPE (Microsoft Point-to-Point Encryption), основанного на алгоритме RSA
ms_l2tp	Туннельный протокол уровня 2. INF-файл netrast.inf, секция Ndi-L2tpProtocol. Протокол туннелирования второго уровня, функционирующий на канальном уровне, представляет собой промышленный стандарт, впервые реализованный в операционной системе Windows 2000 и используемый как передающая среда в VPN (виртуальная частная сеть). В отличие от протокола PPTP (протокол L2TP является дальнейшим развитием PPTP), он использует для шифрования протокол IPSec. Еще одним отличием протокола L2TP является то, что для передачи управляющих сообщений он использует UDP (PPTP в управляющем канале использует протокол TCP)

Название	Добавляемый сетевой протокол
MS_LLDIO	Драйвер в/в тополога канального уровня. INF-файл lldio.inf, секция Install. Данный драйвер используется для создания графического представления сети (карты сети)
MS_NDISUIO	NDIS Usermode I/O Protocol. INF-файл ndisuio.inf, секция Install
MS_wanarpv6	Драйвер удаленного доступа IPv6 ARP. INF-файл netrast.inf, секция Ndi-Wanarpv6
ms_tcpip_tunnel	Протокол Интернета (TCP/IP)-туннели. INF-файл nettcpip.inf, секция MS_TCPIP.Tunnel.PrimaryInstall
ms_Bridge	MAC Bridge. INF-файл netbrdgs.inf, секция Bridge.ndi. Протокол используется во время маршрутизации с помощью стандартных средств операционной системы
MS_IrDA	IrDA Protocol. INF-файл netirda.inf, секция IrDA.Install. Данный протокол предназначен для выполнения подключения между двумя компьютерами с помощью инфракрасной связи. Нужно признать, что на сегодняшний день инфракрасная связь практически не используется, вытесненная беспроводной связью и Bluetooth
*MS_RMCAST	Reliable Multicast Protocol. INF-файл netpgm.inf, секция MS_RMCAST.PrimaryInstall. Протокол представляет собой протокол транспортного уровня, поддерживающий ширококвещательную рассылку

* По умолчанию не установлен.

Сетевые службы

Далее рассмотрены сетевые службы (табл. 9.3).

Таблица 9.3. Сетевые службы

Название	Добавляемая сетевая служба
MS_RasSrv	Сервер удаленного доступа. INF-файл netrass.inf, секция Ndi-RasSrv. Служба обрабатывает запросы от удаленных клиентов, подключаемых к серверу с помощью модема
MS_SERVER	Служба доступа к файлам и принтерам сетей Microsoft. INF-файл netserv.inf, секция Install.ndi. Данная служба является основной службой системы, используемой при доступе к общим папкам и принтерам
ms_steelhead	Сервер маршрутизации и удаленного доступа. INF-файл netrass.inf, секция Ndi-Steelhead. Эта служба представляет собой надстройку над локальной сетью, предназначенную для маршрутизации глобальных сетей (для этого используется Windows Sockets). Она не может заменить собой аппаратные маршрутизаторы, например маршрутизаторы Cisco, однако может превратить сервер в маршрутизатор низкого уровня. Данная служба поддерживает протоколы IP, RIP, OSPF и т. д.
MS_PACER	Планировщик пакетов QoS. INF-файл netpacer.inf, секция Install. Эта служба, основанная на протоколах прикладного уровня, позволяет выделить определенную часть пропускной способности сети для передаваемых данных, которые чувствительны к задержкам
ms_rasman	Диспетчер подключений удаленного доступа. INF-файл netrass.inf, секция Ndi-RasMan. Служба позволяет удаленно подключиться к компьютеру (как правило, с помощью модема)

продолжение ➔

Таблица 9.3 (продолжение)

Название	Добавляемая сетевая служба
MS_NETBIOS	NetBIOS Interface. INF-файл netnb.inf, секция NetBIOS.ndi
MS_NativeWiFiP	Фильтр NativeWiFi. INF-файл: netnwifi.inf, секция MS_NWIFI.Install
*MS_RasCli	Dial-Up Client. INF-файл netrass.inf, секция Ndi-RasCli. Служба выполняет подключение к серверу удаленного доступа с помощью модемного соединения
*MS_Alg	Application Layer Gateway. INF-файл: netrass.inf, секция Ndi-ALG. Эта служба является основой стандартного брандмауэра операционной системы и компонента ICS

* По умолчанию не установлена.

Сетевые клиенты

Последними рассмотрим сетевые клиенты, вернее, один клиент.

MS_MSCLIENT — по умолчанию данный клиент установлен. Определяет клиент для сетей Microsoft. INF-файл netmscli.inf, секция MSClient.ndi.

Принцип работы программы netcfg.exe

Напоследок вкратце рассмотрим принцип работы программы netcfg.exe. При удалении сетевых компонентов данная программа не удаляет никаких файлов или ветвей реестра, принадлежащих сетевым компонентам. Единственной ветвью реестра, которую она использует, является HKLM\System\CurrentControlSet\Control\Network. Эта ветвь включает в себя следующие подразделы:

- {4d36e972-e325-11ce-bfc1-08002be10318} — список сетевых адаптеров, которые установлены в системе;
- {4d36e973-e325-11ce-bfc1-08002be10318} — список сетевых клиентов, установленных в операционной системе;
- {4d36e974-e325-11ce-bfc1-08002be10318} — список сетевых служб, установленных в операционной системе;
- {4d36e975-e325-11ce-bfc1-08002be10318} — список сетевых протоколов, установленных в операционной системе.

Каждый из них содержит вложенные подразделы, которые описывают один установленный сетевой компонент. При удалении сетевого компонента с помощью программы netcfg.exe она удаляет соответствующий подраздел данной ветви реестра.

Фактически действие программы командной строки netcfg.exe аналогично удалению или установке сетевых компонентов с помощью кнопок **Установить** и **Удалить** окна свойств определенного сетевого подключения. Это окно можно вызвать с помощью ссылки **Просмотр состояния** напротив нужного сетевого подключения в папке **Центр управления сетями и общим доступом**. После выбора данной ссылки перед вами отобразится окно состояния данного сетевого подключения, в котором нужно нажать кнопку **Свойства**.

Установка и удаление других сетевых компонентов

Помимо установки стандартных сетевых компонентов, вы можете установить либо удалить другие компоненты сети Windows Vista. Для этого применяются различные варианты синтаксиса программы netsh.exe:

- netsh bridge install|uninstall — установить или удалить программный маршрутизатор;
- netsh interface ipv4 install|uninstall — установить или удалить протокол IPv4;
- netsh interface ipv4 reset — восстановить стандартные настройки сетевого протокола IPv4;
- netsh interface portproxy reset — восстановить стандартные настройки порта прокси-сервера;
- netsh interface tcp reset — восстановить стандартные настройки сетевого протокола TCP;
- netsh winsock reset — восстановить содержимое каталога Winsock;
- netsh winhttp reset proxy — восстановить стандартные параметры WinHTTP.

9.3. Мастер Установка или удаление языков отображения

Расположение: %systemroot%\system32\lpksetup.exe.

Нововведением операционной системы Windows Vista является новый способ изменения языка операционной системы. Если у вас имеется англоязычная версия операционной системы, то для ее, например, русифицирования вам не нужно устанавливать новую операционную систему — достаточно установить новый языковой пакет. При этом предыдущий пакет по умолчанию также останется в системе, поэтому вы сможете оперативно переходить от одного языкового пакета к другому.

Для изменения языкового пакета достаточно воспользоваться программой lpksetup.exe. После ее вызова отобразится мастер, позволяющий либо установить новый языковой пакет (кнопка **Установить языки**), либо удалить один из уже установленных (кнопка **Удалить языки**). Данный мастер также можно вызвать с помощью кнопки **Установить или удалить язык** вкладки **Языки и клавиатуры** окна **Язык и региональные стандарты**.

При установке пакета операционная система попросит указать путь к нему. Если же вы хотите удалить языковой пакет, то мастер выведет список установленных пакетов.

Выбор используемого по умолчанию языкового пакета

После того как вы установите новые языковые пакеты, переключаться между уже установленными языковыми пакетами можно будет с помощью раскрывающегося

списка Выберите язык интерфейса, расположенного на вкладке Языки и клавиатуры окна Язык и региональные стандарты.

Изменить используемый по умолчанию языковой пакет также можно с помощью групповых политик. Для этого достаточно воспользоваться политикой Ограничить язык интерфейса Windows для указанного пользователя раздела Конфигурация пользователя ▶ Административные шаблоны ▶ Панель управления ▶ Язык и региональные стандарты либо политикой Ограничить язык интерфейса Windows для всех зарегистрированных пользователей раздела Конфигурация компьютера ▶ Административные шаблоны ▶ Панель управления ▶ Язык и региональные стандарты.

Удаление языкового пакета

Удалить языковой пакет можно точно так же, как и установить его, — с помощью мастера Установка или удаление языков отображения. Однако удалить текущий языковой пакет можно также с помощью программы командной строки `lpremove.exe`.

При запуске программы `lpremove.exe` без параметров выполняется создание назначенного задания, удаляющего языковой пакет (задание `LPRemove`, расположенное в разделе Библиотека планировщика заданий ▶ Microsoft ▶ Windows ▶ MUI оснастки Планировщик заданий). Если же запустить программу с параметром `/c`, то задание будет отменено.

Если установка языкового пакета завершилась неудачно

Не следует отчаиваться, если установка языкового пакета с помощью мастера Установка или удаление языков отображения привела к ошибке. Возможно, у вас получится установить языковой пакет самостоятельно.

Языковой пакет представляет собой архив, содержащий в себе безумно большое количество (более 6000) файлов с расширением MUI. Именно эти файлы и используются для локализации операционной системы. Названия файлов с расширением MUI соответствуют названию файла, для локализации которого они используются. Например, для локализации файла `regedit.exe` используется файл `regedit.exe.mui`.

Находиться же файлы с расширением MUI должны в папке, названной в соответствии с языком локализации. Например, в папке `ru-RU` для русского языкового пакета или `en-US` — для английского.

Сама папка локализации должна быть внутри папки, где находится файл, который она локализует. То есть, чтобы локализовать файл `regedit.exe`, который содержится в папке `%systemroot%`, необходимо перенести файл `regedit.exe.mui` в папку `%systemroot%\ru-RU` (для русского языкового пакета).

После того как перенос файлов локализации будет выполнен (не имеет значения, перенесете вы все файлы локализации или нет — если операционная система не найдет нужный файл локализации для какого-либо файла, она воспользуется другим файлом локализации, установленным ранее), необходимо будет подключить новый установленный вами языковой пакет. Для этого следует воспользоваться реестром.

Например, для русского языкового пакета нужно присвоить параметру `REG_DWORD`-типа `LCID` значение `0x00000419` (то есть кодировку языкового пакета), а параметру `REG_DWORD`-типа `Type` — значение `0x00000092`. Оба этих параметра находятся в ветви реестра `HKLM\SYSTEM\CurrentControlSet\Control\MUI\UILanguages\ru-RU`. После этого параметру строкового типа `Default` ветви реестра `HKLM\SYSTEM\CurrentControlSet\Control\Nls\Language` нужно присвоить значение `0419` (то есть указать, что по умолчанию должен использоваться русский языковой пакет).

9.4. Установка пакетов обновлений операционной системы

Расположение: `%systemroot%\system32\wusa.exe`.

Пакеты обновлений операционной системы Windows Vista имеют расширение `MSU` и устанавливаются с помощью программы `wusa.exe`. Вряд ли вам когда-нибудь придется самостоятельно пользоваться этой программой, так как такие же пакеты обновлений можно установить и с помощью двойного щелчка кнопкой мыши на них. В данном случае команда установки записана в параметре (По умолчанию) ветви реестра `HKCR\Microsoft.System.Update\shell\open\command`.

Однако программа `wusa.exe` имеет несколько параметров, которые по умолчанию не используются при установке с помощью оболочки. К ним относятся следующие:

- `/quiet` — выполняется автоматическая установка без задания вопросов установки пользователю. Если необходимо, то после установки будет выполнена автоматическая перезагрузка компьютера;
- `/norestart` — используется вместе с предыдущим параметром и говорит о том, что после автоматической установки не будет выполнена перезагрузка компьютера, даже если это нужно для обновленных компонентов операционной системы.

На этом можно закончить рассмотрение параметров работы с компонентами операционной системы.

Приложение 1

Команды rundll32

Программа `rundll32.exe` входит в стандартную поставку всех операционных систем семейства Windows и позволяет выполнять различные функции, описанные в библиотеках, драйверах или ActiveX-объектах так, будто они выполняются из программы. Она также включена в состав Windows Vista, поэтому обзор новой операционной системы, предназначенный для опытных пользователей, никак нельзя назвать законченным, если он не будет содержать описание команд `rundll32.exe`, доступных в Windows Vista.

Многие команды `rundll32.exe` уже были рассмотрены на страницах этой книги. Другие же команды, которые не подошли ни к одной главе, будут описаны в этом приложении.

Синтаксис использования программы `rundll32.exe` следующий: `rundll32.exe <библиотека>. <функция> <параметры>`. При этом очень важно понимать, что название функции, хранящейся в библиотеке, зависит от регистра, которым вы ее написали. Иначе говоря, например, функции `MessageBox` и `messageBox`, в понимании программы `rundll32.exe`, являются различными функциями, поэтому, если в названии функции вы напишете вместо прописной буквы строчную, программа `rundll32.exe` выдаст ошибку. Название же самой программы `rundll32.exe` и название библиотеки можно указывать как строчными, так и прописными символами.

По сравнению с операционной системой Windows XP, операционная система Windows Vista поддерживает меньше команд `rundll32.exe`. Это связано с тем, что Windows Vista больше не использует многие библиотеки, доступные в Windows XP. Ниже будут рассмотрены те команды `rundll32.exe`, которые не упоминались ранее на страницах книги (для экономии места в командах опущено название программы `rundll32.exe`).

- `shell32.dll, SHHelpShortcuts_RunDLL Connect` — вызвать мастер подключения сетевых дисков.
- `shell32.dll, ShellAboutA` — вызвать окно, отображающее версию операционной системы Windows. Его же можно открыть, выполнив команду `winver.exe`.
- `IEAKENG.dll, BrowseForFileA` — отобразить дерево файловой системы жесткого диска компьютера.
- `ndfapi.dll, NdfRunDllHelpTopic <html-страница>` — вызвать окно **Справка и поддержка** на указанной странице. Например, в качестве HTML-страницы можно использовать вызов `mshelp://Windows/?id=33307acf-0698-41ba-b014-ea0a2eb8d0a8`.

- shell32.dll, OpenAs_RunDLL "путь и имя файла" — вызывает окно **Выбор программы** для открытия данного файла.
- shell32.dll, Control_RunDLL Cliconfg.dll — открывает окно **Программа сетевого клиента SQL Server**.
- shell32.dll, Control_RunDLL Hotplug.dll — отображает окно удаления съемного устройства.
- admparse.dll, CheckDuplicateKeysA "путь и имя файла" — переписывает содержимое файла.
- IEAKENG.dll, BToolBar_SaveA "путь к папке" — создает папку. Если данная папка уже существует, то все ее содержимое будет удалено.
- appwiz.cpl, GetProgramsOnline — запускает браузер Internet Explorer и загружает страницу <http://g.msn.com/WMHFUSEN/101724>.
- shell32.dll, ShellExec_RunDLL "команда" — выполняет указанную команду.
- url.dll, FileProtocolHandler "папка" — открывает папку. Если вы не укажете папку, то будет открыта папка вашего профиля.
- WININET.dll, RunOnceUrlCache "путь к папке" — удаляет все файлы указанной папки, имеющие атрибут FILEATTRIBUTETAGINFORMATION.
- hwizards.dll, ProcessXMLFile "путь и имя XML-файла" — выполняет XML-файл.
- user32.dll, mouse_event — выполняет команду нажатия правой кнопки мыши.
- user32.dll, SetCursorPos — устанавливает указатель мыши в правый нижний угол экрана.
- dfshim.dll, KillService — останавливает работу службы фонового обновления dfsvc.exe
- IERNONCE.dll, RunOnceExProcess — выполняет команды, описанные в ветви реестра HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx (или в ветви корневого раздела HKCU).

Приложение 2

Работа с WMI

Если вы не боитесь таких слов, как «сервер сценариев Windows (WSH)» или «инструментарий управления Windows (WMI)», то вам, возможно, будет интересно узнать о новых классах, свойствах и методах, которые появились в репозитории CIM операционной системы Windows Vista. Именно поэтому новые классы (а также некоторые из старых) были описаны на протяжении всей этой книги.

