Уильям Р. Стэнек

Командная строка Microsoft Windows

Справочник администратора

Подробный справочник по управлению Microsoft Windows XP и Microsoft Windows Server 2003 из командной строки

Примеры, помогающие использовать преимущества командной строки

Профессионал

Русская Редакция

Microsoft

William R. Stanek

Microsoft®

Windows® Command-Line

Administrator's Pocket Consultant

Microsoft Press

Уильям Р. Станек

Командная строка Microsoft® Windows®

Справочник администратора

Москва 2004

РУССКАЯ РЕДАКЦИЯ

УДК 004
ББК 32.973.81-018.2
С76

Уильям Р. Станек

С76 Командная строка Microsoft Windows. Справочник администратора.: Пер. с англ. — М.: Издательско-торговый дом «Русская Редакция», 2004. — 480 с.: ил.

ISBN 5-7502-0267-4

Данная книга — краткий, но исчерпывающий справочник, посвященный командной оболочке и базовым средствам командной строки двух операционных систем — Microsoft Windows XP Professional и Microsoft Windows Server 2003. Здесь рассматриваются все основные вопросы, связанные с выполнением стандартных задач администрирования из командной строки, в том числе настройка Windows-служб и управление локальными и удаленными системами, автоматизация мониторинга различных системных параметров, анализ и мониторинг процессов, управление дисками и файловыми системами, создание базовых и динамических дисков, а также RAID-массивов, конфигурирование службы каталогов Active Directory, администрирование TCP/IP-сетей и многие другие насущные вопросы.

Книга адресована системным администраторам и специалистам по технической поддержке Microsoft Windows XP Professional и Microsoft Windows Server 2003, а также пользователям, желающим детально изучить командную оболочку и инструменты командной строки Windows.

Издание состоит из 15 глав, приложения и предметного указателя.

УДК 004
ББК 32.973.81-018.2

ISBN 0-7356-2038-5 (англ)
ISBN 5-7502-0267-4

Оглавление

Благодарности XII
Введение XIV
Часть I Основы командной строки Windows 1
Глава 1 Обзор командной строки Windows 2
Средства работы с командной строкой 3
Введение в командную оболочку Windows 3
Введение в командную оболочку MS-DOS 8
Настройка свойств командной оболочки 9
Хронология команд 11
Windows Support Tools 12
Windows Server 2003 Resource Kit Tools 15
Глава 2 Эффективная работа с командной строкой 17
Управление запуском командной оболочки 18
Путь к командам 20
Использование пути к командам 20
Расширения файлов и файловые сопоставления 22
Перенаправление стандартных ввода, вывода и ошибок 24
Перенаправление стандартного вывода другим командам 24
Перенаправление ввода-вывода в файлы 25
Перенаправление стандартных ошибок 26
Создание цепочек и группирование команд 26
Цепочки команд 27
Группирование последовательностей команд 28
Глава 3 Основы сценариев командной строки 30
Создание сценариев командной строки 31
Стандартные операторы и команды сценариев 32
Очистка окна командной оболочки 33
Добавление комментариев в сценарии 34
Управление отображением текста и команд 35
Управление эхо-отображением с помощью @ 36
Настройка заголовка и цветов консольного окна 37
Передача аргументов в сценарии 38
Знакомство с переменными 40
Использование переменных в сценариях 41
Именование переменных 42
Присвоение значений переменным 43
Подстановка значений переменных 44
Локализация области действия переменных 46
Применение математических выражений 47
Арифметические операторы и операторы присваивания 48

Приоритет операторов ... 49
Имитация возведения в степень ... 50
Операторы выбора в командной строке ... 50
Применение If ... 51
Применение If Not ... 52
Применение If Defined и If Not Defined ... 52
Вложенные условия ... 53
Сравнения в операторах If ... 53
Операторы циклов в командной строке ... 54
Базовые сведения о циклах ... 54
Перебор наборов значений ... 56
Перебор групп файлов ... 57
Перебор каталогов ... 58
Анализ содержимого файлов и вывода ... 60
Создание подпрограмм и процедур ... 63
Подпрограммы ... 63
Процедуры ... 66

Глава 4 Запуск заданий по расписанию ... 68

Планирование заданий на локальных и удаленных системах ... 69
Введение в планирование заданий ... 69
Назначение и доступ к заданиям ... 72
Мониторинг назначенных заданий ... 74
Назначение заданий с помощью Scheduled Task Manager ... 76
Создание заданий с помощью Scheduled Task Manager ... 76
Изменение свойств задания ... 81
Копирование и перемещение заданий
с одной системы на другую ... 82
Включение и выключение заданий ... 83
Немедленный запуск задания ... 83
Удаление ненужных заданий ... 83
Планирование заданий с помощью Schtasks ... 84
Создание назначенных заданий с помощью Schtasks /Create ... 84
Изменение назначенных заданий с помощью Schtasks /Change ... 90
Запрос сведений о назначенных заданиях
через Schtasks /Query ... 93
Немедленный запуск заданий с помощью Schtasks /Run ... 94
Остановка выполняемых заданий с помощью Schtasks /End ... 94
Удаление заданий с помощью Schtasks/Delete ... 95

Часть II Системное администрирование Windows ... 97

Глава 5 Управление Windows-системами ... 98

Изучение системной информации ... 98
Работа с реестром ... 101
Разделы и параметры реестра ... 101
Просмотр параметров реестра ... 104
Сравнение разделов реестра ... 104
Сохранение и восстановление разделов реестра ... 106
Добавление разделов реестра ... 107
Копирование разделов реестра ... 108
Удаление разделов реестра ... 109

Управление системными службами 110
Просмотр настроенных служб 110
Запуск, останов и приостановка служб 113
Настройка запуска службы 114
Настройка регистрации службы 115
Настройка восстановления служб 117
Перезагрузка и выключение систем из командной строки 120
Управление перезагрузкой и выключением локальной системы 121
Управление перезагрузкой и выключением удаленных систем 122
Добавление комментариев по причинам перезагрузки или выключения 123

Глава 6 Регистрация и отслеживание событий, автоматический мониторинг 124
Протоколирование событий Windows 124
Просмотр и фильтрация журналов событий 129
Просмотр событий и формат вывода 129
Фильтрация событий 132
Запись собственных событий в журналы 135
Мониторинг системы с помощью триггеров событий 137
Зачем использовать триггеры событий? 138
Подготовка к работе с триггерами событий 139
Создание триггеров событий 140
Отображение существующих триггеров событий 142
Удаление триггеров событий 143

Глава 7 Мониторинг процессов и производительности 145
Управление приложениями, процессами и производительностью ... 145
Системные и пользовательские процессы 146
Анализ выполняемых процессов 148
Мониторинг процессов и использования системных ресурсов ... 155
Останов процессов 160
Мониторинг для выявления и устранения проблем с производительностью 163
Мониторинг подкачки памяти для индивидуальных процессов ... 164
Мониторинг использования памяти и рабочего набора для индивидуальных процессов 165
Детальный анализ использования памяти и выявление источника утечки памяти 167

Часть III Управление дисками и файловыми системами в Windows 171

Глава 8 Конфигурирование и обслуживание жестких дисков 172
Приступаем к работе с DiskPart 172
Основы DiskPart 173
DiskPart: пример 173
Что такое фокус 174
Команды и сценарии DiskPart 175
DiskPart: пример сценария 178
Установка жестких дисков и управление ими 182
Установка и поиск новых устройств 182
Проверка состояния и конфигурации диска 182
Изменение типа таблицы разделов 185

Работа с базовыми и динамическими дисками 187
Базовые и динамические диски 188
Создание активного раздела 189
Изменение типа диска 190
Обслуживание жестких дисков 192
Получение информации о диске и управление файловыми системами с помощью FSUtil 192
Проверка диска на ошибки и поврежденные сектора 196
Управление автоматической проверкой при загрузке 199
Дефрагментация дисков 200

Глава 9 Разбиение базовых дисков на разделы 203
Получение информации о разделах 203
Создание разделов 204
Создание разделов на MBR-дисках 204
Создание разделов на GPT-дисках 207
Управление назначением букв дисков и точками монтирования 208
Назначение букв диска или точек монтирования 208
Смена букв диска или точек монтирования 209
Удаление букв дисков или точек монтирования 209
Форматирование разделов 210
Команда Format 211
Форматирование: пример 212
Управление разделами 213
Преобразование разделов или томов в NTFS 213
Изменение или удаление метки тома 217
Расширение разделов 218
Удаление разделов 219

Глава 10 Управление томами и RAID на динамических дисках ... 220
Получение сведений о томах и их состоянии 221
Создание простых томов и управление ими 223
Создание простых томов 223
Расширение простых томов 224
Подключение динамических дисков 226
Удаление томов 227
Создание RAID-массивов на динамических дисках 227
Реализация RAID-0 228
Реализация RAID-1 231
Реализация RAID-5 233
Управление RAID и восстановление после сбоя 235
Расформирование зеркального набора 235
Повторная синхронизация и восстановление зеркального набора 236
Восстановление RAID-0 237
Восстановление RAID-5 238

Часть IV Администрирование сетей и управление Active Directory 241

Глава 11 Основы управления Active Directory 242
Управление Active Directory из командной строки 243
Домены, контейнеры и объекты 243

Логические и физические структуры Active Directory 244
Составные имена 245
Средства командной строки, работающие с Active Directory 247
Запросы к каталогам командой DSQUERY 248
Подкоманды и синтаксис DSQUERY 249
Поиск по именам, описаниям и именам учетных записей в SAM ... 251
Задание при поиске разрешений Logon и Run As 252
Задание начального узла, области поиска и максимального числа объектов 254
Формат вывода имен 256
Использование DSQUERY совместно с другими средствами командной строки 258
Поиск проблемных учетных записей пользователей и компьютеров 259
Переименование и перемещение объектов 260
Удаление объектов из Active Directory 261

Глава 12 Управление учетными записями компьютеров и контроллерами домена 263

Общие сведения об управлении учетными записями компьютеров из командной строки 263
Создание учетных записей компьютеров в доменах Active Directory ... 265
Создание учетной записи компьютера 266
Настройка атрибутов учетных записей компьютеров и членства в группах 267
Управление свойствами учетных записей компьютеров 268
Просмотр и поиск учетных записей компьютеров 269
Установка или изменение атрибутов местонахождения и описания 272
Отключение и включение учетных записей компьютеров 272
Восстановление заблокированных учетных записей компьютеров 273
Перемещение учетных записей компьютеров 274
Удаление учетных записей компьютеров 275
Работа с контроллерами домена 275
Установка и удаление контроллеров домена 275
Поиск контроллеров домена в Active Directory 276
Серверы глобального каталога 277
Поиск серверов глобального каталога 278
Добавление или удаление глобального каталога 279
Проверка параметров кэширования и настроек глобального каталога 280
Назначение ролей координаторам операций 282
Поиск координаторов операций 283
Настройка ролей координаторов операций из командной строки 284

Глава 13 Управление пользователями и группами Active Directory 287

Обзор управления учетными записями пользователей из командной строки 287
Добавление учетных записей пользователей 290

Создание доменных учетных записей пользователей 291
Настройка атрибутов доменных учетных записей пользователей и участия в группах 293
Создание локальных учетных записей пользователей 295
Управление учетными записями пользователей 297
Просмотр и поиск учетных записей пользователей 297
Определение членства в группах для индивидуальных учетных записей пользователей 299
Настройка атрибутов учетных записей пользователей 300
Включение и отключение учетных записей 302
Восстановление просроченных учетных записей 303
Управление и восстановление паролей пользователей 304
Перемещение учетных записей пользователей 305
Переименование учетных записей пользователей 306
Удаление учетных записей пользователей 307
Обзор управления учетными записями групп из командной строки 308
Добавление учетных записей групп 310
Создание групп безопасности и распространения 310
Создание локальных групп и включение в них участников 314
Управление учетными записями групп 315
Просмотр и поиск учетных записей групп 315
Определение членства в группах 316
Изменение типа или области групп 317
Добавление, удаление или замена членов групп 318
Перемещение учетных записей групп 321
Переименование учетных записей групп 322
Удаление учетных записей групп 322

Глава 14 Управление сетевыми принтерами и службами печати 324

Получение технических сведений и информации о проблемах в работе принтеров 325
Отслеживание информации о принтерах и их драйверах 325
Отслеживание информации о спулере печати и статистики 330
Управление принтерами 335
Основы управления принтерами 336
Установка физически подключенных принтеров 336
Установка сетевых принтеров 339
Перечисление принтеров, настроенных на компьютере 340
Просмотр и установка принтера по умолчанию 341
Переименование принтеров 341
Удаление принтеров 342
Управление TCP/IP-портами для сетевых принтеров 342
Создание и изменение TCP/IP-портов для принтеров 343
Вывод информации о TCP/IP-портах, используемых принтерами 345
Удаление TCP/IP-портов, используемых принтерами 346
Настройка свойств принтера 346
Добавление комментариев и информации о местонахождении 346
Совместное использование принтеров 347

Публикация принтеров в Active Directory 348
Настройка страниц-разделителей и изменение режима печати .. 348
Планирование заданий на печать и установка приоритетов 349
Настройка спулинга и других дополнительных параметров принтера 350
Устранение проблем со спулингом 352
Проверка службы Print Spooler 352
Коррекция поврежденного спулера 353
Другие службы, требующие проверки 354
Управление очередями печати и индивидуальными заданиями 354
Просмотр заданий в очереди 354
Приостановка принтера и возобновление печати 355
Очистка очереди печати 356
Приостановка, возобновление и перезапуск печати отдельных документов 357
Удаление документа и отмена задания на печать 358

Глава 15 Настройка, поддержка и анализ проблем в TCP/IP-сетях 359

Использование оболочки сетевых сервисов 359
Контексты Netsh 359
Работа с удаленными компьютерами 362
Работа с файлами сценариев 363
Управление параметрами TCP/IP 364
Статический IP-адрес 365
Динамический IP-адрес 366
Добавление IP-адресов и шлюзов 367
Настройка DNS-серверов 367
Задание дополнительных DNS-серверов 369
Задание WINS-сервера 369
Задание дополнительных WINS-серверов 370
Удаление кэша ARP 371
Удаление параметров TCP/IP 372
Поддержка TCP/IP-сетей 373
Получение и сохранение конфигурации TCP/IP 373
Проверка IP-адресов и конфигурации интерфейсов 375
Протокол ICMP и соответствующие сообщения 378
Анализ фрагментации, восстановления и детальных сведений об ошибках 382
Анализ текущих TCP- и UDP-соединений 383
Выявление и устранение проблем в TCP/IP-сетях 389
Просмотр диагностической информации 389
Проблемы с клиентами почты, новостей и прокси 393
Проблемы с базовой конфигурацией компьютера 394
Проблемы с конфигурациями IP, DNS и WINS 404
Проблемы с TCP/IP-соединениями 406
Проверка возможности соединений 408

Приложение Справочник по основным утилитам командной строки 414

Предметный указатель 449

Благодарности

Желание сделать нечто принципиально иное по сравнению с тем, что уже было сделано, оказалось куда сильнее, чем я мог себе представить, и, надеюсь, вам не будет жаль времени, которое вы потратите на чтение этой книги. Знаете, есть масса книг для администраторов Windows и уйма пособий для тех, кто хотел бы автоматизировать рутинные операции средствами поддержки сценариев (scripts) в Windows, но до сих пор никто так и не удосужился написать целую книгу по администрированию Windows из командной строки, где основное внимание уделялось бы не самим командам, а задачам администрирования. Поэтому тщу себя надеждой, что результат кропотливой работы, который вы сейчас держите в руках, получился в своем роде уникальный. Моя книга не относится к тем, где говорится: «вот команда Edit; с ее помощью делается то-то и то-то; а это ее параметры». Конечно, отчасти присутствует и это, как в любой книге для администраторов, но главное в моей книге — другое, а именно: как использовать командную строку для решения рутинных задач администрирования. Вы научитесь выполнять повседневные административные процедуры и узнаете все детали реализации этих процедур средствами командной строки. Так что, если вы хотите понять, как из командной строки управлять рутинными операциями, отслеживать работу Windows, просматривать журналы событий, размечать диски на разделы, настраивать TCP/IP или делать сотни других дел, эта книга для вас.

Я уже неоднократно писал, что команда Microsoft Press выше всяческих похвал. Не могу не отметить важнейшую роль Валери Вулли (Valerie Woolley) во всем процессе работы над книгой. Она помогала мне не сбиться с пути и занималась согласованием материалов по мере сдачи мной очередных глав. Мартин Дельре (Martin DelRe) был выпускающим редактором этого проекта. Он верил в эту книгу и мой уникальный подход. Работать с ним было истинным удовольствием. Закончить и издать книгу без его помощи просто не удалось бы! Сьюзан

Маккланг (Susan McClung) руководила процессом редактирования со стороны корпорации nSight. Ее помощь как менеджера проекта этой и многих других моих книг поистине неоценима. Мой ей поклон!

К несчастью для автора (и к счастью для читателей) написанием книги процесс публикации не ограничивается. Следующий этап — редактирование и авторская правка. Должен заметить, мне еще нигде не довелось встречать столь качественного издательского процесса, как в Microsoft Press, — а я написал немало книг для других издательств. Научным редактором книги был Джим Джонсон (Jim Johnson). По-моему, это была наша первая совместная работа, и, вспоминая ее, я испытываю лишь самые теплые чувства. Он очень тщательно вычитывал материалы и помогал в тестировании, чтобы примеры, приведенные в книге, корректно работали как в Windows XP Professional, так и в Windows Server 2003. Я также хотел бы поблагодарить Питера Тетьена (Peter Tietjen) за качественную литературную правку книги.

Как всегда, хочу выразить свою признательность Майклу Болинджеру (Michael Bolinger), Энн Гамильтон (Anne Hamilton) и Джулиане Элдоус Эткинсон (Juliana Aldous Atkinson). На протяжении всей моей писательской карьеры они часто оказывали мне всяческую помощь и были рядом, когда я больше всего нуждался в них. Спасибо вам и за то, что вы проводили многие мои проекты через весь издательский процесс!

Также благодарю литературное агентство Studio B и моих агентов, Дэвида Рогелберга (David Rogelberg) и Нила Солкайнда (Neil Salkind). Работать с вами — одно удовольствие.

Надеюсь, я никого не забыл, но если и забыл, то не нарочно. *Честно-честно!* ;-)

Введение

Книга «Командная строка Microsoft Windows. Справочник администратора» задумана как краткий и удобный источник информации для администраторов Windows, руководство, которое в любой момент должно быть под рукой. В этой книге есть все, что нужно для успешного решения базовых административных задач с использованием командной строки Windows. Поскольку я постарался вместить максимум информации в небольшой формат, вам не придется перелистывать сотни страниц в поисках нужных сведений. Вы прочитаете только то, что нужно для работы.

Говоря коротко, эта книга — источник, к которому вы обращаетесь за разрешением любых вопросов, связанных с администрированием Windows из командной строки. В ней рассказано об основных процедурах и часто решаемых задачах, описаны типичные примеры и приведены списки параметров, пусть не всегда полные, но достаточно представительные. Одна из моих целей — сочетание краткости и полноты: только так удается создать руководство к действию, а не тысячестраничный талмуд или бесполезную стостраничную памятку. Надеюсь, что мой справочник поможет вам легко и быстро справляться с повседневной работой, возникающими проблемами и с реализацией более сложных задач администрирования, таких как автоматизация мониторинга, анализ утечек памяти, разметка дисков на разделы, управление Active Directory и устранение неполадок в сетях.

Кому адресована эта книга

Книга «Командная строка Microsoft Windows. Справочник администратора» охватывает две операционные системы: Windows Server 2003 и Windows XP Professional. Она адресована:

- администраторам Windows Server 2003;
- специалистам технической поддержки Windows XP Professional;

- опытным пользователям, временами исполняющим функции администратора;
- администраторам, переходящим на Windows Server 2003 с предыдущих версий Windows;
- администраторам, переходящим на Windows Server 2003 с других платформ.

Чтобы не отвлекаться на бесконечные пояснения, я исхожу из того, что вы знакомы с работой сетей, что у вас есть базовое представление о работе Windows и что эта операционная система уже установлена на ваших компьютерах. Иными словами, в книге нет отдельных глав по архитектуре Windows, а также по установке операционной системы, ее запуску и завершению работы с ней. Но я подробно описываю планирование задач, мониторинг Windows-систем, управление учетными записями, администрирование сетевых служб и многое другое.

Я также предполагаю, что вы достаточно хорошо знакомы с командами Windows и ее пользовательским интерфейсом. Если же вы чего-то не знаете, прочитайте документацию Windows.

Структура книги

Эта книга — справочник, а не учебник, поэтому ее основу составляют решения конкретных задач, а не описание компонентов Windows.

Важная черта справочника — легкость, с которой в нем удается отыскать нужную информацию. Книга содержит развернутое оглавление и подробный предметный указатель. Выполняемые задачи расписаны по пунктам и снабжены списками параметров, таблицами и ссылками на другие разделы книги. Книга состоит из частей и глав. Каждая часть начинается с краткого описания освещаемых в ней вопросов.

В части I «Основы командной строки Windows» рассматриваются основные задачи администрирования из командной строки. В главе 1 дается обзор концепций, инструментов и методик администрирования из командной строки. Глава 2 поможет вам эффективно использовать командную оболочку. В ней подробно рассматриваются процедуры запуска командной оболочки с параметрами, управление переменными окружения, задающими пути для команд, способы перенаправления вывода и конвейеризация команд. В главе 3 обсуждаются основы создания сценариев (scripts) командной строки. Вы на-

учитесь задавать значения переменных, работать с операторами ветвления по условию и создавать процедуры. В главе 4 объясняется, как автоматизировать рутинные административные задачи с использованием командной строки.

Microsoft Windows предоставляет множество утилит командной строки, помогающих в управлении повседневными операциями. Часть II «Системное администрирование Windows» посвящена обсуждению базовых утилит и способов управления Windows-системами. В главе 5 рассматриваются многие из важнейших средств администрирования, в том числе тех, которые помогают собирать информацию о системе, работать с реестром, настраивать Windows-службы и удаленно завершать работу систем. В главе 6 изучаются средства мониторинга информации (включая предупреждения и ошибки), записываемой в журналы событий Windows. Вы узнаете, как записывать события в журналы System (Система) и Application (Приложение). В главе 7 вы ознакомитесь со способами и средствами мониторинга приложений, анализа процессов и поддержания производительности на должном уровне.

Часть III называется «Управление дисками и файловыми системами в Windows». Жесткие диски необходимы пользователям для хранения своих документов — с текстом, таблицами и данными других типов. Если вы уже работали с Windows XP или Windows Server 2003, то, вероятно, пользовались службой Disk Management (Управление дисками). Ее эквивалент, работающий в командной оболочке, — утилита для разметки дисков на разделы (DiskPart). С помощью DiskPart можно решать большинство задач, связанных с управлением дисками, а также выполнять некоторые дополнительные операции, недоступные из GUI-оболочки. Глава 8 является введением в функциональность DiskPart; здесь же обсуждаются такие утилиты, как FSUtil, CHKDSK и CHKNTFS. В главе 9 рассказывается о базовых дисках (basic disks), а в главе 10 — о динамических (dynamic disks). Вы узнаете, чем они отличаются и для чего используются. В этой главе также поясняется, как реализовать RAID-массивы, управлять ими и устранять возможные проблемы.

В части IV «Администрирование сетей и управление Active Directory» основное внимание уделяется базовым командам для конфигурирования Active Directory, служб печати и TCP/IP-сетей, а также управления ими и устранения проблем. В главе 11

рассматриваются многие из ключевых средств администрирования службами каталогов, в том числе средств, позволяющих собирать информацию о каталогах. В главе 12 изучаются средства для создания и управления учетными записями компьютеров в Active Directory. Вы узнаете, как настроить контроллеры домена в качестве координаторов глобальных каталогов и операций (global catalogs and operations masters). Глава 13 завершает обсуждение служб каталогов, в ней описываются процедуры создания и управления учетными записями пользователей и групп в Active Directory. Глава 14 посвящена сетевой печати и соответствующим службам. В главе 15 описывается, как из командной строки настраивать TCP/IP-сети, поддерживать их и устранять возникающие проблемы.

Условные обозначения

Чтобы текст было удобнее читать, я ввел в него несколько дополнительных элементов. Код и листинги набраны моноширинным (фиксированным) шрифтом. Команда или текст, которые нужно ввести с клавиатуры, выделены **полужирным** начертанием. Сетевые адреса и новые термины выделяются *курсивом*. Дополнительные сведения приводятся в следующих разделах.

Примечание Комментарий к описываемой процедуре или операции.

Совет Подсказка, дополнительная информация или рекомендация, связанная с описываемой процедурой или концепцией.

Внимание! Предупреждение о потенциальных проблемах.

Я искренне надеюсь, что книга «Командная строка Microsoft Windows. Справочник администратора» поможет вам легко и быстро выполнять основные административные задачи. Я буду счастлив, если вы поделитесь со мной своими впечатлениями, прислав их по адресу *williamstanek@aol.com*. Спасибо!

Поддержка

Издательский коллектив приложил все усилия, чтобы обеспечить точность информации в книге. Список исправлений, если таковые понадобятся, вы найдете по адресу *http://www.microsoft.com/learning/support/*.

Ваши замечания, вопросы и предложения по этой книге направляйте в Microsoft Press.

Наш почтовый адрес:

Microsoft Press

Attn: Editor, Microsoft Windows Command Line Administrator's Pocket Consultant

One Microsoft Way

Redmond, WA 98052-6399

Электронная почта:

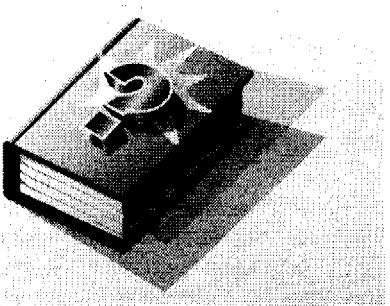
mspinput@microsoft.com

Пожалуйста, обратите внимание, что техническая поддержка продуктов по указанным адресам не осуществляется. Сведения о технической поддержке Windows Server 2003 или Windows XP Professional вы найдете по адресу *http://www.microsoft.com/support/*.

Часть I

Основы командной строки Windows

Книга «Командная строка Microsoft Windows. Справочник администратора» адресована пользователям Microsoft Windows Server 2003 и Microsoft Windows XP Professional. В этой части даются базовые сведения о командной строке. В главе 1 рассказывается, что такое командная строка и какие средства командной строки доступны. В главе 2 я продолжу рассмотрение этой темы и расскажу о пакетных сценариях (batch scripting). Вы ознакомитесь с методами создания и использования сценариев командной строки. В главе 3 вы узнаете, как автоматически выполнять задачи по расписанию. В частности, в ней рассматриваются способы управления системами, находящимися в разных доменах. В последней главе этой части поясняется, как вести мониторинг работоспособности системы и ее состояния средствами командной строки.



Глава 1

Обзор командной строки Windows

Поддержка командной строки встроена в операционную систему Microsoft Windows и доступна через окно командной оболочки. Командная строка поддерживается во всех версиях Windows и служит для запуска встроенных команд, утилит и сценариев. Несмотря на мощь и гибкость командной строки, некоторые администраторы Windows никогда ее не используют. Если вам хватает графических средств администрирования, можно применять только их, щелкая мышью элементы пользовательского интерфейса.

Однако опытные администраторы Windows, квалифицированные специалисты по технической поддержке и «продвинутые» пользователи не могут обойтись без командной строки. Зная, как правильно применять командную строку, в частности, какие средства командной строки выбрать, как и когда их использовать, чтобы они работали эффективно, можно избежать многочисленных проблем и добиться четкого выполнения операций. Если вы занимаетесь поддержкой нескольких доменов или сетей, то для автоматизации ежедневных операций не только важно, но и необходимо иметь представление об экономящих время способах работы с командной строкой.

В этой главе я поясню, как использовать встроенные команды, как запускать утилиты командной строки и как работать со средствами поддержки и инструментами из набора ресурсов (resource kit).

Примечание Читая эту и остальные главы, помните, что книга написана по Windows Server 2003 и Windows XP Professional. Методики, о которых рассказывается в книге, применимы в обеих операционных системах, если не оговорено обратное. Зачастую рассматриваемые методики применимы и к другим операционным системам се-

мейства Windows, хотя задаваемые параметры и выполняемые функции могут быть несколько иными. Во всяком случае всегда следует тестировать команды, параметры и сценарии перед использованием. А для этого лучше всего использовать среду разработки или тестирования, где системы, с которыми вы работаете, изолированы от остальной сети.

Средства работы с командной строкой

С каждой новой версией Windows командная строка совершенствовалась, а ее возможности расширялись. Командная строка претерпела значительные изменения, связанные не только с повышением производительности, но и с увеличением гибкости. Теперь с помощью командной строки Windows можно решать задачи, которые нельзя было решить в предыдущих версиях Windows. Чтобы вы как можно быстрее приступили к продуктивной работе с командной строкой, в следующих разделах рассматриваются параметры командной оболочки и ее настройка, а также даются советы по использованию хронологии команд.

Среду командной оболочки Windows инициализируют разными способами, в частности указывая параметры при запуске Cmd.exe или используя собственный стартовый файл, хранящийся в каталоге *%SystemRoot%*\System32. На рис. 1-1 показано окно командной оболочки. По умолчанию длина командной строки равна 80 символам, а в окне командной оболочки Windows Server 2003 умещается 25 строк текста. В Windows XP Professional число строк по умолчанию зависит от разрешения экрана, но всегда составляет не менее 25. Когда в окно командной оболочки выводится дополнительный текст или вводятся команды, текущий текст показывается в окне, а предыдущий прокручивается вверх, если окно заполнено. Если вам нужно приостановить вывод от команды, нажмите Ctrl+S. Повторное нажатие Ctrl+S возобновляет вывод, а Ctrl+C прекращает выполнение команды.

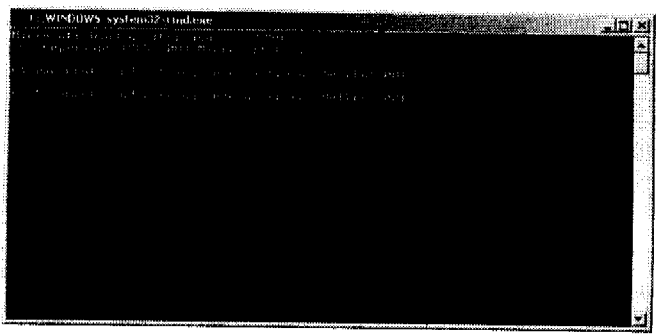


Рис. 1-1. 32-разрядная командная оболочка — основное окно командной строки, с которым вы будете иметь дело

Примечание Собственные стартовые файлы используются при работе с программами MS-DOS, требующими специальных конфигураций. До появления Windows XP эти

В данном случае приглашение командной строки (command prompt) показывает текущий рабочий каталог. По умолчанию это *%UserProfile%*, т. е. каталог профиля текущего пользователя. Мигающий курсор за приглашением означает, что командная строка находится в интерактивном режиме. В этом режиме вы можете вводить команды и нажимать Enter, чтобы их выполнить. Например, чтобы увидеть содержимое текущего каталога, введите **dir** и нажмите Enter.

Кроме того, командная строка может работать в пакетном режиме для выполнения набора команд. В пакетном режиме командная строка считывает и выполняет команды одну за другой. Обычно команды пакетного режима считываются из файла сценария (script file), но их можно вводить и в командной строке, например с помощью команды FOR обрабатывать каждый файл в определенной группе файлов. (Подробнее о пакетных сценариях, циклах и командах ветвления см. в главе 3.)

Работая с командной строкой Windows, вы должны понимать, откуда берутся используемые вами команды. «Родные» команды (встроенные в операционную систему) бывают двух видов:

- внутренние — существуют внутри командной оболочки; у них нет отдельных исполняемых файлов;
- внешние — реализованы в отдельных исполняемых файлах, которые обычно хранятся в каталоге *%SystemRoot%*\System32.

Внутренние команды оболочки Cmd.exe перечислены в табл. 1-1.

Табл. 1-1. Краткий справочник по внутренним командам командной оболочки (Cmd.exe)

Команда	Описание
assoc	Выводит или изменяет сопоставления (associations) типов файлов
break	Задает точки останова при отладке
call	Вызывает из сценария процедуру или другой сценарий
cd (chdir)	Показывает имя текущего каталога или выполняет смену текущего каталога
cls	Очищает окно командной строки и буфер экрана
color	Задает цвета текста и фона окна командной оболочки
copy	Копирует файлы или выполняет конкатенацию файлов
date	Показывает или устанавливает текущую дату

(см. след. стр.)

Табл. 1-1. (*продолжение*)

Команда	Описание
del (erase)	Удаляет заданный файл, группу файлов или каталог
dir	Показывает список подкаталогов и файлов в текущем или заданном каталоге
echo	Выводит текст в окно командной строки или задает, надо ли отображать команды на экране (on|off)
endlocal	Отмечает конец локализации (локальной области видимости) переменных
exit	Выход из оболочки командной строки
for	Выполняет заданную команду для каждого файла в наборе
ftype	Выводит или изменяет текущие типы файлов в сопоставлениях расширений файлов с программами
goto	Указывает, что интерпретатор команд должен перейти на строку с заданной меткой в пакетном сценарии
if	Выполняет команды по условию
md (mkdir)	Создает подкаталог в текущем или заданном каталоге
move	Перемещает файл или группу файлов из текущего или заданного исходного каталога в указанный каталог. Также может переименовывать каталог
path	Показывает или задает путь к командам, используемый операционной системой при поиске исполняемых файлов и сценариев
pause	Приостанавливает выполнение пакетного файла и ожидает ввода с клавиатуры
popd	Делает текущим каталог, имя которого было сохранено командой PUSHD
prompt	Указывает, какой текст должен показываться в строке приглашения
pushd	Сохраняет имя текущего каталога и при необходимости делает текущим заданный каталог
rd (rmdir)	Удаляет каталог или каталог вместе с его подкаталогами
rem	Помечает комментарии в пакетном сценарии или Config.nt
ren (rename)	Переименовывает файл или группу файлов
set	Показывает текущие переменные окружения или задает временные переменные для текущей командной оболочки
setlocal	Отмечает начало локализации (локальной области видимости) переменных в пакетных сценариях
shift	Сдвигает позицию замещаемых параметров в пакетных сценариях

Табл. 1-1. *(окончание)*

Команда	Описание
start	Запускает заданную программу или команду в отдельном окне
time	Показывает или устанавливает системное время
title	Задает заголовок окна командной оболочки
type	Показывает содержимое текстового файла
verify	Включает режим проверки файлов после записи на диск
vol	Показывает метку и серийный номер дискового тома

Синтаксис любой внутренней команды (и большинства внешних) можно получить, введя в командной строке имя команды и /?, например:

```
copy /?
```

Вы увидите, что внешних команд гораздо больше, чем внутренних, и что некоторые внешние команды во многом аналогичны встроенным. Большинство таких аналогичных команд каким-либо образом расширяют или улучшают встроенные команды. Так, внешняя команда XCOPY гибче, чем встроенная COPY: она позволяет копировать не только файлы, но и деревья каталогов, а также указывать массу дополнительных параметров. С помощью внешней команды SETX можно записывать значения переменных окружения прямо в реестр Windows, т. е. изменения сохраняются, а не являются временными, как при выполнении команды SET.

Совет Команда SETX также позволяет получать текущие значения параметров реестра и записывать их в текстовый файл. Эта команда включена в Windows Server 2003, а в Windows XP доступна, только если установлена версия Windows Support Tools для Windows XP Professional.