Настройка WMI

В этом примечании будут описаны способы работы с репозитием CIM. Однако перед тем, как начать описание работы с репозитием, рассмотрим настройки WMI и способы их просмотра.

Использование репозитория CIM

Одним из способов просмотра сведений о настройках WMI является использование класса `Win32_WMISetting`, принадлежащего пространству имен `\\root\cimv2`. Он поддерживает следующие свойства, большинство из которых доступно как для чтения, так и для записи.

- `ASPScriptDefaultNamespace` — пространство имен, которое подразумевается по умолчанию при получении доступа к репозитарию CIM из сценариев, если пространство имен в сценарии не указано. По умолчанию данное свойство содержит значение `root\cimv2`. Данному свойству соответствует параметр реестра строкового типа `Default Namespace`.
- `ASPScriptEnabled` — определяет, разрешено ли использование возможностей WMI в сценариях на страницах ASP. Данному свойству соответствует параметр реестра `REG_DWORD`-типа `Enable for ASP`.
- `AutorecoverMofs` — список MOF-файлов, которые загружены в репозиторий CIM. MOF-файлы, описанные в данном свойстве, также используются при восстановлении репозитория CIM. Данному свойству соответствует параметр реестра `REG_MULTI_SZ`-типа `Autorecover MOFs`.
- `BackupInterval` — интервал создания резервной копии репозитория CIM. Данному свойству соответствует параметр реестра строкового типа `Backup Interval Threshold`.

- BackupLastTime — дата создания резервной копии репозитория CIM.
- BuildVersion — номер версии WMI, установленной в операционной системе. Например, в Windows Vista используется WMI версии 6000.16386, а в Windows XP — версии 2600.0000. Данному свойству соответствует параметр строкового типа Build.
- DatabaseDirectory — путь к репозитарию CIM. Данному свойству соответствует параметр реестра строкового типа Repository Directory.
- DatabaseMaxSize — максимальный размер репозитория CIM. Данному свойству соответствует параметр реестра REG_DWORD-типа Max DB Size.
- EnableEvents — определяет, разрешена ли подсистема событий WMI. Данному свойству соответствует параметр реестра строкового типа EnableEvents.
- EnableStartupHeapPreallocation — определяет, будет ли куча (часть оперативной памяти, используемая для хранения объектов) выделяться сразу при запуске службы WMI. Данному свойству соответствует параметр реестра REG_DWORD-типа EnableStartupHeapPreallocation.
- HighThresholdOnClientObjects — верхнее пороговое значение очереди объектов от поставщиков. Достижение данного порогового значения приводит к прекращению приема объектов от поставщиков и возвращению поставщикам кода WBEM_E_OUT_OF_MEMORY. Данному свойству соответствует параметр реестра REG_DWORD-типа High Threshold On Client Objects (b).
- HighThresholdOnEvents — верхнее пороговое значение очереди событий от поставщиков. Достижение данного порогового значения приводит к прекращению приема событий от поставщиков и возвращению поставщикам кода WBEM_E_OUT_OF_MEMORY. Данному свойству соответствует параметр реестра REG_DWORD-типа High Threshold On Events (b).
- InstallationDirectory — путь к папке, содержащей файлы WMI. Данная папка хранит файлы службы WMI, MOF-файлы, файлы журнала, репозитарий CIM и основные программы командной строки, предназначенные для работы с репозитарием CIM. Свойству соответствует параметр реестра строкового типа Installation Directory.
- LastStartupHeapPreallocation — задает размер выделяемой кучи, если свойство EnableStartupHeapPreallocation равно true.
- LoggingDirectory — путь к папке, содержащей файлы журналов WMI. Данному свойству соответствует параметр реестра строкового типа Logging Directory.
- LoggingLevel — уровень протоколирования ошибок, который может принимать следующие значения: 0 (отключить протоколирование), 1 (краткое протоколирование ошибок) и 2 (полное протоколирование ошибок). Данному свойству соответствует параметр реестра строкового типа Logging.
- LowThresholdOnClientObjects — нижнее пороговое значение очереди объектов от поставщиков, по достижении которого скорость создания объектов замедлится. Данному свойству соответствует параметр реестра REG_DWORD-типа Low Threshold On Client Objects (b).

- `LowThresholdOnEvents` — нижнее пороговое значение очереди событий от поставщиков, по достижении которого скорость создания объектов замедлится. Данному свойству соответствует параметр реестра `REG_DWORD`-типа `Low Threshold On Events (b)`.
- `MaxLogFileSize` — максимальный размер журналов WMI. Данному свойству соответствует параметр реестра строкового типа `Log File Max Size`.
- `MaxWaitOnClientObjects` — время в миллисекундах, в течение которого объект может находиться в очереди. Данному свойству соответствует параметр реестра строкового типа `Max Wait On Client Objects (ms)`.
- `MaxWaitOnEvents` — время в миллисекундах, в течение которого событие может находиться в очереди. Данному свойству соответствует параметр реестра строкового типа `Max Wait On Events (ms)`.
- `MofSelfInstallDirectory` — путь к папке, в которую будут помещаться MOF-файлы, предназначенные для автоматического добавления новых поставщиков WMI, свойств и методов. Данному свойству соответствует параметр реестра строкового типа `MOF Self-Install Directory`.

Использование реестра

В предыдущем подразделе были рассмотрены свойства класса `Win32_WMISetting`, которые определяют настройки базы данных WMI. При этом при описании каждого свойства также указывался параметр реестра, который изменяется данным свойством. Сейчас же будут рассмотрены ветви реестра, в которых содержатся описанные выше параметры, а также несколько параметров, значения которых нельзя изменить с помощью свойств класса `Win32_WMISetting`.

Настройки репозитория. Параметры реестра `BUILD`, `Installation Directory` и `MOF Self-Install Directory` содержатся в ветви `HKLM\SOFTWARE\Microsoft\WBEM`.

Настройки сценариев. Параметры реестра `Default Namespace`, `Enable for ASP` и `MOF Self-Install Directory` хранятся в ветви `HKLM\SOFTWARE\Microsoft\WBEM\Scripting`.

В этой ветви также может находиться параметр `REG_DWORD`-типа `Default Impersonation Level`. Он определяет используемый по умолчанию (если не указан в сценарии) уровень заимствования прав доступа. По умолчанию значение данного параметра равно 3.

Настройки WMI. Остальные параметры реестра содержатся в ветви `HKLM\SOFTWARE\Microsoft\WBEM\CIMOM`.

Кроме параметров, данная ветвь реестра хранит вложенный подраздел `CIMOM`, содержащий сведения о CIM-совместимой базе данных WMI. Данный подраздел включает в себя следующие параметры:

- `DefaultRPCStackSize` — используемый по умолчанию размер буфера для хранения запросов RPC (строковый тип);
- `EnablePrivateObjectHeap` — определяет, будет ли использоваться куча для хранения объектов от поставщиков (тип `REG_DWORD`);
- `EnableObjectValidation` — если параметр равен 1, то будет выполняться проверка целостности объектов от поставщиков (тип `REG_DWORD`);

- Sink Transmit Buffer Size — размер буфера, используемого при передаче ответа на асинхронный запрос (строковый тип).

Использование программы winmgmt.exe

Программа winmgmt.exe, которая расположена в папке %systemroot%\system32\wbem, является службой инструментария управления Windows (WMI). При этом она поддерживает набор параметров, с помощью которых можно управлять работой репозитория CIM и WMI в целом.

- /backup <файл с расширением REP> — выполняет архивирование репозитория CIM.
- /restore <файл с расширением REP> — восстанавливает содержимое репозитория CIM из указанного файла-архива.
- /standalonehost <уровень аутентификации процесса svchost.exe> — перемещает службу инструментария управления Windows в отдельный процесс svchost.exe. Если уровень аутентификации не задан, используется уровень 4.
- /sharedhost — перемещает службу инструментария управления Windows в общий процесс svchost.exe.
- /salvagerepository — проверяет репозиторий CIM на корректность и в случае обнаружения неполадок восстанавливает его на основе стандартных MOF-файлов.
- /verifyrepository <путь к репозиторию CIM> — проверяет репозиторий CIM на корректность. Если не указывать путь к репозиторию CIM (или его копии), то будет выполнена проверка текущего репозитория CIM.
- /resetrepository — восстанавливает состояние репозитория CIM, в котором он находился в момент установки операционной системы.
- /resyncperf <PID службы WMI> — заново регистрирует в инструментарии управления Windows системные библиотеки производительности.
- /clearadap — удаляет из разделов реестра, принадлежащих службам, все параметры, связанные с инструментарием управления Windows.

Использование классов репозитория CIM

Перед тем как описать способы работы с репозиторием CIM с помощью сервера сценариев Windows, стоит рассмотреть несколько простых способов работы с инструментарием управления Windows, чтобы вы смогли вспомнить, что это такое.

Программа WMIC

В стандартную поставку операционной системы Windows Vista входит программа командной строки wmic.exe, расположенная в папке %systemroot%\SYSTEM32\wbem. С помощью этой программы можно получить доступ к репозиторию CIM, даже не зная основ работы с инструментарием управления Windows.

Параметры программы

При первом запуске программы выполняется ее установка в системе, после чего будет выведена командная строка `wmic`. Программа `wmic.exe` поддерживает множество параметров, но рассмотрены будут лишь некоторые из них. Весь же список команд можно просмотреть с помощью команды `wmic /?`.

- `/NAMESPACE:<пространство имен>` — определяет пространство имен, в котором будет выполнен поиск псевдонима.
- `/NODE:<компьютер>` — указывает компьютер, к репозитарию которого будет отправлен запрос WQL.
- `/USER:<пользователь>` — определяет пользователя, от имени которого будет выполнен запрос. Если вы выполняете запрос на локальном компьютере, то указывать пользователя нельзя (можно выполнять запрос только от текущего пользователя).
- `/PASSWORD:<пароль>` — указывает пароль для учетной записи пользователя, от имени которого будет выполняться запрос к репозитарию.
- `/ROLE:<пространство имен>` — определяет пространство имен, содержащее описание доступных псевдонимов. По умолчанию значение данного параметра равно `\\root\cli`.
- `/LOCALE:<язык>` — указывает идентификатор языка, к пространству имен которого будет выполнен запрос.
- `CONTEXT` — данный параметр указывается без косой черты (и без псевдонима). Он отображает список значений всех параметров, используемых в программе `wmic.exe`.

Доступные в программе псевдонимы

Кроме параметров, программа `wmic.exe` также поддерживает набор псевдонимов (ключевых слов, ассоциированных с конкретным запросом WQL). Список доступных псевдонимов можно просмотреть с помощью команды `wmic.exe /?`. Из них наиболее интересны следующие параметры:

- `BASEBOARD` — возвращает сведения о материнской плате, установленной на компьютере: модель, производителя, описание (класс `Win32_BaseBoard`);
- `BIOS` — возвращает сведения о версии BIOS материнской платы: основную и дополнительную версию, производителя, название, язык, описание, дату установки, указание, является ли данная версия BIOS основной (класс `Win32_BIOS`);
- `BOOTCONFIG` — возвращает сведения о настройках загрузочного меню компьютера;
- `COMPUTERSYSTEM` — возвращает сведения об операционной системе компьютера (класс `Win32_ComputerSystem`);
- `GROUP` — возвращает сведения о группах учетных записей, зарегистрированных в системе: SID, присвоенный группе, название группы, ее описание (класс `Win32_Group`);

- LOGICALDISK — возвращает сведения о логических дисках компьютера: о метке диска, букве, серийном номере, описании, размере, указании, включена ли квота, используется ли программное сжатие (класс Win32_LogicalDisk);
- OS — возвращает сведения об операционной системе, установленной на компьютере: системную папку, версию, пакет обновлений, язык операционной системы, размер виртуальной памяти, размер оперативной памяти, количество процессоров, количество запущенных в данный момент процессов, регистрационные данные пользователя (класс Win32_OperatingSystem);
- PAGEFILESET — возвращает сведения о конфигурации файла подкачки: путь к нему, максимальный размер, используемый в данный момент размер (класс Win32_PageFileSetting);
- PARTITION — возвращает сведения о разделах, имеющихся на жестком диске: является ли активным, размер кластера раздела, общее количество блоков, описание раздела (класс Win32_DiskPartition);
- PRINTER — возвращает сведения о настройках принтеров, установленных на компьютере: название принтера, производителя, приоритет, номер порта, поддерживаемая бумага (класс Win32_Printer);
- PROCESS — возвращает сведения о запущенных на компьютере в данный момент процессах: название процесса, PID, командную строку, вызвавшую процесс, учетную запись, от имени которой работает процесс, размер используемой оперативной памяти, виртуальной памяти (класс Win32_Process);
- SHARE — возвращает сведения об общих папках: название, описание, тип (класс Win32_Share);
- STARTUP — возвращает сведения обо всех программах, запускаемых вместе с системой, а также о файлах, которые они запускают (класс Win32_StartupCommand).

Таким образом, программа `wmic.exe` имеет следующий синтаксис: `wmic.exe <параметры> <псевдоним>`. Если же вы вошли в оболочку программы, то достаточно указывать только параметры и псевдоним без названия программы.

Примеры работы с программой

Напоследок рассмотрим несколько примеров работы с данной программой:

- `wmic OS Get DataExecutionPrevention_Available` — показывает, используется ли на данный момент в операционной системе механизм DEP;
- `wmic /OUTPUT:C:\cpu_settings.txt CPU GET /VALUE` — создает на диске C: файл `cpu_settings.txt`, который будет содержать описание процессора, установленного на вашем компьютере;
- `wmic PROCESS WHERE Name="cmd.exe" CALL Terminate` — завершает работу процесса с именем `cmd.exe`;
- `wmic PROCESS CALL Create calc.exe` — запускает процесс `calc.exe`;
- `wmic SERVICE WHERE Name="browser" LIST` — отображает сведения о настройках службы обозревателя компьютеров.

Программа WBEMTEST.exe

Получить доступ к репозитарию CIM можно и с помощью программы `wbemtest.exe`, имеющей графический интерфейс. Эта программа также расположена в папке `%systemroot%\SYSTEM32\wbem`.

Программа `wbemtest.exe` позволяет перечислять, открывать и создавать классы или экземпляры этих классов, а также выполнять WQL-запросы к репозитарию CIM (представляют собой разновидность SQL-запросов) либо методы классов.

Для работы с данной программой необходимо знать названия классов, свойств и методов, реализуемых в этих классах. Ранее на страницах данной книги уже были описаны названия новых классов и свойств, доступных в операционной системе Windows Vista, а также перечень основных свойств некоторых из уже существовавших в предыдущих операционных системах классов.

Подключаемся к пространству имен

Перед тем как начать работу с программой `wbemtest.exe`, необходимо подключить к нужному пространству имен. Для этого в окне программы нужно нажать кнопку **Подключить**. После этого перед вами отобразится окно. В этом окне нужно будет указать пространство имен, к которому выполняется подключение, а также имя пользователя и пароль (если имя пользователя и пароль не указаны, то будет выполнено подключение от имени текущего пользователя).

После того как вы введете необходимые данные, перед вами опять отобразится начальное окно программы, но теперь все ее кнопки будут активны.

Использование запросов WQL для получения экземпляров классов

Обратите внимание на кнопку **Запрос** основного окна программы `wbemtest.exe`. С помощью данной кнопки можно отобразить окно, позволяющее вводить WQL-запросы к репозитарию CIM и получать экземпляры классов, хранящиеся в нем.

После того как вы введете WQL-запрос, перед вами отобразится окно, содержащее список всех найденных экземпляров, удовлетворяющих введенному вами запросу. Если вы дважды щелкнете кнопкой мыши на каком-нибудь из экземпляров, то перед вами отобразится список всех доступных в этом экземпляре свойств, а также их значения.

Запросы WQL делятся на три типа: запросы данных, запросы схем и запросы событий. На страницах данной книги будет рассмотрен только синтаксис запросов данных. Эти запросы предназначены для получения экземпляров класса, а также их ассоциаций.

Самым простым запросом данных является запрос `Select * From <класс>`, с помощью которого можно отобразить список всех экземпляров указанного класса, доступных на компьютере. Например, попробуйте ввести запрос для вывода экземпляров класса `Win32_OperatingSystem`.

Если же вам необходимо просмотреть значения не всех свойств класса, а лишь некоторых из них, то можно воспользоваться более конкретным запросом: `Select`

<список свойств, значения которых нужно вывести> From <класс>. Например, если вы введете запрос `select SystemDevice, LargeSystemCache, EncryptionLevel from Win32_OperatingSystem`, то сможете просмотреть значения только этих свойств класса.