В остальном разница между внутренними и внешними командами не так существенна. У многих утилит Windows есть версии, рассчитанные на командную строку, которые позволяют задавать параметры утилиты в командной строке. Таким образом, эти утилиты используются аналогично внешним командам. Далее в этой главе я рассмотрю два основных источника утилит Windows: Microsoft Windows Support Tools и Microsoft Windows Server 2003 Resource Kit. Вы также можете найти утилиты сторонних поставщиков, имеющие версии для командной строки.

Введение в командную оболочку MS-DOS

В командной оболочке MS-DOS (Command.com) выполняются 16-разрядные команды подсистемы MS-DOS и других подсистем. Для запуска командной оболочки MS-DOS можно открыть меню Start (Пуск), выбрать команду Run (Выполнить) и ввести **command** в поле Open (Открыть). Или ввести **command** в другой командной строке и нажать Enter.

Совет Если вы запускаете командную оболочку MS-DOS из Cmd.exe, заголовок командной оболочки изменится на «Command Prompt — command» (Командная строка — command), показывая, с какой оболочкой вы имеете дело. Закончив работу с Command.com, введите exit для выхода из командной оболочки MS-DOS и возврата в командную строку Windows.

Среду для командной оболочки MS-DOS можно инициализировать несколькими способами, например, передать параметры при запуске Command.com или задать параметры в стартовом файле Config.nt, находящемся в папке *%SystemRoot%* \System32. Как и окно Cmd.exe, окно командной строки MS-DOS имеет ширину 80 символов и в зависимости от операционной системы и разрешения экрана по умолчанию показывает минимум 25 строк текста. При запуске командной оболочки MS-DOS выводится стандартный текст вида:

```
Microsoft(R) Windows DOS
(C) Copyright Microsoft Corp 1990-2001.
C:\>
```

Как и командная оболочка Windows, командная оболочка MS-DOS поддерживает интерактивный и пакетный режимы. Кроме того, она поддерживает «родные» команды, встроенные Microsoft. Эти команды делятся на две категории.

- **Внутренние команды конфигурирования** Команды, используемые для конфигурирования подсистемы MS-DOS, которые следует помещать в стартовые файлы, например в Config.nt или Autoexec.nt, либо в информационные файлы программ (INF-файлы). К командам конфигурирования относятся: BUFFERS, COUNTRY, DEVICE, DEVICEHIGH, DOS, DOSONLY, DRIVEPARM, ECHOCONFIG, FCBS, FILES, INSTALL, LOADHIGH, LASTDRIVE, NTCMDPROMT, SHELL, STACKS и SWITCHES.
- **Стандартные внешние команды** Команды, которые можно вводить в командной строке, выполнять в сценариях и в не-

которых случаях использовать в стартовых файлах. К стандартным внешним командам относятся: APPEND, DEBUG, EDIT, EDLIN, EXE2BIN, EXPAND, FASTOPEN, FORCEDOS, GRAPHICS, LOADFIX, MEM, NLSFUNC, SETVER и SHARE. Эти команды MS-DOS можно выполнять и в Cmd.exe.

Когда вы вводите в оболочке MS-DOS другие команды, они передаются в 32-разрядную командную оболочку, где и выполняются. Поэтому, например, в командной оболочке MS-DOS можно использовать внутреннюю команду COPY. Однако в 64-разрядных версиях Windows Server 2003 стандартные внешние команды оболочки MS-DOS недоступны.

Настройка свойств командной оболочки

Конечно, если вы часто работаете с командной оболочкой, имеет смысл настроить ее свойства. Например, добавить буферы, чтобы просматривать текст, который при прокрутке покинул область просмотра, изменить размер окна командной оболочки, сменить шрифты и т. д.

Чтобы приступить к настройке свойств, щелкните значок командной строки в верхнем левом углу окна командной оболочки или щелкните правой кнопкой мыши строку заголовка консоли и выберите Properties (Свойства). Как показано на рис. 1-2, в диалоговом окне Command Prompt Properties (Свойства: «Командная строка») четыре вкладки.

- **Options (Общие)** Настройка размера курсора, параметров отображения и редактирования, а также хронологии команд. Установите флажок QuickEdit Mode (Выделение мышью), чтобы с помощью мыши выделять и вставлять текст в окне командной строки. Сбросьте флажок Insert Mode (Быстрая вставка), если вы хотите по умолчанию вести редактирование в режиме замещения. В разделе хронологии команд настраивается то, как ранее введенные команды буферизуются в памяти. (Подробнее о хронологии команд см. в разделе «Хронология команд» далее в этой главе.)

Совет Работая с чисто текстовыми командами и средствами, вы, возможно, выберете режим отображения Full Screen (Во весь экран), чтобы уменьшить объем памяти, используемый самой командной строкой. Чтобы впоследствии выйти из командной строки и вернуться в Windows, введите **exit**.

- **Font (Шрифт)** Настройка размера и начертания шрифта в окне командной строки. Для растровых шрифтов задается высота и ширина в пикселах. Например, размер 8 x 12 означает 8 экранных пикселов в ширину и 12 в высоту. Для других шрифтов указывается размер в пунктах, например, вы можете выбрать шрифт Lucida Console размером 10 пт. Интересно, что если выбран размер *n* пт, шрифт будет иметь высоту *n* пикселов; например, у шрифта размером 10 пт высота равна 10 экранным пикселам. Кроме того, можно выбрать полужирное начертание, и тогда ширина шрифта в пикселах увеличится.
- **Layout (Расположение)** Задание размера буфера экрана, размера и позиции окна. Указывайте высоту буфера, достаточную для того, чтобы можно было просмотреть вывод предыдущих команд и сценариев. Для этого параметра подойдет значение в диапазоне 1000–2000. Высоту окна выбирайте такой, чтобы можно было одновременно видеть более одного окна командной оболочки. При разрешении экрана 800 x 600 и шрифте размером 12 пт подойдет высота в 45 строк. Если вы хотите, чтобы окно командной строки появлялось в заданном месте экрана, сбросьте флажок Let System Position Window (Автоматический выбор) и укажите в полях Left (Левый край) и Top (Верхний край) координаты левого верхнего угла окна командной строки.
- **Colors (Цвета)** Настройка цветов текста и фона в окне командной строки. Параметры Screen Text (Текст на экране) и Screen Background (Фон экрана) задают соответственно цвет текста и цвет фона. Параметры Popup Text (Текст всплывающего окна) и Popup Background (Фон всплывающего окна) указывают цвет текста и цвет фона в диалоговых окнах, генерируемых при выполнении команд в окне командной строки.

Завершив настройку свойств командной оболочки, щелкните OK. Windows запросит, как применить указанные вами параметры. Изменения можно применить только к текущему окну или ко всем окнам командных строк, с которыми вы будете работать в дальнейшем. Кроме того, возможно, будет предложено изменить ярлык, с помощью которого вы запустили текущее окно. В этом случае всякий раз, когда вы запускаете командную оболочку двойным щелчком этого ярлыка, будут использоваться заданные вами параметры.

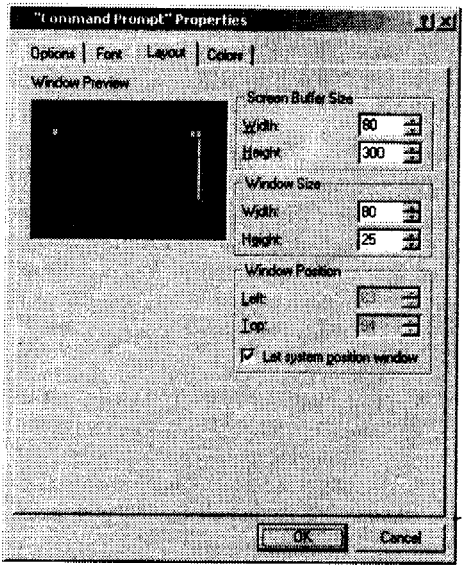


Рис. 1-2. Настройка свойств командной строки

Хронология команд

Буфер хронологии команд — функциональность командной оболочки Windows (Cmd.exe). В нем запоминаются команды, выполненные в текущей командной оболочке, и он позволяет повторно обращаться к этим командам, не вводя их текст заново. Максимальное число команд в буфере задается в диалоговом окне Properties (Свойства) командной оболочки, о котором шла речь в предыдущем разделе. По умолчанию запоминается до 50 команд.

Для изменения размера хронологии выполните следующие операции.

1. Щелкните правой кнопкой мыши строку заголовка окна командной оболочки, выберите Properties (Свойства) и откройте вкладку Options (Общие).
2. В поле Buffer Size (Размер буфера) задайте максимальное количество команд, запоминаемых в хронологии, а затем щелкните OK.
3. Чтобы сохранить этот параметр на будущее, выберите Modify Shortcut That Started This Window (Изменить ярлык

для запуска этого окна) и щелкните OK. Если сохранять параметр на будущее не нужно, просто щелкните OK.

К командам в хронологии можно обращаться несколькими способами.

- **Выбор с помощью клавиш-стрелок** С помощью клавиш «стрелка вверх» и «стрелка вниз» можно перемещаться вверх и вниз по хранящемуся в буфере списку команд. Когда вы найдете команду, которую нужно выполнить, нажмите Enter — она будет выполнена так же, как если бы ее ввели вручную. Кроме того, вы можете отредактировать показанный текст команды, например добавить или изменить параметры, а потом нажать Enter.
- **Просмотр хронологии команд во всплывающем окне** Нажмите F7, чтобы открыть всплывающее окно со списком команд в буфере. Затем с помощью клавиш-стрелок выберите команду. (Также можно нажать F9, набрать на клавиатуре номер команды и нажать Enter.) Нажмите Enter, чтобы выполнить выбранную команду, или Esc, чтобы закрыть всплывающее окно, не выполняя команду.
- **Поиск в хронологии команд** Введите несколько первых букв нужной команды и нажмите F8. Командная оболочка попытается найти в хронологии первую команду, которая начинается с введенных вами символов. Нажмите Enter, чтобы выполнить эту команду, или F8, чтобы найти в буфере следующую команду, начинающуюся с тех же символов.

Работая с хронологией команд, учтите, что у каждого экземпляра Cmd.exe свой набор буферов команд. Таким образом, буферы видны только в контексте соответствующей командной оболочки.

Windows Support Tools

Windows Support Tools — набор утилит, позволяющих решать самые разные задачи: от диагностики системы до мониторинга сети. Эти утилиты можно устанавливать и использовать во всех версиях Windows Server 2003 и Windows XP Professional. Для их установки выполните следующие операции.

1. Вставьте CD-ROM с дистрибутивом соответствующей операционной системы (Windows Server 2003 или Windows XP Professional) в привод CD-ROM.

Внимание! Поскольку установка Support Tools влияет на Help And Support Center (Центр справки и поддержки), следует закрыть любые экземпляры этой консоли, запущенные перед установкой. Если вы этого не сделаете, установка потерпит неудачу.

2. Когда появится окно Autorun, щелкните Perform Additional Tasks, а затем щелкните Browse This CD. Запустится Windows Explorer.
3. В Windows Explorer дважды щелкните Support и Tools.

Примечание В этой книге я часто буду упоминать двойной щелчок — самый распространенный способ открытия папок и запуска программ. При двойном щелчке первый щелчок выбирает элемент, а второй — открывает/запускает элемент. Windows Server 2003 и Windows XP можно настроить на запуск/открытие по одинарному щелчку. В этом случае для открытия/запуска элемента нужно поместить на него курсор и щелкнуть один раз. Параметры щелчка для папки можно изменить в Control Panel (Панель управления) через Folder Options (Свойства папки). Для этого щелкните ярлычок General (Общие), выберите Single-Click To Open An Item (Открывать одним щелчком, выделять указателем) или Double-Click To Open An Item (Открывать двойным, а выделять одним щелчком).

4. Дважды щелкните Suptools.msi, чтобы запустить Windows Support Tools Setup Wizard. Щелкните Next.
5. Прочитайте End User License Agreement и, если вы согласны с его условиями и хотите продолжить установку, щелкните I Agree, затем Next.
6. Введите информацию о пользователе, затем щелкните Next.
7. В Windows XP Professional вы увидите страницу Select An Installation Type. Выберите Complete, затем Next.
8. Укажите каталог, куда будут устанавливаться средства поддержки. По умолчанию это *%ProgramFiles%*\Support Tools. Если вы не хотите использовать каталог по умолчанию, введите другой или щелкните Browse для выбора каталога. В Windows Server 2003 средства поддержки занимают примерно 24 Мб дискового пространства. При установке версии средств поддержки для Windows XP Professional с параметром Complete требуется около 12 Мб дискового пространства.

9. Нажмите Install Now.
10. Щелкните Finish на странице Completing The Windows Support Tools Setup Wizard.

После установки можно приступить к работе со средствами поддержки в Help And Support Center (рис. 1-3). (На этой иллюстрации показано окно в Windows Server 2003; окно в Windows XP Professional выглядит очень похоже.) Щелкните Start (Пуск), Programs (Программы) или All Programs (Все программы), выберите Windows Support Tools, а затем Support Tools Help. Как показано на иллюстрации, средства поддержки упорядочены по именам файлов, названиям утилит и категориям. Щелкнув название утилиты, вы откроете справочную страницу с электронной документацией по данной утилите; эту страницу можно использовать и для запуска утилиты.

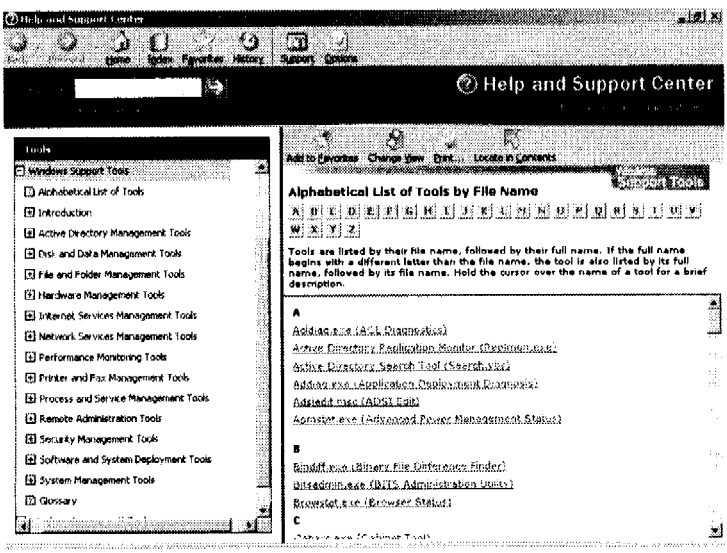


Рис. 1-3. Средства поддержки, применяемые для решения таких задач, как диагностика системы или мониторинг сети

У большинства средств поддержки имеются версии, рассчитанные на командную строку. Исполняемые файлы средств поддержки находятся в каталоге установки. По умолчанию это каталог *%SystemDrive%* \Program Files\Support Tools.

Поскольку в системный путь (переменная path) добавляется каталог установки средств поддержки, запускать утилиты можно не только из каталога установки, но и из командной строки независимо от текущего каталога. Как и в случае других команд и утилит Windows, чтобы увидеть синтаксис командной строки запуска любой утилиты из набора средства поддержки, введите имя команды, пробел и /?, например **spcheck /?**.

Windows Server 2003 Resource Kit Tools

Еще один замечательный ресурс, содержащий утилиты Windows, — Microsoft Windows Server 2003 Resource Kit (Ресурсы Microsoft Windows Server 2003). Как и средства поддержки, инструменты из этого набора ресурсов распространяются с Windows Server 2003 и могут использоваться в любых версиях Windows Server 2003 и Windows XP Professional.

Утилиты из набора ресурсов распространяются на CD-ROM в составе «коробочного» набора ресурсов, кроме того, их можно бесплатно скачать с сайта Microsoft Windows Download Center. Для установки набора ресурсов выполните следующие операции.

1. Вставьте CD-ROM с Windows Server 2003 Resource Kit в привод CD-ROM или дважды щелкните исполняемый файл Rktools.exe, скачанный с сайта Microsoft.

Внимание! Поскольку установка набора ресурсов влияет на консоли Windows, в частности на Help And Support Center, перед установкой набора ресурсов следует закрыть все запущенные программы. Если вы этого не сделаете, установка может закончиться неудачей или же не удастся обновить общие компоненты программ, выполняемых в данный момент (что чревато непредсказуемыми результатами).

2. Когда запустится Windows Resource Kit Tools Setup Wizard, щелкните Next.
3. Прочитайте End User License Agreement и, если вы согласны с его условиями и собираетесь продолжить установку, щелкните I Agree, а затем Next.
4. Введите информацию о пользователе и нажмите Next.
5. Выберите каталог установки. По умолчанию это *%ProgramFiles%*\Windows Resource Kits\Tools. Если вас не устраивает каталог по умолчанию, введите другой каталог или щелкните

Browse для выбора каталога. По умолчанию набор ресурсов занимает примерно 37 Мб дискового пространства.

6. Щелкните Install Now.
7. Щелкните Finish на странице Completing The Windows Resource Kit Tools Setup Wizard.

По окончании установки вы можете работать с утилитами из набора ресурсов через Windows Resource Kit Tools Help. Щелкните Start (Пуск), Programs (Программы) или All Programs (Все программы), а затем выберите Windows Resource Kit Tools и Windows Resource Kit Tools Help. Как и средства поддержки, средства набора ресурсов упорядочены по именам файлов, названиям утилит и категориям. Щелкнув название утилиты, вы откроете справочную страницу с электронной документацией; ее можно использовать и для запуска утилиты.

Исполняемые файлы средств набора ресурсов находятся в каталоге установки. По умолчанию это каталог *%SystemDrive%*\Program Files\Windows Resource Kits\Tools. Как и в случае средств поддержки, в системный путь добавляется каталог установки набора ресурсов. Поэтому средства набора ресурсов можно запускать из командной строки независимо от текущего каталога. Чтобы посмотреть синтаксис команды, введите ее имя, пробел и /?, например **creatfil /?**. Учтите, что не у всех команд имеется справка, вызываемая таким образом.

Глава 2

Эффективная работа с командной строкой

Командная оболочка — весьма мощная среда работы с командами и сценариями. Как рассказывалось в предыдущей главе, в командной строке можно запускать команды разных типов: встроенные команды, утилиты Windows и версии приложений, рассчитанные на командную строку. Независимо от типа каждая команда, которую вы будете использовать, должна соответствовать одним и тем же синтаксическим правилам. Согласно этим правилам, за именем команды идут обязательные или необязательные аргументы. Кроме того, аргументы могут использовать перенаправление ввода, вывода или стандартных ошибок.

Выполняя команду в командной оболочке, вы инициируете такую последовательность событий.

1. Командная оболочка заменяет любые переменные, введенные в тексте команд, их текущими значениями.
2. Если введена группа или цепочка из нескольких команд, строка разбивается на отдельные команды, которые в свою очередь разбиваются на имя и аргументы команды. Далее команды обрабатываются по отдельности.
3. Если в имени команды указан путь, командная оболочка ищет команду по этому пути. Если в указанном каталоге такой команды нет, командная оболочка возвращает ошибку.
4. Если в имени команды не задан путь, командная оболочка сначала пытается разрешить имя команды на внутреннем уровне. Если найдена внутренняя команда с таким именем, значит, вызвана внутренняя команда, которую сразу же можно выполнить. Если внутренней команды с таким именем нет, командная оболочка сначала ищет исполняемый файл команды в текущем каталоге, а затем в каталогах, перечис-

ленных в переменной окружения PATH. Если файла команды нет ни в одном из этих каталогов, командная оболочка возвращает ошибку.

5. Если команда найдена, она выполняется с заданными аргументами и при необходимости ввод считывается из источника, указанного в этих аргументах. Вывод и ошибки команд показываются в окне командной строки или направляются заданному приемнику вывода и ошибок.

Как видите, на выполнение команд влияют многие факторы, в том числе пути к командам, перенаправление ввода-вывода, группирование или создание цепочек команд. В этой главе мы рассмотрим, как с помощью таких средств управления командами добиться максимальной отдачи от командной оболочки. Но, прежде чем углубиться в эту тему, рассмотрим запуск командной оболочки и познакомимся с концепцией вложения командных оболочек.

Управление запуском командной оболочки

При работе с командной оболочкой вы, вероятно, запускали ее, открывая меню Start (Пуск) и выбирая Programs (Программы) или All Programs (Все программы), затем Accessories (Стандартные) и Command Prompt (Командная строка). Другие способы запуска командной строки — диалоговое окно Run (Запуск программы) или ввод **cmd** в другом, уже открытом окне командной оболочки. Эти способы позволяют при запуске командной строки указывать аргументы: ключи, управляющие работой командной строки, и параметры, инициирующие выполнение дополнительных команд. Например, можно запустить командную оболочку в «молчаливом» режиме (т. е. отключить эхо-вывод) командой **cmd /q** или сделать так, чтобы командная оболочка выполнила заданную команду и завершила свою работу, — для этого нужно ввести **cmd /c**, а затем текст команды в кавычках. В следующем примере командная оболочка запускается, выполняет команду ipconfig с выводом результатов в файл и завершается:

```
cmd /c "ipconfig > c:\ipconfig.txt"
```

В табл. 2-1 перечислены основные параметры командной оболочки Windows (Cmd.exe). Заметьте, что для некоторых параметров командной оболочки заданы значения по умолчанию. Поэтому в выводе команд обычно используются стандартные

ANSI-коды символов (а не Unicode-коды) и разрешены расширения команд, благодаря чему ряд встроенных команд предоставляет дополнительные возможности.

Табл. 2-1. Основные параметры командной оболочки

Параметр	Описание
/C	Указывает, что командная оболочка должна выполнить заданную команду и завершить работу
/K	Указывает, что командная оболочка должна выполнить заданную команду и остаться в интерактивном режиме
/A	Устанавливает для вывода команд в файлы (или по конвейеру) ANSI-кодировку (по умолчанию)
/U	Устанавливает для вывода команд в файлы (или по конвейеру) Unicode-кодировку
/U	Включает «молчаливый» режим, отключая эхо-вывод. По умолчанию эхо-вывод включен
/T:FG	Задает цвета текста и фона окна консоли
/E:ON	Активизирует расширения команд (по умолчанию)
/E:OFF	Отключает расширения команд

Примечание Некоторые параметры неприменимы одновременно с другими, например, нельзя задать и ANSI-, и Unicode-кодировку. Если вы укажете */A* и */U* или */E:ON* и */E:OFF*, будет действовать параметр, указанный последним.

Иногда нужно поработать, указав другие значения переменных окружения или параметры командной оболочки, а затем вернуться к исходным значениям, не выходя из консольного окна. Для этого применяется так называемое *вложение* (nesting). При вложении комадная строка запускается из другой командной строки, при этом вложенная командная строка наследует значения переменных окружения текущей комадной строки. Затем можно по своему усмотрению изменить значения переменных окружения и выполнять команды и сценарии с новыми значениями переменных. После ввода **exit** для выхода из вложенного экземпляра командной строки вы вернетесь в предыдущую командную строку, при этом предыдущие значения переменных окружения будут восстановлены.

Совет Учтите, что некоторые символы имеют особый смысл и командная оболочка, обнаружив один из таких символов, пытается выполнить специальную процедуру, сопоставленную с этим символом. К специальным символам относятся < > ( ) & | @ ^. Если вы хотите использовать специальный символ как обычный, с помощью escape-символа укажите, что командная оболочка должна интерпретировать специальный символ как литерал, не вызывая определенные для него специальные процедуры. Escape-символом служит знак ^. Этот знак указывается непосредственно перед специальным символом.

Путь к командам

В операционной системе Microsoft Windows для поиска исполняемых файлов используется путь к командам. Windows определяет, является ли файл исполняемым, по его расширению. Кроме того, расширения файлов могут быть сопоставлены заданным приложениям с помощью файловых сопоставлений (file associations). В следующих двух разделах рассказывается о пути к командам, расширениях файлов и файловых сопоставлениях.

Использование пути к командам

Текущий путь к командам, используемый при поиске исполняемых файлов, можно посмотреть командой PATH. Запустите командную оболочку, введите **path** и нажмите Enter. Если вы установили Windows Support Tools и Windows Resource Kit, вывод команды будет выглядеть примерно так:

```
PATH=C:\Program Files\Windows Resource Kits\Tools\;C:\Program
Files\Support Tools\;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\
System32\Wbem
```


Примечание Обратите внимание, что для отделения путей друг от друга служит точка с запятой (;). По ней командная оболочка определяет, где заканчивается один путь и начинается другой.

Путь к командам задается при входе в систему в переменных окружения пользователя и системы, а именно в переменной *%PATH%*. Командная строка ищет исполняемые файлы в каталогах в порядке, в котором эти каталоги перечислены в пути к командам. В предыдущем примере поиск исполняемых файлов ведется в следующем порядке.

1. C:\Program Files\Windows Resource Kits\Tools\.
2. C:\Program Files\Support Tools\.
3. C:\Windows\System32.
4. C:\Windows.
5. C:\Windows\System32\Wbem.

Сохранить изменение пути к командам в системном окружении позволяет команда SETX. (SETX — «родная» внешняя команда Windows Server 2003, а в Windows XP Professional она доступна только после установки версии Windows Resource Kit для Windows XP Professional с дистрибутивного компакт-диска этой операционной системы.) Например, если вы храните сценарии и приложения в определенных каталогах, то, возможно, есть смысл добавить эти каталоги в путь к командам. Чтобы добавить заданный каталог в существующий путь с помощью команды SETX, введите, например, **setx PATH "%PATH%;C\Scripts"**.

Примечание Обратите внимание на кавычки и точки с запятой. Кавычки нужны для того, чтобы значение *%PATH%*;C:\Scripts воспринималось как второй аргумент команды SETX, а точка с запятой, как уже говорилось, — чтобы указать, где заканчивается один путь и начинается другой.

В этом примере каталог C:\Scripts добавляется в существующий путь к командам, в результате чего путь к командам, показанный в предыдущем примере, изменится так:

```
PATH=C:\Program Files\Windows Resource Kits\Tools\;C:\Program
Files\SupportTools\;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS
\System32\Wbem;C:\Scripts
```

Не забывайте о порядке, в котором Windows ищет команды. Поскольку поиск ведется в том порядке, в каком каталоги перечислены в пути к командам, поиск в каталоге C:\Scripts выполняется в последнюю очередь. Иногда это может замедлить выполнение ваших сценариев. Чтобы Windows быстрее находила сценарии, можно сделать C:\Scripts первым каталогом, просматриваемым при поиске. Для этого путь к командам следует задать следующим образом:

```
setx PATH "C:\Scripts;%PATH%"
```

Указывая путь к командам, соблюдайте осторожность. Очень легко нечаянно перезаписать всю информацию о путях.

Например, если при задании пути к командам не указать переменную окружения *%PATH%*, потеряется вся остальная информация о путях к командам. Один из способов обеспечить возможность без проблем воссоздать путь к командам — создать файл, содержащий копию пути к командам. Чтобы записать в файл текущий путь к командам, введите **path > orig_path.txt**. Чтобы просто вывести путь к командам в окне командной строки, введите **path**.

Теперь у вас есть вывод команды или файл, содержащий исходный путь к командам. Команда path не только выводит текущий путь к командам, но и служит для временного задания пути к командам для текущей командной оболочки. Например, чтобы в текущей командной оболочке добавить в путь к командам каталог C:\Scripts, введите **path %PATH%;C:\Scripts**.

Расширения файлов и файловые сопоставления

Именно благодаря расширениям файлов становится возможным запуск команд простым вводом ее имени в командной строке. Используется два типа расширений файлов.

- **Расширения исполняемых файлов** Задаются переменной окружения *%PATHEXT%*. Чтобы посмотреть ее текущее значение, введите **set pathext** в командной строке. По умолчанию PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH. По значению этой переменной командная строка определяет, какие файлы являются исполняемыми, а какие — нет, поэтому для запуска исполняемого файла не обязательно указывать в командной строке его расширение.
- **Расширения файлов для приложений** Расширения файлов, открываемых приложениями, используются в файловых сопоставлениях. Файловые сопоставления — то, что позволяет передавать аргументы исполняемым файлам и открывать документы, электронные таблицы и другие файлы двойным щелчком значка файла. Для каждого известного системе расширения имеется файловое сопоставление, которое можно увидеть, введя assoc, а затем расширение, например assoc .exe. Каждое сопоставление задает тип файла для данного расширения файла. Информацию о типе файла можно выяснить, введя команду FTYPE и тип файла, например **ftype exefile**.

При поиске исполняемых файлов в каталоге командная оболочка ищет файлы в порядке, в котором перечислены их расширения. Таким образом, если в данном каталоге содержится несколько исполняемых файлов с одинаковыми именами, .com-файл имеет приоритет перед .exe-файлом и т. д.

Для каждого известного системе расширения файла и даже для расширений исполняемых файлов существуют сопоставление и тип файла. В большинстве случаев тип файла — это расширение файла, после которого идет ключевое слово *file* (без точки), например, cmdfile, exefile или batfile, а файловое сопоставление задает, что первый параметр — это имя команды, остальные параметры передаются приложению.

Если известно расширение, можно выяснить файловое сопоставление и тип файла с помощью команд ASSOC и FTYPE. Чтобы найти сопоставление, введите **assoc**, затем расширение файла вместе с точкой. Команда ASSOC выведет тип файла. Если вы введете **ftype *mun*** (где *mun* — вывод команды ASSOC), то увидите команду, соответствующую типу файла. Например, чтобы увидеть команду, соответствующую расширению исполняемого файла .exe, сначала введите **assoc .exe,** затем **ftype exefile**.

Вы получите следующую информацию о команде, соответствующей типу exefile:

```
exefile="%1" %*
```

Таким образом, когда вы запускаете .exe-файл, Windows рассматривает первое значение в строке как команду, а остальные значения — как передаваемые ей параметры.

Совет Сопоставления и типы файлов хранятся в реестре Windows и задаются командами ASSOC и FTYPE. Чтобы создать сопоставление файла, введите assoc, затем расширение и сопоставляемый ему тип файла, например **assoc .pl=perlfile**. Чтобы определить команду для типа файла, введите ftype и укажите тип файла, соответствующую ему команду и передаваемые параметры, например **ftype perlfile=C:\Perl\Bin\Perl.exe "%1" %***. Подробнее о сопоставлениях и типах файлов см. описание команд ASSOC и FTYPE в документации Help And Support Center (Центр справки и поддержки).

Перенаправление стандартных ввода, вывода и ошибок

По умолчанию команды получают ввод из параметров, указываемых при вызове команды в командной оболочке, и направляют свой вывод, в том числе и ошибки, в стандартное окно консоли. Но иногда нужно получить ввод из другого источника либо направить вывод в файл или на другое устройство вывода, например на принтер. Кроме того, может понадобиться вывод ошибок в файл, а не в окно консоли. Для таких задач перенаправления используется синтаксис, показанный в табл. 2-2 и рассматриваемый в следующих разделах.

Табл. 2-2. Синтаксис перенаправления ввода, вывода и ошибок

Синтаксис перенаправления	Описание
команда1 | команда2	Вывод первой команды служит вводом для второй
команда < [путь]имя_файла	Ввод команды поступает из заданного файла
команда > [путь]имя_файла	Вывод направляется в заданный файл; если такого файла нет, он создается, а если есть — перезаписывается
команда >> [путь]имя_файла	Если заданный файл есть, вывод добавляется в него, если нет, файл создается и в него записывается вывод
команда < [путь]имя_файла > [путь]имя_файла	Ввод команды поступает из первого заданного файла, а вывод направляется во второй (с перезаписью)
команда < [путь]имя_файла >> [путь]имя_файла	Ввод команды поступает из первого заданного файла, а вывод дозаписывается во второй
команда 2> [путь]имя_файла	Создается заданный файл, в который направляется вывод ошибок. Если такой файл есть, он перезаписывается
команда 2>&1 имя_файла	Ошибки записываются в тот же файл, что и стандартный вывод

Перенаправление стандартного вывода другим командам

Большинство команд генерирует вывод, который можно перенаправить другим командам в качестве ввода. Для этого используется так называемая *конвейеризация* (piping), при которой

вывод предыдущей команды служит вводом для следующей. Общий синтаксис конвейеризации выглядит так:

```
Команда1 | Команда2
```

где символ конвейеризации означает, что вывод Команды1 служит вводом для Команды2. Но вывод можно перенаправлять неоднократно, например:

```
Команда1 | Команда 2 | Команда 3
```

Две команды, для которых чаще всего используется конвейеризация, — FIND и MORE. Команда FIND ищет строки в файлах или в тексте ввода и выводит строки, соответствующие условию. Например, чтобы получить список всех файлов с расширением .txt в текущем каталоге:

```
dir | find ".txt "
```

Команда MORE принимает вывод других команд и разбивает его на части, каждая из которых умещается в окне консоли. Например, можно разбить файл журнала Dailylog.txt на страницы, выполнив команду вида:

```
type c:\working\logs\dailylog.txt | more
```

Полное описание синтаксиса этих команд можно получить, введя в командной строке **find /?** или **more /?**.

Перенаправление ввода-вывода в файлы

Еще один метод перенаправления — получение входных данных из файла с помощью символа переадресации ввода (<). Например, следующая команда сортирует содержимое файла Usernames.txt и показывает результаты в командной строке:

```
sort < usernames.txt
```

Можно не только считать ввод из файла, но и записать вывод в файл. При этом, чтобы создать или перезаписать выходной файл, используется символ >, а чтобы создать выходной файл или добавить в него вывод — символ >>. Например, если вы хотите вывести в файл данные о текущем сетевом статусе, можно выполнить команду:

```
netstat -a > netstatus.txt
```

К сожалению, если в текущем каталоге уже существует файл с тем же именем, эта команда удалит его и создаст но-

вый файл с таким именем. Если вы хотите добавить эту информацию в существующий файл, не перезаписывая его, измените текст команды следующим образом:

```
netstat -a >> netstatus.txt
```

Перенаправление ввода и вывода можно сочетать. Например, получить ввод команды из одного файла и направить вывод в другой файл. В следующем примере список имен пользователей считывается из файла и сортируется, а отсортированный список имен записывается в новый файл:

```
sort < usernames.txt > usernames-alphasort.txt
```

Перенаправление стандартных ошибок

По умолчанию ошибки команд, как и вывод, показываются в командной строке. Однако, если вы запускаете необслуживаемые сценарии или утилиты, для отслеживания ошибок понадобится вывод стандартных ошибок в файл. Один из способов перенаправить стандартные ошибки — указать в командной строке, что ошибки необходимо записывать туда же, куда и стандартный вывод. Для этого введите символ перенаправления **2>&1**, например:

```
chkdsk /r > diskerrors.txt 2>&1
```

В данном случае стандартный вывод и стандартные ошибки записываются в файл Diskerrors.txt. Если вы хотите видеть только ошибки, перенаправляйте только ошибки. В следующем примере стандартный вывод показывается в командной строке, а стандартные ошибки записываются в файл Diskerrors.txt:

```
chkdsk /r 2> diskerrors.txt
```

Создание цепочек и группирование команд

В предыдущих разделах я рассматривал методы перенаправления ввода-вывода, в частности конвейеризацию команд. Возможно, вас интересует, имеются ли другие способы выполнить последовательность команд. Да, имеются. Вы можете создать цепочку команд и выполнить их последовательно, а также выполнять команды по условию — в зависимости от успеха или неудачи предыдущих команд. Кроме того, можно группировать наборы команд, выполняемых по условию.

В следующих разделах вы подробнее ознакомитесь с этими способами. Однако, прежде чем приступить к их изучению, посмотрите табл. 2-3 с кратким описанием синтаксиса для создания цепочек и группирования команд. Учтите, что приведенный здесь синтаксис не претендует на то, чтобы охватить все случаи. Синтаксис создания цепочек можно расширить, добавив дополнительные команды, выполняемые по условию. Синтаксис группирования также может быть разным в зависимости от ситуации.