Можно также воспользоваться запросом вида `select <свойства> from <класс> where <свойство>=<значение>`. Он применяется в том случае, если нужно отобразить только те экземпляры класса, свойства которых имеют указанные в запросе значения. При этом можно использовать операторы AND, NOT или OR, чтобы указать значения сразу нескольких свойств. Например, `select SystemDevice, EncryptionLevel from Win32_OperatingSystem where Primary="TRUE" AND EncryptionLevel>40`.

Ключевое слово `where` может содержать ключевое слово `__CLASS`, с помощью которого можно ограничить вывод WQL-запроса только одним классом. Например, чтобы ограничить список выводимых экземпляров только классом `Win32_Service`, нужно воспользоваться запросом вида `select * from Win32_BaseService where __CLASS="Win32_Service"`.

Общая схема запроса данных WQL выглядит следующим образом: `select <свойства>[*] from <класс> [where [__CLASS="класс"] [AND|OR] [<свойство>=<значение>]]`.

Выполнение методов классов

Если же вам нужно выполнить метод, который описан в одном из классов соответствующего пространства имен, то необходимо воспользоваться кнопкой **Выполнить метод**. После нажатия этой кнопки программа попросит вас ввести название класса, в котором находится нужный вам метод. После того как вы это сделаете, перед вами отобразится список методов данного класса, среди которых вы должны будете выбрать нужный метод.

Если метод, который вы хотите вызвать, не требует ввода никаких входящих параметров, то после его выбора просто нажмите кнопку **Выполнить**. В противном случае перед нажатием этой кнопки также нажмите кнопку **Входные параметры**. После этого в появившемся окне **Редактор объекта _PARAMETERS** дважды щелкните кнопкой мыши на каждом параметре, значение которого нужно изменить (вызвав тем самым окно **Редактор свойств**), потом измените значение параметра и нажмите кнопку **Сохранить**.

Если метод, который вы выполнили, возвращает какие-либо значения, их можно просмотреть после его выполнения, просто нажав кнопку **Выходные параметры**.

Сервер сценариев Windows

Теперь рассмотрим несколько простых примеров использования инструментария управления Windows в сервере сценариев WSH.

Первый пример достаточно прост. В нем будет произведена попытка отредактировать параметр реестра с помощью одного из методов класса WMI. Например, изменим параметр строкового типа `Auto` из ветви реестра `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug` (листинг П2.1).

Листинг П2.1. Вызов методов инструментария управления Windows

```
Set WMIReg = GetObject("winmgmts:{impersonationLevel=impersonate}!root/Default:StdRegProv")
'если метод возвращает значение, отличное от 0, значит,
'произошла ошибка и изменение параметра не произошло
if WMIReg.SetStringValue (&H80000002,"SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug","Auto", 0) then MsgBox "Произошла ошибка", vbCritical
```

Рассмотрим данный пример детально.

Моникер

С помощью первой строки примера происходит подключение к репозитарию CIM и получение указателя на объект WMI, определенный моникером, являющимся атрибутом метода `GetObject`.

ПРИМЕЧАНИЕ

Моникером называется объект, предназначенный для доступа к другому объекту.

Наиболее простой синтаксис моникера следующий: `Winmgmts:<класс>[.<ключевое свойство>=<значение>]`. С его помощью производится подключение к классу или экземпляру класса (если указывается ключевое свойство и значение) пространства имен `root\cimv2` (именно оно используется по умолчанию). В качестве примера можно привести следующий моникер: `winmgmts:Win32_Service.Name="DHCP"`.

Если же нужно изменить компьютер, а также пространство имен, к которому происходит подключение, то нужно воспользоваться следующим синтаксисом моникера: `Winmgmts:\\<компьютер>\<пространство имен>:<класс>[.<ключевое свойство>=<значение>]`. Если вместо названия компьютера поставить точку, то подключение будет выполнено к локальному компьютеру (например, `winmgmts:.\.\Root\Cimv2:Win32_Service.Name="DHCP"` или `winmgmts://NOUT/Root/Cimv2:Win32_Service.Name="DHCP"` — заметьте, что в первом случае для указания пространства имен и подключаемого компьютера используется символ `\`, а во втором случае — символ `/`; эти символы равнозначны).

Уровень заимствования прав

Следующий пример моникера: `Winmgmts:[{impersonationLevel=<уровень заимствования прав>}!][\\<компьютер>\<пространство имен>:<класс>[.<ключевое свойство>=<значение>]`. В нем используется новое ключевое слово `impersonationLevel`, которое определяет уровень заимствования прав пользователя. Возможны следующие уровни заимствования:

- Anonymous — использовать минимальные привилегии;
- Identify — запросить привилегии пользователя, если нужно, по умолчанию же использовать минимальные привилегии;
- Impersonate — использовать привилегии пользователя.

Уровень аутентификации

Еще один пример моникера: `Winmgmts:[{impersonationLevel=<уровень заимствования прав>, AuthenticationLevel=<уровень аутентификации>}!]\[<компьютер>]\<пространство имен>:<класс>[.<ключевое свойство>=<значение>]`. В нем используется ключевое слово `AuthenticationLevel`, которое определяет уровень аутентификации и может принимать следующие значения.

- `Default` — использовать настройки аутентификации Windows. Данный уровень используется по умолчанию.
- `None` — не производить аутентификацию.
- `Connect` — выполнять аутентификацию клиента только при соединении с сервером.
- `Call` — выполнять аутентификацию клиента при получении любого запроса от сервера.
- `Pkt` — проверять все данные, получаемые от клиента.
- `PktIntegrity` — проверять любые данные, переданные как сервером, так и клиентом.
- `PktPrivacy` — проверять любые данные, переданные как сервером, так и клиентом, а также выполнять шифрование аргументов удаленных вызовов процедур.

Привилегии

Еще один пример синтаксиса моникера: `Winmgmts:[{impersonationLevel=<уровень заимствования прав>, AuthenticationLevel=<уровень аутентификации>}, (привилегия, привилегия и т.д.)!]\[<компьютер>]\<пространство имен>:<класс>[.<ключевое свойство>=<значение>]`. В нем указываются дополнительные привилегии (если перед привилегией поставить восклицательный знак, то ее использование будет запрещено).

Ранее в книге уже рассматривались привилегии (при описании объектов DACL) — именно эти привилегии и можно указывать в данном синтаксисе моникера. Единственное, вы должны удалить от имени привилегии префикс `Se` и суффикс `Privilege`.

Например, чтобы добавить привилегию `SeAuditPrivilege`, которая уже была рассмотрена, ее моникеру, просто добавьте в синтаксис моникера привилегию `Audit`, а для добавления привилегии `SeEnableDelegationPrivilege` нужно добавить `EnableDelegation`.

Чтобы же разрешить выполнение перезагрузки компьютера, нужно использовать моникер `winmgmts:{impersonationLevel = Impersonate, (Shutdown)}!\.\root\cimv2:Win32_OperatingSystem`.

Вызов методов WMI

С помощью второй строки примера вызывается метод класса, позволяющий изменить значение параметра строкового типа. Как правило, каждый метод после своей

работы возвращает код выполнения, который определяет либо ошибку, произошедшую при работе метода, либо то, что метод выполнен без ошибок (в этом случае всегда возвращается значение 0).

Использование WQL-запросов

Еще одним способом работы с репозитарием CIM является выполнение запросов WQL (листинг П2.2).

Листинг П2.2. Использование WQL-запроса

```
'Подключаемся к пространству root\cimv2 репозитария
'указанного компьютера
Set WMIReg = GetObject("winmgmts:{impersonationLevel=impersonate}!/" &
InputBox("Введите название компьютера, к которому производится подключение
(введите точку, если нужно подключиться к локальному компьютеру).", "Connect...")
& "/root/CimV2")
'Выполняем запрос данных WQL, возвращающий все экземпляры
'данного класса
'После этого отображаем содержимое свойств всех найденных
'экземпляров в цикле For Each...Next
Set WMI = WMIReg.ExecQuery ("select * from Win32_Registry")
For Each i in WMI
MsgBox "Подключились к " & i.Name & " : " & i.Status & vbCrLf & "Текущий размер
реестра: " & i.CurrentSize & " Mb" & vbCrLf & "Максимальный размер реестра: " & i.
MaximumSize & " Mb"
Next
```

WQL-запрос к репозитарию CIM выполняется с помощью метода ExecQuery объекта SWbemServices объектной модели WMI. Заметьте, что результат запроса возвращается в виде массива. С помощью же цикла for each in выполняется перечисление каждого из элементов массива.

Редактирование значений свойств

Следующий листинг приводит пример изменения значения свойства экземпляра класса (листинг П2.3).

Листинг П2.3. Запись в свойство инструментария WMI

```
Set WMIReg = GetObject("winmgmts:{impersonationLevel=impersonate}!/" &
InputBox("Введите название компьютера, к которому производится подключение
(введите точку, если нужно подключиться к локальному компьютеру).", "Connect...")
& "/root/CimV2")
Set WMI = WMIReg.ExecQuery ("select * from Win32_Registry")
For Each i in WMI
MsgBox "Подключились к " & i.Name & " : " & i.Status & vbCrLf & "Текущий размер
реестра: " & i.CurrentSize & " Mb" & vbCrLf & "Максимальный размер реестра: " & i.
MaximumSize & " Mb" & vbCrLf & "Желаемый размер реестра: " & i.ProposedSize & " Mb"
Temp = InputBox ("Введите новый желаемый раздел реестра", "Только не перестарайтесь")
'Конечно, размер реестра в виде строки – это ваше дело,
'но мы такие значения все-таки отфильтруем
```



```
if NOT IsNumeric(Temp) then WScript.Quit
if Temp = "" then WScript.Quit
```

'Следующие две строки записывают в свойство, обратите на них внимание

```
i.ProposedSize = temp
```

```
i.Put_
```

```
MsgBox "Подключились к " & i.Name & " : " & i.Status & vbCrLf & "Текущий размер реестра: " & i.CurrentSize & " Mb" & vbCrLf & "Максимальный размер реестра: " & i.MaximumSize & " Mb" & vbCrLf & "Желаемый размер реестра: " & i.ProposedSize & " Mb"
Next
```

Обратите внимание, что после записи в свойство с помощью строки вида <объект>.<свойство>=<значение> необходимо использовать метод Put_ данного объекта. С его помощью новое значение сохраняется в репозитории CIM (если его не использовать, то свойство будет изменяться только на время работы сценария, после чего опять примет свое исходное значение).

Приложение 3

Новые программы командной строки

Данное приложение содержит перечень всех программ командной строки, впервые появившихся в операционной системе Windows Vista (точнее, тех, которых не было в операционной системе Windows XP, хотя, возможно, они уже существовали в Windows Server 2003 или как отдельные программы, которые можно было скачать с сайта Microsoft).

В приложении не описаны стандартные команды командной строки `cmd.exe`, которые являются ее частью, а не отдельными программами. Например, к таким командам относятся команды, представленные в табл. ПЗ.1.

Таблица ПЗ.1. Встроенные команды командной строки

Команда	Описание
<code>chcp <кодировка страницы></code>	Изменяет кодовую страницу, используемую командной строкой. При запуске без параметров отобразится номер текущей кодовой страницы
<code>Date</code>	Отображает текущую дату и предлагает ввести новую
<code>path <папка; папка...></code>	Определяет папки, исполняемые файлы из которых можно выполнять из окна Выполнить, не указывая при этом путь к файлу. Если не указать папку, то отобразится текущий список папок
<code>prompt <строка></code>	Изменить приглашающую строку <code>cmd.exe</code> . Вы можете указать как произвольный текст, так и различные переменные. Список переменных можно просмотреть, выполнив команду <code>prompt /?</code> . По умолчанию используется строка, отображающая путь к текущей папке
<code>set <переменная>=<значение></code>	Позволяет изменить значения переменных сред
<code>Time</code>	Отображает текущее время и предлагает ввести новое
<code>Ver</code>	Выводит версию операционной системы
<code>vol <буква диска></code>	Отображает метку диска и серийный номер
<code>Pause</code>	Приостанавливает работу командной строки <code>cmd.exe</code>
<code>fc</code>	Программа для сравнения двух файлов

Большинство новых программ Windows Vista уже были описаны в главах данной книги, поэтому здесь они описываться не будут. Это такие программы, как: `auditpol.exe`, `icacils.exe`, `wbadmin.exe`, `where.exe`, `bitsadmin.exe`, `msfeedssync.exe`, `netcfg.exe`,

query.exe, quser.exe, RcpPing.exe, winrm.exe, winrs.exe, certreq.exe, certutil.exe, fveupdate.exe, TpmInit.exe, ocsetup.exe, PkgMgr.exe, wusa.exe, wevtutil.exe, WinSAT.exe и takeown.exe.

Описание остальных программ, о которых в книге еще не упоминалось, также будет включать в себя примеры работы с параметрами этих программ командной строки.

Работа с файлами

Convert.exe

Данная программа не является новой. Она предназначена для конвертирования файловой системы раздела диска и поддерживает один недокументированный параметр — /NoChkdsk. С его помощью можно отключить выполнение проверки целостности диска перед его конвертированием.

Forfiles.exe

Данная программа предназначена для обработки файлов, хранящихся в одной папке, и позволяет выполнить определенную команду для этих файлов. Основной синтаксис данной программы следующий: `forfiles /p <путь к папке> /m <маска файлов> /c <команда>`. При этом в значении параметра /c можно использовать следующие константы программы (табл. ПЗ.2).

Таблица ПЗ.2. Константы программы forfiles.exe

Имя	Описание
@file	Заменяется именем текущего файла, удовлетворяющего параметру /m
@fname	Заменяется именем текущего файла без расширения, удовлетворяющего указанной в параметре /m маске
@ext	Заменяется расширением файла, удовлетворяющего параметру /m
@path	Заменяется полным путем к файлу, удовлетворяющему параметру /m
@relpath	Заменяется относительным путем к файлу, удовлетворяющему указанной в параметре /m маске
@isdir	Заменяется значением true, если текущий удовлетворяющий маске файл является папкой. В противном случае заменяется значением false
@fsize	Заменяется размером файла, удовлетворяющего параметру /m
@fddate	Заменяется датой последнего изменения файла, удовлетворяющего указанной в параметре /m маске
@ftime	Заменяется временем последнего изменения файла, удовлетворяющего указанной в параметре /m маске

Например, чтобы открыть в Блокноте все файлы с расширением VBS, расположенные в папке %systemroot%\system32, нужно воспользоваться следующей командой: `forfiles /p c:\windows\system32 /m *.vbs /c "notepad.exe /c @path"`. Заметьте, что в значении параметра /c после названия программы notepad.exe и перед константой идет параметр /c. Если его не указать, то просто будет открыта программа

notepad.exe, а не удовлетворяющие маске файлы в нем. Параметр /с нужно указывать и для других программ, если константы являются их аргументами.

В программе forfiles.exe также можно использовать параметр /D. Его значением должна быть дата в формате dd.MM.yyyy, перед которой может находиться знак + или -.

Если перед датой стоит знак +, то команда из параметра /C будет применяться для всех файлов, созданных позднее даты, указанной в параметре /D. Если же перед датой стоит знак -, то команда будет применяться для всех файлов, созданных до даты, указанной в параметре /D.

О других параметрах данной программы вы можете узнать, введя в командной строке команду forfiles /?.

Robocopy.exe

Данная программа предназначена для выполнения копирования одного или множества файлов, расположенных в определенной папке и ее подпапках. При этом папки могут находиться как на локальном компьютере, так и на удаленном.

После копирования программа отображает таблицу, описывающую каждый скопированный файл: имя файла, процент копирования, размер файла. Она также строит таблицу, отображающую количество скопированных или пропущенных файлов, папок, байт, а также время копирования.

Основной синтаксис программы следующий: ROBOCOPY <папка-источник> <папка назначения> <список копируемых файлов> <параметры>. Если список файлов не указан, то будет выполнено копирование всех файлов (по умолчанию установлена маска файлов *.*). Можно использовать следующие параметры.

Копирование файлов и папок

Чтобы копировать все файлы и папки, включая пустые файлы, нужно использовать параметр /E. Если же пустые папки копировать не нужно, то необходимо указать параметр /S. Можно также указать параметр /LEN:<уровень>, чтобы определить максимальный уровень вложения папок, содержимое которого будет копироваться. Параметр /LEN:<уровень> должен использоваться с одним из предыдущих параметров, иначе копирование будет выполняться только в указанной папке, но не в папках, вложенных в нее.

Например, для копирования файлов можно воспользоваться командой robocopy d:\windows c:\users\1\desktop\testing *.* /S /LEV:3, которая будет просматривать содержимое папки d:\windows вплоть до третьего уровня вложения папок.

Копирование только файлов

Чтобы копировать только файлы, которые не открыты в монопольном режиме другими программами, нужно воспользоваться параметром /Z. Можно также воспользоваться параметром /B, чтобы выполнялось копирование даже тех файлов, которые открыты в монопольном режиме (теневое копирование). Теневое копирование проходит дольше, чем обычное.

Кроме того, можно использовать комбинацию данных параметров (/ZB), чтобы теневое копирование применялось только к тем файлам, доступ к которым при нормальном копировании был бы заблокирован.

Можно также указать параметр /EFSRAW, чтобы выполнялось копирование в режиме EFS RAW всех зашифрованных с помощью EFS файлов.

Определение копируемых сведений

Следующим набором параметров, которые можно указывать в программе, являются те, которые указывают атрибуты и метаданные файлов, копируемые от файлов и папок источников. Например, с помощью параметра /COPY:<флаги> можно указать флаги копируемой у файлов источника информации (также можно использовать параметр /DCOPY:<флаги>, чтобы указать флаги для копируемых папок). По умолчанию значение данного параметра равно DAT. Существуют следующие флаги, позволяющие копировать:

- D — содержимое файлов;
- A — атрибуты файлов;
- T — временные метки файлов;
- S — информацию об ACL файла;
- O — информацию о владельце файла;
- U — сведения аудита для файла.

Можно также указать один из следующих параметров, которые заменяют собой параметр /COPY:<флаги> с его флагами.

- /SEC — заменяет собой параметр /COPY:DATS.
- /COPYALL — заменяет собой параметр /COPY:DATSOU.
- /NOCOPY — указывает на отсутствие необходимости копировать информацию о файлах. В этом случае не будет скопировано ни одного файла, однако в командной строке будет построено дерево папок и файлов папки-источника.

Можно самостоятельно указать атрибуты, которые будут добавляться к копируемым файлам или удаляться из них. Для этого применяются команды /A+:<атрибуты> и /A-:<атрибуты> соответственно. В качестве атрибутов могут указываться стандартные флаги атрибутов, применяемые к программе attrib.exe.

- R — Только чтение.
- A — Архивный.
- S — Системный.
- H — Скрытый.
- C — сжимать файл. Не может устанавливаться вместе с атрибутом E.
- E — шифровать файл. Не может устанавливаться вместе с атрибутом C.

Определение копируемых файлов

Можно также указать параметры, определяющие файлы, которые будут копироваться. Например, чтобы указать копирование файлов, имеющих только определенные атрибуты, нужно воспользоваться одним из следующих параметров.

- /A — копировать только файлы с установленным атрибутом Архивный.
- /M — копировать только файлы с установленным атрибутом Архивный, после чего снимать данный атрибут.

- /IA:<атрибуты> — копировать только файлы с указанными атрибутами.
- /XA:<атрибуты> — копировать только файлы, не имеющие данных атрибутов.
- /XF <файлы> — не копировать файлы с указанными именами.
- /XD <папки> — не копировать папки с указанными именами.
- /MAX:<размер в байтах> — не копировать файлы, размер которых больше указанного в параметре.
- /MIN:<размер в байтах> — не копировать файлы, размер которых меньше указанного в параметре.
- /MAXLAD:<время последнего доступа> — не копировать файлы, последний доступ к которым был осуществлен позднее даты, указанной в атрибуте. Время можно указывать в днях (но не больше чем 1900) или в формате даты уууymmdd.
- /MINLAD:<время последнего доступа> — не копировать файлы, последний доступ к которым был осуществлен ранее даты, указанной в атрибуте. Время можно указывать как в днях, так и в виде даты.
- /MAXAGE:<время создания> — не копировать файлы, созданные позднее даты, указанной в атрибуте. Время можно указывать как в днях, так и в виде даты.
- /MINAGE:<время создания> — не копировать файлы, созданные ранее даты, указанной в атрибуте. Время можно указывать как в днях, так и в виде даты.

Дополнительные атрибуты

Можно также указывать следующие специфические атрибуты.

- /MOV — вместо копирования перемещать файлы.
- /MOVE — вместо копирования перемещать как файлы, так и папки.
- /CREATE — в папке назначения создает только дерево папки-источника: содержащиеся в ней папки, а также все файлы, однако их размер будет равен нулю.
- /MON:<количество минут> — копирует файлы в режиме мониторинга. Иначе говоря, программа через указанное количество минут проверяет содержимое папки-источника и, если находит в нем новые файлы, копирует их в папку назначения.

Об остальных параметрах программы вы сможете узнать, введя команду `robocopy /?`.

Работа в сети

Iscscli.exe

Управляет работой клиента iSCSI и позволяет ему подключаться к серверу. Для просмотра описания параметров программы введите команду `iscsi /?`.

Waitfor.exe

Позволяет отправить сигнал всем компьютерам сети или только одному из компьютеров. Перед отправкой сигнала на удаленных компьютерах нужно запустить программу `waitfor.exe` в режиме ожидания.

Синтаксис данной программы следующий:

- `waitfor /S <удаленный компьютер> /U <пользователь> /P <пароль> /SI <сигнал>` — немедленно отправляет сигнал, указанный в команде, удаленному компьютеру;
- `waitfor <сигнал> /T <количество секунд ожидания>` — запускает программу `waitfor.exe` в режиме ожидания получения сигнала (как только программа получит сигнал, переданный с помощью описанной выше команды, она завершит свою работу, отобразив сообщение о том, что сигнал получен).

Whoami.exe

Выводит сведения об определенном пользователе компьютера. Если вы выполните программу без параметров, то отобразится имя компьютера и ваше имя. Можно также ввести команду `whoami /?`, чтобы просмотреть описание параметров данной программы.

Рассмотрим некоторые возможности данной программы.

- `/user` — выводит список работающих в данный момент пользователей. В списке отображается имя компьютера, имя пользователя и его SID.
- `/groups` — отображает список групп, имеющихся на данном компьютере, а также их SID, тип группы, указания, разрешена ли по умолчанию и разрешена ли в данный момент конкретная группа.
- `/priv` — выводит список возможных привилегий, их описание и указание, установлены ли они для текущего пользователя.
- `/all` — отображает список пользователей, групп и возможных привилегий.
- `/logonid` — выводит SID текущего пользователя.
- `/fqdn` — отображает имя пользователя в формате FQDN.
- `/upn` — выводит имя пользователя в формате UPN.

WSManHTTPConfig.exe

Данная программа имеет три параметра:

- `Install` — создает параметр REG_BINARY-типа `http://+:80/wsman/` в ветви реестра `HKLM\SYSTEM\CurrentControlSet\Services\HTTP\Parameters\UrlAcInfo`;
- `Uninstall` — удаляет параметр REG_BINARY-типа `http://+:80/wsman/` из ветви реестра `HKLM\SYSTEM\CurrentControlSet\Services\HTTP\Parameters\UrlAcInfo`;
- `Quickconfig` — выполняет быструю настройку.

Работа со сценариями

Choice.exe

Эта программа может применяться в сценариях для отображения сообщения выбора определенного действия. Рассмотрим ее возможности на примерах.

- `Choice /M "Select Y for start script. Select N for load script. Select C for exit" /C YNC` — в командной строке отображает строку, определенную в параметре `/M`.

После этой строки идет список клавиш, которые можно нажать, определенный параметром /C. Пользователь должен нажать одну из возможных клавиш, чтобы программа choice.exe завершила свою работу. При завершении работы программа возвращает нажатую пользователем клавишу в переменную ERRORLEVEL. При этом если пользователь нажал первую клавишу набора, то значение этой переменной будет равно 1, если нажал вторую клавишу набора, то 2 и т. д.

Если пользователь нажмет клавишу, которая не определена в параметре /C, то прозвучит звуковой сигнал. В этом случае работа программы choice.exe завершена не будет.

- Choice /M "Select Y for start script. Select C for exit" /C YC /T 3 /D Y — в командной строке выводит строку, определенную в параметре /M. После нее идет список клавиш, которые можно нажать, определенный параметром /C. Если пользователь не сделает выбор в течение количества секунд, определенного параметром /T, то программа choice.exe завершит свою работу, вернув символ, определенный в параметре /D, в переменную ERRORLEVEL (точнее, вернув не символ, а его расположение в наборе, определенном параметром /C).

Если пользователь нажмет клавишу, которая не определена в параметре /C, то прозвучит звуковой сигнал. В этом случае работа программы choice.exe завершена не будет.

- Choice /M "Select button "S" for start script. Select button "E" for exit" /C SE /N — в командной строке отображает строку, определенную в параметре /M. После этой строки не будет идти список клавиш, которые можно нажать. В остальном данная команда аналогична предыдущим рассмотренным командам.

Timeout.exe

Данная программа предназначена для работы в сценариях и позволяет указать время ожидания определенного события. Синтаксис данной команды следующий: timeout /T <количество секунд>. После вызова команды на экране командного процессора отобразится строка обратного отсчета. После окончания отсчета или после нажатия пользователем любой клавиши работа программы timeout.exe будет завершена.

Можно также запретить завершение работы программы после нажатия пользователем любой клавиши. Для этого при запуске программы нужно указать параметр /NOBREAK.

Другие программы

Clip.exe

С помощью данной программы можно поместить содержимое файла или вывод определенной команды в буфер обмена операционной системы Windows Vista.

Например, с помощью команды `diskraid /? | clip` можно поместить описание новой программы операционной системы `diskraid.exe` в буфер обмена Windows, а с помощью команды `clip < "путь к файлу и его имя"` можно поместить в буфер обмена содержимое указанного файла.

Cmdkey.exe

Программа позволяет управлять хранилищем паролей пользователя. Примером ее использования являются следующие команды.

- `Cmdkey /list` — выводит список всех хранимых пользователем паролей. Данные пароли ему больше не нужно будет вводить при аутентификации — операционная система сама будет их предоставлять требующим аутентификации программам.
- `Cmdkey /add:<адрес компьютера> /user:<имя> /pass:<пароль>` — добавляет к хранилищу пользовательских паролей новый пароль для указанного удаленного компьютера.
- `Cmdkey /delete:<адрес компьютера>` — удаляет пароль для подключения к данному компьютеру.
- `Cmdkey /delete /ras` — удаляет пароли для удаленного подключения через модем.

Данным хранилищем можно управлять и с помощью окна **Сохранение имен пользователей и паролей**. Чтобы вызвать его, достаточно запустить программу `netplwiz.exe`, перейти на вкладку **Дополнительно** и нажать кнопку **Управление паролями**.

ПРИМЕЧАНИЕ

Окно **Сохранение имен пользователей и паролей** можно вызвать и с помощью команды `rundll32.exe KEYMGR.dll, KRShowKeyMgr` или команды `rundll32.exe shell32.dll, Control_RunDLL Keymgr.dll`.

Diskraid.exe

Если на вашем компьютере установлен RAID-массив, то для управления им вы можете воспользоваться новой программой `diskraid.exe`. Для ее запуска необходимо, чтобы в операционной системе был зарегистрирован провайдер VDS.

Dispdiag.exe

Эта программа имеет несколько возможностей, которые могут быть полезны пользователям. Если запустить программу `dispdiag.exe` без параметров, то будет создан файл дампа в папке `%userprofile%\AppData\Local\Temp`. Можно также запустить программу с параметром `-testacpi`. После этого будет запущен режим интерпретации клавиш: когда вы будете нажимать клавиши, на экране будет отображаться нажатая клавиша, а также ее код и скан-код. Чтобы выйти из этого режима, достаточно нажать клавишу `Esc`.

Gpscript.exe

Регистрирует сценарии для автоматического запуска при входе или выходе пользователя (или всех пользователей) из системы в GPO. Для этого применяются следующие параметры программы: /RefreshSystemParam, /Shutdown, /Startup, /Logoff, /Logon.

Setx.exe

Позволяет изменять состояние переменных окружения локального или удаленного компьютера. Фактически данная программа изменяет или создает параметры ветви реестра HKCU\Environment (для пользовательских переменных окружения) или для системных переменных окружения параметры в ветви реестра HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment.

ПРИМЕЧАНИЕ

Список переменных окружения компьютера можно просмотреть с помощью свойств класса Win32_Environment пространства имен \\root\cimv2. Он поддерживает свойства: Name (ключевое свойство, название переменной окружения), SystemVariable (является ли данная переменная окружения системной), UserName (ключевое свойство, компьютер и пользователя, создавшего переменную окружения) и VariableValue (значение переменной окружения).

Для просмотра описания всех возможностей данной программы введите команду setx.exe, далее же рассматриваются только основные способы ее использования.

- Setx MACHINE vista — создает на локальном компьютере пользовательскую переменную MACHINE, имеющую значение vista.
- Setx MACHINE vista /M — создает на локальном компьютере системную переменную MACHINE, имеющую значение vista.
- Setx /S vista /U administrator /P 1 MACHINE vista /M — от имени пользователя **Администратор**, имеющего пароль 1, создает на удаленном компьютере vista системную переменную MACHINE, имеющую значение vista.
- Setx MACHINE /K "HKKEY_CURRENT_USER\Control Panel\Desktop\Wallpaper" /M — создает на локальном компьютере системную переменную MACHINE, значение которой равно значению параметра wallpaper ветви реестра HKCU\Control Panel\Desktop.
- Setx MACHINE /F system.ini /A 3,12 /M — создает на локальном компьютере системную переменную MACHINE, значение которой нужно взять из файла system.ini. В нем нужно значение расположено на третьей строке, начиная с 12 столбца.

SLsvc.exe

Представляет собой исполняемый файл службы Лицензирование программного обеспечения. Этот файл поддерживает два параметра: -Install и -Delete. С их помощью можно установить и удалить данную службу соответственно.

Sxstrace.exe

Программа предназначена для настройки трассировки компонента операционной системы WinSxS, содержащего сведения обо всех установленных в системе программах.

Синтаксис данной программы, используемый для запуска трассировки, следующий: `sxstrace trace -logfile:<имя файла с расширением ETL> <параметры>`. Можно использовать следующие параметры программы.

- `-nostop` — не завершать процесс трассировки работы WinSxS.
- `-verbose` — этот параметр недокументирован. Он позволяет указать запись в файл трассировки подробных сведений о работе WinSxS.

После запуска процесса трассировки необходимо нажать клавишу **Enter**, чтобы процесс трассировки был завершен.

Данной программой можно воспользоваться и для фильтрации файла трассировки и его преобразования в удобочитаемый формат. Для этого применяется следующий синтаксис: `sxstrace Parse -logfile:<имя файла с расширением ETL> <параметры>`:

- `-outfile:<дополнительный файл>` — указывает путь к текстовому файлу, в который будут помещаться сведения о трассировке компонента WinSxS, представленные в удобочитаемом формате;
- `-filter:<имя приложения>` — позволяет указать программу, сведения о которой будут заноситься в файл, определенный в параметре `-outfile:<дополнительный файл>`.

Чтобы остановить трассировку, нужно воспользоваться командой `sxstrace stoptrace`.

Tssetup.exe

При использовании параметров `/Install:IIS` и `/Remove:IIS` программа позволяет установить или удалить сервер IIS.

Wecutil.exe

Позволяет управлять подписками на события, создаваемыми с помощью диспетчера Baseboard Management Controller или службы Журнал событий Windows. Для работы данной программы необходимо, чтобы служба Сборщик событий Windows была запущена. Описание всех параметров программы можно просмотреть, введя команду `wecutil.exe /?`. Вот некоторые возможности данной программы.

- `wecutil.exe es` — отображает список всех существующих в данный момент подписок на события.
- `wecutil.exe gs` — выводит информацию о подписке.

- `wecutil.exe ss <имя подписки> <параметры>` — изменяет параметры подписки. Список параметров можно просмотреть, введя команду `wecutil.exe ss -?`.
- `wecutil.exe cs <файл конфигурации подписки> /cup:<имя пользователя> /csp:<пароль>` — создает новую подписку на основе конфигурационного файла, указанного в команде. Пример конфигурационного файла можно просмотреть, введя команду `wecutil.exe cs -?`.
- `wecutil.exe ds <идентификатор подписки>` — удаляет указанную подписку.
- `wecutil.exe qc` — выполняет следующие шаги настройки работы с подписками: включает канал `ForwardedEvents`, устанавливает запуск службы **Сборщик событий Windows** в автоматический режим и запускает данную службу.