Табл. 2-3. Синтаксис для создания цепочек и группирования команд

Символ	Синтаксис	Описание
&	Команда1 & Команда2	Выполняется Команда1, затем Команда2
&&	Команда1 && Команда2	Команда2 выполняется, если Команда1 выполнена успешно
||	Команда1 || Команда2	Команда2 выполняется, если Команда1 не выполнена успешно
( )	(Команда1 & Команда2) && (Команда3)	Команда1 и Команда2 группируются с помощью скобок, а Команда3 выполняется в случае успешного завершения этих команд
	(Команда1 & Команда2) || (Команда3)	Команда1 и Команда2 группируются с помощью скобок, а Команда3 выполняется в случае неудачного завершения этих команд

Цепочки команд

Иногда для большей эффективности нужно выполнять команды в определенной последовательности. Например, перейти в определенный каталог и получить список файлов, отсортированный по дате. Создание цепочки позволяет решить эти две задачи, введя всего одну строку:

```
cd c:\working\docs & dir /O:d
```

Такие цепочки команд часто требуются в сценариях, чтобы быть уверенным, что команды выполняются именно так, как ожидается. Создание цепочек команд еще полезнее, когда запуск последующих команд зависит от того, как завершились предыдущие команды — успешно или неудачно. В следующем примере файл журнала перемещается в другой каталог, только если он существует:

```
dir c:\working\logs\current.log && move current.log
d:\history\logs
```

Зачем это может понадобиться? А чтобы при выполнении сценария не было ошибок, когда перемещать файл не нужно.

Иногда требуется выполнить какую-либо операцию, если предыдущая команда потерпела неудачу. Так, если вы с помощью сценария распространяете файлы среди группы рабочих станций, на одних из которых существует папка C:\Working\Data, а на других — папка C:\Data, то для копирования группы файлов в одну из этих папок независимо от конфигурации рабочей станции можно выполнить следующие команды:

```
cd C:\working\data || cd C:\data
xcopy n:\docs\*.*
```

Группирование последовательностей команд

При выполнении нескольких команд, чтобы избежать конфликтов между ними и обеспечить правильный порядок их выполнения, может потребоваться группирование. Для группирования команд используются скобки. Чтобы понять, зачем нужно группирование, рассмотрим пример. Предположим, вы хотите вывести в файл хост-имя, конфигурацию IP и сетевой статус. Наберите следующий оператор:

```
hostname & ipconfig & netstat -a > current_config.log
```

Однако, когда вы просмотрите файл журнала, окажется, что он содержит только сетевой статус. Это объясняется тем, что командная строка выполняет такую последовательность команд:

1. hostname
2. ipconfig
3. netstat -a > current_config.log

Поскольку команды выполняются последовательно, хост-имя системы и конфигурация IP выводятся в командной строке и только сетевой статус записывается в файл журнала. Чтобы записать вывод всех команд в файл, нужно сгруппировать их так:

```
(hostname & ipconfig & netstat -a) > current_config.log
```

Теперь вывод всех трех команд объединяется и перенаправляется в файл журнала. Группирование можно сочетать с выполнением по условию — в зависимости от успеха или неуда-

чи предыдущей команды. В следующем примере Команда3 выполняется, только если успешно выполнены Команда1 и Команда2:

```
(cd C:\working\data & xcopy n:\docs\*.*) && (hostname >
n:\runninglog.txt)
```

В следующей главе вы увидите, как использовать группирование команд в сочетании с конструкциями *if* и *if else*.

Глава 3

Основы сценариев командной строки

В мире, в котором доминируют сногсшибательные графические пользовательские интерфейсы (GUI), кто-то, наверное, удивится, что такого могут предложить сценарии командной строки, чего не могут Microsoft Windows и диалоговые окна в стиле «укажи и щелкни». Ну, честно говоря, больше, чем многим кажется, особенно если учесть, что большинство воспринимает сценарии командной строки как реинкарнацию командных файлов (batch files) — вроде тех, которыми вы пользовались на компьютерах с процессорами 8088 и MS-DOS. Современная среда сценариев командной строки предоставляет обширные возможности в программировании, в том числе поддерживая:

- переменные;
- арифметические выражения;
- операторы условий;
- операторы управления потоком выполнения;
- процедуры.

Эти программные элементы позволяют автоматизировать рутинные задачи, выполнять сложные операции в ваше отсутствие, находить ошибочно размещенные другими ресурсы и осуществлять многие другие действия, обычно требующие ввода с клавиатуры. Сценарии командной строки не только имеют полный доступ к командной строке, но и могут вызывать любые утилиты, предоставляющие расширения для командной строки, в том числе средства Windows Support Tools и Windows Resource Kit.

Создание сценариев командной строки

Сценарии командной строки — текстовые файлы с командами, которые вы хотите выполнить. Это те же команды, которые обычно вводятся в командной оболочке Windows. Однако, вместо того чтобы вводить команды каждый раз, когда они понадобятся, можно создать соответствующий сценарий и упростить себе жизнь.

Поскольку сценарии состоят из стандартных текстовых символов, их можно создавать и редактировать в любом стандартном текстовом редакторе, скажем, в Notepad (Блокнот). Вводя команды, убедитесь, что каждая команда или группа команд, которые должны выполняться совместно, размещаются с новой строки. Это обеспечит их корректное выполнение. Закончив создание сценария командной строки, сохраните файл сценария с расширением .bat или .cmd. Оба расширения работают одинаково. Например, если вам надо создать сценарий для вывода имени системы, версии Windows и конфигурации IP, включите в файл SysInfo.bat или SysInfo.cmd следующие три команды:

```
hostname
ver
ipconfig -all
```

Сохранив сценарий, его можно запустить так, будто это Windows-утилита: просто наберите имя сценария в командной оболочке и нажмите Enter. После этого командная оболочка считывает файл сценария и последовательно выполняет его команды. Выполнение сценария прекращается, как только достигается конец файла или встречается команда EXIT. На листинге 3-1 показан образец вывода, который дает сценарий, приведенный выше.

Листинг 3-1. Вывод сценария-примера

```
C:\>hostname
mailer1

C:\>ver
Microsoft Windows [Version 5.2.3790]

C:\>ipconfig -all
Windows IP Configuration
```

```
   Host Name . . . . . . . . . . . . : mailer1
   Primary Dns Suffix  . . . . . . . : adatum.com
   Node Type . . . . . . . . . . . . : Unknown
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : adatum.com

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/100 VE
      Network Connection
   Physical Address. . . . . . . . . : X0-EF-D7-AB-E2-1E
   DHCP Enabled. . . . . . . . . . . : No
   IP Address. . . . . . . . . . . . : 192.168.10.50
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.10.1
   DNS Servers . . . . . . . . . . . : 192.168.10.155
```

Изучив листинг, вы увидите, что вместе с выводом команд отображается приглашение командной строки и сами команды. Причина в том, что при стандартной обработке сценариев командная оболочка закулисно выполняет дополнительную работу. Прежде всего она показывает приглашение командной строки. Затем считывает строку из сценария, выводит ее и интерпретирует. Если командная оболочка достигает конца файла или встречает команду EXIT, выполнение прекращается. В ином случае командная оболочка повторяет этот процесс заново, отображая приглашение командной строки и готовясь считать следующую строку из сценария.

Хотя стандартный режим обработки с включенным эхо-отображением команд полезен при устранении проблем в сценариях, вам вряд ли захочется применять такой режим в регулярно используемых сценариях. К счастью, стандартное поведение можно изменить, выключив эхо-отображение, как я покажу в разделе «Управление отображением текста и команд» далее в этой главе.

Стандартные операторы и команды сценариев

До сих пор я говорил о командах, но не рассказывал о том, что такое операторы. Хотя эти термины часто взаимозаменяемы, термин *оператор* (statement) формально обозначает ключевое

слово для команды, как, например, оператор *rem*, но может обозначать и строку кода, включая весь текст команды в этой строке. В некоторых языках программирования, таких как Java, каждый оператор должен завершаться специальным символом. В Java символом завершения служит точка с запятой. Командная строка не требует специального символа завершения помимо символа конца строки, которым считается любой из следующих символов:

- разрыв строки (как при нажатии Shift+Enter);
- возврат каретки и разрыв строки (как при нажатии Enter);
- маркер конца файла (end-of-file marker).

Теперь, когда мы обсудили, как создавать сценарии, рассмотрим стандартные операторы и команды для использования в сценариях, которые включают:

- **cls** — очищает консольное окно и сбрасывает буфер экрана;
- **rem**— помечает комментарии в сценариях;
- **echo** — выводит сообщения в командной строке, а также включает и выключает эхо-отображение команд;
- **@** — управляет построчным эхоотображением команд;
- **title** — устанавливает заголовок окна командной оболочки;
- **color** — задает цвета текста и фона, используемые в окне командной оболочки.

Очистка окна командной оболочки

Обычно, прежде чем отображать вывод сценария, неплохо очистить окно командной оболочки. Это делается командой CLS. Почему бы не попробовать? В командной строке введите **cls** и нажмите Enter. Консольное окно очищается, а курсор перемещается в верхний левый угол окна*. Весь текст в буфере экрана тоже очищается.

Вы могли бы добавить команду CLS в показанный ранее пример сценария:

```
cls
hostname
ver
ipconfig -all
```

* При условии, что отключен вывод приглашения командной строки, иначе в верхний левый угол сначала выводится приглашение и только потом курсор. — *Прим. перев.*

Добавление комментариев в сценарии

Оператор *rem* позволяет добавлять комментарии в сценарий. В каждом создаваемом вами сценарии должны быть комментарии, поясняющие:

- когда сценарий был создан и в последний раз изменен;
- кто создал сценарий;
- для чего предназначен сценарий;
- как связаться с автором сценария;
- сохраняется ли вывод сценария и, если да, то где.

Ответы на эти вопросы важны не только для того, чтобы создаваемые вами сценарии могли использоваться другими администраторами; они также помогут вам вспомнить, что делает конкретный сценарий, особенно если с момента его последнего использования прошли недели или месяцы. Пример сценария с комментариями, отвечающими на эти вопросы, показан на листинге 3-2.

Листинг 3-2. Модифицированный пример сценария с комментариями

```
rem ************************
rem Сценарий: SystemInfo.bat
rem Дата создания: 2/2/2004
rem Последнее изменение: 3/15/2004
rem Автор: William R. Stanek
rem E-mail: williamstanek@aol.com
rem ************************
rem Описание: отображает информацию о конфигурации системы,
rem           в том числе имя системы, IP-конфигурацию
rem           и версию Windows
rem ************************
rem Файлы: сохраняет вывод в c:\current-sys.txt
rem ************************

hostname > c:\current-sys.txt
ver >> c:\current-sys.txt
ipconfig -all >> c:\current-sys.txt
```

В разделе «Передача аргументов в сценарии» далее в этой главе я покажу, как использовать комментарии в качестве автоматизированной справочной документации. Но, прежде чем мы перейдем к этому, запомните, что операторы *rem* пригодны и для:

- вставки в сценарии поясняющего текста, например описания работы процедуры;
- предотвращения выполнения команды. Добавьте rem перед командой, чтобы закомментировать ее;
- скрытия части строки от интерпретации. Добавьте rem в строке, чтобы заблокировать интерпретацию всего, что расположено после оператора *rem*.

Управление отображением текста и команд

Команда ECHO служит двум целям: для записи текста в вывод (например в окно командной оболочки или текстовый файл) и для включения/выключения эхо-отображения команд. Обычно при выполнении команд сценария сами команды и вывод этих команд отображаются в консольном окне. Это называется *эхо-отображением команд* (comand echoing).

Чтобы использовать команду ECHO для отображения текста, введите **echo** и текст, который надо вывести:

```
echo The system host name is:
hostname
```

Чтобы с помощью ECHO управлять эхо-отображением команд, введите **echo off** или **echo on**, например:

```
echo off
echo The system host name is:
hostname
```

Чтобы направить вывод в файл, а не в окно командной оболочки, используйте перенаправление вывода, например:

```
echo off
echo The system host name is: > current.txt
hostname >> current.txt
```

Теперь посмотрим, как подавляется эхо-отображение команд. Запустите командную оболочку, введите **echo off**, затем другие команды. Вы увидите, что приглашение командной строки больше не выводится. Вместо него появляется только то, что набирается в консольном окне, и вывод выполненных команд. В сценариях команда ECHO OFF отключает эхо-отображение команд и приглашение командной строки. Добавляя в свои сценарии команду ECHO OFF, вы предотвращаете загромождение окна командной оболочки или файла текстом команд, если вас интересует лишь вывод от этих команд.

Совет Кстати, если вы хотите узнать, включено ли эхо-отображение команд, просто введите команду ECHO. Попробуйте! Если эхо-отображение включено, вы увидите сообщение Echo Is On; в ином случае — Echo Is Off. Экспериментируя с командой ECHO OFF в своих сценариях, вы можете обнаружить небольшую проблему. Если команда ECHO OFF выключает эхо-отображение команд, то как же предотвратить эхо-отображение самой команды ECHO OFF? Не волнуйтесь: об этом рассказывается в следующем разделе.

Примечание Программисты часто задают мне вопрос: как вывести пустую строку в командной оболочке? Вероятно, вы подумали, что, поместив в отдельную строку команду ECHO, можно добиться нужного результата, но это не так. Ввод в строке одной команды **echo** сообщает о состоянии эхо-отображения команд, как я уже говорил. Ввод команды **echo** с пробелами после нее также не сработает, потому что командная строка Windows считает пробелы в данном случае ненужными и вы получите тот же результат, что и при вводе **echo** без пробелов. Чтобы ECHO отобразила пустую строку, эту команду следует ввести с точкой: **echo.** (Заметьте, что между командой ECHO и точкой не должно быть пробела.)

Управление эхо-отображением с помощью @

Команда @ предотвращает эхо-отображение команд в отдельной строке, и ее можно считать оператором *echo off* для конкретной строки. Вы можете использовать @ для отключения эхо-отображения команд так:

```
@echo The system host name is:
@hostname
```

С помощью @ вывод, показанный здесь:

```
C:\>echo The system host name is:
The system host name is:

C:\>hostname
mailer1
```

становится таким:

```
The system host name is:
mailer1
```

Но истинная ценность @ в том, что она сообщает командной оболочке не отображать приглашение командной строки или команду ECHO OFF и тем самым обеспечивает присутствие в выводе только результатов работы ваших сценариев. Вот пример сценария, использующего @ для скрытия команды ECHO OFF:

```
@echo off
echo The system host name is:
hostname
```

Этот сценарий выводит:

```
The system host name is:
mailer1
```

Совет Рекомендую использовать *@echo off* в начале всех сценариев командной строки. Кстати, если запустить командную оболочку и ввести **@echo off**, можно погасить и приглашение командной строки.

Настройка заголовка и цветов консольного окна

Если вы собираетесь потратить время на написание сценария, можете слегка «оживить» его. К числу базовых приемов, уже описанных мной, относятся применение команды ECHO OFF и очистка консольного окна перед записью вывода. Также можно установить заголовок окна или изменить используемые в окне цвета.

Строка заголовка командной оболочки расположена вверху консольного окна. Обычно там выводится «Command Prompt» («Командная строка») или путь к файлу командной оболочки. Заголовок можно настроить командой TITLE. Она работает почти так же, как ECHO: отображает любой текст, следующий за командой, в строке заголовка консоли. Например, если вы хотите установить для текущей консоли заголовок «System Information», то можете сделать это, введя в командной строке:

```
title System Information
```

Команда TITLE позволяет не только сообщать имя выполняемого сценария, но и отражать ход его обработки, например:

```
rem Блоки рабочих команд
title Gathering Information
```

```
rem Блоки команд протоколирования
title Logging System Information
```

По умолчанию консольное окно отображает белый текст на черном фоне. Как вы уже знаете из главы 1 «Обзор командной строки Windows», эти цвета можно изменить на вкладке Colors (Цвета) в диалоговом окне свойств командной строки. Также цвета консоли можно установить командой COLOR. Для этого ей передается двухразрядный шестнадцатеричный код. Первая цифра соответствует цвету фона, вторая — цвету текста, например:

```
color 21
```

устанавливает синий цвет текста на зеленом фоне.

Коды цветов, используемые в команде COLOR, показаны в табл. 3-1. Помните, что для цвета текста и цвета фона нельзя устанавливать одинаковые значения. Если вы попытаетесь сделать это, цвет не изменится. Кроме того, можно восстановить стандартные цвета командой COLOR без аргументов:

```
color
```

Табл. 3-1. Коды цветов для окна командной оболочки

Код	Цвет	Код	Цвет
0	Black (черный)	8	Gray (серый)
1	Blue (синий)	9	Bright Blue (ярко-синий)
2	Green (зеленый)	A	Bright Green (салатовый)
3	Aqua (бирюзовый)	B	Bright Aqua (голубой)
4	Red (красный)	C	Bright Red (ярко-красный)
5	Purple (фиолетовый)	D	Bright Purple (ярко-фиолетовый)
6	Yellow (желтый)	E	Bright Yellow (ярко-желтый)
7	White (белый)	F	Bright White (ярко-белый)

Передача аргументов в сценарии

Как и большинству утилит командной строки, сценариям при запуске можно передавать аргументы. Аргументы используются для установки специальных параметров в сценарии или для передачи сценарию необходимой информации. Каждый аргумент должен располагаться за именем сценария и отделяется пробелом (а при необходимости заключается в кавычки).

В следующем примере сценарию с именем Check-sys передаются параметры Mailer1 и Full:

```
check-sys mailer1 full
```

Каждое значение, передаваемое сценарию, может быть определено через формальные параметры. Имя самого сценария представляется параметром *%0*. Параметр *%1* представляет первый переданный сценарию аргумент, *%2* — второй и так далее до *%9* для девятого аргумента. Например, если вы создадите сценарий с именем Check-sys, а затем вызовете его командой:

```
check-sys mailer1 full actual
```

то обнаружите, что соответствующие значения параметров будут такими:

- *%0* — check-sys;
- *%1* — mailer1;
- *%2* — full;
- *%3* — actual.

Доступ к аргументам в сценариях выполняется по имени параметра: *%0* — для имени сценария, *%1* — для первого параметра сценария и т. д. Например, если хотите отобразить имя сценария и первый переданный ему аргумент, введите:

```
echo %0
echo %1
```

Если передать более девяти параметров, дополнительные параметры не теряются. Они сохраняются в специальном параметре: *%** (процент + звездочка). Параметр *%** представляет все аргументы, переданные сценарию, и команда SHIFT позволяет просмотреть дополнительные параметры. Если вызвать SHIFT без аргументов, параметры сценария сдвигаются на 1. То есть значение, представляемое *%0*, отбрасывается и заменяется значением, представляемым *%1*, а значение, представляемое *%2*, теперь представляется параметром *%1* и т. д. Также можно указать, где начинается сдвиг, чтобы при необходимости сохранить предшествующие параметры. Например, если указать:

```
shift /3
```

то *%4* станет *%3*, *%5* — *%4* и т. д. Но *%0*, *%1* и *%2* не изменятся.

Знакомство с переменными

То, что в сценариях командной строки мы обычно называем переменными, точнее было бы называть *переменными окружения* (environment variables). Переменные окружения могут поступать от многих источников. Некоторые встроены в операционную систему или формируются драйверами аппаратного обеспечения при загрузке системы. Эти переменные, называемые *встроенными системными* (built-in system variables), доступны всем Windows-процессам независимо от того, вошел ли кто-то в систему в интерактивном режиме. Системные переменные могут берутся из реестра Windows. Другие переменные устанавливаются при входе в систему и называются *встроенными пользовательскими* (built-in user variables). Встроенные пользовательские переменные одинаковы, кто бы ни вошел в систему. Как можно догадаться, они действительны только в течение рабочего сеанса, т. е. пока пользователь зарегистрирован в системе.

Чтобы увидеть список всех известных переменных в текущем экземпляре командной оболочки, введите в командной строке **set**. В дополнение к обычным системным и пользовательским переменным можно создавать собственные переменные, что вы и будете делать при программировании в командной строке. Переменные для текущего экземпляра командной оболочки определяются командой SET со следующим синтаксисом:

```
set имя_переменной=значение_переменной
```

Например:

```
set working=C:\Work\Data
set value=5
set string="Hello World"
```

Некоторые переменные, включая системные и пользовательские, имеют особый смысл для командной оболочки. К ним относятся *path*, *computername*, *homedrive* и многие другие важные переменные окружения. Одна из переменных окружения, о которой следует знать больше, — *errorlevel* — отслеживает код завершения (exit code) последней использованной команды. Если команда выполнена нормально, код завершения равен нулю. Если при выполнении команды возникла ошибка, код завершения получает соответствующее ненулевое значение. Коды ошибок включают:

- 1 — указывает на общую ошибку;
- 2 — указывает на ошибку выполнения, т. е. команда завершилась неудачей;
- –2 — указывает на математическую ошибку, когда, например, создается число, слишком большое для обработки в командной оболочке.

Работать с переменной *errorlevel* можно несколькими способами. Например, проверять конкретное условие ошибки:

```
if "%ERRORLEVEL%"=="2" echo "An error occurred!"
```

Или использовать специальный синтаксис и проверять условие, когда код завершения равен или больше указанного:

```
if errorlevel 2 echo "An error occurred!"
```

Примечание Подробнее об операторах *errorlevel* и *if* см. в разделе «Операторы выбора в командной строке» далее в этой главе.

По окончании работы с переменными хорошим стилем считается их очистка. Это делается для того, чтобы освободить память, занимаемую переменными, и избежать проблем или неожиданных результатов при случайной ссылке на такую переменную в будущем. Чтобы очистить переменную, просто присвойте ей пустое значение:

```
set working=
```

Теперь переменная удалена из памяти и больше не доступна.

Использование переменных в сценариях

В сценариях переменные используются для хранения значений при выполнении различных типов операций. В отличие от большинства языков программирования в сценарии командной строки нельзя объявить переменную, не присвоив ей значения. В этом есть смысл, поскольку с практической точки зрения пустая переменная не нужна. В следующих разделах поясняются ключевые концепции, необходимые для работы с переменными, в том числе:

- имена переменных;
- их значения;

- подстановка;
- область действия.

Именование переменных

Командная оболочка отслеживает имена переменных на случай, если вы используете буквы разных регистров, но не следите за регистром при работе с переменными. Это значит, что имена переменных не чувствительны к регистру букв, но допускают наличие букв разных регистров. Помимо этого на имена переменных накладывается очень мало ограничений, и вы можете использовать практически любое сочетание букв, цифр и символов для формирования имен. В принципе, все перечисленные ниже имена переменных формально допустимы:

```
2six
85
!
?
```

Но с какой стати может возникнуть желание использовать такие жуткие имена для переменных — это выше моего понимания. Так как же именовать переменные? Главное правило, о котором надо помнить, — имена переменных должны быть информативными, например:

```
System-name
CurrentStats
mergetotal
Net_Address
```

Такие информативные имена переменных пригодятся, когда вам или кому-то еще понадобится модифицировать сценарий. И заметьте, что есть много способов создания имен переменных из нескольких слов. Хотя вы вправе использовать любой стиль по своему усмотрению, большинство программистов формируют имена переменных из нескольких слов, делая первую букву в первом слове строчной, а первую букву в каждом последующем слове прописной. Почему? Причина проста: это стандартная схема именования. Согласно этой схеме, показанные выше имена переменных должны иметь такой вид:

```
systemName
currentStats
mergeTotal
netAddress
```

Примечание Помните, что командная оболочка не чувствительна к регистру букв. Имена переменных просто допускают буквы разных регистров. А значит, вы можете ссылаться на переменную *systemName* как на *SYSTEMNAME*, *systemname* или даже *sYStemNAMe*.

Присвоение значений переменным

Как я уже говорил, новые переменные определяются по следующему синтаксису:

```
set имя_переменной=значение_переменной
```

Пробелы допустимы и в именах, и в значениях, так что ставьте пробелы слева и/или справа от знака равенства, только если вы хотите, чтобы они входили в имя и/или значение переменной.

В отличие от многих языков программирования, командная оболочка игнорирует типы данных. Все переменные хранятся как символьные строки. Это так, даже если вы присваиваете переменной числовое значение. Таким образом, следующие значения сохраняются как строки:

```
Current status:
311
"Error!"
12.75
```

при помощи команд вроде:

```
set varA=Current status:
set varB=311
set varC="Error!"
set varD=12.75
```

Помните, что в командной строке некоторые символы зарезервированы, в том числе @ < > & | ^. Прежде чем использовать эти символы, их следует предварять символом ^, как сказано в главе 2 «Эффективная работа с командной строкой». Например, чтобы установить эти строковые значения:

```
2 & 3 = 5
2^3
```

следует присваивать значения переменным так:

```
2 ^& 3 = 5
2^^3
```

используя операторы вроде:

```
set example1=2 ^& 3 = 5
set example3=2^^3
```

Примечание При попытке эхо-отображения приведенных значений происходят странные вещи. Вместо ожидаемых уравнений вы получаете ошибку или непонятное значение. Дело в том, что при эхо-отображении значений специальные символы анализируются повторно. Если вы хотите присвоить переменной значение, включающее специальные символы и пригодное для отображения пользователю, следует помещать три escape-кода, т. е. писать так: set example1=2 ^^^& 3 = 5 или set example2=2^^^^3. Это необходимо потому, что значение анализируется дважды (один раз при присваивании и один — при отображении).

Подстановка значений переменных

Вряд ли от переменных был бы какой-нибудь толк, если бы единственным способом обращения к ним была команда SET. К счастью, получить доступ к значениям переменных можно и другими способами. Один из них — использовать подстановку переменных для сравнения имени переменной с ее действительным значением. Вы видели такой тип подстановки в одном из предыдущих примеров:

```
if "%ERRORLEVEL%"=="2" echo "An error occurred!"
```

Здесь определяется, содержит ли переменная *errorlevel* значение 2, и, если да, отображается текст, сообщающий об ошибке. Знаки процента, окружающие имя переменной, сообщают командной оболочке, что вы ссылаетесь на переменную. Без этих знаков Windows выполнила бы символьное сравнение «ERRORLEVEL» и «2». Также обратите внимание на использование в примере символов кавычек. Символы кавычек гарантируют точное сравнение строковых значений.

Другой способ подстановки — замена имени переменной ее реальным значением. Например, если у вас возникнет желание написать сценарий, который можно запускать на разных компьютерах, то вместо вписывания пути к корневому системному каталогу вроде C:\Windows используйте переменную окружения *systemroot*, которая ссылается на корневой системный каталог на данном компьютере. То есть пишите:

```
cd %SYSTEMROOT%\System32
```

вместо:

```
cd C:\Windows\System32
```

Подстановка возможна и при присвоении значений переменным, например:

```
systemPath=%SystemRoot%\System32
```

Подстановка переменных вполне эффективна. Рассмотрим фрагмент кода, приведенный на листинге 3-3.

Листинг 3-3. Заголовок сценария-примера

```
@echo off
@if not "%OS%"=="Windows_NT" goto :EXIT
@if"%1"=="" (set INFO=echo && set SEXIT=1) else (
set INFO=rem && set SEXIT=0)

%INFO% ************************
%INFO% Сценарий: SystemInfo.bat
%INFO% Дата создания: 2/2/2004
%INFO% Последнее изменение: 3/15/2004
%INFO% Автор: William R. Stanek
%INFO% E-mail: williamstanek@aol.com
%INFO% ************************
%INFO% Описание: отображает информацию о конфигурации
%INFO% системы, в том числе имя системы, IP-конфигурацию
%INFO% и версию Windows
%INFO% ************************
%INFO% Файлы: сохраняет вывод в c:\current-sys.txt
%INFO% ************************

@if "%SEXIT%"=="1" goto :EXIT

@title "Configure Scheduling..."
cls
color 07
```

Листинг 3-3 — стандартный заголовок, который я применяю в некоторых своих сценариях. Первый оператор *if* проверяет версию операционной системы. Если это Windows 2000 или более поздняя, т. е. Windows 2000, Windows XP или Windows Server 2003, сценарий продолжает работу. Иначе вызывается подпрограмма *goto*. Второй оператор *if* проверяет значение первого аргумента, переданного сценарию. Если сцена-

рий вызывается без аргументов, экземпляры *%INFO%* заменяются на *echo*, что приводит к записи в вывод документации сценария. Если сценарий вызывается с одним или несколькими аргументами, экземпляры *%INFO%* заменяются на *rem*, чтобы обозначить соответствующие строки как комментарии.

Примечание Не волнуйтесь, если пример вам понятен не полностью. Вы все узнаете о выполнении по условию и о подпрограммах в разделах «Операторы выбора в командной строке» и «Создание подпрограмм и процедур» далее в этой главе.

Локализация области действия переменных

Изменения, вносимые в переменные в командной оболочке с помощью *set*, локализованы, т. е. применяются только к текущему экземпляру командной оболочки или к командным оболочкам, запущенным из текущей командной оболочки (вложенным командным оболочкам). Эти изменения недоступны другим системным процессам. После выхода из командной оболочки, в котором были созданы переменные, они перестают существовать.

Иногда нужно ограничить область действия (видимости) переменных еще больше, чем просто текущим процессом их командной оболочки. Для этого в сценарии создается *локальная область действия* (local scope), гарантирующая, что все изменения переменной локализуются в этой особой области сценария. Далее локальная область действия завершается и восстанавливаются исходные значения переменных.

Начало локальной области действия в сценарии отмечается командой SETLOCAL, а конец — командой ENDLOCAL. При использовании этих команд происходит несколько событий. Вызов SETLOCAL создает статический набор (snapshot) переменных окружения. Любые изменения, сделанные в этой области, теперь локализуются и отбрасываются при вызове ENDLOCAL. Вот пример использования SETLOCAL и ENDLOCAL:

```
@echo off
set sysCount=0
set deviceCount=0
```

```
rem Начало локализации
setlocal
set sysCount=5
set deviceCount=5
echo Local count: %sysCount% system edits ^& %deviceCount%
device checks
endlocal

echo Count: %sysCount% system edits ^& %deviceCount% device
checks
```

Вывод этого сценария будет таким:

```
Local count: 5 system edits & 5 device checks
Count: 0 system edits & 0 device checks
```

Как видите, локальные области действия во многом похожи на вложенные командные оболочки. Как и в случае вложенных командных оболочек, вы можете создавать несколько вложенных уровней локализации. И хотя каждый уровень наследует параметры окружения своего предка, любые изменения во вложенном уровне не затрагивают окружение предка.

Применение математических выражений

Время от времени требуется выполнить в сценарии какие-либо математические операции и присвоить результат переменной. Как и большинство языков программирования, командная оболочка позволяет писать математические выражения с использованием множества операторов, включая:

- арифметические операторы для выполнения стандартных математических операций (таких как сложение, вычитание, умножение и деление);
- операторы присваивания, сочетающие присваивание (символизируемое знаком равенства) с арифметической операцией;
- операторы сравнения, которые сравнивают значения и обычно используются вместе с операторами *if*;
- побитовые (bitwise) операторы, позволяющие манипулировать последовательностями двоичных значений.

Математические операции выполняются через set с параметром */A* (арифметика), например:

```
set /a theTotal=18+2
set /a theTotal=18*2
set /a theTotal=18/2
```

Все математические выражения вычисляются с применением 32-разрядной целочисленной арифметики со знаком. Это позволяет работать со значениями от -2^{32} до $2^{32}-1$. При выходе за этот диапазон вместо нужного значения вы получите арифметическую ошибку (код –2).

Чаще всего используются арифметические операторы, операторы присваивания и сравнения. Первые два вида операторов описываются в следующих разделах, а операторы сравнения — в разделе «Сравнения в операторах If» далее в этой главе. Обратите особое внимание на дополнительную информацию о приоритете операторов и имитации в сценариях операций возведения в степень.

Арифметические операторы и операторы присваивания

Арифметические операторы служат для выполнения основных математических операций над числовыми значениями. Эти значения могут выражаться буквально как числа, скажем, 5 или как переменные, содержащие значения, например *%TOTAL%*.

В табл. 3-2 собраны все доступные арифметические операторы и операторы присваивания. Большинство арифметических операторов вполне очевидно. С помощью «*» выполняется умножение, «/» — деление, «+» — сложение, а «–» — вычитание. Знак равенства (=) присваивает значения переменным. Оператор % (модуль) позволяет определить остаток от деления. Например, если разделить 60 на 8, то получится 7 с остатком 4; значение 4 и будет результатом от применения этого оператора.

Вот примеры работы с арифметическими операторами:

```
set /a theCount=5+3

set /a theCount=%nServers% + %nWstations%

set /a theCount=%nServers% - 1
```

Совет Ранее я сказал, что в переменной все хранится в виде строки, и это по-прежнему так. Однако командная оболочка умеет определять, когда строка содержит только цифры, и это позволяет использовать переменные в выражениях. Главное, помните о правильном синтаксисе подстановки — *%имяПеременной%*.

Табл. 3-2. Операторы арифметики и присваивания

Арифметические операторы	Операторы присваивания
+ (сложение)	+= (увеличение, т. е. сложение и присваивание)
- (вычитание)	-= (уменьшение, т. е. вычитание и присваивание)
* (умножение)	*= (увеличение масштаба, т. е. умножение и присваивание)
/ (деление)	/= (уменьшение масштаба, т. е. деление и присваивание)
% (модуль)	%= (деление по модулю и присваивание)

Операторы присваивания используются для увеличения, уменьшения и масштабирования. Эти операторы сочетают в себе арифметические функции и присваивание. Так, оператор += используется для увеличения значения и сочетает действия оператора + и оператора =. Таким образом, следующие два выражения эквивалентны и при вводе в командной строке дают одинаковые результаты:

```
set /a total=total+1
set /a total+=1
```

Приоритет операторов

Используя математические операции, вы должны понимать приоритет операторов; он определяет, что происходит, когда командной оболочке надо вычислить выражение, включающее более одного оператора. Например:

```
set /a total=8+3*4
```

Если вычислять слева направо, выражение равно 44 (8+3=11, 11*4=44). Но, как и в обычной математике, в командной строке выражение вычисляется не так. Вместо этого командная оболочка дает результат 20 (3*4=12, 8+12=20), поскольку приоритет операторов следующий:

1. Модуль.
2. Умножение и деление.
3. Сложение и вычитание.

Примечание Если выражение содержит несколько операций одинакового приоритета, эти операции выполняются слева направо. Таким образом, set /a total=10-4+2 даст результат 8 (10–4=6, 6+2=8).

Однако, чтобы числа обрабатывались в определенном порядке, можно применять группирование скобками. То есть вы можете использовать выражение:

```
set /a total=(8+3)*4
```

чтобы командная строка интерпретировала его как (8+3=11, 11*4=44).

Имитация возведения в степень

Хотя в командной строке допустимы многие математические операции, возводить значения в степень нельзя. Но вы можете выполнять эти операции вручную. Например, простейший способ получить значение 2^3 — ввести:

```
set /a total=2*2*2
```

Результат — 8. Так же можно получить значение 10^5, введя:

```
set /a total=10*10*10*10*10
```

Результат — 100 000.

Операторы выбора в командной строке

Теперь, когда вы знаете, как работать с переменными и составлять выражения, обратимся к чему-нибудь посложнее: операторам выбора, используемым в командной строке. При необходимости управлять потоком выполнения на основе параметров, известных только в период выполнения, используются:

- *if* — для выполнения оператора, когда условие истинно (true), например, если операционная система соответствует Windows 2000 или выше. В ином случае оператор пропускается;
- *if not* — для выполнения оператора, когда условие ложно (false), например, если в системе отсутствует каталог C:\Windows. В ином случае оператор пропускается;

- *if...else* — для выполнения оператора при совпадении условия (истинно или ложно) и выполнения второго оператора в ином случае.

Хотя в некоторых предыдущих примерах из этой главы встречалось выполнение по условию, мы не обсуждали синтаксис этих операторов и соответствующие операторы сравнения. Если у вас нет опыта программирования, вы, наверное, удивитесь эффективности и гибкости этих операторов.