Wisptis.exe

Устанавливает значения параметров ветви реестра `HKCU\Software\Microsoft\Wisp`, предназначенных для настройки пера при использовании планшетного ПК.

Приложение 4

Использование ActiveX-объектов

История развития ActiveX-объектов началась еще в 1991 году. Именно тогда появилась такая технология, как OLE. Она представляет собой специальные объекты, предназначенные для выполнения операций связывания и внедрения. Уже в 1993 году появилась вторая версия объектов OLE, которая вобрала в себя понятия нескольких технологий, основанных на модели СОМ (модели компонентного объекта, предназначенной для создания и редактирования компонентных объектов). Особенностью модели СОМ является независимость СОМ-объектов от используемого в их создании языка программирования. Например, если раньше какой-то класс был создан с помощью языка С++, то для него приходилось писать дополнительные классы на этом же языке. СОМ-объект же преобразуется в промежуточный байт-код, не зависящий от языка программирования, на котором был изначально написан СОМ-объект. Второй же особенностью СОМ-объектов является то, что они будут одинаково работать, независимо от того, находятся они на локальном компьютере или на удаленном. Все эти особенности СОМ-объектов переняли и OLE-объекты второй версии.

Через три года, в 1996 году, на свет появилась технология ActiveX. ActiveX-объекты представляют собой полностью скомпилированные программы, которые не могут быть запущены непосредственно пользователем, но могут запускаться браузером Internet Explorer, самой операционной системой или любой другой программой. Несмотря на то что корпорация Microsoft позиционирует ActiveX-объекты как совершенно новую технологию, они являются развитием OLE-объектов и переняли от них основные функции и возможности. Каждый ActiveX-объект должен содержать подпись своего создателя, которая производится одним из центров сертификации и гарантирует, что данный ActiveX-объект не выполняет никаких вредоносных или подозрительных действий.

Каждый ActiveX-объект имеет уникальный идентификатор, называемый CLSID-номером, который позволяет идентифицировать этот ActiveX-объект среди других. CLSID-номера ActiveX-объектов строятся на основе даты создания объекта, случайного числа и конфигурации компьютера. Эти имена состоят из 32 чисел

в шестнадцатеричной системе счисления. Первые восемь чисел генерируются случайным образом. Следующие четыре числа создаются на основе текущей даты и времени. Остальные числа генерируются на основе конфигурации компьютера.

Все ActiveX-объекты, чтобы ими можно было пользоваться, должны быть описаны в дочернем подразделе ветви реестра `HKCR\CLSID`. Данный подраздел реестра включает в себя набор подразделов, названных в соответствии с CLSID-номерами ActiveX-объектов. В этих подразделах как раз и описываются настройки конкретных ActiveX-объектов.

Для чего же используются ActiveX-объекты? В операционной системе Windows Vista с помощью ActiveX-объектов можно создать специальную вкладку окна **Свойства** файлов определенного расширения, специальную команду контекстного меню файлов определенного расширения или различных элементов операционной системы Windows, специальный значок с расширенными возможностями настройки, надстройки для операционной системы или сторонних программ и т. д. В контексте данной книги основное внимание будет сконцентрировано на описании стандартных ActiveX-объектов операционной системы и на их использовании, создание же простых ActiveX-объектов, не требующих знания языков программирования будет рассмотрено лишь поверхностно.

Установка/удаление ActiveX-объекта. Каждый ActiveX-объект представляет собой файл с расширением `OCX`. Чтобы выполнить установку либо удаление ActiveX-объекта на основе данного файла, необходимо воспользоваться программой `regsvr32.exe`, имеющей следующие варианты синтаксиса.

- `regsvr32.exe <OCX-файл>` — зарегистрировать с помощью функции `DllRegisterServer` ActiveX-объекты, описанные в данном `OCX`-файле. Как правило, именно эта команда используется для установки ActiveX-файла.
- `regsvr32.exe /i[:<строка команд>] <OCX-файл>` — зарегистрировать с помощью функций `DllRegisterServer` и `DllInstall` ActiveX-объекты, описанные в данном `OCX`-файле.
- `regsvr32.exe /n /i[:<строка команд>] <OCX-файл>` — зарегистрировать с помощью функции `DllInstall` ActiveX-объекты, описанные в данном `OCX`-файле.
- `regsvr32.exe /u <OCX-файл>` — удалить с помощью функции `DllUnRegisterServer` ActiveX-объекты, описанные в данном `OCX`-файле. Как правило, именно эта команда используется для удаления ActiveX-файла.
- `regsvr32.exe /u /i[:<строка команд>] <OCX-файл>` — удалить с помощью функций `DllUnRegisterServer` и `DllUnInstall` ActiveX-объекты, описанные в данном `OCX`-файле.

Все, что делает программа `regsvr32.exe`, — вызывает функции `DllRegisterServer` и `DllInstall` (или `DllUnRegisterServer` и `DllUnInstall`), описанные в файле `OCX`. То есть альтернативным способом установки/удаления ActiveX-объектов являются следующие команды:

- `Rundll32.exe <OCX-файл> DllInstall` — зарегистрировать ActiveX-объекты, описанные в данном `OCX`-файле, с помощью функции `DllInstall`;

- Rundl132.exe <ОСХ-файл> DllRegisterServer — зарегистрировать ActiveX-объекты, определенные в данном ОСХ-файле, с помощью функции DllRegisterServer;
- Rundl132.exe <ОСХ-файл> DllInstall — удалить ActiveX-объекты, определенные в данном ОСХ-файле, с помощью функции DllUnInstall;
- Rundl132.exe <ОСХ-файл> DllUnRegisterServer — удалить ActiveX-объекты, определенные в данном ОСХ-файле, с помощью функции DllUnRegisterServer.

Структура подразделов реестра, определяющих ActiveX-объекты

Перед тем как начать описание существующих ActiveX-объектов, необходимо хотя бы поверхностно ознакомиться со структурой подразделов реестра, определяющих ActiveX-объекты, и параметрами, хранящимися в них.

Параметры подраздела ActiveX-объекта

Сведения обо всех ActiveX-объектах, зарегистрированных в операционной системе, содержатся в ветви реестра HKCR\CLSID. Каждый подраздел данной ветви определяет настройки одного ActiveX-объекта. Подраздел назван в соответствии с CLSID-номером ActiveX-объекта. Подраздел реестра HKCR\CLSID\{CLSID-номер ActiveX-объекта} может включать в себя следующие параметры.

- (По умолчанию) — определяет название ActiveX-объекта. Значение данного параметра может использоваться и в качестве названия значка, если, конечно, ActiveX-объект создает значок.
- AppID — этот параметр строкового типа определяет CLSID-номер COM-приложения, которое использует данный ActiveX-объект.
- InfoTip — параметр имеет строковый тип и определяет подсказку, отображаемую при удержании указателя мыши над значком ActiveX-объекта.
- LocalizedString — этот параметр типа REG_EXPAND_SZ определяет название значка, создаваемого ActiveX-объектом. Значение этого параметра переопределяет значение параметра (По умолчанию) ветви ActiveX-объекта.

Некоторые возможные дочерние подразделы подраздела ActiveX-объекта

В подразделе ActiveX-объекта могут также находиться дочерние подразделы.

Подраздел DefaultIcon

Параметр (По умолчанию) данного подраздела позволяет указать путь к файлу изображения, которое будет использоваться в качестве значка ActiveX-объекта (если ActiveX-объект создает значок).

Подраздел shell

Данный подраздел определяет команды контекстного меню соответствующего ActiveX-объекта. Параметр (По умолчанию) данного подраздела может содержать либо название команды, которая будет выполняться при двойном щелчке кнопкой мыши на значке ActiveX-объекта, либо перечисление последовательности всех команд через запятую. В этом случае определяется расположение команд контекстного меню.

Подраздел shell включает в себя дочерние подразделы, каждый из которых определяет название одного элемента контекстного меню. Параметр (По умолчанию) этих подразделов определяет название команды, отображаемое в контекстном меню значка ActiveX-объекта. Если же параметр (По умолчанию) подраздела будет пуст, то в контекстном меню значка ActiveX-объекта будет отображаться название подраздела.

Наконец, в подразделе названия элемента контекстного меню должен содержаться подраздел command, параметр (По умолчанию) которого содержит команду, выполняющуюся при выборе соответствующего элемента контекстного меню.

Подраздел shellEx

Кроме shell, в подразделе ActiveX-объекта может находиться еще один подраздел, содержимое которого влияет на контекстное меню значка ActiveX-объекта — подраздел shellEx. Он включает в себя дополнительные команды контекстного меню, которые при своей работе вызывают другие ActiveX-объекты.

Подраздел shellEx должен содержать другие подразделы, которые, в свою очередь, должны включать в себя подразделы с названием CLSID-номера ActiveX-объекта, вызываемого с помощью соответствующей команды контекстного меню. В зависимости от названия подраздела, являющегося дочерним по отношению к подразделу shellEx, изменяется и результат выполнения ActiveX-объекта. Далее описаны основные подразделы, дочерние к подразделу shellEx.

- IconHandler — определяет собственный обработчик значка для ActiveX-объекта. Например, так можно использовать рисунок, содержащийся в файле, как значок для этого файла.
- PropertySheetHandlers — указывает новую вкладку окна **Свойства** данного ActiveX-объекта или файла.
- ContextMenuHandlers — определяет новую команду контекстного меню значка ActiveX-объекта или файла.

Добавление вкладок к окну Свойства

Как вы уже знаете, сведения об ActiveX-объектах, добавляющих вкладки к окну **Свойства**, должны содержаться в ветви реестра HKCR\CLSID\{CLSID-номер ActiveX-объекта}\shellEx\PropertySheetHandlers (или, если нужно добавить вкладку к окну **Свойства** для файла стандартного расширения, в ветви реестра HKCR\<идентификатор файла>\shellEx\PropertySheetHandlers).

Стандартные ActiveX-объекты

Теперь рассмотрим стандартные ActiveX-объекты операционной системы Windows Vista, которые добавляют новые вкладки к окну **Свойства** файлов или значков ActiveX-объектов.

- {00020D75-0000-0000-C000-000000000046} — при открытии окна **Свойства** отображается дополнительное окно настройки почты, с помощью которого можно добавить новую конфигурацию почты либо выбрать из списка. По умолчанию данная вкладка добавлена к ActiveX-объекту {00020D75-0000-0000-C000-000000000046}.
- {2206CDB2-19C1-11D1-89E0-00C04FD7A829} — добавляет вкладки **Подключение**, **Поставщик данных**, **Дополнительно** и **Все** окна **Свойства**. По умолчанию данная вкладка добавлена к файлам с расширением UDL.
- {3FC0B520-68A9-11D0-8D77-00C04FD70822} — добавляет вкладку **Рабочий стол** окна **Свойства**.
- {42071712-76d4-11d1-8b24-00a0c9068ff3} — добавляет вкладку **Адаптер** окна **Свойства**.
- {42071713-76d4-11d1-8b24-00a0c9068ff3} — добавляет вкладку **Монитор** окна **Свойства**.
- {596AB062-B4D2-4215-9F74-E9109B0A8153} — добавляет вкладку **Предыдущие версии** окна **Свойства**. По умолчанию вкладка **Previous Version** добавляется к файлам всех типов расширений с помощью ветви реестра HKCR\AllFileSystemObjects\shell\PropertySheetHandlers.
- {5F5295E0-429F-1069-A2E2-08002B30309D} — добавляет вкладки **Общие** (отображает сведения о файловой системе), **Сервис** (содержит кнопки для выполнения архивации данных либо проверки и дефрагментации диска) и **Оборудование** (содержит список всех установленных в данный момент на компьютер жестких дисков и устройств записи данных) окна **Свойства**. По умолчанию данные вкладки добавляются ко всем значкам разделов диска.

Например, если добавить данные вкладки к папкам операционной системы, то всегда можно будет просмотреть сведения о занятом и свободном пространстве на диске, на котором расположена соответствующая папка.

- {60254CA5-953B-11CF-8C96-00AA00B8708C} — добавляет вкладку **Сценарий** окна **Свойства**. С помощью данной вкладки можно настроить параметры выполнения сценариев WSH. По умолчанию данная вкладка добавлена к файлам с расширением JSE, JS, VBE, VBS, WSF, WSH.
- {645FF040-5081-101B-9F08-00AA002F954E} — добавляет вкладку **Общие** окна **Свойства**. С помощью данной вкладки можно изменить параметры настройки работы **Корзины** в разных разделах диска операционной системы. По умолчанию данная вкладка добавлена к ActiveX-объекту {645FF040-5081-101B-9F08-00AA002F954E}.
- {6D5313C0-8C62-11D1-B2CD-006097DF8C11} — добавляет вкладки **Общие**, **Вид** и **Поиск** окна **Свойства**. Фактически это те же вкладки с теми же возможностями, что и отображаемые в окне **Свойства** папки, которое можно открыть с помощью команды **Свойства папок и файлов** из меню **Упорядочить** любой папки.

- {86422020-42A0-1069-A2E5-08002B30309D} — добавляет вкладки **Общие**, **Сервис** и **Оборудование** окна **Свойства**. По умолчанию данные вкладки добавляются ко всем значкам разделов диска.
- {F04CC277-03A2-4277-96A9-77967471BDFF} — добавляет вкладку **Общие** окна **Свойства**, с помощью которой можно просмотреть информацию о синхронизации файла или папки. По умолчанию данная вкладка добавляется с помощью ветви реестра HKCR\ConflictFolder\shell\PropertySheetHandlers.
- {f92e8c40-3d33-11d2-b1aa-080036a75b03} — добавляет вкладку **Диагностика** окна **Свойства**.
- {FA3E1D55-16DF-446d-872E-BD04D4F39C93} — добавляет вкладку **COM+** окна **Свойства**.

Пример добавления вкладок

В качестве примера будет создано собственное расширение файла и к его окну **Свойства** добавлен набор основных вкладок.

Создание нового расширения файла

Первый шаг реализации примера является наиболее творческой задачей — нужно придумать новое расширение, создание файлов которого должно быть зарегистрировано в операционной системе.

Стало традицией создавать расширения файлов, состоящие из трех символов (реже из четырех или двух), в данном же случае будет создано расширение из четырех символов (чтобы вероятность совпадения нового расширения с уже существующим была минимальной), например PARA.

Результат действий, которые нужно выполнить для создания своего расширения, представлен в листинге П4.1 в виде REG-файла.

Листинг П4.1. REG-файл двух ветвей реестра, регистрирующих новое расширение REGEDIT4

```
[HKEY_CLASSES_ROOT\.para]
@="parafile"
```

```
[HKEY_CLASSES_ROOT\parafile]
"NeverShowExt"=""
@="Test-File"
```

```
[HKEY_CLASSES_ROOT\parafile\shell]
```

```
[HKEY_CLASSES_ROOT\parafile\shell\PropertySheetHandlers]
```

Добавление вкладок к окну **Свойства**

Для примера добавим следующие стандартные вкладки к окну **Свойства** создаваемого расширения файлов:

- {645FF040-5081-101B-9F08-00AA002F954E} — позволяет изменить параметры настройки работы **Корзины**;
- {6D5313C0-8C62-11D1-B2CD-006097DF8C11} — отображает вкладки окна **Свойства** папки.

Для выполнения этого шага достаточно создать в ветви реестра `HKCR\parafile\shellex\PropertySheetHandlers` дочерние подразделы `{645FF040-5081-101B-9F08-00AA002F954E}` и `{6D5313C0-8C62-11D1-B2CD-006097DF8C11}`.

Проверка результатов

Чтобы проверить результат творчества, достаточно создать обычный текстовый файл (либо любой другой), после чего изменить расширение TXT этого файла на расширение PARA. После того как вы это сделаете и зайдете в окно **Свойства** созданного файла, перед вами отобразится окно, представленное на рис. П4.1.