Применение If

Оператор *if* используется для ветвления по условию. Его основной синтаксис:

```
if условие (оператор1) [else (оператор2)]
```

Здесь каждый оператор внутри скобок может быть одной командой или несколькими, объединенными в цепочку (chained), конвейер (piped) или группу (grouped). Условие — это любое выражение, возвращающее при оценке булево значение True или False. Блок *else* не обязателен, т. е. можно использовать синтаксис:

```
if условие (оператор)
```

Совет Формально скобки не обязательны, но лучше их ставить, особенно если условие включает оператор *echo* или команду с параметрами. Если не использовать скобки в таких случаях, то все, что идет за оператором в этой строке будет интерпретировано как часть оператора, что обычно приводит к ошибке.

Оператор *if* работает так: если *условие* — True, выполняется *оператор1*. В ином случае выполняется *оператор2* (если он указан). Ни при каких обстоятельствах оба раздела *if* и *else* не выполняются вместе. Рассмотрим пример:

```
if "%1"=="1" (echo is one) else (echo is not one)
```

Здесь, если первый переданный сценарию параметр — это 1, то в вывод записывается «is one». Иначе в вывод записывается «is not one».

Командная оболочка ожидает наличия лишь одного оператора после каждого условия. Обычно оператором является одна команда. Если вы хотите выполнить несколько команд, следует воспользоваться одним из способов объединения — кон-

вейеризацией, созданием цепочки или группированием команд, как в примере ниже:

```
if "%1"=="1" (hostname & ver & ipconfig /all) else (netstat -a)
```

Если значение первого параметра равно 1, будут выполнены все три команды в скобках.

Применение If Not

Для выполнения оператора только в том случае, когда условие ложно, используйте *if not*. Вот основной синтаксис:

```
if not условие (оператор1) [else (оператор2)]
```

Здесь командная оболочка вычисляет *условие*. Если результат — False, командная оболочка выполняет оператор. В ином случае команда не выполняется, и командная оболочка переходит к следующему оператору. Раздел *else* не обязателен, т. е. допустим и такой синтаксис:

```
if not условие (оператор1)
```

Возьмем пример:

```
if not errorlevel 0 (echo An error has occurred!) & (goto :EXIT)
```

Здесь проверяется, отлично ли от нуля состояние ошибки. Если ошибки не было (т. е. код завершения равен нулю), командная оболочка переходит к следующему оператору. Иначе она записывает в вывод «An error has occurred!» и выходит из сценария. (Вы узнаете все о *goto* и подпрограммах далее в этой главе.)

Применение If Defined и If Not Defined

Последние типы операторов *if*, доступных для использования, — это *if defined* и *if not defined*. Они созданы для проверки наличия переменных; их синтаксис выглядит соответственно:

```
if defined переменная оператор
```

и

```
if not defined переменная оператор
```

Для сценариев командной строки полезны оба оператора. В первом случае команда выполняется, если указанная переменная существует, во втором — если ее нет. Рассмотрим пример:

```
if defined numServers (echo Servers: %numServers%)
```

Если переменная *numServers* определена, сценарий записывает вывод. Иначе сценарий переходит к следующему оператору.

Вложенные условия

Вложенное условие — это оператор *if* в другом операторе *if*. Вложенные условия весьма распространены в программировании, и программирование в командной оболочке не исключение. Вкладывая операторы *if*, обратите внимание на следующее.

1. Используйте скобки для определения блоков кода и символ @, отмечающий начало вложенного оператора *if*.
2. Помните, что оператор *else* всегда ссылается на ближайший оператор *if*, который находится в том же блоке, что и оператор *else*, и не сопоставлен другому оператору *else*.

   Вот пример:

```
if "%1"=="1" (
@if "%2"=="2" (hostname & ver) else (ver)) else (hostname
& ver & netstat -a)
```

Первый оператор *else* сопоставлен с if "%2"=="2". Последний оператор *else* сопоставлен с if "%1"=="1".

Сравнения в операторах If

Часто выражения, используемые для управления операторами *if*, будут включать операторы сравнения, как показано в предыдущих примерах. Наиболее распространено сравнение двух строк с помощью оператора равенства (=):

```
if stringA==stringB оператор
```

Здесь осуществляется литеральное сравнение строк, и, если они идентичны, выполняется оператор. Такой синтаксис подходит для литеральных строк, но не идеален для сценариев. Параметры и аргументы могут содержать пробелы, или у переменной может вовсе не быть значения. В таких случаях при литеральном сравнении возможна ошибка. Поэтому используйте двойные кавычки, чтобы выполнять строковое сравнение и избежать большинства ошибок, например:

```
if "%varA%"=="%varB%" оператор
```

или

```
if "%varA%"=="string" оператор
```

Строковые сравнения всегда чувствительны к регистру букв, если не указано обратное ключом */i*. Ключ */i* предписывает командной оболочке игнорировать регистр букв при сравнении, и он используется так:

```
if /I "%1"=="a" (echo A) else (echo is not A)
```

Для более сложных проверок вам потребуются операторы сравнения, показанные в табл. 3-3. Эти операторы используются по аналогии со стандартным оператором проверки равенства (equ):

```
if "%varA%" equ "%varB" (echo The values match!)
```

Табл. 3-3. Операторы сравнения

Оператор	Описание
equ	«Равно». Дает True, если значения равны
neq	«Не равно». Дает True, если значения не равны
lss	«Меньше». Дает True, если *значение1* меньше, чем *значение2*
leq	«Меньше или равно». Дает True, если *значение1* равно или меньше, чем *значение2*
gtr	«Больше». Дает True, если *значение1* больше, чем *значение2*
geq	«Больше или равно». Дает True, если *значение1* равно или больше, чем *значение2*

Операторы циклов в командной строке

Оператор *for* позволяет выполнить команду или набор команд несколько раз. Это мощная конструкция, и, прежде чем пропустить данный раздел, полагая, будто вы знаете, как работает оператор *for*, подумайте еще разок. Оператор *for* создан специально для работы в среде командной оболочки и сильно отличается от любых других операторов *for*, с которыми вам, наверное, доводилось работать в других языках программирования. В отличие от большинства других операторов *for* этот оператор командной строки помогает перебирать группы файлов и каталогов и построчно анализировать текстовые файлы, вывод команд, а также строки.

Базовые сведения о циклах

В командной оболочке несколько форм операторов *for*. И все же основной формой для всех операторов *for* является:

```
for итератор do (оператор)
```

Здесь итератор управляет выполнением цикла *for*. Для каждого шага или элемента итератора обрабатывается указанный *оператор*. Этот оператор может быть одной или несколькими командами, объединенными в цепочку, конвейер или группу (внутри скобок).

Итератор обычно состоит из инициализирующей переменной и набора перечисляемых элементов, таких как группа файлов или диапазон значений. Инициализирующие переменные, по сути, являются полями подстановки для значений, с которыми вы хотите работать. Имея дело с инициализирующими переменными, помните следующее:

- переменные итератора существуют только в контексте цикла *for*;
- имена переменных итератора должны лежать в диапазоне от a до z или от A до Z, например *%%A*, *%%B* или *%%C*;
- имена переменных итератора чувствительны к регистру букв, т. е. *%%a* отличается от *%%A*.

Как показано в табл. 3-4, с операторами *for* используются разные структуры, каждая из которых рассчитана на свое применение. При инициализации оператора *for*, переменные итератора, такие как *%%B*, заменяются их действительными значениями. Эти значения берутся из набора элементов, указанного в операторе *for*, и могут состоять из списка файлов, списка каталогов, диапазона значений и т. п.

Табл. 3-4. Формы циклов

Цель	Синтаксис формы
Наборы файлов	for *%%переменная* in (*наборФайлов*) do *оператор*
Наборы каталогов	for /D *%%переменная* in (*наборКаталогов*) do *оператор*
Файлы в подкаталогах	for /R [*путь*] *%%переменная* in (*наборФайлов*) do *оператор*
Перебор наборов значений	for /L *%%переменная* in (*диапазонПеребора*) do *оператор*
Анализ текстовых файлов, строк и вывода команд	for /F ["*параметры*"] *%%переменная* in (*источник*) do *оператор*

Примечание Представленные формы предназначены для сценариев. Вы также можете использовать операторы *for* интерактивно из командной строки. В таком случае применяйте синтаксис *%переменная* вместо *%%переменная*. Помимо этого, операторы *for* и в сценариях, и в командной строке обрабатываются совершенно одинаково.

Перебор наборов значений

«Традиционный» способ применения операторов *for* заключается в переборе набора значений и выполнения задач с использованием этих значений. Это можно сделать и в командной оболочке. Основной синтаксис цикла *for* этой формы таков:

```
for /l %%переменная in (начало,шаг,конец) do (оператор)
```

Этот тип оператора *for* работает так. Сначала командная оболочка инициализирует внутренние переменные *начало*, *шаг* и *конец* указанными вами значениями. Затем сравнивает начальное значение с конечным значением, определяя, следует ли выполнять оператор. Положительное решение на этот счет принимается, если начальное значение может быть увеличено или уменьшено в соответствии с шагом; в ином случае принимается отрицательное решение. В случае положительного решения командная оболочка выполняет оператор, используя начальное значение, затем увеличивает или уменьшает начальное значение на указанную величину шага, после чего повторяет весь процесс. При отрицательном решении командная оболочка выходит из оператора *for*, переходя к следующему оператору сценария.

Рассмотрим следующий пример, где идет счет от 0 до 10 с шагом 2:

```
for /l %%B in (0,2,10) do echo %%B
```

Его вывод:

```
0
2
4
6
8
10
```

Кроме того, можно использовать отрицательную величину шага для перебора диапазона уменьшающихся значений. Вот счет от 10 до 0 через –2:

```
for /l %%B in (10,-2,0) do echo %%B
```

Вывод:

```
10
8
6
4
2
0
```

Перебор групп файлов

Более эффективный способ применения операторов *for* в командной оболочке — работа с файлами и каталогами. Соответствующий синтаксис оператора *for* для обработки групп файлов выглядит так:

```
for %%переменная in (наборФайлов) do (оператор)
```

Переменная *наборФайлов* может быть:

- отдельным файлом, указанным по имени, например MyFile.txt;
- группой файлов, заданной шаблоном, например *.txt;
- несколькими файлами или группами файлов, разделенными пробелами, например *.txt *.rtf *.doc.

Теперь, зная основные правила, вам будет легко работать с файлами. Так, чтобы перечислить все текстовые файлы в каталоге приложения, можно использовать в сценарии следующую команду:

```
for %%B in (C:\Working\*.txt) do (echo %%B)
```

Здесь *B* — инициализирующая переменная, C:\Working*.txt указывает, что вы хотите работать со всеми текстовыми файлами в каталоге C:\Working, а исполняемый оператор *echo %%B* предписывает командной оболочке отображать текущее значение *%%B* при каждой итерации цикла *for*. В результате в вывод записывается список текстовых файлов каталога.

Этот пример можно расширить, чтобы просматривать все файлы .txt, .rtf и .doc:

```
for %%B in (%AppDir%\*.txt %AppDir%\*.rtf %AppDir%\*.doc) do
(echo %%B)
```

Вы также вправе задействовать несколько команд путем объединения их в цепочку, конвейеризации и группирования:

```
for %%B in (%AppDir%\*.txt %AppDir%\*.rtf %AppDir%\*.doc) do
(echo %%B & move C:\Data)
```

Здесь в каталоге, указанном переменной *AppDir*, перечисляются файлы .txt .rtf и .doc, которые одновременно перемещаются в каталог C:\Data.

Перебор каталогов

Если вместо файлов вам нужно работать с каталогами, используйте следующую форму оператора *for*:

```
for /d %%переменная in (наборКаталогов) do (оператор)
```

Здесь *наборКаталогов* указывает набор каталогов, с которым вы собираетесь работать. Перечисление каталогов выполняется точно так же, как и перебор файлов, за исключением того, что вы задаете пути к каталогам, а не к файлам. Если вы хотите перечислить все базовые каталоги в *%SystemRoot%*, сделайте так:

```
for /d %%B in (%SystemRoot%\*) do echo %%B
```

В Windows Server 2003 часть полученного списка будет выглядеть так:

```
C:\Windows\AppPatch
C:\Windows\Cluster
C:\Windows\Config
C:\Windows\Cursors
C:\Windows\Debug
```

Примечание Обратите внимание, что в цикле *for /d* перебирается указанный набор каталогов, но подкаталоги этих каталогов не включаются. Для доступа к подкаталогам (и вообще ко всей структуре дерева каталогов) предназначены циклы *for /r*, о которых я вскоре расскажу.

Вы можете задать несколько базовых каталогов, разделив их имена пробелами:

```
for /d %%B in (%SystemRoot% %SystemRoot%\*) do echo %%B
```

Здесь просматривается сам каталог *%SystemRoot%*, а затем каталоги на один уровень ниже. Так что теперь список каталогов начнется с C:\Windows (если это системный корневой каталог) и продолжится каталогами, приведенными выше.

Командная оболочка допускает совмещение перебора файлов и каталогов с выполнением действий над всеми файлами в наборе каталогов, например:

```
for /d %%B in (%APPDATA% %APPDATA%\*) do (
@for %%C in ("%%B\*.txt") do echo %%C)
```

Первый оператор *for* возвращает список каталогов верхнего уровня в *%APPDATA%*, куда входит и сам *%APPDATA%*. Второй оператор *for* перебирает все файлы .txt в каждом из этих каталогов. Обратите внимание на символ @ перед вторым оператором *for*. Как и в случае с операторами *if*, это значит, что второй оператор *for* является вложенным. Двойные кавычки в наборе файлов ("%%B*.txt") гарантируют правильную обработку имен с пробелами.

Поскольку наряду с каталогами часто требуется работать с подкаталогами, командная оболочка предоставляет оператор *for /r*. Он позволяет просматривать все дерево каталогов от стартовой точки, указанной как путь. Вот его синтаксис:

```
for /r [путь] %%переменная in (наборФайлов) do (оператор)
```

Здесь путь отмечает базу дерева каталогов, с которым вы хотите работать, например C:\. Однако путь не обязателен, и, если он не указан, в качестве базы считается текущий рабочий каталог.

С помощью оператора *for /r* можно расширить предыдущий пример для перечисления всех файлов .txt на диске C: и обойтись без двойного цикла *for*:

```
for /r C:\ %%B in (*.txt) do echo %%B
```

Как видите, оператор *for /r* проще и эффективнее двойного цикла *for*. Кроме того, разрешается использовать комбинацию */r* и */d*. Ниже создается список всех каталогов и подкаталогов в *%SystemRoot%*:

```
for /r %SystemRoot% /d %%B in (*) do echo %%B
```

Анализ содержимого файлов и вывода

С содержимым файлов и выводом команд можно работать точно так же, как с именами файлов и каталогов. Для этого применяется следующая форма оператора *for*:

```
for /f ["параметры"] %%переменная in (источник) do (оператор)
```

Здесь *параметры* определяют параметры поиска текста, *источник* указывает источник текста, которым может быть текстовый файл, строка или вывод команды, а *оператор* задает команды, выполняемые над найденным текстом. Каждая строка текста в источнике обрабатывается как запись, поля которой разделяются специальным символом, например табулятором или пробелом (это стандартные разделители). Затем, используя подстановку, командная оболочка заменяет переменные полей подстановки в операторе на реальные значения.

Рассмотрим следующую строку текста из файла-источника:

```
William Stanek Engineering Williams@adatum.com 3408
```

Один из способов представления этой строки текста — запись с пятью полями:

- **Имя** William
- **Фамилия** Stanek
- **Отдел** Engineering
- **Адрес электронной почты** Williams@adatum.com
- **Добавочный телефон** 3408

Для синтаксического разбора этой и подобных строк в соответствующем файле годится следующий оператор *for*:

```
for /f "tokens=1-5" %%A in (current-users.txt) do (
@echo Имя и фамилия: %%A %%B Отдел: %%C E-mail: %%D Доб. тел.:
%%E)
```

Здесь указывается, что нужно работать с первыми пятью полями (лексемами, по умолчанию разделенными пробелами или табуляторами) и сопоставить их с переменными итератора, начиная с *%%A*, т. е. первое поле — *%%A*, второе — *%%B* и т. д. Итоговый вывод будет выглядеть так:

```
Имя и фамилия: William Stanek Отдел: Engineering E-Mail:
Williams@adatum.com Доб. тел.: 3408
```

В табл. 3-5 приведен полный список доступных параметров с примерами.

Табл. 3-5. Параметры анализа содержимого файлов и вывода команд

Параметр	Описание	Пример	Описание примера
eol	Устанавливает признак концевого комментария (end-of-line comment character). Все, что находится за ним, считается комментарием	`"eol=#"`	Задает # в качестве признака концевого комментария
skip	Указывает число строк, пропускаемых от начала файлов	`"skip=5"`	Сообщает командной оболочке пропустить в файле-источнике строки с первой по пятую
delims	Задает разделители полей. По умолчанию — пробел и табулятор	`"delims=, . :"`	Указывает, что запятые, точки и двоеточия являются разделителями
tokens	Указывает, какие поля лексем (token fields) использовать в каждой исходной строке. Можно указать до 26 лексем, если первой переменной итератора назначить а или A. По умолчанию анализируется только первая лексема	`"tokens=1,3"` `"tokens=2-5"`	В первом примере указываются поля 1 и 3, а во втором — поля 2, 3, 4 и 5
usebackq	Определяет возможность использования символов кавычек в указателе источника (source designator): двойных для имен файлов, обратных (') для исполняемых команд и одинарных для литеральных строк	`"usebackq"`	—

Чтобы понять, как пользоваться дополнительными параметрами, разберем следующий пример:

```
for /f "skip=3 eol=; tokens=3-5" %%C in (current-users.txt) do (
@echo Отдел: %%C E-mail: %%D Доб. тел.: %%E)
```

Здесь задействовано три параметра. Параметр *skip* заставляет пропустить первые три строки файла. Параметр *eol* указывает в качестве признака концевого комментария точку с запятой (;). Наконец, параметр *tokens* предписывает поместить лексемы с третьей по пятую в переменные итератора, начиная с *%%C*.

Указывать нужные поля с помощью лексем можно по-разному. Вот несколько примеров:

- **tokens=2,3,7** — поля 2, 3 и 7;
- **tokens=3-5** — поля 3, 4 и 5;
- **tokens=*** — строка просматривается целиком без разбиения на поля.

Учтите, что все пустые строки текстовых файлов пропускаются и что можно указать несколько файлов-источников, используя шаблоны или вводя список имен файлов, разделенных пробелами, например:

```
for /f "skip=3 eol=; tokens=3-5" %%C in (data1.txt data2.txt)
do (
@echo Отдел: %%C E-mail: %%D Доб. тел.: %%E)
```

Если имя файла содержит пробелы или если вы хотите выполнить команду, укажите параметр *usebackq* и кавычки, например:

```
for /f "tokens=3-5 usebackq" %%C in ("user data.txt") do (
@echo Отдел: %%C E-mail: %%D Доб. тел.: %%E)
```

или

```
for /f "tokens=3-5 usebackq" %%C in (`type "user data.txt"`)
do (
@echo Отдел: %%C E-mail: %%D Доб. тел.: %%E
```

Совет Запомните, что обратные кавычки (`) используются для команд, а одинарные (') — для строковых литералов. Конечно, на бумаге эти символы очень похожи. Однако на стандартной клавиатуре обратная кавычка (`) находится на одной клавише с тильдой (~), а одинарная (') — на одной клавише с двойной (").

Примечание Во втором примере я записывал содержимое файла в стандартный вывод командой TYPE, просто чтобы проиллюстрировать применение обратных кавычек.

Кстати, о кавычках. Они применяются при обработке строк и значений переменных. При этом строка или имя переменной заключается в двойные кавычки, чтобы обеспечить их правильную обработку. Для этого параметр *usebackq* не требуется.

Рассмотрим следующий пример:

```
set value=All,Some,None
for /f "delims=, tokens=1,3" %%A in ("%VALUE%") do (echo %%A
%%B)
```

Вот его вывод:

```
All None
```

Создание подпрограмм и процедур

Обычно командная оболочка Windows исполняет сценарии построчно от начала файла и до конца. Порядок выполнения можно изменить, и для этого применяются:

- **подпрограммы** — при вызове подпрограммы осуществляется переход на метку (label) в текущем сценарии, и выполнение продолжается до конца файла;
- **процедуры** — при обращении к процедуре вызывается другой сценарий, который выполняется от начала до конца, а затем управление возвращается в исходный сценарий на следующую строку после оператора вызова.

Как видите, основное различие между подпрограммой и процедурой в том, что именно вы хотите сделать. Переданные в сценарий аргументы в подпрограммах *goto* доступны напрямую, а список аргументов в вызванной процедуре меняется, и в аргументе 0 (*%0*) содержится имя процедуры вместо имени сценария.

Подпрограммы

Подпрограммы состоят из двух частей:

- вызова *goto*, указывающего подпрограмму, в которую надо перейти;
- метки, обозначающей начало подпрограммы.

Рассмотрим следующий вызов подпрограммы:

```
if "%1"=="1" goto SUB1
```

Здесь, если первым параметром в сценарий передается 1, вызывается подпрограмма с именем *SUB1* и командная оболочка должна перейти на соответствующую метку подпрограммы. Чтобы создать метку, введите на отдельной строке двоеточие и имя метки:

```
:SUB1
```

Хотя метки могут состоять практически из любых допустимых символов, обычно используют алфавитно-цифровые символы, поскольку так легче читать метки при разборе кода.

При использовании *goto* выполнение сценария продолжается со строки, следующей за указанной меткой, и заканчивается в конце файла, если по пути не придется обрабатывать вызовы процедур или дополнительные операторы *goto*. Если метка в сценарии находится выше текущей позиции, командная оболочка может вернуться к предыдущему участку сценария. Так можно создать бесконечный цикл (в отсутствие управляющей конструкции, которая пропускает оператор *goto*). Вот пример бесконечного цикла:

```
:START
.
.
.
goto START
```

Если метка находится после оператора *goto*, можно пропустить команды и перейти вперед к новому разделу сценария, например:

```
goto MIDDLE
.
.
.
:MIDDLE
```

Здесь происходит переход на метку :MIDDLE, и выполнение продолжается до конца файла. Вернуться обратно к невыполненным командам можно только с помощью другого оператора *goto*.

Иногда выполнять оставшуюся часть сценария не требуется, и после выполнения операторов подпрограммы следует выйти из сценария. Для этого создайте метку выхода и в конце подпрограммы перейдите на эту метку:

```
goto MIDDLE
.
.
.
:MIDDLE
.
.
.
goto EXIT
.
.
.
:EXIT
```

На листинге 3-4 показан детальный пример работы с *goto* и метками. В этом примере значение первого параметра сценария определяет исполняемую подпрограмму. Первый оператор *if* обрабатывает ситуацию, когда параметр не передан. Он отображает сообщение об ошибке и завершает сценарий. Оператор *goto EXIT*, расположенный после операторов *if*, обрабатывает ситуацию, когда передан неверный параметр. Здесь сценарий просто переходит на метку :EXIT.

Листинг 3-4. Применение goto

```
@echo off
if "%1"=="" (
echo Error: No parameter passed with script!) & (goto EXIT)
if "%1"=="1" goto SUBROUTINE1
if "%1"=="2" goto SUBROUTINE2
if "%1"=="3" goto SUBROUTINE3
goto EXIT

:SUBROUTINE1
echo In subroutine 1
goto EXIT

:SUBROUTINE2
echo In subroutine 2
goto EXIT
```

```
:SUBROUTINE3
echo In subroutine 3
goto EXIT

:EXIT
echo Exiting...
```

Совет Помните: если вызываемой вами метки нет, возникнет ошибка и сценария завершится, не выполнив оставшиеся команды. Бывалые программисты командной строки (вроде меня) предпочитают использовать *goto EXIT* и создавать метку :EXIT, как показано в предыдущем примере. Однако интерпретатор команд в Windows Server 2003 и Windows XP поддерживает метку :EOF, которая передает управление в конец файла. Это позволяет при помощи :EOF легко выйти из сценария, не определяя метку.

Процедуры

Процедуры позволяют вызывать другие сценарии, не выходя из текущего. При этом командная оболочка запускает указанный сценарий, выполняет его команды и возвращает управление в исходный сценарий на следующую после вызова строку. Рассмотрим пример:

```
if "%1"=="1" call system-checks
if "%1"=="2" call C:\scripts\log-checks
```

Внимание! Если вы забудете поставить оператор *call* и сделаете ссылку на имя другого сценария, второй сценарий запустится, но управление не будет возвращено.

Здесь первый вызов предполагает, что сценарий находится в текущем рабочем каталоге или что путь к нему указан в переменной окружения PATH. Второй вызов адресован сценарию, путь к файлу которого — c:\scripts\log-checks.

Все аргументы, переданные исходному сценарию, передаются вызываемому сценарию с одним изменением: список аргументов слегка модифицируется и в аргумент 0 (%0) помещается имя процедуры. Эти аргументы, специфичные для процедуры, действуют до тех пор, пока не будет достигнут конец файла и управление не будет возвращено исходному сценарию.

Передавать аргументы вызываемому сценарию можно и так:

```
set Arg1=mailer1
set Arg2=dc2
set Arg3=web3
call system-checks Arg1 Arg2 Arg3
```

Теперь в вызываемом сценарии будут доступны переменные Arg1, Arg2 и Arg3.

Глава 4

Запуск заданий по расписанию

Как администратору, вам, вероятно, приходится выполнять одни и те же или похожие задачи практически каждый день. Возможно, вы даже вынуждены приходить на работу раньше или оставаться допоздна, чтобы выполнять эти задачи в нерабочее время. Такими задачами могут быть рутинные операции обслуживания, например, удаление временных файлов, чтобы избежать переполнения дисков, или резервное копирование важных данных. Среди этих задач встречаются процессы и посложнее, в частности просмотр журналов событий на всех бизнес-серверах для выявления проблем, требующих решения. Хорошая новость в том, что если вы можете разбить эти задачи на этапы, есть шанс их автоматизировать. Microsoft Windows предоставляет для этого несколько способов, включая:

- **Schtasks** — «продвинутая» утилита командной строки для запуска команд, сценариев и программ на основе расписания. Задания могут быть назначены для запуска однократно, поминутно, через определенный интервал (например ежечасно, ежедневно или ежемесячно), при загрузке системы, при входе в систему или во время простоя системы;
- **Task Scheduler (Планировщик заданий)** — GUI-утилита для запуска команд, сценариев и программ на основе расписания. Task Scheduler выполняет те же операции, что и утилита командной строки Schtasks, что позволяет использовать их совместно и управлять заданиями, созданными в любой из этих утилит при помощи любого инструмента.

Поскольку Schtasks и Task Scheduler взаимозаменяемы, в данной главе рассказывается, как использовать обе утилиты для автоматизации запуска программ, утилит командной строки и сценариев. Вы увидите, что в большинстве случаев полезно знать обе утилиты, и даже используя Task Scheduler в удоб-

ной графической среде, вы по-прежнему будете работать с командной строкой.

Планирование заданий на локальных и удаленных системах

Расписание можно настроить для всего, что можно запустить в командной строке, включая утилиты командной строки, сценарии, приложения, ярлыки и документы. Также можно указывать аргументы командной строки. Иногда задания назначаются для компьютера, с которым вы работаете в данный момент (т. е. для локальной системы). Однако чаще при планировании заданий вы делаете это для удаленных систем через сеть со своего локального компьютера (т. е. для удаленных компьютеров).

Введение в планирование заданий

Планирование локальных и удаленных заданий обеспечивается совместной работой следующих средств Windows.

- **Task Scheduler (Планировщик заданий)** Служба Windows, управляющая планированием заданий. Она должна работать на каждой системе, где нужно планировать задания. По умолчанию планировщик заданий регистрируется под учетной записью LocalSystem и обычно не имеет нужных разрешений для выполнения задач администрирования. Поэтому вы должны отдельно настраивать каждое задание, чтобы оно выполнялось под адекватной учетной записью с соответствующими разрешениями и правами доступа. Вы также должны убедиться, что служба Task Scheduler настроена на автоматический запуск во всех системах, где вам нужно планировать задания. Убедитесь в правильности параметров запуска и регистрации Task Scheduler.
- **File and Printer Sharing for Microsoft Networks (Служба доступа к файлам и принтерам сетей Microsoft)** Сетевой компонент Windows, позволяющий другим компьютерам получать доступ к ресурсам системы. Этот компонент должен быть установлен и активизирован на каждой системе, к которой вы хотите удаленно обращаться и управлять ею через планировщик заданий. Он также поддерживает множество других функций удаленного управления.

Примечание Ошибки, возникающие, если служба доступа к файлам и принтерам сетей Microsoft не установлена или не активизирована, включают «RPC Server unavailable» («Сервер RPC недоступен») и «Network path not found» («Сетевой путь не найден»). Подробнее об установке и конфигурировании сетевых компонентов см. в разделе «Настройка дополнительных сетевых компонентов» главы 16 в книге «Microsoft Windows Server 2003. Справочник администратора».

Каждое назначенное задание запускает лишь одну программу, утилиту или сценарий и может быть настроено для запуска:

- в указанное время, например 25 октября 2004 г. в 17:45;
- с заданным интервалом, например каждый понедельник, среду и пятницу в 17:45;
- при возникновении определенного системного события, например, когда кто-то входит в систему.

Событийно-управляемые задания заслуживают особого внимания, поскольку не всегда работают, как ожидается; к ним относятся задания, инициируемые при следующих событиях.

- **Запуске системы** — если вы настраиваете задание на запуск при старте системы, планировщик заданий запускает его как неинтерактивный процесс. Задание выполняется до полной обработки, принудительного завершения или до выключения системы. Помните, что завершить выполняемые задания может только их владелец или администратор.
- **Входе в систему** — если вы настраиваете задание на запуск при входе пользователя в систему, планировщик заданий запускает его, когда кто-то входит в систему. Задание выполняется до полной обработки, принудительного завершения или до выхода пользователя из системы. В зависимости от настройки задания, запускаемые при входе (logon tasks), могут выполняться интерактивно или неинтерактивно.

Совет Если пользователь настраивает интерактивное задание под своей учетной записью, а потом в систему входит кто-то другой, задание выполняется с разрешениями настроившего его пользователя и может не завершиться при выходе другого пользователя из системы (поскольку у текущего пользователя может не оказаться соответствующих разрешений для завершения задания и он не является его владельцем). Кроме того, при быстрой смене пользователей (Fast User Switching) в Windows XP задания, запускаемые при входе в систему, не выполняются. Такие задания выполняются только при наличии в системе одного зарегистрированного пользователя.

- **Простое системы** — если вы настраиваете задание на запуск в простое системы, планировщик заданий запускает его при отсутствии действий пользователя за указанное время. Например, можно создать задание, которое запускается, если система простаивает пять минут. Но помните, что дальнейшие действия пользователя не завершат задание. Оно будет выполняться либо до конца, либо до принудительного завершения.

Чтобы запустить несколько команд, программ и утилит, можно создать сценарий командной строки, выполняющий необходимые задачи. Здесь сценарий следует запускать под учетными данными конкретного пользователя или администратора, чтобы у сценария были необходимые разрешения и права доступа. Сценарий также должен настраивать любые необходимые пользовательские параметры, чтобы получить контроль за выполняемыми операциями и чтобы при необходимости ему были доступны доменные пользовательские параметры вроде подключений сетевых дисков.

Примечание Настраивая запуск заданий, вы можете указать учетную запись и пароль пользователя. Для часто повторяемых заданий такая тактика создает проблемы, особенно если разрешения или пароли меняются, — а они обязательно меняются. При изменении разрешений или смене паролей вам придется модифицировать свойства задания и предоставить новые учетные данные.

Назначение и доступ к заданиям

Любой пользователь может назначать задания на локальном компьютере, а также просматривать и изменять назначенные им задания. Администраторы могут назначать, просматривать и менять все задания на локальном компьютере. Чтобы назначать, просматривать или менять задания на удаленном компьютере, вам надо быть членом группы Administrators (Администраторы) на удаленном компьютере или иметь возможность предоставить учетные данные администратора удаленного компьютера при запросе.

С помощью Scheduled Task Wizard (Мастер планирования заданий) или Schtasks можно получать доступ и управлять системными заданиями, используя папку Scheduled Tasks (Назначенные задания); считайте ее центральным интерфейсом планирования заданий. Получить доступ к этой папке в локальной системе позволяет любой из следующих способов:

- откройте Control Panel (Панель управления) и выберите Scheduled Tasks (Назначенные задания);
- откройте Windows Explorer (Проводник), щелкните My Computer (Мой компьютер), затем Control Panel и Scheduled Tasks.

В Windows-домене доступ к папке Scheduled Tasks в удаленной системе можно получить, выполнив следующие действия.

1. Откройте Windows Explorer, затем через My Network Places (Сетевое окружение) перейдите к нужному компьютеру.
2. Щелкните значок компьютера, затем Scheduled Tasks (рис. 4-1).

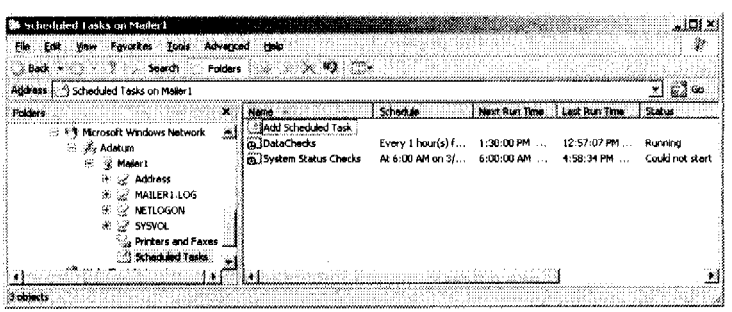


Рис. 4-1. Папка Scheduled Tasks показывает названия, время запуска и состояние всех настроенных в системе заданий

Примечание В рабочей группе проще всего получить доступ к папке Scheduled Tasks через соединение Remote Desktop (Удаленный рабочий стол). Конечно, такой прием годится и для компьютеров в домене, но он не обязателен. В любом случае сначала убедитесь, что на вкладке Remote (Удаленное использование) утилиты System (Система) в Control Panel установлен флажок Allow Users To Connect Remotely To This Computer (Разрешить удаленный доступ к этому компьютеру). Затем установите соединение с удаленным рабочим столом, выполнив следующую процедуру.

1. Откройте меню Start (Пуск), выберите Programs (Программы) или All Programs (Все программы), затем Administrative Tools (Администрирование), и, наконец, Remote Desktops (Удаленные рабочие столы). Это приведет к запуску консоли Remote Desktops.
2. Щелкните правой кнопкой мыши Remote Desktops под Console Root (Корень консоли) в левой части окна и выберите Add New Connection (Добавить подключение).
3. Введите имя или IP-адрес компьютера в рабочей группе или домене, к которому хотите подключиться, а также имя пользователя и пароль, затем щелкните OK.

В рабочей группе используйте учетную запись на компьютере, к которому вы подключаетесь. Затем подключитесь к компьютеру через соединение Remote Desktop, щелкнув его правой кнопкой мыши и выбрав команду Connect (Подключить). Теперь запустите Windows Explorer на удаленном компьютере и перейдите в папку Scheduled Tasks (Назначенные задания).

Обратите внимание, что на рис. 4-1 в папке Scheduled Tasks показываются:

- **Name (Имя)** — название задания, которое может быть любой строкой символов и которое, как и другие свойства задания, устанавливается при создании;
- **Schedule (Расписание)** — расписание запуска задания. Disabled (Остановлено) означает, что задача остановлена и не будет запускаться;
- **Next Run Time (Время следующего запуска)** — дата и время следующего запуска задания. Never (Никогда) указывает, что после назначенного времени запуска задание больше не запустится и, вероятно, является разовым;

- **Last Run Time (Время прошлого запуска)** — дата и время предыдущего запуска задания. Never указывает, что задание еще не запускалось;
- **Status (Состояние)** — текущее состояние задачи. Running (Выполняется) указывает, что задание запущено планировщиком заданий и выполняется, а Could not start (Нельзя запустить) — что задание не удается запустить из-за какой-то проблемы;
- **Last Result (Прошлый результат)** — код ошибки при завершении. Код, равный нулю, сообщает, что ошибки не было. Любое другое значение указывает на ошибку какого-либо типа;
- **Creator (Создан)** — имя пользователя, создавшего назначенное задание.

Мониторинг назначенных заданий

Служба Task Scheduler не проверяет предоставляемую вами информацию и доступность программ, команд или утилит. Если вы укажете неправильные данные, задание просто не запустится или будет приводить к ошибкам при запуске. Один из способов проверки заданий — просмотреть их состояние и прошлый результат в папке Scheduled Tasks. Эта информация относится к последнему запуску задания. Однако она ничего не говорит о том, были ли проблемы при запусках задания, предшествующих последнему запуску. Для более глубокого анализа и лучшего понимания того, как выполняются задания, периодически проверяйте файл журнала планировщика заданий, Schedlgu.txt, который находится в папке *%SystemRoot%*/Tasks.

Файл журнала можно просмотреть, выбрав команду View Log (Просмотр журнала) из меню Advanced (Дополнительно), когда в Windows Explorer выбрана папка Scheduled Tasks (Назначенные задания). Просмотрев журнал планировщика заданий, вы найдете записи, указывающие:

- когда была запущена служба Task Scheduler и когда она была завершена (остановлена);
- когда запускались задания, когда они завершались и с каким кодом ошибки. Код ошибки, равный нулю, означает, что задача выполнена нормально. Любой другой код указывает на возможную ошибку.

Примечание *%SystemRoot%*\Tasks — особая папка с представлением, позволяющим работать с назначенными заданиями. Если открыть эту папку в Windows Explorer, вы не обнаружите файла журнала SchedLgU.txt. Но он там, и, если зайти в этот каталог из командной строки, вы увидите журнал в списке.

Также вам могут понадобиться более подробные сведения о том, что происходит при выполнении сценариев. Для этого записывайте вывод команд и утилит в отдельный файл журнала, что позволит понять, дают ли эти команды и утилиты нужные результаты. Как было сказано в предыдущей главе, вывод команд можно записывать в именованный файл, перенаправляя стандартный вывод и стандартные ошибки. В следующих примерах вывод команды DEFRAG добавляется в файл Stat-log.txt; туда же записываются все ошибки DEFRAG:

```
defrag c: >> c:\logs\stat-log.txt 2>&1
defrag d: >> c:\logs\stat-log.txt 2>&1
```

Внимание! Если вы работаете с каталогами, как в этом примере, то они должны существовать. За вас они не создаются, а все ошибки, возникающие из-за отсутствия каких-либо каталогов, в журнал не записываются.

Примечание Запись вывода в журнал не решает все возможные проблемы, но помогает определить, что назначенные задания выполняются, как ожидалось. Если вы пытаетесь выявить и устранить проблему, помните, что задания могут не выполняться по многим причинам, некоторые из которых вне вашей власти. Например, назначенные задания не запустятся, если в нужное время система выключена. Если вы хотите получать уведомления о пропущенных заданиях, выберите Notify Me Of Missed Tasks (Уведомлять о пропущенных заданиях) из меню Advanced (Дополнительно), когда в Windows Explorer (Проводник) выбрана папка Scheduled Tasks (Назначенные задания). Если эта функция включена, при входе в систему отображается сообщение, информирующее о наличии заданий, пропущенных за время отключения системы.

Назначение заданий с помощью Scheduled Task Manager

Для назначения заданий в локальной или удаленной системе, к которой вы подключены, используйте Scheduled Task Manager (Мастер планирования заданий). Доступ к этому мастеру и текущим назначенным заданиям осуществляется через папку Scheduled Tasks (Назначенные задания).

Элементы папки Scheduled Tasks показывают текущие назначенные задания. Получить доступ к этой папке в локальной системе можно любым из следующих способов:

- откройте Control Panel (Панель управления) и щелкните Scheduled Tasks (Назначенные задания);
- последовательно откройте Windows Explorer (Проводник), My Computer (Мой компьютер), Control Panel и Scheduled Tasks.

В домене Windows доступ к папке Scheduled Tasks на удаленной системе можно получить, выполнив следующие действия.

1. Откройте Windows Explorer, затем через узел My Network Places (Сетевое окружение) перейдите к нужному компьютеру.
2. Щелкните значок компьютера, затем Scheduled Tasks.

В случае рабочей группы установите подключение к удаленному рабочему столу, как описано выше, затем через Windows Explorer откройте папку Scheduled Tasks (Назначенные задания). Этот способ годится и для компьютеров в домене, но предыдущая процедура удобнее.

Создание заданий с помощью Scheduled Task Manager

Чтобы назначить задание с помощью Scheduled Task Manager (Мастер планирования заданий), выполните следующую процедуру.

1. Запустите Scheduled Task Manager, дважды щелкнув Add Scheduled Task (Добавить задание) в папке Scheduled Tasks (Назначенные задания). Прочитайте страницу приветствия и щелкните Next (Далее).
2. На странице, показанной на рис. 4-2, выберите программу, планируемую для запуска в качестве задания. Для этого

щелкните кнопку Browse (Обзор), чтобы открыть диалоговое окно Select Program To Schedule (Выберите приложение, для которого следует составить расписание) и найдите нужную программу, утилиту командной строки или сценарий. После этого нажмите кнопку Open (Открыть).

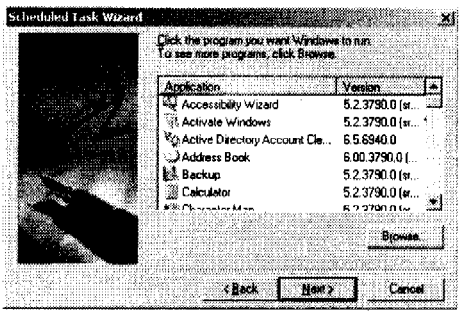


Рис. 4-2. Выберите в Scheduled Task Manager запускаемую программу или щелкните Browse, чтобы найти сценарии и другие приложения

3. Введите название задания, как показано на рис. 4-3. Оно должно быть коротким, но информативным, чтобы вы могли быстро определить, для чего предназначено это задание.

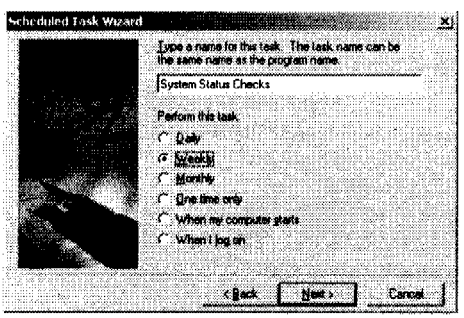


Рис. 4-3. С помощью Scheduled Task Manager присвойте заданию имя и укажите, как часто вы хотите выполнять это задание

4. Выберите расписание запуска. Задания могут выполняться периодически (ежедневно, еженедельно, ежемесячно) или при определенном событии, например при загрузке компьютера или входе пользователя задания в систему.

5. Щелкните Next (Далее). Если вам нужен периодический запуск, выберите дату и время для запуска назначенного задания. Следующая страница, которую вы увидите, зависит от выбранного расписания запуска задания.
6. Если вы выбрали ежедневный запуск, появляется страница с датой и временем, как на рис. 4-4.

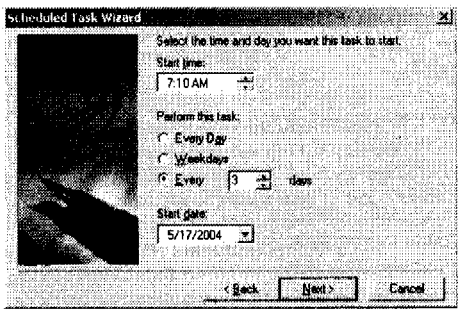


Рис. 4-4. На этой странице настраивается ежедневный запуск задания

Установите время и дату запуска. Задания, выполняемые по дням, могут быть настроены для запуска:

- *every day (ежедневно)* — семь дней в неделю;
- *weekdays (по рабочим дням)* — только с понедельника по пятницу;
- *every ...days (каждый ... день)* — каждый 2-й, 3-й, *N-й* день.

7. Если вы выбрали еженедельный запуск задания, появляется страница с датой и временем, как на рис. 4-5.

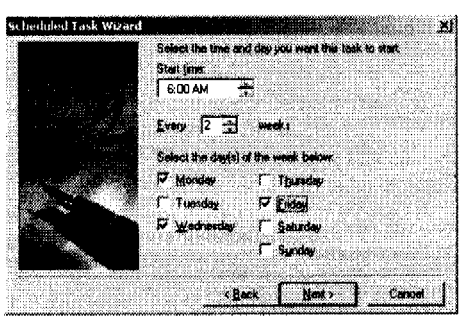


Рис. 4-5. На этой странице настраивается еженедельный запуск задания

Настройте задание с помощью этих полей:

- *start time (время начала)* — устанавливает время начала выполнения задания;
- *every ... weeks (каждую ... неделю)* — позволяет запускать задание каждую неделю, каждые две недели или каждые *N* недель;
- *select the day(s) of the week below (выберите дни недели)* — устанавливает день (дни) недели для запуска задания, например по понедельникам или по понедельникам и пятницам.

8. Если вы выбрали ежемесячный запуск задания, появляется страница с датой и временем, как на рис. 4-6.

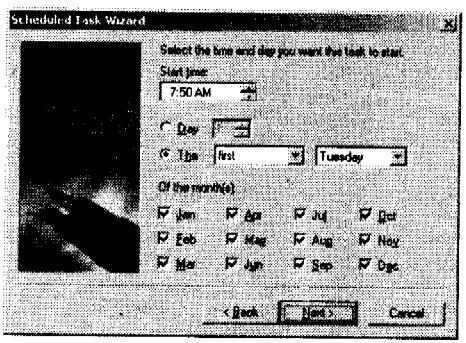


Рис. 4-6. На этой странице настраивается ежемесячный запуск задания

Настройте задание с помощью этих полей:

- *start time (время начала)* — устанавливает время начала выполнения задания;
- *day (число)* — указывает число месяца для запуска задания. Например, если выбрать 5, задание будет запускаться пятого числа каждого месяца;
- *the... (или...)* — устанавливает запуск задания по *N*-ым числам месяца, скажем, по вторым понедельникам или по третьим вторникам каждого месяца;
- *of the month(s) (следующих месяцев)* — эти флажки позволяют выбрать месяцы для запуска задания.

9. Если для запуска задания вы выбрали One Time Only (Однократно), появляется страница с датой и временем, как на рис. 4-7. Укажите время и дату начала.

Совет Для заданий, автоматически запускаемых при загрузке компьютера или при входе пользователя задания, время и дату начала устанавливать не надо. Если вы хотите с помощью мастера настроить стартовую задачу (startup task) для определенного пользователя, то должны войти в систему под учетной записью этого пользователя и запустить мастер.

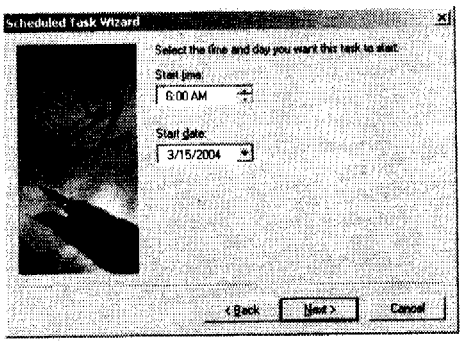


Рис. 4-7. На этой странице настраивается однократный запуск задания

10. Настроив дату и время, щелкните Next.
11. В следующем диалоговом окне введите имя пользователя и пароль, которые могут понадобиться для запуска назначенного задания. У этого пользователя должны быть соответствующие разрешения и привилегии на запуск задания.

Примечание В доменах Windows вводите имя пользователя в виде Домен\ИмяПользователя, например Adatum\wrstanek, где Adatum — имя домена, а wrstanek — учетная запись пользователя.

12. Последняя страница мастера предоставляет сводную информацию о назначенном задании, и кроме того:
    - если вы хотите задать аргументы командной строки, установите флажок Open Advanced Properties For This Task When I Click Finish (Установить дополнительные параметры после нажатия кнопки «Готово») или дважды щелкните задание в Windows Explorer после нажатия кнопки Finish (Готово). Затем в поле Run (Выполнить) введите параметры после пути. Если путь для за-

дания включает пробелы, не забудьте заключить его в двойные кавычки ("). Кроме того, если пробелы есть в аргументе командной строки, его тоже надо заключить в двойные кавычки (");

- если при создании задания возникла ошибка, вы увидите соответствующее сообщение. Щелкните OK. Задание все равно будет создано. Однако вам скорее всего придется модифицировать его свойства. В Windows Explorer дважды щелкните задание, чтобы исправить указанную ошибку. Наиболее распространенная ошибка — Access Denied (Доступ запрещен). Она возникает, если предоставлены неверные учетные данные пользователя, например, введен неверный пароль или указанной учетной записи пользователя в домене нет.

Щелкните Finish (Готово), чтобы завершить создание задания.

Примечание Многие проблемы с запуском заданий проявляются весьма четко. В частности, проверяйте любое задание, для которого указано «Could not start» (Нельзя запустить). Но иногда проблемы с заданиями не столь очевидны. Задание, обозначенное как выполняемое («Running»), на самом деле может не выполняться, а зависнуть. Проверить наличие зависших процессов можно по столбцу Last Run Time (Время прошлого запуска); он сообщает, когда было запущено задание. Если задание выполняется больше одного дня, обычно это проблема. Сценарий может ожидать ввода, ему не удается считать или записать файлы, или это просто задание, контроль над которым потерян и которое следует остановить. Чтобы остановить задание, щелкните его правой кнопкой мыши в папке Scheduled Tasks (Назначенные задания) и выберите End Task (Снять задачу). Или можно подождать, пока задачу снимет система. По умолчанию таймаут для всех заданий наступает через 72 часа выполнения. Изменить время ожидания можно на вкладке Settings (Параметры) диалогового окна свойств задания.

Изменение свойств задания

Чтобы изменить параметры задания откройте папку Scheduled Tasks (Назначенные задания). Для локальной системы дважды щелкните Scheduled Tasks в Control Panel. Для удаленной системы запустите Windows Explorer, через узел My Network

Places (Сетевое окружение) перейдите к нужному компьютеру, щелкните его значок, а затем Scheduled Tasks. Потом дважды щелкните нужное задание и используйте диалоговое окно свойств для внесения необходимых изменений.

Примечание При изменении способа выполнения задания у вас может быть запрошена информация об учетной записи. Тогда вы можете принять текущую учетную запись, под которой выполняется задание, или ввести новую учетную запись в формате Домен\ИмяПользователя. В любом случае вам надо будет ввести и подтвердить пароль для используемой учетной записи.

Копирование и перемещение заданий с одной системы на другую

Чтобы задания, созданные в одной системе, задействовать в другой, необязательно еще раз создавать их вручную. Вместо этого в домене действуйте по следующей схеме.

1. Через узел My Network Places (Сетевое окружение) перейдите к компьютеру, на котором хранятся нужные вам задания. Щелкните его значок и Scheduled Tasks (Назначенные задания).
2. Щелкните правой кнопкой мыши одно из заданий, которые вы хотите использовать, и выберите Copy (Копировать).
3. Вновь воспользуйтесь узлом My Network Places для навигации. На этот раз перейдите к компьютеру, на который вы хотите скопировать задание.
4. Выбрав значок нужного компьютера, щелкните правой кнопкой мыши Scheduled Tasks и выберите Paste (Вставить).
5. Назначенное задание будет скопировано в новое место. Не забудьте проверить свойства задания на компьютере-адресате, и убедиться, что все параметры корректны.

Также можно перемещать задания с одного компьютера на другой. Для этого придерживайтесь следующей схемы.

1. Через узел My Network Places перейдите к компьютеру, на котором хранятся нужные вам задания. Щелкните его значок, затем Scheduled Tasks.
2. Щелкните правой кнопкой мыши одно из заданий, которые вы хотите задействовать, и выберите Cut (Вырезать).

3. Вновь воспользуйтесь узлом My Network Places для навигации. На этот раз перейдите к компьютеру, на который вы хотите переместить задание.
4. Выбрав значок нужного компьютера, щелкните правой кнопкой мыши Scheduled Tasks и выберите Paste (Вставить).
5. Назначенное задание переместится на новое место. Не забудьте проверить свойства задания на компьютере-адресате, и убедиться, что все параметры корректны.

Включение и выключение заданий

В зависимости от предпочтений задачи можно включать и выключать по мере необходимости. Если вы временно не хотите использовать задание, отключите его. А когда оно вновь вам потребуется, включите это задание. Включая и выключая задания вместо их удаления, вы экономите время на повторной настройке параметров заданий.

Включать и выключать задания можно так.

1. Откройте папку Scheduled Tasks (Назначенные задания). Для локальной системы выберите или дважды щелкните Scheduled Tasks в Control Panel (Панель управления). Для удаленной системы запустите Windows Explorer (Проводник), через узел My Network Places (Сетевое окружение) перейдите к нужному компьютеру, щелкните его значок, а затем Scheduled Tasks.
2. Выберите или дважды щелкните нужное задание. Откроется диалоговое окно свойств с именем задания. По умолчанию должна быть выбрана вкладка Task (Задание).
3. Чтобы включить задачу, установите флажок Enabled (Включено), а чтобы отключить — сбросьте его. Щелкните OK.

Немедленный запуск задания

Необязательно ждать назначенного времени, чтобы запустить задание. Запустить задание можно в любой момент, открыв папку Scheduled Tasks (Назначенные задания), щелкнув правой кнопкой мыши нужное задание и выбрав Run (Выполнить).

Удаление ненужных заданий

Если задание вам больше не нужно, удалите его, открыв папку Scheduled Tasks (Назначенные задания), щелкнув задание правой кнопкой мыши и выбрав Delete (Удалить).

Планирование заданий с помощью Schtasks

Schtasks позволяет выполнять те же операции по планированию заданий, что и Scheduled Tasks Wizard (Мастер планирования заданий). Все задания, созданные с помощью Schtasks, отображаются как назначенные в папке Scheduled Tasks (Назначенные задания) и могут управляться из командной строки или через GUI.

Schtasks поддерживает несколько наборов подкоманд (subcommands) и является одной из самых сложных утилит командной строки. В следующих разделах я расскажу о каждой из этих подкоманд, перечисленных ниже.

- **Schtasks /Create** — создание заданий.
- **Schtasks /Change** — изменение параметров существующих заданий.
- **Schtasks /Query** — отображение назначенных заданий на локальном или указанном компьютере.
- **Schtasks /Run** — немедленный запуск назначенного задания.
- **Schtasks /End** — остановка выполняемого задания.
- **Schtasks /Delete** — удаление назначенных заданий, которые больше не нужны.

Создание назначенных заданий с помощью Schtasks /Create

Schtasks/Create позволяет создавать разовые и регулярно выполняемые задания, а также задания, которые запускаются на основе особых системных событий, таких как вход в систему или ее запуск. Вот основной синтаксис для определения этих типов заданий:

```
schtasks /create /tn ИмяЗадания /tr ИсполняемоеЗадание /sc
Расписание [/mo Модификатор]
```

где *ИмяЗадания* — строка с названием задания, *ИсполняемоеЗадание* — путь к файлу утилиты командной строки или сценария, который надо запустить, *Расписание* — расписание запуска, а *Модификатор* — необязательный параметр, меняющий расписание запуска на основе типа расписания. Любые задания, создаваемые по такому синтаксису, создаются на локальном компьютере и используют ваши разрешения.

Если вы не предоставили пароль своей учетной записи, он будет запрошен при создании задания.

Допустимые значения для *Расписания* показаны в табл. 4-1. Обратите внимание на примеры использования и модификаторы для разных типов расписания. Далее в этой главе я подробно расскажу обо всех типах расписания и модификаторах. Также обратите внимание на следующее:

- дни недели можно вводить списком, разделяя их запятыми, например Mon, Wed, Fri, или через дефис, указывая последовательность дней, скажем, Mon-Fri задает интервал с понедельника по пятницу;
- месяцы можно вводить списком, разделяя их запятыми, например Jan, Mar, Jun, или через дефис, указывая последовательность месяцев, например Jan-Jun указывает интервал с января по июнь;
- для недели месяца можно задать только одно значение, например FIRST или LAST.

Табл. 4-1. Типы расписаний для Schtasks /Create

Тип расписания	Описание	Значения модификатора
MINUTE	Задание запускается через указанный интервал в минутах. По умолчанию — раз в минуту	/mo 1–1439; количество минут между запусками задания. Стандартный модификатор — 1
HOURLY	Задание запускается через указанный интервал в часах. По умолчанию — раз в час	/mo 1–23; количество часов между запусками задания. Стандартный модификатор — 1
DAILY	Задание запускается каждый день или каждые *N* дней. По умолчанию — каждый день	/mo 1–365; число дней между запусками задания. Стандартный модификатор — 1
WEEKLY	Задание запускается каждую неделю или каждые *N* недель в указанные дни. По умолчанию — каждую неделю по понедельникам	/mo 1–52; число недель между запусками задания. Необязательный параметр /d указывает дни недели для выполнения. Для задания дней используйте MON (понедельник), TUE (вторник), WED (среда), THU (четверг), FRI (пятница), SAT (суббота) и SUN (воскресенье). Символ * задает все дни недели

(см. след. стр.)

Табл. 4-1. (*окончание*)

Тип расписания	Описание	Значения модификатора
MONTHLY	Задание запускается каждый месяц или каждые *N* месяцев в указанные дни. По умолчанию — в первый день каждого месяца	/mo 1–12; число месяцев между запусками задания. Чтобы указать дни недели для запуска, используйте /d MON-SUN, а чтобы задача запускалась ежедневно — символ *
	Второй вариант ежемесячного запуска в указанный день. Используйте /mo и /m или /m и /d	/mo LASTDAY — последний день месяца; /m JAN, FEB, ..., DEC — месяц (месяцы); /d 1–31 — число месяца
	Третий вариант для определенной недели месяца	/mo FIRST | SECOND | THIRD | FOURTH | LAST — неделя месяца; /d MON–SUN — день недели; /m JAN, FEB, ..., DEC — месяц (месяцы)
ONCE	Задание запускается один раз в указанные время и дату	—
ONSTART	Задание запускается при старте системы	—
ONLOGON	Задание запускается при входе пользователя в систему	—
ONIDLE	Задание запускается при простое системы в течение заданного периода	/i 1–999 — интервал простоя системы до запуска задания (в минутах)

Чтобы понять, как использовать Schtasks /Create, рассмотрим несколько примеров.

Задание выполняется немедленно и больше не запускается:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
once
```

Задание запускается при старте системы:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
onstart
```

Задание запускается при простое системы более 10 минут:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
onidle /i 10
```

Задание запускается на локальном компьютере каждые 15 минут:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
minute /mo 15
```

Задание запускается на локальном компьютере каждые пять часов:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
hourly /mo 5
```

Задание запускается на локальном компьютере каждые два дня:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
daily /mo 2
```

Задание запускается через каждые две недели по понедельникам (стандартный день запуска):

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
weekly /mo 2
```

Задание запускается каждую неделю по понедельникам и пятницам:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
weekly /d mon,fri
```

Задание запускается первого числа каждого месяца:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
monthly
```

Задание запускается пятого числа каждые два месяца:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
monthly /mo 2 /d 5
```

Задание запускается в последний день каждого месяца:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sysch.bat /sc
monthly/mo lastday
```

Задание запускается в первый понедельник апреля, августа и декабря:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sysch.bat /sc
monthly /mo first /d mon /m apr,aug,dec
```

Когда в пути к заданию есть пробелы, заключайте путь к файлу в двойные кавычки:

```
schtasks /create /tn "SysChecks" /tr "c:\My Scripts\sch.bat"
/sc onstart
```

Иначе, когда Schtasks попытается запустить задание, возникнет ошибка. Далее, если вы хотите передать программе, утилите или сценарию аргументы, просто укажите их в пути к исполняемому заданию. Любые аргументы, содержащие пробелы, следует заключать в кавычки, чтобы они правильно интерпретировались как один, а не как несколько аргументов. Вот примеры:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat 1 Y
LAST /sc onstart
schtasks /create /tn "SysChecks" /tr "c:\My Scripts\sch.bat" Y
N /sc onstart
schtasks /create /tn "SysChecks" /tr "c:\My Scripts\sch.bat"
"Full Checks"
```

Кроме того, можно назначать задания для удаленных компьютеров и задания, которые должны выполняться с другими пользовательскими разрешениями. При назначении заданий для удаленных компьютеров главное — помнить о том, что ваш компьютер должен быть в том же домене, что и удаленный, или в домене, которому доверяет удаленный компьютер. Для этого применяется расширенный синтаксис, включающий параметры:

```
/s Компьютер, /u [Домен\]Пользователь [/p Пароль]
```

где *Компьютер* — имя или IP-адрес удаленного компьютера, *Домен* — необязательное имя домена, где находится учетная запись пользователя, *Пользователь* — имя учетной записи пользователя, чьи разрешения вы хотите задействовать, а *Пароль* — необязательный пароль для данной учетной записи. Если домен не указан, берется текущий домен. А если вы не задаете пароль учетной записи, он будет запрошен у вас.

Совет Чтобы задание запускалось только при входе в систему определенного пользователя, указывайте необязательный параметр /It, который сообщает, что задание следует запускать, только когда в систему входит пользователь — владелец этого задания. Параметр /It действует лишь в Windows Server 2003. В Windows XP он не применим.

Чтобы понять, как добавить информацию о компьютере и пользователе, рассмотрим пару примеров.

При установке разрешений для задания на локальном компьютере используется учетная запись adatum\wrstanek:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
onstart /s mailer01 /u adatum\wrstanek
```

Используется удаленный компьютер mailer01 и учетная запись adatum\wrstanek:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
onstart /s mailer01 /u adatum\wrstanek
```

Наконец, при желании можно указать нужные дату и время начала, а также дату и время завершения, используя:

- /st ВремяНачала, где *ВремяНачала* задается в 24-часовом формате (*ЧЧ:ММ*), например 15:00;
- /et ВремяЗавершения, где *ВремяЗавершения* задается в 24-часовом формате (*ЧЧ:ММ*), например 15:00. Используется только с /sc minute и /sc hourly. Применимо только к Windows Server 2003;
- /du Длительность, где *Длительность* — число часов и минут для выполнения в формате *ЧЧЧЧ:ММ*. Используется только с /sc minute и /sc hourly. Применимо только к Windows Server 2003;
- /sd ДатаНачала, где *ДатаНачала* — дата начала в стандартном формате системы, например *ММ/ДД/ГГГГ*;
- /ed ДатаЗавершения, где *ДатаЗавершения* — дата завершения в стандартном формате системы, например *ММ/ДД/ГГГГ*.

Совет Если вы указываете дату и время завершения, можете указать и параметр /Z, который предписывает планировщику заданий удалить задание по завершении его расписания. Это относится только к Windows Server 2003.

Чтобы понять, как использовать дату и время начала и дату и время завершения, рассмотрим несколько примеров.

Начать выполнение ежечасного задания в полночь:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
hourly /st 00:00
```

Начать выполнение ежечасного задания в 3 утра и прекратить в 7 утра:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
hourly /st 03:00 /et 07:00
```

Начать выполнение еженедельного задания в 3 утра 20 февраля 2004 года:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
weekly /st 03:00 /sd 02/20/2004
```

Начать выполнение еженедельного задания в 3 утра 20 февраля 2004 года и прекратить в 2:59 утра 15 марта 2004 года:

```
schtasks /create /tn "SysChecks" /tr c:\scripts\sch.bat /sc
weekly /st 03:00 /sd 02/20/2004 /et 02:59 /ed 03/15/2004
```

Примечание Форматы даты и времени определяются параметрами диалогового окна Regional And Language options (Язык и региональные стандарты), установленными в данной системе. В моих примерах дата представлена в формате «Английский (США)».

Изменение назначенных заданий с помощью Schtasks /Change

Schtasks /Change служит для изменения ключевых параметров назначенных заданий. Основной синтаксис Schtasks /Change таков:

```
schtasks /change /tn ИмяЗадания ИзменяемыеПараметры
```

где *ИмяЗадания* — это название изменяемого задания, а *ИзменяемыеПараметры* — параметры, которые вы хотите изменить. К доступным параметрам относятся:

- /ru *Домен\Пользователь* — изменяет учетную запись пользователя, под которой выполняется задание, например /ru adatum/wrstanek;

- /rp *Пароль* — задает пароль для ранее указанной или вновь назначенной учетной записи для выполнения задания;
- /tr *ИсполняемаяЗадача* — служит для смены утилиты командной строки или сценария, выполняемого в указанном задании;
- /st *ВремяНачала* — устанавливает время начала для ежеминутных или ежечасных заданий. Применим только в Windows Server 2003;
- /ri *Интервал* — задает интервал запуска в минутах. Применим только в Windows Server 2003;
- /et *ВремяЗавершения* — устанавливает время завершения для ежеминутных или ежечасных заданий. Применим только в Windows Server 2003;
- /du *Длительность* — задает, сколько часов и минут отводится на выполнение задания. Применим только к ежеминутным и ежечасным заданиям и только в Windows Server 2003;
- /sd *ДатаНачала* — указывает дату начала выполнения задания. Применим только в Windows Server 2003;
- /ed *ДатаЗавершения* — задает дату завершения для задания. Применим только в Windows Server 2003;
- /k — указывает, что при достижении времени завершения или по истечении срока выполнения задание не следует запускать вновь, но не останавливает задание, если оно уже выполняется (текущий запуск будет последним). Применим только в Windows Server 2003;
- /it — указывает, что задание следует выполнять только при входе в систему владеющего им пользователя. Применим только в Windows Server 2003.

Вот примеры изменения параметров заданий.

Изменение запускаемого сценария:

```
schtasks /change /tn "SysChecks" /tr
c:\scripts\systemchecks.bat
```

Смена имени пользователя и пароля для выполнения задания:

```
schtasks /change /tn "SysChecks" /ru adatum\hthomas /rp
gophers
```

Изменение задания для еженедельного запуска с 7 утра 1 марта 2004 до 6:59 утра 30 марта 2004:

```
schtasks /change /tn "SysChecks" /st 07:00 /sd 03/01/2004 /et
06:59 /ed 03/30/2004
```

Примечание Как уже говорилось, формат даты и времени определяется параметрами диалогового окна Regional And Language options (Язык и региональные стандарты), применяемыми на компьютере. Здесь используется формат «Английский (США)».

При изменении задания Schtasks выводит сообщение, информирующее об успехе или неудаче этой операции, например:

```
SUCCESS: The parameters of the scheduled task "SysChecks" have
been changed.
```

Если вы работаете с удаленным компьютером или не зарегистрированы под учетной записью, обладающей разрешением на изменение заданий, то можете при необходимости указать информацию о компьютере и учетной записи. Синтаксис выглядит так:

```
schtasks /change /tn ИмяЗадания /s Компьютер /u
[Домен\]Пользователь [/p Пароль]
```

где *Компьютер* — имя или IP-адрес удаленного компьютера, *Домен* — необязательное имя домена, где находится учетная запись пользователя, *Пользователь* — имя учетной записи пользователя, чьи разрешения вы хотите задействовать, а *Пароль* — необязательный пароль для этой учетной записи. Если домен не указан, берется текущий домен. А если вы не задаете пароль учетной записи, он будет запрошен у вас.

В следующем примере удаленным компьютером является mailer1, а учетная запись пользователя с полномочиями на изменение задания SysChecks, — wrstanek из домена Adatum:

```
schtasks /change /tn "SysChecks" /tr
c:\scripts\systemchecks.bat
/s mailer1 /u adatum\wrstanek
```

Поскольку пароль не указан, Schtasks запросит его. В Windows Server 2003 можно быстро включать и выключать задания по имени. Для включения заданий используется следующий синтаксис:

```
schtasks /change /tn ИмяЗадания /enable
```

А для выключения — такой синтаксис:

```
schtasks /change /tn ИмяЗадания /disable
```

где *ИмяЗадания* — имя включаемого или отключаемого задания, например:

```
schtasks /change /tn "SysChecks" /disable
```

Запрос сведений о назначенных заданиях через Schtasks /Query

Введя в командной строке **schtasks /query**, можно быстро определить, какие задания назначены на компьютере. Также можно (а для удаленного компьютера даже нужно) указать информацию о компьютере и учетной записи для доступа к компьютеру в следующем виде:

```
schtasks /query /s Компьютер /u [Домен\]Пользователь [/p Пароль]
```

где *Компьютер* — имя или IP-адрес удаленного компьютера, *Домен* — необязательное имя домена, где находится учетная запись пользователя, *Пользователь* — имя учетной записи пользователя с соответствующими правами доступа на удаленном компьютере, а *Пароль* — необязательный пароль для выбранной учетной записи.

В следующем примере удаленный компьютер это mailer1, а учетная запись пользователя — wrstanek из домена Adatum:

```
schtasks /query /s mailer1 /u adatum\wrstanek
```

Поскольку пароль не указан, Schtasks запросит его.

Основной вывод Schtasks /Query представляется в виде таблицы со столбцами Task Name (Имя задания), Next Run Time (Время следующего запуска) и Status (Состояние). Вы также можете форматировать вывод в список или в строки разделенных запятыми значений, используя соответственно /Fo List или /Fo Csv. Вывод в виде списка идеально сочетается с параметром /V (verbose), который предоставляет полную информацию обо всех свойствах заданий и может применяться, как показано в следующем примере:

```
schtasks /query /s mailer1 /u adatum\wrstanek /fo list /v
```

Другой полезный параметр, /Nh, указывает, что табличный или CSV-форматированный вывод не должен содержать заголовков.

Совет Вероятно, вас интересует, зачем нужны разные форматы. Хороший вопрос. Я рекомендую использовать формат детального списка (/Fo List /V), когда вам нужны все подробности о назначенных в системе заданиях или когда вы занимаетесь устранением неполадок. Разделенные запятыми значения лучше использовать при записи вывода в файл, который впоследствии может быть экспортирован в таблицу или в плоскую (неиерархическую) базу данных. Помните, что вывод Schtasks можно перенаправлять в файл с помощью > или >>.

Немедленный запуск заданий с помощью Schtasks /Run

Задание можно запустить в любой момент:

```
schtasks /run /tn ИмяЗадания
```

где *ИмяЗадания* — название запускаемого задания, например:

```
schtasks /run /tn "SysChecks"
```

Запуск задания не влияет на его расписание и не изменяет время следующего запуска. Если задание удастся успешно запустить, вы увидите сообщение об этом. Кроме того, можно указать имя удаленного компьютера, на котором назначено задание, и при необходимости учетную запись, под которой запускается задание, включая необязательный пароль. Вот примеры:

```
schtasks /run /tn "SysChecks"/s 192.168.1.100
schtasks /run /tn "SysChecks"/s 192.168.1.100 /u adatum/
wrstanek
```

Примечание Если вы укажете имя пользователя и не предоставите пароль, вам будет немедленно предложено ввести его.

Остановка выполняемых заданий с помощью Schtasks /End

Задание можно остановить в любой момент:

```
schtasks /end /tn ИмяЗадания
```

где *ИмяЗадания* — название выполняемого задания, которое следует остановить, например:

```
schtasks /end /tn "SysChecks"
```

Задание останавливается, только если оно выполнялось, и при успехе этой операции выводится примерно такое сообщение:

```
SUCCESS: The scheduled task "SysChecks" has been terminated
successfully.
```

Кроме того, можно указать имя удаленного компьютера, на котором назначено задание, и при необходимости учетную запись с полномочиями для остановки задания, а также необязательный пароль, например:

```
schtasks /end /tn "SysChecks"/s 192.168.1.100
```

или

```
schtasks /end /tn "SysChecks"/s 192.168.1.100 /u adatum/
wrstanek
```

Поскольку пароль не указан, Schtasks запросит его.