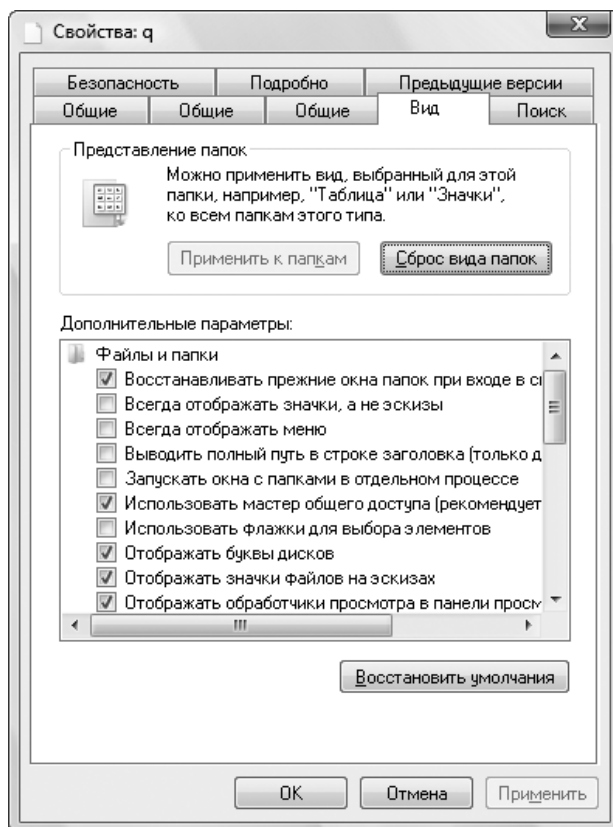


Рис. П4.1. Добавление дополнительных вкладок к окну Свойства

Добавление команд к контекстному меню

Как вы уже знаете, чтобы добавить команду к контекстному меню файлов или ActiveX-объектов, нужно добавить соответствующий команде CLSID-номер ActiveX-объекта к ветви реестра `HKCR\CLSID\{CLSID-номер ActiveX-объекта}\shellex\ContextMenuHandlers` либо к ветви реестра `HKCR\<идентификатор файла>\shellex\ContextMenuHandlers`.

Стандартные ActiveX-объекты

Ниже рассмотрены стандартные ActiveX-объекты операционной системы Windows Vista, которые добавляют новые элементы контекстного меню значка ActiveX-объекта или файла.

- {0006F019-0000-0000-C000-000000000046} — команда **С помощью Microsoft Outlook**. С помощью данной команды можно отобразить окно **Настройка Outlook**.
- {09799AFB-AD67-11d1-ABCD-00C04FC30936} — команда **Открыть с помощью**. По умолчанию данная команда добавляется ко всем зарегистрированным в операционной системе расширениям файлов с помощью ветви реестра HKCR*\shell\ex\ContextMenuHandlers.
- {1a184871-359e-4f67-aad9-5b9905d62232} — команда **Установить**. С помощью данной команды можно установить шрифт, поэтому она добавляется к контекстному меню файлов различных шрифтов.
- {2559a1f0-21d7-11d4-bdaf-00c04f60b9f0} — команда **Поиск**. Данная команда позволяет отобразить окно **Результаты поиска**.
- {2559a1f1-21d7-11d4-bdaf-00c04f60b9f0} — команда **Справка и поддержка**.
- {C2FBB630-2971-11D1-A18C-00C04FD75D13} — команда **Копировать в папку**. С помощью данной команды можно отобразить окно, а в нем указать папку, в которую будет скопирован соответствующий файл.
- {C2FBB631-2971-11d1-A18C-00C04FD75D13} — команда **Переместить в папку**. С помощью данной команды можно отобразить окно, а в нем указать папку, в которую будет перемещен соответствующий файл.
- {b8cdcb65-b1bf-4b42-9428-1dfdb7ee92af} — команда **Извлечь все**. По умолчанию она добавляется с помощью ветви системного реестра HKCR\CompressedFolder\ShellEx\ContextMenuHandlers.
- {f81e9010-6ea4-11ce-a7ff-00aa003ca9f6} — команда **Общий доступ**. По умолчанию она добавляется ко всем зарегистрированным в операционной системе расширениям файлов с помощью ветви реестра HKCR*\shell\ex\ContextMenuHandlers.
- {2559a1f3-21d7-11d4-bdaf-00c04f60b9f0} — команда **Выполнить**.
- {2559a1f4-21d7-11d4-bdaf-00c04f60b9f0} — команда **Обзор Интернета**, а также команда **Свойства Интернета**.
- {2559a1f5-21d7-11d4-bdaf-00c04f60b9f0} — команда **Чтение почты**. Она запускает почтовый клиент Outlook Express.
- {2559a1f7-21d7-11d4-bdaf-00c04f60b9f0} — команда **Настройка доступа программ и умолчаний**. Данная команда запускает одноименное окно.
- {3080F90D-D7AD-11D9-BD98-0000947B0257} — команда **Показывать поверхность рабочего стола**. После выбора данной команды все окна программ будут минимизированы и отобразится **Рабочий стол** операционной системы.
- {32714800-2E5F-11d0-8B85-00AA0044F941} — команда **Найти людей**. Данная команда отображает окно поиска в адресной книге.

- {596AB062-B4D2-4215-9F74-E9109B0A8153} — команда **Восстановить прежнюю версию**. По умолчанию команда добавляется к файлам всех типов расширений с помощью ветви реестра HKCR\AllFileSystemObjects\shell\PropertySheetHandlers.
- {645FF040-5081-101B-9F08-00AA002F954E} — команда **Очистить корзину**. Данная команда позволяет очистить содержимое **Корзины**.
- {7444C717-39BF-11D1-8CD9-00C04FC29D45} — команда **Открыть**, с помощью которой открываются файлы публичных ключей, имеющие расширение РКО.
- {7968860a-1ac6-482e-8f90-0874a1b6a79f} — команда **Изменить**.
- {7BA4C740-9E81-11CF-99D3-00AA004AE837} — команда **Отправить**. С ее помощью можно переместить файлы и папки или объекты, ссылки на которые указаны в папке %userprofile%\AppData\Roaming\Microsoft\Windows\SendTo. По умолчанию данная команда добавляется ко всем объектам файловой системы Windows Vista с помощью ветви реестра HKCR\AllFileSystemObjects\shell\ContextMenuHandlers.
- {7D4734E6-047E-41e2-AEAA-E763B4739DC4} — команда **Воспроизвести с помощью проигрывателя Windows Media**. Данная команда запускает Проигрыватель Windows Media и, если обнаруживает музыкальный файл среди содержимого папки (если команда добавлена к контекстному меню папки), пытается его воспроизвести. По умолчанию команда добавляется с помощью ветви реестра HKCR\SystemFileAssociations\Directory.Audio\shell\ContextMenuHandlers.
- {8DD448E6-C188-4aed-AF92-44956194EB1F} — команда **Добавить в список записи**. С ее помощью можно добавить файл к списку файлов, которые будут записываться на диск. По умолчанию данная команда добавляется с помощью ветви реестра HKCR\SystemFileAssociations\Directory.Audio\shell\ContextMenuHandlers.
- {A8E64375-B645-4314-9EFC-C085981786FA} — команда **Очистить список последних элементов**.
- {CE3FB1D1-02AE-4a5f-A6E9-D9F1B4073E6C} — команда **Воспроизвести с помощью проигрывателя Windows Media**. Данная команда запускает Проигрыватель Windows Media и воспроизводит соответствующий музыкальный файл. По умолчанию данная команда добавляется к файлам с расширениями AIF, ASF, ASX, AU, AVI, M3U, MID и т. д.
- {F1B9284F-E9DC-4e68-9D7E-42362A59F0FD} — команда **Добавить в список проигрывателя Windows Media**. По умолчанию она добавляется к файлам с расширениями AIF, ASF, ASX, AU, AVI, M3U, MID и т. д.
- {f3d06e7c-1e45-4a26-847e-f9fcdee59be0} — команда **Копировать как путь**. Данная команда отобразится после того, как вы нажмете клавишу Shift и, удерживая ее, вызовете контекстное меню файлов. По умолчанию команда добавляется к файлам всех типов расширений с помощью ветви реестра HKCR\AllFileSystemObjects\shell\PropertySheetHandlers.
- {FBF23B40-E3F0-101B-8488-00AA003E56F8} — команда **Открыть**, предназначенная для открытия ярлыков страниц Интернета. По умолчанию данная команда добавляется к файлам с расширением URL.

Пример добавления команды к контекстному меню

В качестве примера будет зарегистрирован собственный ActiveX-объект, после чего он будет отображен в какой-нибудь папке файловой системы компьютера.

Регистрация своего ActiveX-объекта

Первым этапом будет регистрация своего ActiveX-объекта в операционной системе Windows Vista. Для этого будет использован листинг П4.2, отображающий REG-файл тех действий, которые нужно произвести.

ПРИМЕЧАНИЕ

В контексте данной книги не объясняется процесс создания уникального ActiveX-объекта. В примере используется не уникальный CLSID-номер ActiveX-объекта, а введенный наугад.

Листинг П4.2. Файл, регистрирующий новый ActiveX-объект

Windows Registry Editor Version 5.00

```
[HKEY_CLASSES_ROOT\CLSID\{FFFFFFFF-FFFF-FFFF-FFFF-F53F46C2B1A7}]
@="TEST_ActiveX"
```

```
[HKEY_CLASSES_ROOT\CLSID\{FFFFFFFF-FFFF-FFFF-FFFF-F53F46C2B1A7}\DefaultIcon]
@="%SystemRoot%\System32\imageres.dll,-33"
```

```
[HKEY_CLASSES_ROOT\CLSID\{FFFFFFFF-FFFF-FFFF-FFFF-F53F46C2B1A7}\shell]
```

```
[HKEY_CLASSES_ROOT\CLSID\{FFFFFFFF-FFFF-FFFF-FFFF-F53F46C2B1A7}\shell\
ContextMenuHandlers]
```

Формирование команды контекстного меню

Для примера добавим следующие стандартные команды к контекстному меню ActiveX-объекта:

- {32714800-2E5F-11d0-8B85-00AA0044F941} — позволяет отобразить окно поиска среди контактов адресной книги;
- {645FF040-5081-101B-9F08-00AA002F954E} — позволяет очистить содержимое Корзины;
- {7D4734E6-047E-41e2-AEAA-E763B4739DC4} — позволяет запустить Проигрыватель Windows Media.

Для этого достаточно добавить соответствующие дочерние подразделы к ветви реестра HKCR\CLSID\{FFFFFFFF-FFFF-FFFF-FFFF-F53F46C2B1A7}\shell\ContextMenuHandlers.

Отображение зарегистрированного ActiveX-объекта

Последний этап — отображение созданного ActiveX-объекта.

В следующем подразделе будут описаны способы добавления ActiveX-объекта к одной из специальных папок. Сейчас же мы рассмотрим пример добавления ActiveX-объекта к панели быстрого запуска. Для этого достаточно создать в папке

%userprofile%\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch папку, после чего добавить к ее названию строку `{FFFFFFFF-FFFF-FFFF-FFFF-F53F46C2B1A7}`. Результат этого действия показан на рис. П4.2.

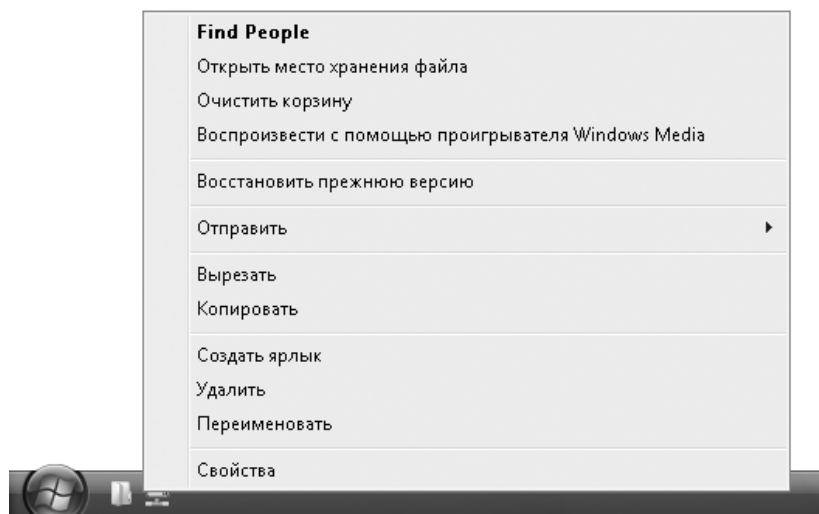


Рис. П4.2. Отображение свойств ActiveX-объекта для папки панели быстрого доступа

ActiveX-объекты, отображающие значки

Еще одним примером использования ActiveX-объектов является отображение значков или мастеров и окон операционной системы Windows Vista. Например, вы можете добавить описанные ниже значки ActiveX-объектов на Рабочий стол, в папку Компьютер или на Панель управления. Для этого нужно в специальной ветви реестра создать подраздел, названный в соответствии с CLSID-номером ActiveX-объекта. Вы можете воспользоваться следующими ветвями реестра:

- `HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Desktop\namespace` (или в ветви корневого раздела HKLM) — добавляет ActiveX-объект на Рабочий стол;
- `HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\mycomputer\namespace` (или в ветви корневого раздела HKLM) — добавляет ActiveX-объект в папку Компьютер;
- `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ControlPanel\NameSpace` — добавляет ActiveX-объект в папку Панель управления.

Еще одним способом создания значка ActiveX-объекта является использование папки. Для этого достаточно создать папку и к ее названию добавить `{CLSID-номер соответствующего ActiveX-объекта}`.

- `{00f2886f-cd64-4fc9-8ec5-30ef6cdbc8c3}` — значок Сканеры и камеры. Этот же значок можно вызвать с помощью окна Выполнить, введя в нем команду `::{21EC2020-3AEA-1069-A2DD-08002B30309D}\::{00f2886f-cd64-4fc9-8ec5-30ef6cdbc8c3}`.

- {025A5937-A6BE-4686-A844-36FE4BEC8B6D} — значок **Электропитание**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду ::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {025A5937-A6BE-4686-A844-36FE4BEC8B6D}.
- {0DF44EAA-FF21-4412-828E-260A8728E7F1} — значок **Панель задач и меню "Пуск"**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду ::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {0DF44EAA-FF21-4412-828E-260A8728E7F1}.
- {1443904b-34e4-40f6-b30f-6beb81267b80} — запускает модуль Speech Recognition при запуске папки.
- {15eae92e-f17a-4431-9f28-805e482dafd4} — значок, вызывающий мастер **Получение программ**. Этот мастер позволяет установить программы, которые были опубликованы по сети администратором домена Active Directory. Данный ActiveX-объект также можно использовать в виде команды ::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {15eae92e-f17a-4431-9f28-805e482dafd4}.
Можно также создать папку со свойствами данного объекта (она должна называться "название". {15eae92e-f17a-4431-9f28-805e482dafd4}).
- {17cd9488-1228-4b2f-88ce-4298e93e0966} — значок **Программы по умолчанию**. Этот ActiveX-объект можно использовать и в виде команды ::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {17cd9488-1228-4b2f-88ce-4298e93e0966} окна **Выполнить**.
Свойства данного окна можно также присвоить папке.
- {1FA9085F-25A2-489B-85D4-86326EEDCD87} — значок **Управление беспроводными сетями**. Этот же значок можно вызвать, присвоив свойства ActiveX-объекта папке.
- {208D2C60-3AEA-1069-A2D7-08002B30309D} — значок **Сеть (WORKGROUP)**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду ::{208D2C60-3AEA-1069-A2D7-08002B30309D}.
Это окно также можно вызвать, присвоив его свойства папке.
- {20D04FE0-3AEA-1069-A2D8-08002B30309D} — значок **Компьютер**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду ::{20D04FE0-3AEA-1069-A2D8-08002B30309D}.
Это окно также можно вызвать, присвоив его свойства папке.
- {21EC2020-3AEA-1069-A2DD-08002B30309D} — значок **Панель управления**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду ::{21EC2020-3AEA-1069-A2DD-08002B30309D}.
- {2227A280-3AEA-1069-A2DE-08002B30309D} — значок **Принтеры**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду ::{2227A280-3AEA-1069-A2DE-08002B30309D}.
Это окно также можно вызвать с помощью окна **Выполнить**, введя в нем команду ::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {2227A280-3AEA-1069-A2DE-08002B30309D}.
- Те же самые действия выполняются, если присвоить свойства папки **Принтеры** обычной папке.
- {2559a1f0-21d7-11d4-bdaf-00c04f60b9f0} — значок **Поиск**.