Удаление заданий с помощью Schtasks/Delete

Удалять задания по имени на локальном или удаленном компьютере позволяет следующий синтаксис:

```
schtasks /delete /tn ИмяЗадания [/s Компьютер /u [Домен/]
Пользователь [/p Пароль]]
```

где *ИмяЗадания* — название задания, которое следует удалить, а остальные параметры при необходимости указывают удаленный компьютер, учетную запись пользователя, применяемую при удалении задания, и пароль для учетной записи, например:

```
schtasks /delete /tn "SysChecks"
```

или

```
schtasks /delete /tn "SysChecks"/s 192.168.1.100 /u adatum/
wrstanek /p frut5
```


Примечание Если вы укажете имя пользователя и не предоставите пароль, вам будет немедленно предложено ввести его.

Введя команду Schtasks /Delete, вы увидите предупреждение с запросом подтверждения на удаление задания. Нажмите нужную букву на клавиатуре. Если вы хотите отключить запрос на подтверждение, используйте параметр /f, например:

```
schtasks /delete /tn "SysChecks" /f
```

Здесь вы сообщаете Schtasks удалить задание без предупреждения.

Кроме того, если вы хотите удалить все назначенные задания на локальном или указанном удаленном компьютере, то вместо имени задания укажите звездочку (*), например:

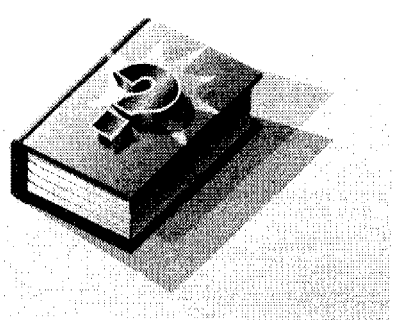
```
schtasks /delete /tn *
```

В ответ на запрос подтвердите выполняемое действие.

Часть II

Системное администрирование Windows

Microsoft Windows предоставляет множество инструментов командной строки, помогающих управлять рутинными операциями. Главы в этой части книги посвящены основным средствам и методам управления Windows-системами. В главе 5 описываются важнейшие средства администрирования, в том числе для сбора информации о системе, работы с реестром Windows, настройки служб Windows и удаленного выключения систем. В главе 6 рассматриваются инструменты, помогающие отслеживать информацию, которая записывается в журналы событий Windows, включая предупреждения и ошибки. Вы также научитесь регистрировать события в журналах системы и приложений. Наконец, в главе 7 вы освоите средства и методы мониторинга приложений, анализа процессов и поддержания производительности систем на должном уровне.



Глава 5

Управление Windows-системами

Вы, как администратор, должны планировать, организовывать и контролировать все компоненты, от которых зависит работа сети. И не просто добросовестно выполнять свои задачи, но и знать, как делать это быстро и эффективно. К счастью, Windows предоставляет множество средств командной строки, помогающих решать такие задачи, и эта глава посвящена некоторым из важнейших инструментов, предназначенных для повседневного управления системами.

Изучение системной информации

Часто при работе с компьютером пользователя или удаленным сервером возникает необходимость в получении базовой информации о системе вроде имени зарегистрированного в ней пользователя, текущего системного времени или местоположения определенного файла. Команды, которые позволяют собрать основную информацию о системе, включают:

- NOW — отображает текущую системную дату и время в 24-часовом формате, например Sat May 9 12:30:45 2003. Доступна только в Windows Server 2003 Resource Kit;
- WHOAMI — сообщает имя пользователя, зарегистрированного в системе на данный момент, например adatum\administrator;
- WHERE — выполняет поиск файлов по шаблону поиска (search pattern) и возвращает список совпавших результатов.

Чтобы использовать NOW или WHOAMI, просто введите команду в окне командной оболочки и нажмите Enter. Наиболее распространенный синтаксис для WHERE выглядит так:

```
where /r базовыйКаталог имя_файла
```

Здесь параметр */r* указан для рекурсивного поиска, начиная от указанного каталога (*\базовыйКаталог*) и включая все его подкаталоги, а *имя_файла* — полное или частичное имя искомого файла, которое может включать символы подстановки (wildcards): знак *?* заменяет один символ, а знак * — группу символов, например data???.txt или data*.*. В следующем примере в каталоге C:\ и всех его подкаталогах выполняется поиск всех текстовых файлов, имена которых начинаются с *data*:

```
where /r C:\ data*.txt
```

Также можно найти файлы всех типов, имена которых начинаются с *data*:

```
where /r C:\ data*.*
```

Иногда нужно получить информацию о конфигурации системы или о системном окружении. В критически важных системах эту информацию можно сохранить или распечатать для справки. Ниже перечислены команды, позволяющие собирать информацию о системе.

- **DRIVERQUERY** — выводит список всех установленных драйверов устройств и их свойства, в том числе имя модуля (module name), отображаемое имя (display name), тип драйвера и дату сборки (driver link date). В режиме отображения всей информации (/V) сообщается статус (status) и состояние (state) драйвера, режим запуска, сведения об использовании памяти и путь в файловой системе. Параметр /V также включает вывод детальной информации обо всех неподписанных драйверах.
- **SYSTEMINFO** — выдает подробную информацию о конфигурации системы, в том числе сведения о версии, типе и изготовителе операционной системы, процессоре, версии BIOS, объеме памяти, региональных стандартах, часовом поясе и конфигурации сетевого адаптера.
- **NLSINFO** — отображает подробную информацию о региональных стандартах, включая язык по умолчанию (default language), кодовую страницу Windows, форматы отображения времени и чисел, часовой пояс и установленные кодовые страницы. Эта команда доступна лишь в Windows Server 2003 Resource Kit.

Чтобы использовать эти команды на локальном компьютере, просто введите имя нужной команды в окне командной оболочки и нажмите Enter. В команде DRIVERQUERY параметр

/V вызывает вывод всей информации, а параметр /Si — отображение свойств только подписанных драйверов, например:

```
driverquery /v /si
```

В командах DRIVERQUERY и SYSTEMINFO можно также указать опрашиваемый удаленный компьютер и разрешения Run As (Запуск от имени). Для этого следует использовать расширенный синтаксис, который включает следующие параметры:

```
/S Компьютер /U [Домен\]Пользователь [/P Пароль]
```

где *Компьютер* — имя или IP-адрес удаленного компьютера, *Домен* — необязательное имя домена, где находится учетная запись пользователя, *Пользователь* — имя учетной записи пользователя, чьи разрешения вы хотите задействовать, а *Пароль* — необязательный пароль для учетной записи. Если домен не указан, берется текущий домен. А если вы не зададите пароль учетной записи, он будет запрошен у вас.

Чтобы понять, как добавлять в синтаксис команд информацию о компьютере и пользователе, рассмотрим пару примеров.

Использование учетной записи adatum\wrstanek при опросе MAILER1 о параметрах драйверов:

```
driverquery /s mailer1 /u adatum\wrstanek
```

Использование учетной записи adatum\administrator при запросе информации о системе CORPSERVER01:

```
systeminfo /s corpserver01 /u adatum\administrator
```

Совет Основной вывод этих команд представляется в форме таблицы. Вы также можете форматировать вывод в список или в строки разделенных запятыми значений, используя соответственно /Fo List или /Fo Csv. Вероятно, вас интересует, зачем нужны разные форматы. Хороший вопрос. Я рекомендую использовать формат детального списка (/Fo List /V), когда вам нужны все подробности о системе или когда вы занимаетесь устранением неполадок. Разделенные запятыми значения лучше использовать при записи вывода в файл, который впоследствии может быть экспортирован в таблицу или в плоскую (неиерархическую) базу данных. Помните, что вывод команд DRIVERQUERY и SYSTEMINFO можно перенаправлять в файл с помощью > или >>.

Работа с реестром

Реестр Windows хранит параметры конфигурации. С помощью утилиты командной строки Reg можно просматривать, добавлять, удалять, сравнивать и копировать элементы реестра. Поскольку реестр Windows крайне важен для корректной работы операционной системы, вносите изменения в реестр, лишь точно зная, как они повлияют на систему. Прежде чем редактировать реестр каким-либо способом, выполните полное резервное копирование системы и создайте снимок данных для восстановления системы (system recovery data snapshot). Тогда в случае ошибки вы сможете восстановить реестр и систему.

Внимание! Некорректная модификация реестра Windows чревата серьезными проблемами. При повреждении реестра может потребоваться переустановка операционной системы. Дважды проверьте используемые команды, прежде чем запускать их. Убедитесь, что они делают именно то, что было задумано.

Разделы и параметры реестра

Реестр Windows хранит конфигурационную информацию операционной системы, приложений, пользователей и оборудования. Эти данные содержатся в разделах (keys) и параметрах (values) реестра, которые размещаются в определенном корневом разделе (root key), который контролирует, как и когда используются разделы и параметры.

В табл. 5-1 приведены корневые разделы реестра, их описания и ссылочные имена (reference names) для ссылок на корневые разделы при работе с командой REG. В корневых разделах вы найдете главные разделы (main keys), управляющие параметрами системы, пользователей, приложений и оборудования. Эти разделы организованы в древовидную структуру, где разделы представлены папками. Так, в разделе HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services находятся папки для всех служб, установленных в системе. В этих папках содержатся разделы реестра, хранящие важную конфигурационную информацию, и их подразделы (subkeys).

Табл. 5-1. Разделы реестра Windows

Корневой раздел	Имя для ссылок	Описание
HKEY_CURRENT_USER	HKCU	Конфигурационные параметры для текущего пользователя
HKEY_LOCAL_MACHINE	HKLM	Конфигурационные параметры системного уровня
HKEY_CLASSES_ROOT	HKCR	Конфигурационные параметры для приложений и файлов. Также обеспечивает открытие нужного приложения при обращении к файлу
HKEY_USERS	HKU	Параметры для пользователя по умолчанию (default user) и для других пользователей по профилям
HKEY_CURRENT_CONFIG	HKCC	Информация об используемом профиле оборудования

Нужные разделы реестра задаются через их пути к папкам. Например, путь к разделу DNS выглядит как HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS, и вы, используя сокращенный путь HKLM\SYSTEM\CurrentControlSet\Servies\DNS, можете просматривать этот раздел и манипулировать над ним.

Для значений параметров определены специальные типы данных. В табл. 5-2 дана сводная информация об основных типах данных.

Табл. 5-2. Значения и типы данных параметров реестра

Тип данных	Описание	Пример
REG_BINARY	Представляет двоичный параметр. Двоичные параметры хранятся как двоичные (только комбинация нулей и единиц), но отображаются и вводятся в шестнадцатеричном формате	01 00 14 80 90 00 00 9c 00
REG_SZ	Представляет строковый параметр, содержащий последовательность символов	DNS Server

Табл. 5-2. *(окончание)*

Тип данных	Описание	Пример
REG_DWORD	Представляет параметр типа DWORD, который состоит из шестнадцатеричных данных с максимальной длиной в четыре байта	0x00000002
REG_MULTI_SZ	Представляет параметр из нескольких строк	Tcpip Afd RpcSc
REG_EXPAND_SZ	Представляет раскрываемый строковый параметр (expandable string value), который обычно используется в путях к каталогам	*%SystemRoot%*\dns.exe

Если вы знаете пути к разделам и понимаете допустимые типы данных в разделах, то можете использовать команду REG для просмотра разделов и параметров и манипуляций над ними самыми разнообразными способами. REG поддерживает несколько подкоманд, и мы рассмотрим некоторые из них. В следующих разделах рассказывается о каждой из подкоманд REG, перечисленных ниже.

- **REG add** — добавляет в реестр новый подраздел или элемент.
- **REG delete** — удаляет из реестра подраздел или элемент.
- **REG query** — выводит список элементов раздела и имена подразделов (если они есть).
- **REG compare** — сравнивает подразделы или элементы реестра.
- **REG copy** — копирует элемент реестра по указанному пути раздела на локальной или удаленной системе.
- **REG restore** — записывает в реестр ранее сохраненные подразделы, элементы и параметры.
- **REG save** — сохраняет копию указанных подразделов, элементов и параметров реестра в файл.

Примечание Команда REG выполняется с разрешениями текущего пользователя. Самый простой способ перейти на другой набор разрешений — войти в систему под учетной записью нужного пользователя.

Просмотр параметров реестра

REG query позволяет читать параметры реестра, указав полный путь и имя нужного раздела или параметр. Основной синтаксис таков:

```
reg query ИмяРаздела [/v ИмяПараметра]
```

где *ИмяРаздела* — имя нужного раздела, а *ИмяПараметра* — необязательный параметр, указывающий определенный параметр раздела. В следующем примере отображается раздел DNS из текущего набора управления (current control set):

```
reg query HKLM\SYSTEM\CurrentControlSet\Services\DNS
```

Или, если вам требуется определенный параметр реестра, можно ограничить результаты запроса, применив параметр /V. В этом примере отображается значение элемента ImagePath из раздела DNS:

```
reg query HKLM\SYSTEM\CurrentControlSet\Services\DNS /v
ImagePath
```

Путь раздела также может включать UNC-имя или IP-адрес удаленного компьютера, который вы хотите проверить, например \\Mailer1 или \\192.168.1.100. Но помните: на удаленном компьютере можно работать только с корневыми разделами HKLM и HKU. В следующем примере просматривается раздел DNS на MAILER1:

```
reg query \\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS
```

Примечание Если указан несуществующий раздел или параметр, появится сообщение об ошибке. Обычно оно выглядит так: «ERROR: The system was unable to find the specified registry key or value».

Сравнение разделов реестра

С помощью REG compare можно сравнивать элементы и параметры реестров двух систем или разных разделов в одной системе. Сравнение элементов реестра полезно в следующих ситуациях.

- **При устранении проблем в конфигурации служб и приложений.** В таких случаях полезно сравнить конфигурации реестров двух разных систем. Идеально, если одна из этих систем настроена правильно, а на другой вы подозреваете

ошибку в конфигурации. Тогда можно сравнить те параметры, которые по вашему мнению вызывают проблемы.

- **Когда надо убедиться, что приложения или службы на разных системах настроены одинаково.** Здесь одна система используется как эталон для проверки конфигурации остальных систем. В идеале эталонная система до начала сравнения ее конфигурации с другими системами настраивается именно так, как нужно.

Основной синтаксис REG compare таков:

```
reg compare ИмяРаздела1 ИмяРаздела2 [/v ИмяПараметра]
```

где *ИмяРаздела1* и *ИмяРаздела2* — имена сравниваемых подразделов, а *ИмяПараметра* — необязательный параметр, указывающий для сравнения определенный параметр в разделе. Имя раздела может включать UNC-имя или IP-адрес удаленного компьютера, который вы хотите проверить. В следующем примере на MAILER1 и MAILER2 сравниваются разделы DNS из текущих наборов управления:

```
reg compare
\\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS
\\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS
```

Если разделы настроены одинаково, выводится:

```
Results Compared: Identical
The operation completed successfully.
```

А если по-разному — в выводе показываются отличия. Все отличия, предваряемые символом <, принадлежат первому из указанных разделов, а отличия, начинающиеся с символа >, — второму. Также появится сообщение:

```
Results Compared: Different
The operation completed successfully.
```

Совет Различия показываются потому, что по умолчанию предполагается параметр /Od. Используя дополнительные параметры, можно также добиться вывода всех различий и совпадений (/Oa), только совпадений (/Os) или только результата сравнения (/On).

Кроме того, если хотите рекурсивно сравнить все подразделы и элементы, добавьте параметр /S:

```
reg compare
\\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS
\\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS /s
```

Теперь будет выполнено сравнение разделов, всех подразделов и соответствующих элементов разделов DNS на MAILER1 и MAILER2.

Сохранение и восстановление разделов реестра

Прежде чем модифицировать элементы реестра, полезно сохранить разделы, с которыми вы будете работать. Если что-то пойдет не так, вы сможете восстановить исходное состояние этих разделов. Чтобы сохранить копию подраздела реестра и всех относящихся к нему подразделов и параметров, используйте команду REG save:

```
reg save ИмяРаздела "ИмяФайла"
```

где *ИмяРаздела* — путь к сохраняемому подразделу, а *ИмяФайла* — текстовое имя файла куста реестра (registry hive), который вы хотите создать. Путь к подразделу может включать UNC-имя или IP-адрес удаленного компьютера. Но на удаленном компьютере можно работать только с корневыми разделами HKLM и HKU. Кроме того, имя файла следует заключать в двойные кавычки, и оно должно заканчиваться расширением .hiv, обозначающим файл куста реестра, как показано в следующем примере:

```
reg save HKLM\SYSTEM\CurrentControlSet\Services\DNS
"DNSKey.hiv"
```

Здесь в файл с именем Dnskey.hiv сохраняются подраздел DNS и все его подразделы и параметры. Имя файла может включать путь к каталогу:

```
reg save \\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS
"\\Mailer1\SavedData\DNSKey.hiv"
```

Если файл куста реестра существует, у вас будет запрошено подтверждение на перезапись этого файла. Нажмите Y, чтобы перезаписать файл. Для перезаписи без запроса подтверждения используйте параметр /Y.

Для восстановления ранее сохраненных разделов реестра используется команда Reg restore. Ее синтаксис выглядит так:

```
reg restore ИмяРаздела "ИмяФайла"
```

где *ИмяРаздела* — путь к подразделу, который следует восстановить, а *ИмяФайла* — текстовое имя файла куста реестра, который служит источником восстановления. В отличие от REG copy команда REG restore применима лишь на локальном компьютере, т. е. с ее помощью нельзя восстанавливать разделы реестра на удаленном компьютере. Однако вы можете запустить сеанс удаленного рабочего стола на удаленном компьютере и использовать вход через удаленный рабочий стол для восстановления раздела реестра на удаленном компьютере.

Вот пример использования REG restore:

```
reg restore HKLM\SYSTEM\CurrentControlSet\Services\DNS
"DNSKey.hiv"
```

Здесь восстанавливается раздел DNS, ранее сохраненный в файл DNSKey.hiv.

Добавление разделов реестра

Для добавления подразделов и параметров в реестр Windows используется команда REG add. Основной синтаксис для создания раздела или параметра имеет вид:

```
reg add ИмяРаздела /v ИмяПараметра /t ТипДанных /d Данные
```

где *ИмяРаздела* — имя нужного раздела, *ИмяПараметра* — создаваемый параметр раздела или подраздела, *ТипДанных* — используемый тип данных, а *Данные* — реальное значение, помещаемое в реестр. Кажется, что значений многовато, но они вполне очевидны. Рассмотрим пример:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\DNS /v
DisplayName /t REG_SZ /d "DNS Server"
```

Здесь в раздел DNS реестра добавляется параметр DisplayName. Элемент раздела — строка со значением «DNS Server». Обратите внимание на двойные кавычки. В этом примере они необходимы, поскольку строка содержит пробел.

При установке значений для раскрываемых строк (REG_EXPAND_SZ) следует указывать знак ^, предваряя им символы процента (%), которые обозначают используемую переменную окружения. Рассмотрим пример:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\DNS /v
ImagePath /t REG_EXPAND_SZ /d ^%SystemRoot^%\System32\dns.exe
```

Как видите, чтобы переменная окружения *SystemRoot* была правильно введена и интерпретирована, указывается **^%SystemRoot^%**.

При установке значений, отличных от строковых, кавычки не нужны:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\DNS /v
ErrorControl /t REG_DWORD /d 0x00000001
```

Копирование разделов реестра

REG copy позволяет копировать элемент реестра в новое место в локальной или удаленной системе. Основной синтаксис REG copy выглядит так:

```
reg copy ИмяРаздела1 ИмяРаздела2
```

где *ИмяРаздела1* — путь к копируемому разделу, а *ИмяРаздела2* — путь к подразделу-приемнику. Хотя пути к подразделам могут включать UNC-имя или IP-адрес удаленного компьютера, REG copy имеет ограниченную область действия в отношении корневых разделов, используемых при работе с удаленным источником или приемником копируемого раздела.

- В качестве удаленного источника возможны лишь подразделы из корневых разделов HKLM или HKU.
- В качестве удаленного приемника возможны лишь подразделы из корневых разделов HKLM или HKU.

Ниже подраздел DNS из локальной системы копируется в подраздел DNS на MAILER2:

```
reg copy HKLM\SYSTEM\CurrentControlSet\Services\DNS
    \\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS
```

Добавив параметр /S, можно копировать указанный подраздел вместе со всеми его подразделами и элементами:

```
reg copy HKLM\SYSTEM\CurrentControlSet\Services\DNS
    \\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS /s
```

Если параметры в месте назначения уже существуют, REG copy попросит подтвердить перезапись каждого из них. Нажимайте Y или N в соответствии с нужным действием. Кроме того, можно нажать A, чтобы перезаписать все существующие параметры без дальнейших вопросов.

Примечание Если вы не хотите появления запросов на подтверждение, используйте параметр /F, чтобы выполнять перезапись без подтверждения. Но, прежде чем копировать поверх существующих разделов реестра, желательно сначала сохранить их, чтобы в случае проблем их можно было восстановить. Для этого предназначены команды REG save и REG restore, которые описаны в разделе «Сохранение и восстановление разделов реестра» ранее в этой главе.

Удаление разделов реестра

Для удаления подразделов и параметров реестра Windows используется команда REG delete. Она поддерживает несколько синтаксисов. Чтобы удалить подраздел вместе со всеми его подразделами и элементами, используйте синтаксис:

```
reg delete ИмяРаздела
```

где *ИмяРаздела* — имя удаляемого раздела. Хотя путь к подразделу может включать UNC-имя или IP-адрес удаленного компьютера, на удаленном компьютере можно использовать только подразделы корневых разделов HKLM или HKU. Рассмотрим пример:

```
reg delete
\\Mailer1\HKLM\SYSTEM\CurrentControlSet\Services\DNS2
```

Здесь на MAILER1 удаляется подраздел DNS2 вместе со всеми его подразделами и элементами.

Чтобы ограничить область действия команды, при помощи следующего синтаксиса укажите, что в подразделе следует удалить только определенный элемент:

```
reg delete ИмяРаздела /v ИмяПараметра
```

где *ИмяРаздела* — имя нужного подраздела, а *ИмяПараметра* — имя удаляемого элемента. И вновь путь к подразделу может включать UNC-имя или IP-адрес удаленного компьютера, но на удаленном компьютере допускаются операции только с подразделами корневых разделов HKLM или HKU. В этом примере на MAILER2 из подраздела DNS2 удаляется элемент Description:

```
reg delete
\\Mailer2\HKLM\SYSTEM\CurrentControlSet\Services\DNS2 /v
Description
```

Совет В обоих случаях вам потребуется подтвердить окончательное удаление указанного элемента. Чтобы сделать это, нажмите Y. Используя параметр /F, можно выполнять удаление без запроса на подтверждение. Еще один полезный параметр — /Va. Он указывает, что удалять следует только элементы подраздела. В таком случае никакие подразделы этого подраздела не удаляются.

Управление системными службами

Службы обеспечивают ключевые функции рабочих станций и серверов. Для управления системными службами на локальных и удаленных системах используется команда контроллера служб (service controller command) SC, имеющая набор подкоманд; здесь описывается лишь их часть. В следующих разделах я расскажу о подкомандах:

- **SC config** — настройка учетных записей регистрации и запуска служб;
- **SC query** — вывод списка всех служб, настроенных на компьютере;
- **SC qc** — отображение конфигурации определенной службы;
- **SC start** — запуск служб;
- **SC stop** — останов служб;
- **SC pause** — приостановка работы служб;
- **SC continue** — возобновление работы служб;
- **SC failure** — задание действий, выполняемых при сбое службы;
- **SC qfailure** — просмотр действий, выполняемых при сбое службы.

Во всех командах можно указывать имя удаленного компьютера, со службами которого вы хотите работать. Для этого вставьте UNC-имя или IP-адрес компьютера перед используемой подкомандой. Вот синтаксис:

```
sc ИмяСервера Подкоманда
```

Просмотр настроенных служб

Чтобы получить список всех служб, настроенных в системе, введите в командной строке команду:

```
sc query type= service state= all
```

или

```
sc ИмяСервера query type= service state= all
```

где *ИмяСервера* — UNC-имя или IP-адрес удаленного компьютера, например \\Mailer1 или \\192.168.1.100, как показано в следующих примерах:

```
sc \\Mailer1 query type= service state= all
sc \\192.168.1.100 query type= service state= all
```

Примечание После знака равенства должен быть пробел, как в *type= service* и *state= all*. Если пробел не вставить, команда не выполнится.

Флагу *state* можно присвоить значение *active* (для отображения только работающих служб) или *inactive* (для отображения всех приостановленных или остановленных служб). Вот примеры:

```
sc \\Mailer1 query type= service state= active
sc \\Mailer1 query type= service state= inactive
```

В первом примере у MAILER1 запрашивается список всех работающих служб, а во втором — список всех остановленных служб.

SC query выводит информацию о службах и их конфигурациях. Записи для каждой службы форматируются так:

```
SERVICE_NAME: W3SVC
DISPLAY_NAME: World Wide Web Publishing Service
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 4  RUNNING
                              (STOPPABLE, PAUSABLE,
ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

Как администратору, вам чаще всего будут нужны следующие поля*.

- **Service Name** Сокращенное имя службы. Здесь перечисляются только установленные в системе службы. Если нужная вам служба не указана, ее придется установить.
- **Display Name** Описательное имя службы.
- **State** Состояние службы: Running (работает), Paused (приостановлена) или Stopped (остановлена).

---

* Команда SC на русский язык не переведена. — *Прим. перев.*

Запустив команду SC query, вы увидите, что ее вывод очень длинный, поэтому лучше использовать фильтры для получения только нужной информации. Например, если в выводе следующей команды будут содержаться лишь самые важные поля:

```
sc query type= service | find /v "x0"
```

Здесь вывод SC query проходит через команду FIND и фильтруется, так что записи для служб отображаются, как показано в этом примере:

```
SERVICE_NAME: W3SVC
DISPLAY_NAME: World Wide Web Publishing Service
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 4  RUNNING
                              (STOPPABLE, PAUSABLE,
ACCEPTS_SHUTDOWN)
```

Примечание Параметр /V "x0" предписывает команде FIND отображать лишь строки вывода, не содержащие текста *x0*, который обычно присутствует в полях WIN32_Exit_Code, Service_Exit_Code, Checkpoint и Wait_Hint.

Если вам известно имя нужной службы, вы можете получить информацию о ее конфигурации командой SC qc. Ее синтаксис таков:

```
sc qc ИмяСлужбы
```

где *ИмяСлужбы* — имя службы, информацию о которой вы хотите получить. Вывод для отдельной службы выглядит так:

```
SERVICE_NAME: w3svc
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\WINDOWS\System32\svchost.exe -k
         iissvcs
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : World Wide Web Publishing Service
        DEPENDENCIES       : RPCSS
                           : HTTPFilter
                           : IISADMIN
        SERVICE_START_NAME : LocalSystem
```

Обратите внимание, что в выводе не сообщается о текущем состоянии службы. Но в нем показывается следующее.

- **Binary Path Name** — путь и имя исполняемого файла службы.
- **Dependencies** — службы, которые должны работать, чтобы работала и указанная вами служба.
- **Display Name** — описательное имя службы.
- **Service Start Name** — имя учетной записи пользователя, от имени которого служба регистрируется в системе.
- **Start Type** — стартовая конфигурация службы.

Примечание Если служба настроена на автоматический запуск, сообщается AUTO_START, а если она настроена на запуск вручную — DEMAND_START. Для отключенных служб указывается DISABLED.

- **Type** — тип службы и является ли она общим процессом (shared process).

Примечание Конфигурируя для службы вход в систему, иногда важно знать, работает процесс в собственном контексте или является общим. Для общих процессов указывается WIN32_SHARE_PROCESS. Для процессов, выполняемых в собственном контексте, сообщается WIN32_OWN_PROCESS.

Запуск, останов и приостановка служб

Как администратору, вам придется часто запускать, останавливать или приостанавливать службы Windows. Вот соответствующие команды SC и их синтаксис.

Запуск службы:

```
sc start ИмяСлужбы
```

Приостановка службы:

```
sc pause ИмяСлужбы
```

Возобновление работы приостановленной службы:

```
sc continue ИмяСлужбы
```

Останов службы:

```
sc stop ИмяСлужбы
```

где *ИмяСлужбы* — сокращенное имя нужной вам службы, например:

```
sc start w3svc
```

Как и во всех командах SC, вы можете также задать имя удаленного компьютера, со службами которого вы хотите работать. Например, чтобы запустить w3svc на MAILER1, используйте команду:

```
sc \\Mailer1 start w3svc
```

В выводе должно быть указано состояние START_PENDING. При останове, паузе или возобновлении работы службы вы увидите соответственно STOP_PENDING, PAUSE_PENDING или CONTINUE_PENDING. Если в результате возникла ошибка, сообщается FAILED и приводится текст ошибки, описывающий причину сбоя более подробно. Если вы попытаетесь запустить уже запущенную службу, то увидите сообщение об ошибке:

```
An instance of the service is already running.
```

Если вы попытаетесь остановить уже остановленную службу, то увидите сообщение об ошибке:

```
The service has not been started.
```

Настройка запуска службы

Службы Windows можно настроить на автоматический запуск или запуск вручную. Кроме того, их можно вовсе выключить. Запуск служб настраивается командой:

```
sc config ИмяСлужбы start= флаг
```

где *ИмяСлужбы* — сокращенное имя нужной службы, а *флаг* — используемый тип запуска. Для служб допустимыми значениями флага являются:

- **Auto** — запуск службы при старте системы;
- **Demand** — позволяет запустить службу вручную;
- **Disabled** — отключает службу.

Таким образом, настроить службу для автоматического запуска можно так:

```
sc config w3svc start= auto
```

или

```
sc \\Mailer1 config w3svc start= auto
```

Примечание После знака равенства должен быть пробел, как в *start= auto*. Если пробел не вставить, команда не выполнится. Также заметьте, что команда сообщает лишь об успехе (SUCCESS) или неудаче (FAILURE) операции. Она не информирует, что служба уже была настроена на указанный режим запуска.

Внимание! Отключение службы не остановит запущенную службу. Оно лишь предотвратит запуск службы при следующей загрузке компьютера. Чтобы отключить и остановить службу, используйте SC stop, а затем SC config.

Настройка регистрации службы

Службы Windows можно настраивать для регистрации под системной учетной записью или учетной записью определенного пользователя. Чтобы служба регистрировалась под учетной записью LocalSystem, используйте:

```
sc config ИмяСлужбы obj= LocalSystem
```

где *ИмяСлужбы* — имя службы, настраиваемой для использования учетной записи LocalSystem. Если служба предоставляет пользовательский интерфейс, которым можно управлять, добавьте флаги **type= interact type= own**, как в следующем примере:

```
sc config w3svc obj= LocalSystem type= interact type= own
```

Флаг *type= interact* указывает, что служба может взаимодействовать с рабочим столом Windows, а флаг *type= own* — что она выполняется в собственном процессе. Если служба использует свои исполняемые файлы совместно с другими службами, следует указывать флаг *type= share*, как показано в примере:

```
sc config w3svc obj= LocalSystem type= interact type= share
```


Совет Если вы не знаете, работает служба в общем процессе или в собственном контексте, используйте SC qc, чтобы определить тип запуска службы. Эта команда описывается в разделе «Просмотр настроенных служб» ранее в этой главе.

Службы также могут регистрироваться по учетным записям пользователей. Для этого применяется синтаксис:

```
sc config ИмяСлужбы obj= [Домен\]Пользователь password= Пароль
```

где *Домен* — необязательное имя домена, где находится учетная запись пользователя, *Пользователь* — имя учетной записи пользователя, чьи разрешения вы хотите задействовать, а *Пароль* — пароль для этой учетной записи. Рассмотрим пример:

```
sc config w3svc obj= adatum\webbies password= blue5!CraZy
```

Здесь служба W3svc настраивается на использование учетной записи Webbies из домена Adatum. Вывод команды сообщает SUCCESS или FAILED. Изменение не удастся (FAILED) при недопустимом или несуществующем имени учетной записи, а также при неверном пароле для учетной записи.

Примечание Если служба ранее была настроена на взаимодействие с рабочим столом под учетной записью LocalSystem, эту службу нельзя перенастроить на запуск под учетной записью домена без флага *type= own*. Поэтому синтаксис становится таким: sc config **ИмяСлужбы** obj= **[Домен\]Пользователь** password= **Пароль** type= own.

Внимание! Как администратор, вы должны следить за всеми учетными записями, которые используются службами. При неправильной настройке эти учетные записи могут стать источником серьезных проблем для безопасности. Учетным записям служб следует назначать жесточайшие параметры безопасности и предоставлять минимально возможное число разрешений — только чтобы службы могли выполнять необходимые функции. Обычно учетным записям служб не требуются многие разрешения, назначаемые типичным учетным записям пользователей. Так, большинству учетных записей служб не нужны права на локальный вход в систему (log on locally). Каждый администратор должен знать о применяемых учетных записях служб (чтобы лучше следить за их использованием) и обращаться с ними как с учетными записями администраторов. То есть назначать надежные пароли, тщательно следить за использованием этих учетных записей, осторожно применять разрешения и привилегии для учетной записи и т. д.

Настройка восстановления служб

Команда SC failure позволяет настроить службы Windows для выполнения определенных действий при сбое. Например, можно попытаться перезапустить службу или запустить какое-то приложение.

Параметры восстановления можно настроить для первой, второй и последующих попыток восстановления. Текущий счетчик неудач увеличивается при каждом сбое. Кроме того, можно задать время, которое должно пройти до сброса счетчика неудач (failure counter). Например, сбросить счетчик неудач, если с момента последнего сбоя прошло 24 часа.

Прежде чем настраивать восстановление службы, проверьте текущие параметры восстановления с помощью SC qfailure. Вот ее синтаксис:

```
sc qfailure ИмяСлужбы
```

где *ИмяСлужбы* — имя нужной службы, например:

```
sc qfailure w3svc
```

Разумеется, можно указать и удаленный компьютер:

```
sc \\Mailer1 qfailure w3svc
```

или

```
sc \\192.168.1.100 qfailure w3svc
```

В выводе действия при неудачах перечисляются в том порядке, в каком они выполняются. В следующем примере вывода служба W3svc была настроена на попытку перезапуска при первом и втором сбое службы и на перезапуск компьютера при третьем:

```
[SC] QueryServiceConfig2 SUCCESS

SERVICE_NAME: w3svc
       RESET_PERIOD (in seconds)    : 86400
       REBOOT_MESSAGE               :
       COMMAND_LINE                 :
       FAILURE_ACTIONS              : RESTART - Delay = 1
milliseconds.
                                      RESTART - Delay = 1
milliseconds.
                                      REBOOT - Delay = 1000
milliseconds.
```

Примечание При установке Windows восстановление некоторых критически важных системных служб настраивается автоматически (обычно на попытку перезапуска, но некоторые из них — на запуск определенных программ). Например, служба IIS Admin настраивается на запуск программы iisreset.exe при сбое. Эта программа устраняет проблемы, возникшие с данной службой, и безопасно управляет зависимыми IIS-службами в процессе перезапуска IIS Admin.

Для настройки восстановления службы предназначена команда SC failure. Вот ее основной синтаксис:

```
sc failure ИмяСлужбы reset= ИнтервалСбросаСчетчика actions=
Действия
```

где *ИмяСлужбы* — имя настраиваемой службы, *ИнтервалСбросаСчетчика* указывает время в секундах, которое должно пройти без сбоев, чтобы счетчик неудач был сброшен, а *Действия* — это действия, которые следует предпринять при сбое, плюс время задержки (в миллисекундах) перед инициацией этих действий. Действия могут быть следующими.

- **Take No Action** — операционная система не будет пытаться восстанавливать службу при этом сбое, но может пытаться восстанавливать службу при предыдущих или последующих сбоях.
- **Restart The Service** — останов и последующий запуск службы после короткой паузы.
- **Run A Program** — позволяет в случае сбоя запустить программу или сценарий. Сценарием может быть командный файл или Windows-сценарий. При выборе этого варианта укажите полный путь к файлу программы, которую следует запустить, и установите все необходимые параметры командной строки, передаваемые программе при запуске.
- **Reboot The Computer** — выход из операционной системы и перезагрузка компьютера по истечении указанного времени задержки.

Совет Настраивая параметры восстановления для критически важных служб, попробуйте перезапускать службу при первом и втором сбое, а на третьей попытке перезагружать сервер.

Работая с SC failure, помните о следующем.

- **Интервал сброса счетчика задается в секундах.** Интервал сброса обычно устанавливают на несколько часов или дней. В одном часе 3600 секунд, а в одном дне — 86400. Например, чтобы установить двухчасовой период сброса, укажите значение 7200.
- **После каждого действия, связанного с восстановлением, надо указывать время ожидания (в миллисекундах) до выполнения этого действия.** Для перезапуска службы, вероятно, лучше подойдет короткая задержка, например 1 миллисекунда (без задержки), 1 секунда (1000 миллисекунд) или 5 секунд (5000 милисекунд). Для перезагрузки компьютера, видимо, лучше использовать больший период, скажем, 15 секунд (15000 миллисекунд) или 30 (30000 миллисекунд).
- **Действия и соответствующие интервалы задержки вводятся как одна текстовая запись с разделением косой чертой (/).** Например, вы можете указать значение restart/1000/restart/1000/reboot/15000. Здесь на первой и второй попытках служба перезапускается с задержкой в одну секунду, а на третьей попытке после задержки в 15 секунд перезагружается компьютер.

Рассмотрим пару примеров.

```
sc failure w3svc reset= 86400 actions= restart/1/restart/1/
reboot/30000
```

Здесь при первой и второй попытках служба перезапускается почти сразу же, а на третьей попытке после 30 секундной задержки перезагружается компьютер. Кроме того, счетчик неудач сбрасывается, если прошло 24 часа без сбоев (86400 секунд). Также можно указать удаленный компьютер, вставив UNC-имя или IP-адрес, как было показано в предыдущих примерах.

При использовании действия Run, исполняемая команда или запускаемая программа указываются с помощью параметра *Command=*. Укажите после *Command=* полный путь для исполняемой команды и все передаваемые аргументы. Не забудьте заключить путь команды и текст в двойные кавычки, как показано в следующем примере:

```
sc failure w3svc reset= 86400 actions= restart/1/restart/1/
run/30000 command= "c:\restart_w3svc.exe 15"
```

Перезагрузка и выключение систем из командной строки

Системы нередко приходится перезагружать или выключать. Один из способов — использовать для этого утилиту Shutdown, которая позволяет работать с локальной и удаленными системами. Другой способ управлять выключением или перезагрузкой системы — назначить задание для выключения. Здесь можно использовать Schtasks, чтобы указать время выключения, или создать сценарий со списком команд выключения для индивидуальных систем.

Примечание Хотя Windows-системы обычно запускаются и выключаются без проблем, иногда во время этих процессов они перестают отвечать. Если так и случилось, попробуйте определить причину. Вот некоторые из причин, по которым система перестает отвечать:

1. Система пытается выполнить или выполняет сценарий загрузки/выключения, который еще не завершен или сам не отвечает (в таком случае система может ждать таймаута сценария).
2. Причиной проблемы может быть служба или файл стартовой инициализации; в этом случае вам может потребоваться отладка стартовых элементов с помощью утилиты конфигурирования системы — System Configuration Utility (Msconfig). Также проблему может решить отключение службы, стартового элемента или записи в файле стартовой инициализации.
3. В системе может находиться антивирусная программа, создающая проблему. В некоторых случаях антивирусные программы при выключении системы пытаются сканировать дисковод. Для решения этой проблемы настройте антивирусную программу так, чтобы при выключении она не сканировала дисковод или другие устройства со сменными носителями. Кроме того, попробуйте временно отключить или остановить антивирусную программу.
4. Проблемы при запуске или выключении могут вызывать неверно настроенные звуковые устройства. Чтобы определить возможный источник проблем, проверьте каждое такое устройство по очереди. Отключите звуковое устройство и перезапустите компьютер. Если проблема исчезла, вам

следует установить новые драйверы для используемого звукового устройства; также возможно, что у вас поврежден звуковой файл, проигрываемый при запуске (Start Windows) или завершении работы Windows (Exit Windows).

5. Проблемы при запуске или выключении могут вызывать неверно настроенные сетевые адаптеры. Попробуйте отключить сетевой адаптер и перезагрузиться. Если это сработало, вам может потребоваться удалить и переустановить драйвер адаптера или получить новый драйвер от производителя.
6. Проблемы при запуске или выключении могут вызывать неверно настроенные драйверы видеоадаптера. Удаленно зарегистрируйтесь в системе с другого компьютера и попробуйте откатить (roll back) текущие видеодрайверы до предыдущей версии. Если это невозможно, попробуйте удалить и переустановить видеодрайверы.

Управление перезагрузкой и выключением локальной системы

Управлять перезагрузкой и выключением локальной системы позволяют следующие команды.

Выключение локальной системы:

```
shutdown /s /t ЗадержкаВыключения /l /f
```

Перезагрузка локальной системы:

```
shutdown /r /t ЗадержкаВыключения /l /f
```

Отмена задержанного выключения локального компьютера:

```
shutdown /a
```

где /T *ЗадержкаВыключения* используется для установки произвольного числа секунд задержки перед выключением или перезагрузкой, необязательный параметр /L обеспечивает немедленный выход текущего пользователя из системы, а необязательный параметр /F закрывает все работающие приложения без предварительного предупреждения пользователей. В этом примере выполняется перезагрузка локальной системы после 60-секундной задержки:

```
shutdown /r /t 60
```

Совет В большинстве сетевых сред главным приоритетом является время непрерывной работы системы. Перезагружающиеся или выключенные системы недоступны пользователям, из-за чего, возможно, кто-то не сумеет закончить свою работу и будет недоволен. Вместо того чтобы выключать системы посреди рабочего дня, старайтесь делать это до или после рабочего времени. Но если вам все же необходимо выключить систему в рабочее время, по возможности предупредите пользователей заранее, позволив им сохранить текущую работу и выйти из системы.

Управление перезагрузкой и выключением удаленных систем

Для удаленных систем через параметр /M следует указать UNC-имя или IP-адрес системы, которую вы хотите выключить или перезагрузить. Таким образом, основной синтаксис для выключения, перезагрузки и отмены задержанного выключения принимает следующий вид.

Выключение удаленной системы:

```
shutdown /s /t ЗадержкаВыключения /l /f /m \\Система
```

Перезагрузка удаленной системы:

```
shutdown /r /t ЗадержкаВыключения /l /f /m \\Система
```

Отмена задержанного выключения удаленного компьютера:

```
shutdown /a /m \\Система
```

В этом примере MAILER1 перезагружается после 30-секундной задержки:

```
shutdown /r /t 30 /m \\Mailer1
```

А здесь система с IP-адресом 192.168.1.105 перезагружается немедленно, и все запущенные приложения принудительно останавливаются:

```
shutdown /r /f /m \\192.168.1.105
```

Добавление комментариев по причинам перезагрузки или выключения

В большинстве сетевых сред желательно документировать причины выключения или перезагрузки компьютеров. При внеплановых выключениях документируйте причины в системном журнале, расширив синтаксис следующими параметрами:

```
/e /c "ВнеплановаяПричина" /d ОсновнойКод:ДополнительныйКод
```

где /C "*ВнеплановаяПричина*" сообщает детальное описание (которое может быть длиной до 127 символов) причины выключения или перезагрузки, а /D *ОсновнойКод:ДополнительныйКод* — код причины выключения. Коды причин могут быть произвольными — со значениями основного кода 0–255 и дополнительного кода 0–65535. Рассмотрим следующий пример:

```
shutdown /r /e /m \\Mailer1 /c "System Reset" /d 5:15
```

В этом примере MAILER1 перезагружается, и причина внеплановой перезагрузки документируется как «System Reset» с кодом причины 5:15.

Для плановых выключений и перезагрузок используйте в коде причины префикс **p:**, указывающий на плановое выключение:

```
/e /c "ПлановаяПричина" /d p:ОсновнойКод:ДополнительныйКод
```

Например:

```
shutdown /r /e /m \\Mailer1 /c "Planned Application Upgrade"
/d p:4:2
```

В этом примере MAILER1 перезагружается, и причина плановой перезагрузки документируется как «Planned Application Upgrade» с кодом причины 4:2.

Глава 6

Регистрация и отслеживание событий, автоматический мониторинг

До сих пор в центре нашего внимание были средства и способы управления локальными и удаленными системами из командной строки. Теперь обсудим использование журналов событий для мониторинга и оптимизации. Мониторингом называется процесс регулярной проверки систем на наличие проблем. А оптимизация — это процесс тонкой настройки системы для поддержания или достижения максимальной производительности.

В этой главе рассматриваются средства протоколирования событий, доступные в Windows-системах. Они помогают обнаруживать проблемы в системе и определять их причины, следить за приложениями и сервисами, а также поддерживать безопасность системы. Когда система замедляется, ведет себя непредсказуемо или демонстрирует другое ошибочное поведение, нелишне заглянуть в журналы событий и попытаться определить потенциальный источник проблем. После того как причина ошибок обнаружена, можно провести плановое или внеочередное обслуживание для их устранения. С помощью триггеров событий, следящих за событиями и выполняющих требуемые действия при их возникновении, можно даже автоматизировать процесс мониторинга и обслуживания.

Протоколирование событий Windows

В Microsoft Windows *событие* (event) — это любое значительное происшествие в операционной системе, которое требует уведомления пользователей или администраторов. События сохраняются в журналах событий Windows и предоставляют важные хронологические сведения, помогающие вести мони-

торинг системы, поддерживать ее безопасность, устранять ошибки и выполнять диагностику. Необходимо регулярно анализировать информацию, содержащуюся в этих журналах; это очень важно. Администраторам следует тщательно следить за журналами событий всех бизнес-серверов и настраивать рабочие станции на сохранение важных системных событий. На серверах надо следить за безопасностью системы, нормальной работой приложений и сервисов, а также проверять сервер на наличие ошибок, способных ухудшить производительность. На рабочих станциях следует убедиться в том, что события, необходимые для поддержки систем и устранения ошибок, протоколируются и что соответствующие журналы вам доступны.

Windows-служба, управляющая протоколированием событий, называется Event Log (Журнал событий). При ее запуске Windows записывает важные данные в журналы. Доступность журналов в системе определяется ее ролью, а также установленными службами. Существует несколько журналов, в том числе следующие.

- **Application (Приложение)** — хранит важные события, связанные с конкретным приложением. Например, Exchange Server сохраняет события, относящиеся к пересылке почты, в том числе события информационного хранилища, почтовых ящиков и запущенных служб. По умолчанию помещается в *%SystemRoot%*\System32\Config\Appevent.evt.
- **Directory Service (Служба каталогов)** — на контроллерах домена этот журнал хранит события службы каталогов Active Directory, в том числе относящиеся к ее запуску, глобальным каталогам и проверкам целостности. По умолчанию помещается в *%SystemRoot%*\System32\Config\Ntds.evt.
- **DNS Server (DNS-сервер)** — на DNS-серверах в этом журнале сохраняются DNS-запросы, ответы и прочие события DNS. По умолчанию помещается в *%SystemRoot%*\System32\Config\Dnsevent.evt.
- **File Replication Service (Служба репликации файлов)** — на контроллерах домена и других серверах, использующих репликацию, этот журнал регистрирует действия в системе, связанные с репликацией файлов, в том числе события состояния и управления службой, сканирования данных на системных томах, а также управления наборами репликации. По умолчанию помещается в *%SystemRoot%*\Sysem32\Config\Ntfrs.evt.

- **Security (Безопасность)** — хранит события, связанные с безопасностью, такие как вход/выход из системы, использование привилегий и обращение к ресурсам. По умолчанию помещается в *%SystemRoot%*\System32\Config\Secevent.evt.

Внимание! Для доступа к журналам безопасности у пользователей должно быть право Manage Auditing And Security Log (Управление аудитом и журналом безопасности). По умолчанию оно выдается членам группы администраторов. Подробнее о назначении прав пользователям см. в разделе «Настройка политик прав пользователей» главы 9 в книге «Microsoft Windows Server 2003. Справочник администратора».

- **System (Система)** — хранит события операционной системы или ее компонентов, например неудачи при запусках служб или инициализации драйверов, общесистемные сообщения и прочие сообщения, относящиеся к системе в целом. По умолчанию помещается в *%SystemRoot%*\System32\Config\Sysevent.evt.

Значимость событий варьируется от уведомлений до предупреждений общего характера и серьезных инцидентов вроде критических сбоев и ошибок. Категория события обозначается его типом. Существуют следующие типы:

- **Information (Уведомление)** — указывает на возникновение информационного события, обычно связанного с успешным действием;
- **Warning (Предупреждение)** — предупреждение общего характера. Зачастую помогает избежать последующих проблем в системе;
- **Error (Ошибка)** — критическая ошибка, например неудача при запуске службы;
- **Success Audit (Аудит успехов)** — успешное выполнение действий, которые вы отслеживаете через аудит, например использование какой-либо привилегии;
- **Failure Audit (Аудит отказов)** — неудачное выполнение действий, которые вы отслеживаете через аудит, например ошибка при входе в систему.

Примечание Из множества типов событий внимательнее всего следует наблюдать за ошибками и предупреждениями. Если возникает событие этих типов и его причина неизвестна, нужно детально проанализировать его и определиться с дальнейшими действиями.

Кроме типа, у каждого события есть следующие связанные с ним свойства.

- **Date/Time (Дата/Время)** — определяет дату/время возникновения события.
- **Event (Событие)** — детализирует конкретное событие с числовым идентификатором, который называется кодом события (event ID). Последний генерируется источником события и служит для уникальной идентификации события.
- **Source (Источник)** — определяет источник события, например приложение, службу или компонент системы. Помогает выяснить причину события.
- **Computer (Компьютер)** — идентифицирует компьютер, ставший причиной события.
- **Category (Категория)** — определяет категорию события, иногда используемую для последующего описания допустимого действия. У каждого источника событий свои категории. Так, источник безопасности имеет следующие категории: вход/выход, использование привилегий, изменение политики и управление учетной записью.
- **User (Пользователь)** — определяет учетную запись пользователя, вызвавшую событие. К пользователям относятся особые сущности, например Local Service, Network Service и Anonymous Logon, а также учетные записи реальных пользователей. В этом поле может стоять N/A (Н/Д); это означает, что в данной ситуации учетная запись неприменима.
- **Description (Описание)** — содержит подробное описание события, возможно, со сведениями о местонахождении дополнительной информации, которая поможет устранить проблему. Это поле доступно при двойном щелчке записи журнала в Event Viewer (Просмотр событий).

GUI-средство управления событиями называется Event Viewer (Просмотр событий). Для его запуска наберите в командной строке **eventvwr** для просмотра событий на локальном компьютере или **eventvwr /computer=*ИмяКомпьютера***,

где *ИмяКомпьютера* — имя удаленного компьютера, чьи события вы хотите проанализировать. Как и большинство GUI-средств, Event Viewer прост в обращении и полезен для определенных задач управления. Например, он используется для управления размером журналов событий, способами обработки протоколирования, а также архивированием журналов событий. Эти действия нельзя выполнить из командной строки.

Однако Event Viewer плохо умеет фильтровать события и работать с журналами на удаленных компьютерах. Конечно, для этих задач можно применять и Event Viewer, но существуют другие, более подходящие для этих задач утилиты, в том числе следующие.

- **Eventquery** — просматривает журналы событий и отбирает записи, удовлетворяющие определенным требованиям. В сценарии Eventquery позволяет анализировать события на множестве систем и сохранять результаты в файле, облегчая поиск информации, а также ошибок и предупреждений во всей сети.
- **Eventcreate** — создает пользовательские события в журналах. При запуске собственных сценариев по расписанию или при плановом обслуживании вам может потребоваться регистрация какого-либо действия в журналах, и в этом поможет Eventcreate.
- **Eventtriggers** — следит за определенными событиями в журналах и при их возникновении реагирует запуском заданий или команд. Используя триггеры событий, можно настроить систему на самонаблюдение. Триггеры событий похожи на задания, запускаемые по расписанию, с тем исключением, что они выполняются при возникновении событий, а не периодически или однократно.

Примечание Мониторинг системных событий следует проводить последовательно и тщательно, а не от случая к случаю. На серверах журналы событий нужно анализировать минимум раз в день и настраивать триггеры событий на немедленное оповещение о любых критических ошибках. На рабочих станциях журналы можно анализировать по мере необходимости, например, если пользователь сообщает о какой-то проблеме.

Просмотр и фильтрация журналов событий

Для просмотра событий, сохраненных в журналах Windows, служит утилита Eventquery. Ее параметры управляют форматом вывода и уровнем детализации, а также позволяют применять фильтры для включения или исключения событий из набора результатов. Работая с Eventquery, не забудьте о том, что она дает возможность автоматизировать выполнение различных задач. Незачем каждый раз запускать команду вручную. Вместо этого создайте сценарий, запрашивающий журналы событий на множестве систем и сохраняющий результаты в файл. Если скопировать этот файл в каталог, опубликованный на сервере интрасети, можно использовать Web-браузер при изучении списка событий. А это не только сбережет время, но и позволит создать единую точку просмотра всех журналов и выявления проблем, требующих дальнейшего анализа.

Просмотр событий и формат вывода

Базовый синтаксис Eventquery таков:

```
eventquery /l "ИмяЖурнала"
```

где *ИмяЖурнала* — название требуемого журнала, например «Application», «System» или «Directory Service». В этом примере был запрошен журнал Application (Приложение):

```
eventquery /l "Application"
```

и запрос выдает примерно такой результат:

```
----------------------------------------
Listing the events in 'application' log of host 'MAILER1'
----------------------------------------
Type        Event Date       Time       Source            ComputerName
----------- --- ------------ ---------- ----------
Warning     9220  5/19/2004  4:38:01PM  MSExchange        MTAMAILER1
Information 1001  5/19/2004  4:28:50PM  MSExchangeIS      MAILER1
Information 9600  5/19/2004  4:28:50PM  MSExchangeIS      MAILER1
Information 9523  5/19/2004  4:28:50PM  MSExchangeIS Publ MAILER1
Information 9523  5/19/2004  4:28:49PM  MSExchangeIS Mail MAILER1
Information 9523  5/19/2004  4:28:48PM  MSExchangeIS Publ MAILER1
Information 9523  5/19/2004  4:28:47PM  MSExchangeIS Mail MAILER1
Information 9523  5/19/2004  4:28:46PM  MSExchangeIS Mail MAILER1
Information 3000  5/19/2004  4:28:45PM  MSExchangeIS Publ MAILER1
Information 1133  5/19/2004  4:28:41PM  MSExchangeIS Publ MAILER1
```

Как видите, в листинге присутствуют свойства Type, Event, Date Time, Source и ComputerName событий. С помощью параметра /V (verbose) можно добавить к выводимым свойствам категорию, пользователя и описание. Таким образом, если вам нужен подробный отчет о журнале Application, воспользуйтесь следующей командой:

```
eventquery /l "Application" /v
```

Примечание С технической точки зрения, кавычки нужны, только если в названии журнала имеется пробел, как в случае с журналами DNS Server, Directory Service и File Replication Service. Однако лучше всегда ставить их, тогда вы не забудете о них в тот момент, когда они действительно нужны, и тем самым избежите ошибок в сценариях или назначенных заданиях.

Совет В отличие от утилит командной строки, с которыми вы работали ранее, Eventquery настроена в качестве Windows-сценария. Если вы впервые работаете с Windows-сценариями из системной командной строки или настроили WScript как основной хост сценариев, вам потребуется сделать основным хостом сценариев CScript. Для этого наберите в командной строке cscript //h:cscript //s. Это необходимо потому, что вы собираетесь работать с командной строкой, а не с GUI.

Примечание Хост сценариев устанавливается для каждого пользователя отдельно, и при запуске сценария под конкретной учетной записью может оказаться, что в ней CScript не является хостом по умолчанию. Эффективное решение этой проблемы — начинать сценарий со строки **cscript //h:cscript //s**, а уж потом вводить свои запросы событий.

По умолчанию Eventquery выполняется на локальном компьютере с разрешениями зарегистрированного пользователя. При необходимости вы можете указать удаленной компьютер, события которого вы собираетесь запрашивать, а также разрешения Run As (Запустить от имени). Для этого применяется расширенный синтаксис, содержащий следующие параметры:

```
/s Компьютер /u [Домен\]Пользователь [/p Пароль]
```

где *Компьютер* — имя или IP-адрес удаленного компьютера, *Домен* — необязательное имя домена, где находится учетная запись пользователя, *Пользователь* — имя учетной записи, разрешения которой вы собираетесь задействовать, а *Пароль* — необязательный пароль для этой учетной записи. Например, если вы хотите проанализировать события службы каталогов на MAILER1 под учетной записью Adatam\WRStanek, введите команду:

```
eventquery /l "Directory Service" /s Mailer1 /u
Adatam\WRStanek
```

Примечание Если домен не указан, предполагается текущий. Если пароль не задан, вам будет предложено ввести его.

В синтаксис можно включать и следующие форматирующие параметры.

- **/Nh** — удаляет строку заголовка из таблицы или из CSV-отформатированных данных.
- **/Fo *Формат*** — изменяет формат вывода. По умолчанию используется табличный формат (/Fo Table). Чтобы вывести данные, разделенные запятыми, укажите /Fo Csv. Параметр /Fo List выводит результат в виде списка.

В Eventquery интересна поддержка диапазонов и фильтрации. При использовании диапазонов можно просмотреть:

- ***N* последних записей** — введите /r *N*, где *N* — число последних записей, например /r 50 для 50 последних записей;
- ***N* самых старых записей** — введите /r -*N*, где -*N* — число самых старых записей, например /r -50 для просмотра 50 самых старых записей;
- **события с *N1* по *N2*** — введите /r *N1*-*N2*, где *N1* — первое событие, а *N2* — последнее из требуемых, причем 1 — самое недавнее событие, 2 — предыдущее и т. д.; например, чтобы просмотреть события с 10 по 20, используйте /r 10-20.

Способы фильтрации событий рассматриваются в следующем разделе.

Фильтрация событий

Одна из основных причин для использования Eventquery — способность этой утилиты фильтровать события для исключения или включения их в набор результатов. Обычно все сгенерированные в системе события не нужны. Чаще вас интересуют только предупреждения или критические ошибки, и именно для этого нужны фильтры. При помощи фильтров можно включать только события, удовлетворяющие заданному критерию.

Фильтровать можно по любому информационному полю, даже если это поле перечисляется только при задании флага детального вывода (/V) и вы не указали его в командной строке. То есть поддерживается фильтрация по типу, дате и времени, источнику, имени компьютера, коду события, категории и пользователю.

Применением фильтров к конкретным информационным полям Eventquery управляют операторы фильтров. Доступны следующие операторы.

- **Eq** — равно; если поле содержит указанное значение, событие включается в результат.
- **Ne** — не равно; если поле содержит указанное значение, событие исключается из результата.
- **Gt** — больше; если поле содержит числовое значение и оно больше указанного, событие включается в результат.
- **Lt** — меньше; если поле содержит числовое значение и оно меньше указанного, событие включается в результат.
- **Ge** — больше или равно; если поле содержит числовое значение и оно больше или равно указанному, событие включается в результат.
- **Le** — меньше или равно; если поле содержит числовое значение и оно меньше или равно указанному, событие включается в результат.

Как показано в табл. 6-1, значения, которые можно использовать в операторах фильтров, зависят от анализируемого информационного поля события. Не забудьте, что доступны все поля, даже если они не выводятся при заданных параметрах. Например, можно проверять поле состояния без флага /V.

Табл. 6-1. Операторы фильтров и допустимые значения для Eventquery

Имя фильтруемого поля	Допустимые операторы	Допустимые значения
Category	eq, ne	Любая допустимая строка символов
Computer	eq, ne	Любая допустимая строка символов
Datetime	eq, ne, gt, lt, ge, le	Любое допустимое время в формате *мм/дд/гг, чч:мм:сс*AM или *мм/дд/*гг, *чч:мм:*ссPM
ID	eq, ne, gt, lt, ge, le	Любое допустимое положительно целое число до 65 535
Source	eq, ne	Любая допустимая строка символов
Type	eq, ne	Information, Warning, Error, SuccessAudit, FailureAudit
User	eq, ne	Любое допустимое имя пользователя (только имя или в формате *домен\пользователь*)

Строки фильтров нужно заключать в кавычки. Рассмотрим несколько примеров использования фильтров.

Просмотр ошибок в журнале приложений:

```
eventquery /l "application" /fi "type eq error"
```

Просмотр событий в системном журнале на MAILER1, произошедших после полуночи 05/06/04:

```
eventquery /s Mailer1 /l "system" /fi "date gt 05/06/
04,00:00:00AM"
```

Просмотр ошибок в журнале DNS-сервера на MAILER1 с кодом события 4004:

```
eventquery /s Mailer1 /l "dns server" /fi "id eq 4004"
```

Несколько параметров /Fi указывают, что результат должен удовлетворять нескольким фильтрам:

```
eventquery /l "system" /fi "date gt 05/06/04,00:00:00AM" /fi
"type eq error"
```

В данном случае Eventquery ищет в системных журналах события ошибок, зарегистрированные после полуночи 05/06/04. Обратите внимание, что фильтры взаимно исключают друг друга. Нельзя указывать в одной команде, что вам нужны и ошибки, и предупреждения. Для этого следует вводить две разные команды: одну — с параметром /Fi «type eq error», а другую — с /Fi «type eq warning».

Однако, если вы работаете с каким-либо журналом, кроме журнала безопасности (в котором фиксируются только события успешного и неудачного аудита), то можете просто указать, что вам не нужны уведомления. Тем самым вы получите только предупреждения и ошибки, как показано в следующем примере:

```
eventquery /l "system" /fi "type ne information"
```

Процесс запроса событий можно автоматизировать, создав сценарий, который получает требуемую информацию о событии и сохраняет ее в текстовый файл. Например, так:

```
@echo off
eventquery /s Mailer1 /l "system" /r 100 /fi "type ne
information" > \\CorpIntranet01\www\currentlog.txt

eventquery /s Mailer1 /l "application" /r 100 /fi "type ne
information" >> \\CorpIntranet01\www\currentlog.txt

eventquery /s Mailer1 /l "directory service" /r 100 /fi "type
ne information" >> \\CorpIntranet01\www\currentlog.txt
```

Здесь анализируются журналы System, Application и Directory Service на MAILER1, а результаты записываются в общий сетевой ресурс на CorpIntranet01. Если какие-либо из этих журналов среди 100 последних событий содержат предупреждения или ошибки, они сохраняются в файле Currentlog.txt. Так как первое перенаправление указывает перезапись (>), а последующие — дозапись (>>), текущее содержимое файла Currentlog.txt перезаписывается при каждом запуске сценария. Этим обеспечивается его актуальность. В целях еще большей автоматизации можно создать назначенное задание, запускающее сценарий ежедневно или с определенными интервалами в течение дня.

Запись собственных событий в журналы

При работе с автоматизированными сценариями, заданиями по расписанию или собственными приложениями вам может потребоваться, чтобы они записывали собственные события в журналы. Например, при нормальном выполнении сценария вы хотите записать событие уведомления в журнал приложения, чтобы в дальнейшем легко определить, выполнен сценарий и нормально ли он завершился. И наоборот, если сценарий не сработал и в результате его выполнения возникли ошибки, вам может понадобиться сохранить событие ошибки или предупреждения в журнале — тогда вы узнаете, что нужно проанализировать сценарий и выяснить, что случилось.

Совет Вы можете отслеживать ошибки в сценариях с помощью переменной *%ErrorLevel%*. Эта переменная окружения хранит код завершения последней команды. Если команда выполнена нормально, переменная равна нулю. Если при выполнении команды произошла ошибка, переменная устанавливается в ненулевое значение. Подробнее о работе с этой переменной см. раздел «Знакомство с переменными» главы 3.

Для создания собственных событий используется утилита Eventcreate. Собственные события можно сохранять в любом доступном журнале за исключением журнала безопасности. Такие события могут содержать источник, код и нужное описание. Синтаксис Eventcreate:

```
eventcreate /l ИмяЖурнала /so ИсточникСобытия /t ТипСобытия /
id КодСобытия /d ОписаниеСобытия
```

- **ИмяЖурнала** — название журнала для записи события; если оно содержит пробелы, заключите его в кавычки, например «DNS Server».

Совет В журнал безопасности нельзя записывать собственные события. Но их можно записать в журналы DNS Server, Directory Service, File Replication Service и другие журналы, относящиеся к службам. Начните с записи фиктивного события, используя источник, который вы хотите зарегистрировать для выбранного журнала. Начальное событие из этого источника будет зафиксировано в журнале приложения. После этого вы можете использовать источник с указанным журналом и собственными событиями.

- **ИсточникСобытия** — указывает источник события и может быть любой строкой. Если строка содержит пробелы, заключите ее в кавычки, например «Event Tracker». В большинстве случаев источник указывает на приложение, задание или сценарий, вызвавший ошибку.

Внимание! Тщательно выбирайте источник, прежде чем записывать события в журнал. Каждый источник должен быть уникальным, его имя не должно совпадать с именем существующего источника, используемого установленной службой или приложением. Например, DNS, W32Time или Ntfrs в качестве источников не годятся, так как они уже задействованы установленными службами или приложениями. Также учтите: как только вы используете источник событий с конкретным журналом, этот источник регистрируется для данного журнала на указанной системе. Например, нельзя работать с «EventChecker» в качестве источника событий для журнала приложений и системного журнала на MAILER1. Если вы попытаетесь записать событие в системный журнал, применяя «EventChecker» после того, как событие из этого источника было зафиксировано в журнале приложений, то увидите сообщение об ошибке «ERROR: Source already exists in 'Application' log. Source cannot be duplicated».

- **ТипСобытия** — задает тип события. Может принимать значения Information, Warning или Error. Типы событий «Success Audit» и «Failure Audit» неприменимы, так как используются в журнале безопасности, в который записывать собственные события нельзя.
- **КодСобытия** — задает числовой код события. Может принимать любое значение от 1 до 1000. Чем случайно назначать идентификаторы, лучше составить список общих событий, которые могут возникнуть, а затем разбить его на категории. Тогда каждой категории можно присвоить свой диапазон кодов событий. Например, события из первой сотни могут быть общими, из второй — событиями состояния, из пятой — предупреждениями, а из девятой — ошибками.
- **ОписаниеСобытия** — задает описание события и может быть любой строкой. Не забудьте заключить строку в кавычки.

Примечание По умолчанию Eventcreate выполняется на локальном компьютере с разрешениями пользователя, зарегистрированного в этой системе на данный момент. При необходимости можно указать удаленный компьютер, чьи события вы хотите запрашивать, а также разрешения Run As, используя синтаксис /S *Компьютер* /u [*Домен*\]*Пользователь* [/P *Пароль*], где *Компьютер* — имя или IP-адрес удаленного компьютера, *Домен* — необязательное имя домена, где хранится учетная запись пользователя, *Пользователь* — имя пользовательской учетной записи, чьи разрешения вы хотите задействовать, а *Пароль* — необязательный пароль для этой учетной записи.

Рассмотрим применение Eventcreate на нескольких примерах.

Создать событие-уведомление в журнале приложения с источником Event Tracker и кодом события 209:

```
eventcreate /l "application" /t information /so "Event
Tracker" /id 209 /d "evs.bat script ran without errors."
```

Создать событие-предупреждение в системном журнале с источником CustApp и кодом события 511:

```
eventcreate /l "system" /t warning /so "CustApp" /id 511 /d
"sysck.exe didn't complete successfully."
```

Создать событие-ошибку в системном журнале на MAILER1 с источником SysMon и кодом события 918:

```
eventcreate /s Mailer1 /l "system" /t error /so "SysMon" /id
918 /d "sysmon.exe was unable to verify write operation."
```

Мониторинг системы с помощью триггеров событий

Теперь, когда вы знаете, как просматривать, фильтровать и создавать события, обсудим способы применения триггеров событий, позволяющих автоматизировать процесс мониторинга событий. Благодаря триггерам можно настроить системные задания, отслеживающие журналы событий и выполняющие определенное действие при возникновении события. Например, создать триггер, который отслеживает запись в журналы событий, связанных с нехваткой места, и который при возникновении таких событий выполняет сценарий, удаляющий вре-

менные или ненужные файлы, освобождая место на диске. Таким образом, триггеры событий способны не только автоматизировать процесс мониторинга, но и помочь в устранении проблем по мере их возникновения, чтобы поддерживать производительность системы на должном уровне, обеспечить ее целостность и т. д.

Создание триггеров событий — не то дело, которое можно делать мимоходом, без тщательного предварительного планирования. Вам понадобится четкий план действий — набор целей, которых вы рассчитываете достигнуть, применяя триггеры событий. Рассмотрим причины, по которым вам могут потребоваться триггеры событий, а также средства управления ими.

Зачем использовать триггеры событий?

Основная причина для использования триггеров событий — поддержание производительности приложений и системы на должном уровне. Например, если с серверным приложением связаны проблемы, которые вы обычно разрешаете вручную, возможно, вам удастся настроить триггеры событий, отслеживающие возникновение в журналах соответствующих ошибок и запускающие сценарии, которые предпринимают адекватные действия для устранения проблемы. В этом случае определить известные проблемы приложения можно, проанализировав журналы событий, опросив других администраторов или изучив статьи в базах знаний, описывающих проблемы. После этого сопоставьте проблемы конкретным событиям или типам событий, для которых вы можете настроить триггеры мониторинга, а потом напишите сценарий, который уведомляет администраторов о проблеме или предпринимает определенные действия, решающие проблему. Этот сценарий используется в качестве задания, запускаемого триггером.

Еще одна распространенная причина использования триггеров — быстрое обнаружение перерывов в работе приложений и служб, возможно, с восстановлением нормального функционирования. Когда приложение или служба останавливается, пользователи не могут работать с ресурсом, что может стоить организации времени и денег. В этом случае полезно просмотреть документацию на предмет событий, возникающих, когда приложение или служба ведет себя ненормально. Затем, проанализировав журналы событий на наличие таких же или сходных событий, определите источники, коды событий и опи-

сания, чтобы создать триггеры для наблюдения за такими событиями. И наконец, вы можете написать сценарий, который перезапускает приложение или предпринимает другие подходящие действия для устранения проблем.

Триггеры событий также полезны для поддержания безопасности и целостности системы. При атаке на систему в журналы могут записываться события, указывающие на атакуемое приложение, компонент или службу. При атаке методом грубой силы злоумышленник, вероятно, будет подбирать комбинации имен и паролей, пытаясь получить доступ. Если вы ведете мониторинг атакуемой системы, то увидите неудачные попытки регистрации в системе, соответствующие попыткам хакера получить доступ. Злоумышленник может попытаться остановить систему, приложение или службу, применяя атаку типа «отказ в обслуживании» (denial-of-service). Обычно хакеры делают это, посылая последовательные потоки неправильно сформированных запросов. Такие попытки будут отражены в журналах соответствующих приложений, служб или системы в виде ошибок. Для борьбы с такими атаками настройте триггеры, следящие за соответствующими событиями, например за блокировкой учетной записи вследствие серии неудачных попыток регистрации.