- {2559a1f1-21d7-11d4-bdaf-00c04f60b9f0} — значок Справка и поддержка.
- {2559a1f3-21d7-11d4-bdaf-00c04f60b9f0} — значок Выполнить.
- {2559a1f7-21d7-11d4-bdaf-00c04f60b9f0} — значок Настройка доступа программ и умолчаний.
- {26EE0668-A00A-44D7-9371-BEB064C98683} — значок Панель управления, открывающий Панель управления с отсортированными по категориям компонентами. Этот же значок можно вызвать с помощью окна Выполнить, введя в нем команду : : {26EE0668-A00A-44D7-9371-BEB064C98683}.
- {289978AC-A101-4341-A817-21EBA7FD046D} — значок Просмотр конфликтов синхронизации. Этот же значок можно отобразить, присвоив его свойства папке.
- {2E9E59C0-B437-4981-A647-9C34B9B90891} — значок Настройка новой связи синхронизации. Этот же значок можно отобразить, присвоив его свойства папке.
- {3080F90D-D7AD-11D9-BD98-0000947B0257} — значок Свернуть все окна.
- {335a31dd-f04b-4d76-a925-d6b47cf360df} — значок Центр архивации и восстановления. Этот же значок можно вызвать с помощью окна Выполнить, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {335a31dd-f04b-4d76-a925-d6b47cf360df}.
- Это окно также можно отобразить, присвоив его свойства папке.
- {36eef7db-88ad-4e81-ad49-0e313f0c35f8} — значок Центр обновления Windows. Этот же значок можно вызвать с помощью окна Выполнить, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {36eef7db-88ad-4e81-ad49-0e313f0c35f8}.
- Это окно также можно вызвать, присвоив его свойства папке.
- {37efd44d-ef8d-41b1-940d-96973a50e9e0} — значок Свойства боковой панели Windows. Этот же значок можно вызвать с помощью окна Выполнить, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {37efd44d-ef8d-41b1-940d-96973a50e9e0}.
- {3e7efb4c-faf1-453d-89eb-56026875ef90} — значок Приобретение программ через Интернет. Этот же значок можно вызвать с помощью окна Выполнить, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {3e7efb4c-faf1-453d-89eb-56026875ef90}.
- {3f6bc534-dfa1-4ab4-ae54-ef25a74e0107} — значок Восстановление системы.
- {4026492F-2F69-46B8-B9BF-5654FC07E423} — значок Брандмауэр Windows. Этот же значок можно вызвать с помощью окна Выполнить, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {4026492F-2F69-46B8-B9BF-5654FC07E423}.
- Это окно также можно вызвать, присвоив его свойства папке.
- {437ff9c0-a07f-4fa0-af80-84b6c6440a16} — значок Command Folder.
- {450D8FBA-AD25-11D0-98A8-0800361B1103} — значок Документы. Этот же значок можно вызвать с помощью окна Выполнить, введя в нем команду : : {450D8FBA-AD25-11D0-98A8-0800361B1103}.
- {48e7caab-b918-4e58-a94d-505519c795dc} — значок Главное меню. Этот же значок можно отобразить, присвоив его свойства папке.

- {4D1209BD-36E2-4e2f-840D-6C7FB879DD9E} — значок **Windows Ultimate**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду `::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {4D1209BD-36E2-4e2f-840D-6C7FB879DD9E}`.
Это окно также можно вызвать, присвоив его свойства папке.
- {5399E694-6CE5-4D6C-8FCE-1D8870FDCBA0} — значок **Панель управления**.
- {58E3C745-D971-4081-9034-86E34B30836A} — значок **Speech Recognition**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду `::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {58E3C745-D971-4081-9034-86E34B30836A}`.
Его также можно отобразить, присвоив свойства данного ActiveX-объекта папке.
- {59031a47-3f72-44a7-89c5-5595fe6b30ee} — значок доступа к папке **%userprofile%**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду `::{59031a47-3f72-44a7-89c5-5595fe6b30ee}`.
Его также можно отобразить, присвоив свойства данного ActiveX-объекта папке.
- {5ea4f148-308c-46d7-98a9-49041b1dd468} — значок **Центр устройств Windows Mobile**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду `::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {5ea4f148-308c-46d7-98a9-49041b1dd468}`.
Это окно также можно отобразить, присвоив свойства ActiveX-объекта папке.
- {60632754-c523-4b62-b45c-4172da012619} — значок **Учетные записи пользователей**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду `::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {60632754-c523-4b62-b45c-4172da012619}`.
Это окно также можно вызвать, присвоив его свойства папке.
- {645FF040-5081-101B-9F08-00AA002F954E} — значок **Корзина**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду `::{645FF040-5081-101B-9F08-00AA002F954E}`.
Это окно также можно вызвать, присвоив свойства ActiveX-объекта папке.
- {67718415-c450-4f3c-bf8a-b487642dc39b} — значок **Компоненты Windows**. Этот же значок можно вызвать, присвоив свойства ActiveX-объекта папке.
- {692F0339-CBAA-47e6-B5B5-3B84DB604E87} — присвоение свойств данного ActiveX-объекта папке приводит к тому, что эта папка будет ссылаться на другую папку, содержащую набор XML-файлов. Данная папка содержит все установленные надстройки браузера Internet Explorer.
- {6DFD7C5C-2451-11d3-A299-00C04F8EF6AF} — значок **Свойства папки**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду `::{21EC2020-3AEA-1069-A2DD-08002B30309D}\:: {6DFD7C5C-2451-11d3-A299-00C04F8EF6AF}`.
Это окно также можно вызвать, присвоив его свойства папке.
- {7007ACC7-3202-11D1-AAD2-00805FC1270E} — значок **Сетевые подключения**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду `::{7007ACC7-3202-11D1-AAD2-00805FC1270E}`.

Это окно также можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {7007ACC7-3202-11D1-AAD2-00805FC1270E}.

- {71D99464-3B6B-475C-B241-E15883207529} — значок **Результаты выполнения синхронизации**. Этот же значок можно вызвать, присвоив его свойства папке.
- {74246bfc-4c96-11d0-abef-0020af6b0b7a} — значок **Диспетчер устройств**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {74246bfc-4c96-11d0-abef-0020af6b0b7a}.
- {78F3955E-3B90-4184-BD14-5397C15F1EFC} — значок **Счетчики и средства производительности**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {78F3955E-3B90-4184-BD14-5397C15F1EFC}.

Это окно также можно отобразить, присвоив его свойства папке.

- {7A979262-40CE-46ff-AEEE-7884AC3B6136} — значок **Установка оборудования**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {7A979262-40CE-46ff-AEEE-7884AC3B6136}.

Это окно также можно отобразить, присвоив его свойства папке.

- {7A9D77BD-5403-11d2-8785-2E0420524153} — значок **Учетные записи пользователей**.
- {7b81be6a-ce2b-4676-a29e-eb907a5126c5} — значок **Программы и компоненты**. Его же можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {7b81be6a-ce2b-4676-a29e-eb907a5126c5}.

Это окно также можно отобразить, присвоив его свойства папке.

- {7be9d83c-a729-4d97-b5a7-1b7313c39e0a} — значок, который комбинирует в себе содержимое сразу двух папок — подпапки, хранящейся в папке %userprofile%\Главное меню, и подпапки, находящейся в папке %systemdrive%\Users\All Users\Главное меню. Данный значок можно отобразить, только присвоив его свойства папке.
- {85BBD920-42A0-1069-A2E4-08002B30309D} — значок **Портфель**. Этот значок можно отобразить, только присвоив его свойства папке.
- {865e5e76-ad83-4dca-a109-50dc2113ce9a} — если присвоить свойства данного ActiveX-объекта папке, то она будет отображать папку, содержащую все программы из меню **Пуск**.
- {8E908FC9-BECC-40f6-915B-F4CA0E70D03D} — значок **Центр управления сетями и общим доступом**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {8E908FC9-BECC-40f6-915B-F4CA0E70D03D}.

Это окно также можно отобразить, присвоив его свойства папке.

- {9343812e-1c37-4a49-a12e-4b2d810d956b} — значок **Результаты поиска**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {9343812e-1c37-4a49-a12e-4b2d810d956b}.

Данный значок можно отобразить, только присвоив его свойства папке.

- {96AE8D84-A250-4520-95A5-A47A7E3C548B} — значок **Родительский контроль**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {96AE8D84-A250-4520-95A5-A47A7E3C548B}.
Это окно также можно отобразить, присвоив его свойства папке.
- {992CFFA0-F557-101A-88EC-00DD010CCC48} — значок **Сетевые подключения**.
- {9C60DE1E-E5FC-40f4-A487-460851A8D915} — значок **Автозапуск**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {9C60DE1E-E5FC-40f4-A487-460851A8D915}.
Это окно также можно отобразить, присвоив его свойства папке.
- {9C73F5E5-7AE7-4E32-A8E8-8D23B85255BF} — значок **Центр синхронизации**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {9C73F5E5-7AE7-4E32-A8E8-8D23B85255BF}.
Это окно также можно отобразить, присвоив его свойства папке.
- {9f433b7c-5f96-4ce1-ac28-aea1cc04d7c} — значок **Центр обеспечения безопасности**. Этот же значок можно отобразить, присвоив его свойства папке.
- {A304259D-52B8-4526-8B1A-A1D6CECC8243} — значок **Инициатор iSCSI**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {A304259D-52B8-4526-8B1A-A1D6CECC8243}.
- {AFDB1F70-2A4C-11d2-9039-00C04F8EEB3E} — значок **Автономные файлы**.
- {b155bdf8-02f0-451e-9a26-ae317cfd7779} — значок **delegate folder that appears in Computer**.
- {B2C761C6-29BC-4f19-9251-E6195265BAF1} — значок **Управление цветом**, вызывающий одноименный мастер. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {B2C761C6-29BC-4f19-9251-E6195265BAF1}.
- {BB06C0E4-D293-4f75-8A90-CB05B6477EEE} — значок **Система**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {BB06C0E4-D293-4f75-8A90-CB05B6477EEE}.
- {CB1B7F8C-C50A-4176-B604-9E24DEE8D4D1} — значок **Центр начальной настройки**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {CB1B7F8C-C50A-4176-B604-9E24DEE8D4D1}.
- {D20EA4E1-3957-11d2-A40B-0C5020524152} — значок **Шрифты**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {D20EA4E1-3957-11d2-A40B-0C5020524152}.
- {D20EA4E1-3957-11d2-A40B-0C5020524153} — значок **Администрирование**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду :: {21EC2020-3AEA-1069-A2DD-08002B30309D}\ : : {D20EA4E1-3957-11d2-A40B-0C5020524153}.
- {d34a6ca6-62c2-4c34-8a7c-14709c1ad938} — значок **Common Places FS Folder**. Этот же значок можно отобразить, присвоив его свойства папке.
- {d450a8a1-9568-45c7-9c0e-b4f9fb4537bd} — значок **Установленные обновления**. Этот же значок можно отобразить, присвоив его свойства папке.

- {D555645E-D4F8-4c29-A827-D93C859C4F2A} — значок **Центр специальных возможностей**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {D555645E-D4F8-4c29-A827-D93C859C4F2A}.

Это окно также можно отобразить, присвоив его свойства папке.

- {D8559EB9-20C0-410E-BEDA-7ED416AEC2A} — значок **Защитник Windows**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {D8559EB9-20C0-410E-BEDA-7ED416AEC2A}.

Его также можно отобразить, присвоив свойства папке.

- {D9EF8727-CAC2-4e60-809E-86F80A666C91} — значок **Шифрование диска BitLocker**, вызывающий одноименный мастер. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {D9EF8727-CAC2-4e60-809E-86F80A666C91}.

Его также можно отобразить, присвоив свойства папке.

- {E7E4BC40-E76A-11CE-A9BB-00AA004AE837} — перезагружает оболочку.
- {E95A4861-D57A-4be1-AD0F-35267E261739} — значок **Windows SideShow**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {E95A4861-D57A-4be1-AD0F-35267E261739}.

Это окно также можно отобразить, присвоив его свойства папке.

- {ED228FDF-9EA8-4870-83b1-96b02CFE0D52} — значок **Игры**. Этот значок также можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {ED228FDF-9EA8-4870-83b1-96b02CFE0D52}.

Это окно также можно отобразить, присвоив его свойства папке.

- {ED834ED6-4B5A-4bfe-8F11-A626DCB6A921} — значок **Персонализация**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {ED834ED6-4B5A-4bfe-8F11-A626DCB6A921}.

Его также можно отобразить, присвоив его свойства папке.

- {F02C1A0D-BE21-4350-88B0-7367FC96EF3C} — значок **Сеть**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {F02C1A0D-BE21-4350-88B0-7367FC96EF3C}.

- {FCFECAE-EE1B-4849-AE50-685DCF7717EC} — значок **Отчеты о проблемах и их решениях**. Этот же значок можно вызвать с помощью окна **Выполнить**, введя в нем команду : : {21EC2020-3AEA-1069-A2DD-08002B30309D} \ : : {FCFECAE-EE1B-4849-AE50-685DCF7717EC}.

Это окно также можно вызвать, присвоив его свойства папке.

На этом закончено рассмотрение ActiveX-объектов операционной системы Windows Vista.

Приложение 5

Дополнительные материалы

На протяжении всей книги вы не раз должны были встречать ссылки на дополнительные материалы, которые можно скачать с сайта издательства «Питер». Не встречали? Тогда вам следует заново перечитать книгу. На этот раз внимательнее.

Если же ссылки на дополнительные материалы вы все-таки заметили, тогда сейчас вкратце рассмотрим, что же вы можете скачать с сайта издательства «Питер».

Скачав архив с дополнительными материалами, вы сможете обнаружить в нем следующие папки: **Дополнительные главы**, **Сценарии**, **Свойства и методы WMI** и **Site**.

Дополнительные главы

Второе издание данной книги было дополнено многими новыми методами работы с операционной системой Windows Vista. При этом объем полученной книги существенно не изменился. Решить такое противоречие удалось лишь путем удаления отдельных частей книги, которые, по представлению автора, могут быть интересны только узкому кругу читателей. Чтобы не обижать узкий круг читателей, удаленные главы были размещены среди дополнительных материалов к книге, а именно, в папке **Дополнительные главы**.

В этой папке вы сможете найти следующие файлы в формате DOC.

- **Глава 6. Стандартные службы.doc** — как видно из названия файла, в нем находится шестая глава первого издания книги. Она содержит в себе описание всех служб операционной системы Windows Vista, тогда как шестая глава текущего издания книги включает в себя описание лишь большинства служб Windows Vista. Кроме того, шестая глава из данного файла содержит подробное описание упоминаемых групповых политик, тогда как шестая глава текущего издания книги представляет лишь названия групповых политик и параметры реестра, которые они изменяют.
- **Групповые политики.doc** — в этом документе можно прочитать подробное описание групповых политик. В первом издании книги были подробно рассмотрены все групповые политики операционной системы Windows Vista. В этом же издании, как вы уже знаете, для большинства политик приводится только их название, а также параметр реестра, который они изменяют.

Сценарии

Второе издание данной книги отличается от первого не только наличием нового материала. Теперь в дополнение к книге можно скачать огромное количество сценариев, позволяющих быстро настроить операционную систему Windows Vista, даже не прибегая в программе Редактор реестра.

Свойства и методы WMI

Первое издание книги содержало в себе описание большинства новых свойств и методов операционной системы Windows Vista. Во втором издании описаний новых свойств и методов WMI практически нет. Не стоит огорчаться, если вас не устраивают такие изменения. Описание всех новых свойств и методов WMI, которые упоминались в первом издании книги, теперь можно найти среди дополнительных материалов к книге. Они хранятся в виде отдельных файлов формата DOC в папке Свойства и методы WMI.

Site

Еще одним изменением второго издания книги является полностью работоспособная версия сайта <http://www.onestyle.com.ua>, которая поставляется вместе с веб-сервером «Денвер». На этом сайте вы сможете найти как материал данной книги, так и множество дополнительных материалов, относящихся к раскрываемой теме. Кроме того, с помощью этого сайта вы сможете автоматизировать или облегчить многие задачи.

Версия сайта <http://www.onestyle.com.ua> находится в папке **site** дополнительных материалов к книге. Чтобы начать работу с сайтом, необходимо скопировать папку **site** на локальный компьютер (если вы этого еще не сделали) и затем запустить веб-сервер «Денвер».

Запуск сервера

Еще раз обратите внимание, что перед запуском веб-сервера папку **site** необходимо скопировать на локальный компьютер или в любое другое хранилище, доступ к которому разрешен как в режиме записи, так и чтения. Дело в том, что в процессе работы веб-сервера будет создано множество временных файлов.

После того как вы скопируете папку **site**, перейдите к следующей папке: `site\index\etc` (здесь и далее в пути к папке **site** не указывается буква диска и начальный путь, так как начальный путь к папке может различаться и зависит только от вас). В данной папке вы должны найти три файла: `Restart.exe`, `Run.exe` и `Stop.exe` (рис. П5.1).

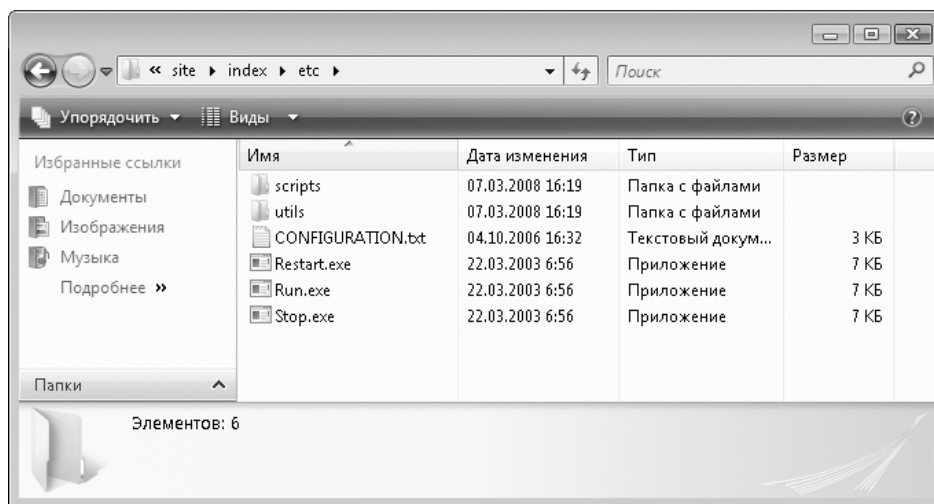


Рис. П5.1. Файлы для управления веб-сервером

Файл `Run.exe` используется для запуска веб-сервера, файл `Stop.exe` — для остановки веб-сервера, а файл `Restart.exe` — для его остановки и повторного запуска.

Если вы планируете использовать данный веб-сервер довольно часто, тогда рекомендуется создать ярлыки файлов `Run.exe` и `Stop.exe`, а пока просто запустите веб-сервер. Для этого запустите файл `Run.exe`. Это приведет к отображению окна командной строки с описанием выполняемых действий. Не пугайтесь, если увидите в этом окне иероглифы — так и должно быть.

Если ваша учетная запись обладает административными привилегиями, тогда запуск веб-сервера должен пройти успешно.

Если же вы не обладаете административными привилегиями, тогда запуск веб-сервера завершится неудачно. Дело в том, что веб-сервер добавляет новые строки в файл `hosts`, расположенный в папке `%systemroot%\system32\drivers\etc`, а этот файл могут изменять только администраторы, поэтому без административных прав веб-сервер не сможет зарегистрировать зеркало сайта, издаст звуковой сигнал и попросит для продолжения нажать клавишу `Enter`. В этом случае веб-сервер также будет запущен, однако вы не сможете получить доступ к зеркалу сайта.

Поэтому перед запуском веб-сервера убедитесь, что вы обладаете административными правами. Если же административные права получить нельзя, тогда переименуйте папку `onestyle.loc` в `localhost`. Данная папка находится по адресу `site\index\home`. После этого запустите сервер и, после того как он издаст звуковой сигнал, нажмите клавишу `Enter`. Теперь вы все-таки сможете запустить зеркало сайта.

В результате совершения данных действий в области уведомлений будет отображен значок веб-сервера Apache (рис. П5.2). Если после запуска сервера этот значок появился, значит, все отлично.

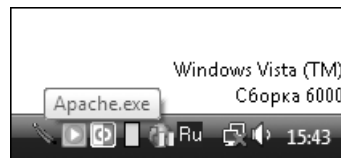


Рис. П5.2. Значок запущенного веб-сервера Apache

Запуск зеркала сайта

После того как вы запустили комплект для разработчиков «Денвер» и в области уведомлений появился значок веб-сервера Apache, пора попробовать открыть зеркало сайта. Для этого запустите какой-либо браузер и введите адрес onestyle.loc (если вы переименовали папку onestyle.loc в localhost, тогда введите адрес localhost). В ответ на эти действия должна открыться главная страница зеркала сайта. Собственно, главная страница сайта нас пока не интересует — мы с вами рассмотрим только возможность работы с разделом сайта **Онлайн**. Чтобы отобразить данный раздел, перейдите по одноименной ссылке в верхней части страницы либо же просто воспользуйтесь адресом onestyle.loc/online (рис. П5.3).

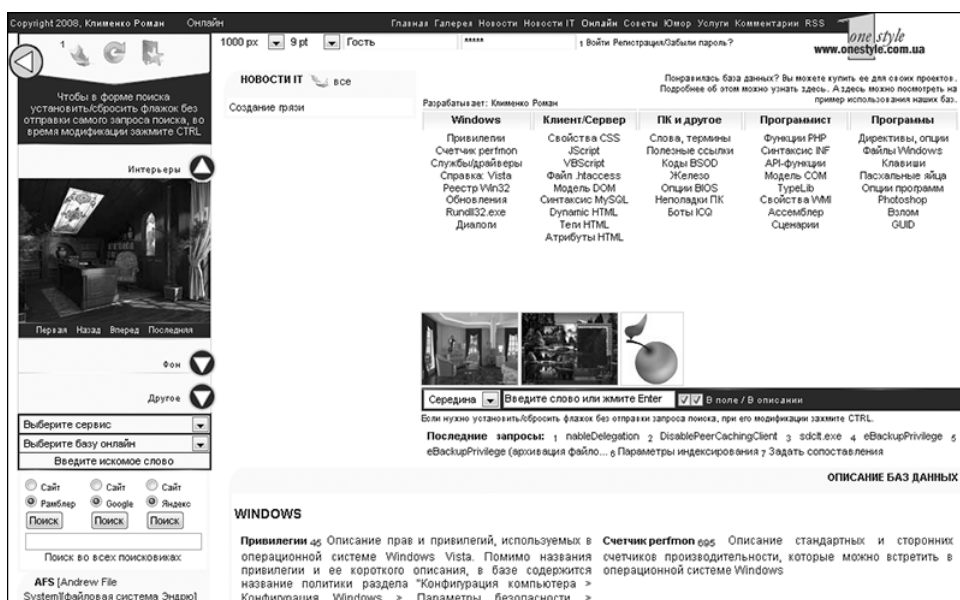


Рис. П5.3. Раздел Онлайн локального сайта onestyle.loc

Раздел **Онлайн** предоставляет вам возможность выполнить одновременный поиск по множеству баз данных, которые поставляются вместе с сайтом, либо же перейти к конкретной базе данных.

Все базы данных сайта разбиты по группам: **Windows**, **Клиент/Сервер**, **ПК и другое**, **Программист** и **Программы**. Локальная версия сайта содержит в себе не все базы данных, а лишь те, содержание которых касается темы данной книги. Ниже будет рассмотрен полный список баз данных, доступных на сайте, но перед этим попробуйте поработать с любой базой данных.

Работа с базами данных

Например, обучение работе с базами данных можно провести на основе базы **Реестр Win32** группы **Windows**. Как видно из названия, в этой базе содержится описание параметров и ветвей реестра.

Чтобы начать работу с базой данных, необходимо перейти к ее странице, то есть щелкнуть кнопкой мыши на ссылке **Реестр Win32** группы **Windows**. После этого перед вами отобразится страница базы данных, на которой присутствует форма поиска, а также отображаются семь последних добавленных в базу записей (рис. П5.4).

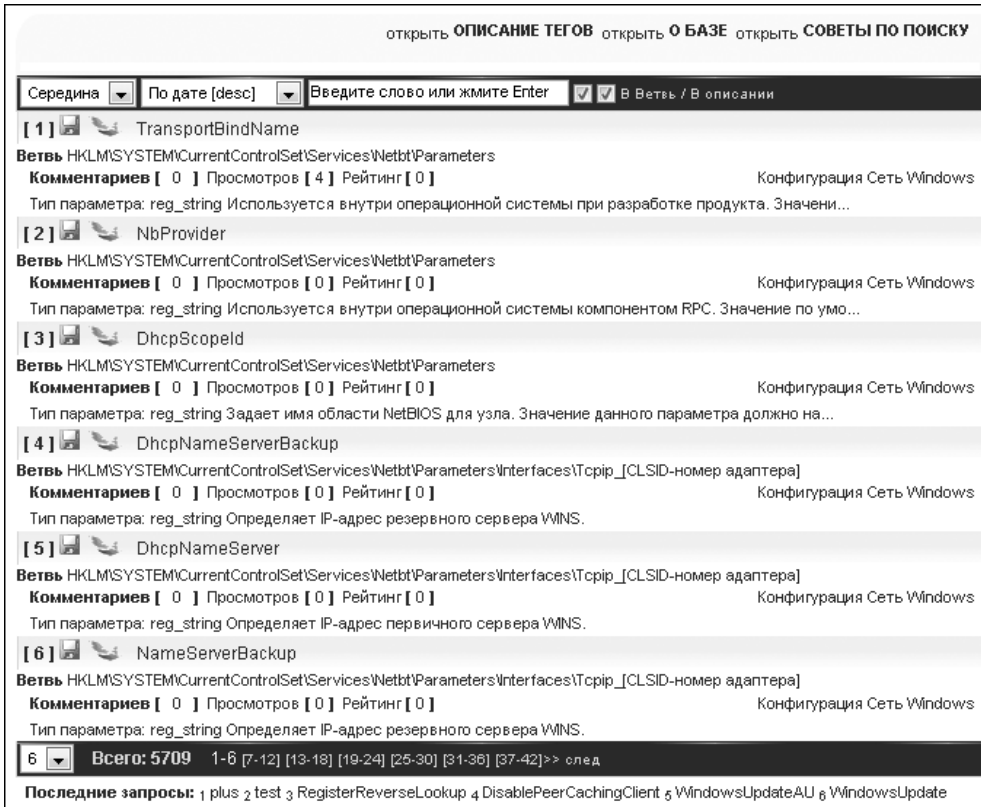


Рис. П5.4. Форма поиска в базе данных, а также результат поиска

Чтобы выполнить поиск по базе, достаточно ввести искомую фразу в форму поиска (в поле, содержащее фразу **Введите слово или жмите Enter**), после чего нажать клавишу **Enter**. Поиск можно выполнять не только по части названия параметра или ветви реестра, но и по другим полям. Чтобы эффективнее выполнять поиск, прочитайте советы по поиску. Для этого воспользуйтесь ссылкой **Открыть** слева от фразы **Советы по поиску** (выше формы поиска).

После того как вы введете искомую фразу и нажмете **Enter**, перед вами отобразится результат поиска. Если поиск окажется удачным и вы найдете то, что искали, щелкните на названии найденного параметра — вы перейдете на страницу с описанием параметра. Помимо самого описания параметра, с помощью данной страницы вы можете создать различные сценарии с участием параметра (рис. П5.5).

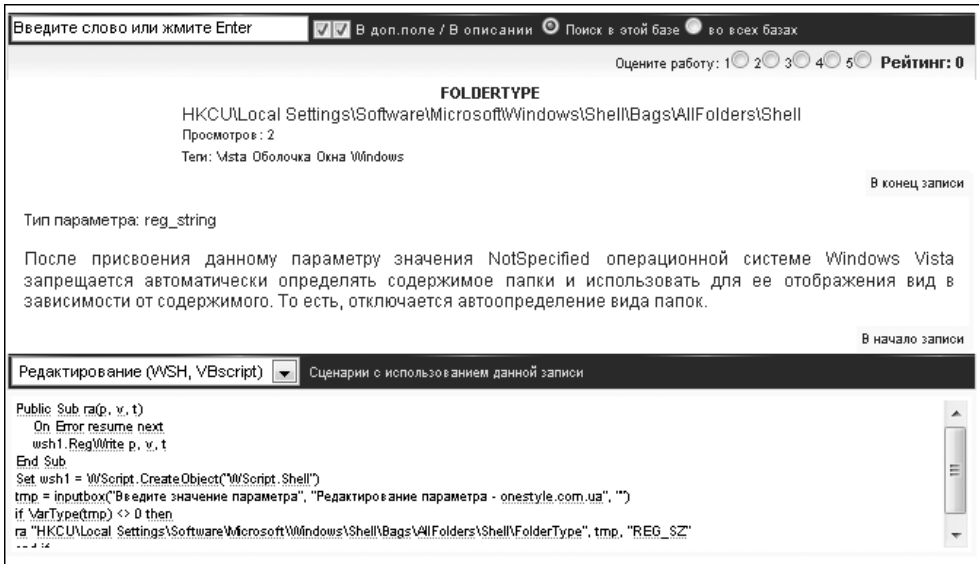


Рис. П5.5. Страница с описанием конкретной записи базы данных

Доступные базы данных

Мы с вами рассмотрели основы работы с базами данных, доступными на сайте. Уверен, что в остальном вы и сами сумеете разобраться, поэтому перейдем к рассмотрению содержимого тех баз данных, которые поставляются вместе с сайтом.

Группа Windows

В данной группе содержатся базы данных, содержимое которых полностью посвящено операционной системе Windows.

- **Привилегии** — содержит описание всех прав и привилегий операционной системы Windows Vista.
- **Счетчик perfmon** — содержит описание стандартных и сторонних счетчиков производительности, которые можно встретить в операционной системе Windows.
- **Службы/драйверы** — содержит описание служб и стандартных драйверов, установленных в операционной системе Windows Vista. Для каждой записи данной базы существует набор сценариев WSH, которые позволяют выполнить следующие действия: отключить и включить службу, запустить и приостановить работу службы, а также отобразить текущие параметры работы службы.
- **Справка: Vista** — содержит описание разделов справки операционной системы Windows Vista, а также сведения о командах rundll32.exe, с помощью которых эти разделы можно отобразить.
- **Реестр Win32** — содержит описание параметров и ветвей реестра операционной системы Windows. Для каждой записи данной базы существует набор сценариев, которые позволяют выполнить следующие действия:

- изменить значение параметра (VBS-файл);
 - прочитать текущее значение параметра (VBS-файл);
 - создать или удалить параметр или ветвь реестра (VBS, INF или REG-файл);
 - просмотреть различные команды программы reg.exe для работы с данной записью.
- **Обновления** — содержит перечень последних (на момент написания книги) обновлений операционных систем семейства Windows, начиная с Windows 2000. Для каждой записи данной базы приводится набор ссылок, с помощью которых вы можете как прочитать подробное описание обновления на сайте Microsoft или в базе знаний Microsoft, так и скачать обновление.
 - **Rundll32.exe** — содержит описание команд rundll32.exe, доступных в операционной системе Windows Vista. Для каждой записи данной базы существует набор сценариев, позволяющих выполнить команду (INF- и VBS-файлы) либо же создать ярлык на команду (VBS-файл).
 - **Диалоги** — содержит краткое описание стандартных окон Windows Vista и способы доступа к ним.

Группа ПК и другое

Содержимое баз данных этой группы самое разнообразное: в ней собраны все базы данных, которые нельзя было отнести к другим группам.

- **Слова, термины** — содержит описание около 3000 терминов, посвященных компьютерным технологиям и дизайну.
- **Полезные ссылки** — содержит ссылки на интересные сайты или отдельные страницы с интересным содержанием. Кроме того, в данной базе собран набор внутренних адресов популярных браузеров: Internet Explorer, Opera и Mozilla. Каждую ссылку можно добавить в **Избранное** вашего браузера.

Группа Программист

В данной группе вам доступны лишь две базы данных — **Свойства WMI** и **Сценарии**. Содержимое первой базы данных понятно из ее названия. Вторая же база данных содержит в себе различные сценарии, которые мне приходилось создавать либо для себя, либо для различных книг. Помимо самого кода сценария, также приводится его описание. Если какой-то сценарий вам понравился, тогда просто скопируйте его содержимое в отдельный файл и присвойте полученному файлу нужное расширение (в соответствии с тегами, которые ассоциированы со сценарием).

Группа Программы

В данную группу попали базы данных, содержимое которых может относиться к различным программам (как к стандартным программам Windows, так и к сторонним программам).

- **Файлы Windows** — содержит краткое описание стандартных файлов операционной системы и различных сторонних программ.

-
- **Клавиши** — содержит описание сочетаний клавиш операционной системы Windows Vista, многих ее стандартных программ, а также некоторых сторонних программ.
 - **Опции программ** — содержит описание документированных и undocumented параметров стандартных программ операционной системы Windows Vista, а также некоторых сторонних программ.
 - **GUID** — содержит описание стандартных ActiveX-объектов операционной системы Windows Vista. В зависимости от тегов, присвоенных ActiveX-объекту, с записью ассоциированы различные сценарии.