Подготовка к работе с триггерами событий

Прежде чем приступить к созданию триггеров событий, вы должны решить, чего именно вы надеетесь достичь путем автоматического мониторинга, а также оценить его возможное влияние на исследуемую систему и сеть в целом. Вот что для этого нужно сделать.

1. Определить, за какими событиями вы хотите следить и по каким причинам. Проанализируйте журналы событий на нескольких системах и документацию на предмет известных проблем и ошибок, в том числе статьи в базе знаний. Это поможет вам понять, с чего надо начинать.
2. Укажите действия, которые следует предпринять при наступлении события. Вначале оформите их в виде списка. Убедитесь, что вы учли влияние корректирующих действий на систему или на сеть в целом.
3. Напишите сценарий или приложение для выполнения требуемых корректирующих действий или для уведомления

пользователей. Пока не реализуйте их в виде триггеров. Вначале протестируйте сценарии в изолированной сети или в системе разработки для обнаружения возможных недочетов.

4. Определите триггеры и исполняемые задания, после чего реализуйте триггеры. Тщательно следите за системой в течение нескольких последующих дней или недель и убедитесь, что нежелательные эффекты отсутствуют.
5. Поддерживайте и удаляйте триггеры при необходимости, обеспечивая бесперебойную работу.

В осуществлении этапов 1, 2 и 3 вам помогут сведения, приведенные ранее в этой и других главах. Однако этапы 4 и 5 связаны с определением, поддержкой и удалением триггеров. Эти действия реализуются с помощью следующих подкоманд утилиты Eventtriggers:

- **Eventtriggers /create** — создает новый триггер и задает предпринимаемое действие;
- **Eventtriggers /query** — выводит триггеры, настроенные в указанной системе на данный момент;
- **Eventtriggers /delete** — удаляет триггер события, когда он больше не нужен.

Примечание В отличие от большинства других команд с подкомандами в Eventtriggers подкоманды предваряются знаком /.

Эти подкоманды и их применение обсуждаются в следующих разделах.

Создание триггеров событий

Триггеры событий можно настроить на запуск исполняемых файлов с расширением .exe и сценариев с расширением .bat или .cmd при возникновении события. Триггеры создаются по синтаксису:

```
eventtriggers /create /tr Имя /l ИмяЖурнала [Ограничения] /d
Описание /tk Задание
```

- **Имя** — задает имя триггера в виде строки символов в кавычках, например «Ошибка соединения».

- **ИмяЖурнала** — задает имя отслеживаемого журнала. Если имя журнала содержит пробелы, например «DNS Server», заключите его в кавычки. По умолчанию подставляется звездочка (*) для мониторинга всех журналов.
- **Ограничения** — указывает ограничения, по которым определяется, соответствует ли событие триггеру. Они ограничивают область действия триггера по коду события, его источнику или типу. Для этого используются параметры /Eid *КодСобытия*, /So *ИсточникСобытия* или /T *ТипСобытия* соответственно.

Совет Допускается несколько ограничений. В этом случае для срабатывания триггера событие должно соответствовать каждому ограничению. То есть дополнительные ограничения сужают область действия триггера.

- **Задание** — указывает запускаемую программу или сценарий. Обязательно задайте полный путь к программе или сценарию.

Примечание Eventtriggers не проверяет пути, и при вводе неправильного пути предупреждение не выводится. Чтобы передать программе или сценарию аргументы, заключите путь к файлу и аргументы команды в двойные кавычки, например «c:\scripts\trackerror.bat system y».

- **Описание** — задает описание триггера и может быть любой строкой символов. Заключается в двойные кавычки.

Не пугайтесь этой кучи параметров. Все гораздо проще, чем кажется на первый взгляд. Рассмотрим несколько примеров.

Создать триггер события, который следит за всеми журналами и событиями с кодом 9220 и запускает Record-prob.bat:

```
Eventtriggers /create /tr "Monitor 9220 Errors" /eid 9220 /tk
\\Mailer1\scripts\record-prob.bat
```

Создать триггер события, который следит за журналом DNS-сервера и событиями с источником DNS и кодом 4004 и запускает Dns-adfix.bat:

```
Eventtriggers /create /tr "DNS AD Fix" /l "DNS Server" /so
"DNS" /eid 4004 /tk c:\admin\scripts\dns-adfix.bat
```

Создать триггер события, следящий за журналом безопасности и событиями неудачного аудита с источником Security:

```
Eventtriggers /create /tr "Failure Audit Checks" /l "Security"
/so "Security" /t Failureaudit
```

По умолчанию триггеры событий и сопоставляемые с ними задания создаются на локальном компьютере с разрешениями зарегистрированного в данный момент пользователя. Так как эта команда используется в основном для администрирования, перед добавлением триггера запрашивается пароль. Если задание, связанное с триггером, нужно выполнять с другими разрешениями, воспользуйтесь синтаксисом /u [*Домен*\]*Пользователь* [/p *Пароль*], где *Домен* — необязательное имя домена, хранящего учетную запись пользователя, *Пользователь* — имя учетной записи, чьи разрешения вы хотите задействовать, а *Пароль* — необязательный пароль для этой учетной записи, например:

```
Eventtriggers /create /u adatam\wrstanek /p R4Runner! /tr
"Exchange Monitor" /l "Application" /so "MSExchangeMTA" /t
warning /tk c:\admin\scripts\exe-errlog.bat
```

При необходимости можно создать триггер и на удаленном компьютере, указав /S *Компьютер*, где *Компьютер* — имя или IP-адрес удаленного компьютера, например:

```
Eventtriggers /create /s 192.168.1.150 /tr "Exchange Monitor"
/l "Application" /so "MSExchangeMTA" /t warning /tk
c:\admin\scripts\exe-errlog.bat
```

Отображение существующих триггеров событий

Чтобы получить информацию обо всех настроенных в данный момент триггерах, используйте команду Eventtriggers /query. Просто введите ее в командной строке:

```
eventtriggers /query
```

Базовый вывод содержит код триггера события, его имя и исполняемое задание, как показано в следующем примере:

```
Trigger ID                      Event Trigger Name Task
========== ====================  ================================
       4   Failure Audit Checks  c:\admin\scripts\auditing.bat
       2   Monitor 9220 Errors   \\Mailer1\scripts\record-prob.bat
       3   DNS AD Fix            c:\admin\scripts\dns-adfix.bat
       1   Disk Cleanup          d:\windows\system32\cleanmgr.exe
```

Примечание Код триггера используется для его удаления. Формат вывода по умолчанию — таблица (/Fo Table). Формат /Fo Csv предназначен для вывода разделенных запятыми значений, а /Fo List — для вывода значений в виде списка. Кроме того, параметр /Nh отключает отображение заголовков, если указан формат Table или Csv.

Для получения более подробных сведений служит флаг */V* (verbose). Если он указан, выводятся следующие дополнительные столбцы информации.

- **Hostname (Имя узла)** — имя или IP-адрес компьютера, на котором настроен триггер события.
- **Query (Запрос)** — полный текст команды, которой был создан триггер.
- **Description (Описание)** — описание триггера, если оно было задано при его создании.
- **Run As (Запустить от имени)** — учетная запись Run As, указанная при определении задания и используемая для выполнения связанного с триггером задания.

При необходимости можно запросить триггеры на удаленном компьютере, используя /s *Компьютер*, где *Компьютер* — имя или IP-адрес удаленного компьютера, например:

```
eventtriggers /query /s Mailer1
```

Можно также задействовать другие разрешения, используя синтаксис /U [*Домен*\]*Пользователь* [/P *Пароль*], где *Домен* — необязательное имя домена, хранящего учетную запись пользователя, *Пользователь* — имя учетной записи, чьи разрешения вы хотите задействовать, а *Пароль* — необязательный пароль для этой учетной записи, например:

```
eventtriggers /query /s Mailer1 /u adatam\administrator /p
dataset5
```

Удаление триггеров событий

Если триггер больше не нужен, его можно удалить командой eventtriggers /delete. Вот ее синтаксис:

```
eventtriggers /delete /tid Код
```

где *Код* — код удаляемого триггера. Подставив звездочку (*) вместо конкретного кода, вы удалите все триггеры. Рассмотрим пару примеров.

Удалить триггер события 5:

```
eventtriggers /delete /tid 5
```

Удалить все триггеры событий:

```
eventtriggers /delete /tid *
```

При необходимости можно запросить триггеры на удаленном компьютере, используя /s *Компьютер*, где *Компьютер* — имя или IP-адрес удаленного компьютера, а также указать другие разрешения по синтаксису /U [*Домен*\]*Пользователь* [/P *Пароль*], где *Домен* — необязательное имя домена, хранящего учетную запись пользователя, *Пользователь* — имя учетной записи, чьи разрешения вы хотите задействовать, а *Пароль* — необязательный пароль для этой учетной записи, например:

```
eventtriggers /delete /tid 3 /s Mailer1 /u adatam\wrstanek /p
outreef7
```

Внимание! Удаленные триггеры восстановить нельзя. Если вы считаете, что в будущем вам может понадобиться данный триггер, запишите вывод команды Eventtriggers /query /v в файл и сохраните его. Благодаря параметру /V файл будет содержать полный текст команды, использованной для создания этого триггера.

Глава 7

Мониторинг процессов и производительности

Важной частью работы каждого администратора является мониторинг сетевых систем и обеспечение нормальной работы всех процессов — по крайней мере такой, насколько можно ожидать. Как вы видели в предыдущей главе, внимательное наблюдение за журналами событий помогает выявлять и отслеживать проблемы в приложениях, безопасности и важных службах. Обнаружив или предполагая проблему, вы должны докопаться до ее причины и устранить. Точное определение причины проблемы предотвратит ее повторное появление.

Управление приложениями, процессами и производительностью

Всякий раз, когда операционная система или пользователь запускает службу, приложение или команду, Microsoft Windows запускает один или более процессов для управления соответствующей программой. Несколько утилит командной строки упростят вам мониторинг программ и управление ими. К этим утилитам относятся:

- **Pmon (Process Resource Manager)** — показывает статистические данные по производительности, включая использование памяти и процессора, а также список всех процессов, выполняемых в локальной системе. Позволяет получать детальные «снимки» задействованных ресурсов и выполняемых процессов. Pmon поставляется с Windows Resource Kit;
- **Tasklist (Task List)** — перечисляет все выполняемые процессы по имени и идентификатору процесса, сообщает информацию о сеансе пользователя и занимаемой памяти;

- **Taskkill (Task Kill)** — останавливает выполнение процесса, заданного по имени или идентификатору. С помощью фильтров можно останавливать процессы в зависимости от их состояния, номера сеанса, процессорного времени, занимаемой памяти, имени пользователя и других параметров.

В следующих разделах подробно обсуждается, как применяются эти утилиты командной строки. Но сначала рассмотрим, как выполняются процессы и каковы наиболее распространенные проблемы, которые могут встретиться при работе с ними.

Системные и пользовательские процессы

Обычно процесс, запускаемый операционной системой, называется *системным*, а процесс, запускаемый пользователем, — *пользовательским*. Большинство пользовательских процессов выполняется в интерактивном режиме. То есть пользователь запускает процесс непосредственно при помощи клавиатуры или мыши. Если программа активна, связанный с ней интерактивный процесс контролирует клавиатуру и мышь до тех пор, пока вы не переключите управление, завершив эту программу или выбрав другую. Процесс, получивший контроль над клавиатурой и мышью, называют активным.

Процессы могут работать и в фоновом режиме независимо от сеансов зарегистрированных пользователей. Фоновые процессы не имеют контроля над клавиатурой, мышью или другими устройствами ввода и обычно запускаются операционной системой. Но с помощью Task Scheduler (Планировщик заданий) пользователи тоже могут запускать процессы в фоновом режиме, и эти процессы способны работать независимо от того, зарегистрирован ли пользователь в системе. Например, если планировщик заданий запускает назначенное задание при наличии зарегистрированного пользователя, процесс может продолжить выполнение даже после завершения сеанса пользователя.

Windows отслеживает каждый процесс, выполняемый в системе, по имени образа, идентификатору процесса, приоритету и другим параметрам. Имя образа — это имя исполняемого файла, используемого для запуска процесса, скажем, Msdtc.exe или Svchost.exe. Идентификатор процесса — числовой идентификатор процесса, например 2588. Приоритет процесса указывает на то, какую часть системных ресурсов должен получить процесс по сравнению с другими выполняемыми процессами. Процесс с более высоким приоритетом полу-

чает преимущества над процессами с более низким приоритетом, и, вероятно, ему не придется дожидаться получения процессорного времени, доступа к памяти или обращения к файловой системе. Процесс с более низким приоритетом, напротив, обычно вынужден ждать завершения текущей задачи, выполняемой процессом с более высоким приоритетом, и только после этого он может получить доступ к процессору, памяти или файловой системе.

В идеале процессы должны работать без всяких проблем. В действительности, однако, проблемы возникают — и зачастую в самый неподходящий момент. Обычно проблемы заключаются в следующем.

- Процессы перестают отвечать, например, когда приложение прекращает обработку запросов. Как только это происходит, пользователи могут сказать вам, что они не в состоянии работать с конкретным приложением, что их запросы к приложению остаются необработанными или что оно их игнорирует.
- Процессы не освобождают процессор — например, у вас появляется неконтролируемый процесс, который захватывает все процессорное время. Когда это происходит, система может показаться работающей медленно или вообще зависшей, так как этот процесс пожирает все процессорное время и не дает другим процессам выполнять свои задачи.
- Процессы используют больше памяти, чем нужно, например, когда в приложении есть утечка памяти. В этом случае процессы не освобождают занятую ими память. В результате свободная память в системе может постепенно уменьшаться с течением времени, и, поскольку доступной памяти становится мало, может замедлиться отклик системы на запросы, или она просто перестанет на них отвечать. Утечки памяти также могут нарушить работу других програм, выполняемых в данной системе.

В большинстве случаев, когда обнаруживаются эти или другие проблемы, процесс обычно останавливают и запускают вновь. Также нужно проверить журналы событий для выявления причины проблемы. При утечках памяти следует известить об этом разработчиков и проверить, не появилось ли обновление, решающее данную проблему.

Совет Периодический перезапуск приложений, вызывающих утечку памяти, весьма полезен. Такой перезапуск позволяет операционной системе восстановить контроль над «утерянной» памятью.

Анализ выполняемых процессов

При помощи утилиты командной строки Tasklist можно проверить процессы, работающие в локальной или удаленной системе. Tasklist позволяет:

- получить идентификатор процесса, его состояние и другие важные сведения о процессах в системе;
- увидеть зависимости между выполняемыми процессами и службами, настроенными в системе;
- просмотреть список DLL, задействованных выполняемыми в системе процессами;
- использовать фильтры для включения или исключения процессов, показываемых Tasklist.

Все эти задачи рассматриваются в следующих разделах.

Получение подробных сведений о процессах

Чтобы увидеть список выполняемых задач в локальной системе, просто введите в командной строке **tasklist**. Как и многие другие утилиты командной строки, Tasklist по умолчанию выполняется с разрешениями зарегистрированного пользователя. Вы можете указать удаленный компьютер, а также запустить эту утилиту от имени другого пользователя. Для этого применяется расширенный синтаксис:

```
/s Компьютер /u [Домен\]Пользователь [/p Пароль]
```

где *Компьютер* — имя или IP-адрес удаленного компьютера, *Домен* — необязательное имя домена, где находится учетная запись пользователя, *Пользователь* — имя учетной записи пользователя, разрешения которой вы хотите задействовать, а *Пароль* — необязательный пароль для этой учетной записи. Если домен не указан, подразумевается текущий. А если вы не задали пароль, вам будет предложено ввести его.

Чтобы понять применение в синтаксисе команды информации о компьютере и пользователе, рассмотрим пару примеров.

Проверка выполняемых задач на компьютере с именем Mailer1:

```
tasklist /s mailer1
```

Проверка выполняемых задач на компьютере с IP-адресом 192.168.1.5 с использованием учетной записи adatum\wrstanek:

```
tasklist /s 192.168.1.5 /u adatum\wrstanek
```

Совет Как правило, эти команды выводят данные в табличном виде. Вы можете задать вывод в виде списка или строк со значениями, разделенными запятыми, при помощи параметров /Fo List или /Fo Csv соответственно. Помните, что вывод можно перенаправить в файл, используя символы перенаправления вывода (> или >>), например tasklist /s mailer1 >> current-tasks.log.

Независимо от того, работаете вы с локальным или удаленным компьютером, результат будет иметь примерно такой вид:

```
Image Name                     PID Session Name        Session#Mem Usage
========================= ====== ================ =========== ============
System Idle Process              0 Console                    0         16 K
System                           4 Console                    0        216 K
smss.exe                       420 Console                    0        480 K
csrss.exe                      472 Console                    0      4,420 K
sqlgea.exe                     496 Console                    0      3,352 K
services.exe                   540 Console                    0      3,288 K
sqlmon.exe                     552 Console                    0     32,508 K
sdman.exe                      728 Console                    0      2,856 K
sdman.exe                      788 Console                    0      3,840 K
sdman.exe                      988 Console                    0      4,016 K
sdman.exe                     1036 Console                    0      2,032 K
sdman.exe                     1048 Console                    0     15,624 K
spoolsv.exe                   1348 Console                    0      4,728 K
msdtc.exe                     1380 Console                    0      3,808 K
```

Поля, выводимые Tasklist, содержат следующую информацию.

- **Image Name (Имя образа)** — имя процесса или исполняемого образа.

Примечание Первый процесс называется System Idle Process. Это особый системный процесс, предназначенный для отслеживания объема незадействованных системных ресурсов. Подробнее об этом процессе см. в разделе «Мониторинг процессов и использования системных ресурсов» далее в этой главе.

- **PID (PID)** — числовой идентификатор процесса.
- **Session Name (Имя сессии)** — имя сеанса, из которого запущен процесс. Значение *console* говорит о том, что процесс запущен локально.
- **Session# (№ сеанса)** — числовой идентификатор сеанса.
- **Mem Usage (Память)** — общий объем памяти, занимаемой процессом на момент запуска Tasklist.

Если вам нужна более подробная информация, включите детальный вывод при помощи параметра /V (verbose). При этом дополнительно отображаются следующие столбцы данных.

- **Status (Статус)** — текущее состояние процесса: Running (Работает), Not Responding (Не отвечает) или Unknown (Не известно). Процесс может быть в состоянии Unknown и продолжать нормально работать и отвечать на запросы. Однако процесс в состоянии Not Responding должен быть остановлен или перезапущен.
- **User Name (Пользователь)** — учетная запись, от имени которой выполняется процесс; указывается в виде *домен\пользователь*. Для процессов, которые запускаются системой, указывается имя системной учетной записи, такое как SYSTEM, LOCAL SERVICE или NETWORK SERVICE с NT AUTHORITY в качестве домена.
- **CPU Time (Время ЦП)** — общее количество процессорного времени, использованного процессом с момента его запуска.
- **Window Title (Заголовок окна)** — название окна процесса в GUI, если оно есть. В ином случае вместо названия выводится N/A (Н/Д). Например, для процесса Helpctr.exe заголовок окна — Help and Support Center (Центр справки и поддержки).

Просмотр зависимостей между процессами и службами

Запуская Tasklist с параметром /Svc, вы можете проверить взаимосвязь между выполняемыми процессами и сконфигурированными в системе службами. При этом выводятся имя образа процесса, идентификатор процесса и список всех служб, с которыми взаимодействует процесс, примерно в таком виде:

```
Image                  Name    PID Services
====================== ======  ============================================
System Idle Process         0  N/A
System                      4  N/A
smss.exe                  408  N/A
csrss.exe                 456  N/A
winlogon.exe              484  N/A
services.exe              528  Eventlog, PlugPlay
lsass.exe                 540  HTTPFilter, kdc, Netlogon, NtLmSsp,
                               PolicyAgent, ProtectedStorage, SamSs
svchost.exe               800  RpcSs
svchost.exe               956  Dnscache
svchost.exe               984  LmHosts
svchost.exe               996  AudioSrv, Browser, CryptSvc,
                               dmserver, EventSystem, helpsvc,
                               lanmanserver, lanmanworkstation,
                               Netman, Nla, Schedule, seclogon,
                               SENS, ShellHWDetection, W32Time,
                               winmgmt, wuauserv, WZCSVC
spoolsv.exe              1300  Spooler
msdtc.exe                1332  MSDTC
dfssvc.exe               1400  Dfs
dns.exe                  1436  DNS
svchost.exe              1492  ERSvc
inetinfo.exe             1552  IISADMIN, IMAP4Svc, POP3Svc, RESvc,
                               SMTPSVC
ismserv.exe              1568  IsmServ
ntfrs.exe                1584  NtFrs
svchost.exe              1688  RemoteRegistry
mad.exe                  1724  MSExchangeSA
mssearch.exe             1784  MSSEARCH
exmgmt.exe               1824  MSExchangeMGMT
svchost.exe              2000  W3SVC
store.exe                2108  MSExchangeIS
```

По умолчанию вывод форматируется в виде таблицы, и этот формат нельзя сменить на *list* или *CSV*. Помимо форматирования, важно отметить, что службы перечисляются по

сокращенным именам, что соответствует стилю именования, принятому в утилите командной строки Sc (service controller), управляющей службами.

Знание взаимосвязи между процессами и службами поможет в управлении системой. Например, если вы считаете, что возникли проблемы со службой World Wide Web Publishing Service (Служба публикаций World Wide Web) (W3svc), то для выяснения причины проблемы нужно начать мониторинг процесса или процессов, связанных с этой службой. Вы должны проверить:

- состояние процесса;
- объем занимаемой процессом памяти;
- используемое процессорное время.

Отслеживая эту статистику какое-то время, вы, возможно, заметите изменения, которые могут указывать на не отвечающий или неконтролируемый процесс, захватывающий все процессорное время, либо на утечку памяти.

Просмотр DLL, используемых процессами

Tasklist с параметром /M позволяет проверить взаимосвязь между выполняемыми процессами и DLL, зарегистрированными в системе. В результате будет показано имя образа процесса, идентификатор процесса и список всех DLL, которые используются процессом, например:

```
Image Name                   PID    Modules
=================== ======  ======
System Idle Process            0    N/A
System                         4    N/A
smss.exe                     408    ntdll.dll
csrss.exe                    456    ntdll.dll, CSRSRV.dll,
                                    basesrv.dll, winsrv.dll,
                                    KERNEL32.dll, USER32.dll,
                                    GDI32.dll, sxs.dll, ADVAPI32.dll,
                                    RPCRT4.dll, Apphelp.dll,
                                    VERSION.dll
```

Информация о DLL-модулях, загружаемых процессом, облегчает выяснение причин, по которым процесс не отвечает, захватывает процессорное время или использует больше памяти, чем должен. В некоторых случаях потребуется проверить версии DLL, чтобы удостовериться, что в системе работают DLL нужных версий. В качестве справочного материала ис-

пользуйте базу знаний Microsoft или документацию поставщиков DLL.

Если вы ищете процессы, использующие определенную DLL, — укажите ее имя. Например, если вы предполагаете, что причиной зависания процесса является драйвер спулера принтера Winspool.drv, найдите процессы, работающие с Winspool.drv, проверьте их состояние и задействованные ими ресурсы.

Синтаксис для перечисления процессов, загрузивших указанную DLL, выглядит так:

```
tasklist /m ИмяDLL
```

где *ИмяDLL* — имя необходимой DLL. Tasklist безразлично, в каком регистре введено имя DLL. Рассмотрим следующий пример*:

```
tasklist /m winspool.drv
```

В этом примере мы ищем процессы, использующие Winspool.drv. Команда перечисляет процессы, работающие с этим модулем, и их идентификаторы, как показано ниже.

```
Image Name                     PID    Modules
=========================  =======    =======
winlogon.exe                   484    WINSPOOL.DRV
spoolsv.exe                   1300    winspool.drv
explorer.exe                  3516    WINSPOOL.DRV
mshta.exe                     3704    WINSPOOL.DRV
```

Фильтрация вывода Tasklist

Параметр /Fi позволяет ограничить список выводимых процессов на основании значений любых доступных полей, даже если эти поля, принимая во внимание заданные параметры, не выводятся. Это значит, что вы, к примеру, можете указать, чтобы были показаны процессы Svchost.exe только с состоянием Not Responding (Не отвечает) или те, у которых используемое процессорное время превышает определенную величину.

Вы задаете условия применения фильтра к конкретным значениям выводимой информации при помощи следующих операторов.

---

* По крайней мере в русской версии Windows XP с SP1 указание любого модуля в команде tasklist /m не срабатывает, и появляется сообщение «Ошибка: фильтр поиска не опознан». — *Прим. перев.*

- **Eq** — равно. Если поле содержит указанное значение, процесс будет включен в вывод.
- **Ne** — не равно. Если поле содержит указанное значение, процесс будет исключен из вывода.
- **Gt** — больше. Если поле содержит числовое значение, превышающее указанную величину, процесс будет включен в вывод.
- **Lt** — меньше. Если поле содержит числовое значение, меньшее указанной величины, процесс будет включен в вывод.
- **Ge** — больше или равно. Если поле содержит числовое значение, которое больше или равно указанной величине, процесс будет включен в вывод.
- **Le** — меньше или равно. Если поле содержит числовое значение, которое меньше или равно указанной величине, процесс будет включен в вывод.

Как показано в табл. 7-1, значения, допустимые в операторах фильтра, зависят от информации, содержащейся в поле. Помните, что для наложения фильтра доступны все поля, даже если они не отображаются при заданных параметрах. Например, поле Status (Статус) можно проверять без применения флага */V* (verbose).

Табл. 7-1. Операторы фильтра и их допустимые значения для Tasklist

Имя поля фильтра	Допустимые операторы	Допустимые значения
CPUTime	eq, ne, gt, lt, ge, le	Любое допустимое значение времени в формате *чч:мм:СС*
Services	eq, ne	Любая допустимая строка символов
ImageName	eq, ne	Любая допустимая строка символов
MemUsage	eq, ne, gt, lt, ge, le	Любое целое, выраженное в килобайтах (Кб)
PID	eq, ne, gt, lt, ge, le	Любое положительное целое
Session	eq, ne, gt, lt, ge, le	Любой действительный номер сеанса
SessionName	eq, ne	Любая допустимая строка символов
Status	eq, ne	Running, Not Responding, Unknown
Username	eq, ne	Любое допустимое имя пользователя (только имя или в формате *домен**пользователь*)
WindowTitle	eq, ne	Любая допустимая строка символов

Строка фильтра должна быть заключена в двойные кавычки. Рассмотрим несколько примеров, поясняющих применение фильтров.

Поиск не отвечающих процессов:

```
tasklist /fi "status eq not responding"
```

Примечание При работе с удаленными системами нельзя задать фильтр по полям статуса или заголовка окна. Чтобы обойти эту проблему, иногда вывод перенаправляют в команду FIND, например **tasklist /v /s Mailer1 /u adatum\wrstanek | find /i "not responding"**. Заметьте, что в этом случае поле, по значению которого осуществляется отбор, должно быть в числе выводимых. Именно поэтому в приведенном примере был добавлен параметр /V. Более того, при помощи параметра /I нужно указать, чтобы команда find игнорировала регистр букв.

Поиск процессов на компьютере Mailer1, занимающих процессорное время более 30 минут:

```
tasklist /s Mailer1 /fi "cputime gt 00:30:00"
```

Поиск процессов на компьютере Mailer1, использующих более 20000 Кб памяти:

```
tasklist /s Mailer1 /u adatum\wrstanek /fi "memusage gt 20000"
```

Ввод нескольких параметров /Fi "Filter", чтобы указать, что вывод должен удовлетворять нескольким условиям:

```
tasklist /s Mailer1 /fi "cputime gt 00:30:00" /fi "memusage gt
20000"
```

Мониторинг процессов и использования системных ресурсов

Process Resource Monitor (Pmon) показывает «моментальный снимок» используемых системных ресурсов и выполняемых процессов. После запуска (вводом **pmon** в командной строке) эта утилита собирает информацию об использовании ресурсов и выполняемых процессах в локальной системе и выводит результаты в консольное окно. Статистика автоматически обновляется каждые пять секунд. Pmon продолжает работу, пока вы не нажмете клавишу Q для выхода; нажатие любой другой клавиши приводит к обновлению информации.

Примечание Перенаправлять вывод утилиты Pmon нельзя, ее применение возможно только на локальном компьютере. Для проверки ресурсов удаленного компьютера используется удаленный доступ к компьютеру через Remote Desktop (Удаленный рабочий стол). Кроме того, Pmon не совместима с командой REMOTE.

Pmon выводит данные в табличном формате:

```
Memory:  523248K Avail: 300516K  PageFlts:   905 InRam Kernel: 2444K P:11496K

Commit: 337868K/ 214648K Limit:1280320K Peak: 345720K  Pool N: 8372K P:11648K

                  Mem  Mem   Page  Flts Commit  Usag       Pri  Hnd   Thd      Image
CPU Cpu Time     Usage Diff Faults  Diff Charge  NonP Page       Cnt   Cnt      Name
                 39448   64  59570   282                                     FileCache
 96 0:38:42      16    0     0     0     0     0    0     0     0     1 IdleProcess
  0 0:00:03      216   0     4080  0     28    0    0     8  1810    59 System
  0 0:00:00      480   0     197   0     164   0    5     11   17     3 smss.exe
  2 0:00:09      5236  56    2803  24    3216  5    48    13  756    10 csrss.exe
  0 0:00:01      4624  0     12878 0     7620  8    50    13  537    21 sqlgea.exe
  0 0:00:05      4740  0     2181  0     3932  12   52    9   388    19 servi-
                                                                        ces.exe
  0 0:00:04 30676  0     19113 0     28856 83   80    9  1040    61 sqlmon.exe
  0 0:00:00      2860  0     780   2     1040  23   21    8   242    11 sdman.exe

  0 0:00:00      3788  0     1076  0     1272  4    28    8   127    14 sdman.exe
  0 0:00:00      944   0     232   0     340   1    7     13    7     1 pmon.exe
  0 0:00:00      1776  0     464   0     536   1    25    8   15     1 notepad.exe
```

Как видите, в первых двух строках сообщаются сводные данные об использовании памяти. Значения даются в килобайтах (Кб) и предоставляют следующую информацию.

- **Memory, Avail** — информация об общем объеме оперативной памяти (RAM) в системе. *Memory* показывает физический объем RAM, а *Avail* — не используемую в данный момент RAM.
- **InRam Kernel** — информация о памяти, занятой ядром операционной системы. Критически важные блоки памяти ядра всегда должны помещаться в RAM и не могут быть выгружены в виртуальную память. Этот тип памяти ядра обозначается как InRam Kernel. Остальная часть памяти ядра может быть выгружена в виртуальную память и показывается после InRam Kernel.

- **Commit, Limit, Peak** — сведения об объеме переданной (committed) физической и виртуальной памяти. *Commit* указывает объем физической памяти, для которой зарезервировано место в страничном файле (page file) на диске. Далее сообщается текущий объем переданной виртуальной памяти. *Limit* указывает объем виртуальной памяти, которая может быть передана без увеличения размера страничного файла (или файлов). *Peak* отражает максимальный объем памяти, использовавшийся системой с момента ее запуска. Если разница между общим объемом доступной памяти и используемой переданной памятью почти все время невелика, увеличение объема физической памяти повысит производительность системы. Если максимальное использование памяти отличается не более чем на 10% от значения *Limit*, то целесообразно увеличить объем физической или виртуальной памяти (либо и той, и другой).
- **Pool N и P** — сведения о пулах подкачиваемой (paged pool) и неподкачиваемой памяти (non-paged pool) соответственно. Пул первого вида — это системная память, которая может быть сброшена на диск, как только перестает использоваться, а пул второго — системная память, не выгружаемая на диск ни при каких условиях. *Pool N* соответствует размеру пула неподкачиваемой памяти, а значение, следующее за ним (*Pool P*), — размеру пула подкачиваемой.

Вслед за двумя строками статистики, относящейся к памяти, отображается информация об использовании ресурсов индивидуальными процессами. Эти данные предоставляют массу сведений о выполняемых процессах, которые помогут выявить процесс, пожирающий такие системные ресурсы, как процессорное время и память. Соответствующие поля содержат значения:

- **CPU** — процентная доля процессорного времени для процесса;
- **CpuTime** — общее процессорное время, выделенное процессу с момента его запуска;
- **Mem Usage** — объем памяти, занимаемой процессом;
- **Mem Diff** — изменение объема занимаемой процессом памяти с момента последнего обновления данных;
- **Page Faults** — число ошибок страниц. Ошибка страницы происходит, когда процесс запрашивает страницу памяти,

а система не может ее найти в заданном месте. Если страница находится в другом месте памяти, ошибку страницы называют программной (soft page fault), а если запрошенная страница должна быть прочитана с диска — аппаратной (hard page fault). Большинство процессоров способно обрабатывать большое количество программных ошибок страниц. Однако аппаратные ошибки страниц вызывают значительные задержки, и, если их много, следует подумать об увеличении объема физической памяти или об уменьшении размера системного кэша. Как определить число аппаратных ошибок страниц, см. в разделе «Мониторинг подкачки памяти для индивидуальных процессов» далее в этой главе;

- **Flts Diff** — отражает изменение числа ошибок страниц для процесса с момента последнего обновления данных;
- **Commit Charge** — показывает объем виртуальной памяти, переданной процессу;
- **Usage NonP/Page** — показывает задействованные процессом объемы пулов неподкачиваемой и подкачиваемой памяти. Вы должны обращать внимание на процессы, которые расходуют большой объем пула неподкачиваемой памяти. Если на сервере недостаточно свободной памяти, эти процессы могут быть причиной большого числа ошибок страниц;
- **Pri** — сообщает приоритет процесса. Приоритет определяет, сколько системных ресурсов может быть выделено процессу. Стандартные приоритеты таковы: Low (низкий) (4), Below Normal (ниже обычного) (6), Normal (обычный) (8), Above Normal (выше обычного) (10), High (высокий) (13) и Real-Time (реального времени) (24)*. У большинства процессов по умолчанию обычный приоритет. Наивысший приоритет предоставляется процессам **реального времени.** Вы можете увидеть и другие приоритеты. Например, у потока

* В Task Manager русской версии Windows XP эти приоритеты называются соответственно низкий, ниже среднего, средний, выше среднего, высокий и реального времени. — *Прим. перев.*

простоя (процесса Idle)* значение приоритета равно 0**, так как этот поток не расходует процессорное время, а отслеживает, когда процессор не используется. Некоторые процессы системных служб имеют приоритет 9 или 11, что обеспечивает важным процессам приоритет, чуть превышающий обычный уровень или уровень выше обычного;

- **Hnd Cnt** — общее количество описателей (handles) файлов, открытых процессом. Используйте это значение, чтобы проверить, насколько зависим от файловой системы данный процесс. У некоторых процессов, например Microsoft Internet Information Services (IIS), тысячи открытых описателей файлов. Для каждого открытого описателя файла требуется определенный объем системной памяти;
- **Thd Cnt** — текущее число потоков, используемых процессом. Большинство серверных приложений являются многопоточными. Многопоточность (multithreading) обеспечивает параллельное выполнение запросов процесса. Некоторые приложения для большей производительности умеют динамически управлять количеством параллельно выполняемых потоков. Однако слишком большое число потоков приводит к чрезмерно частому переключению контекстов потоков, фактически понижая производительность;
- **Image Name** — имя процесса или исполняемого файла, запустившего процесс.

Исследуя процессы, помните, что одно приложение может запустить несколько процессов. Как правило, эти процессы зависят от основного процесса, начиная с которого формируется дерево зависимых процессов. Поэтому при завершении процессов нужно выполнять эту операцию по отношению к основному процессу приложения или непосредственно к самому приложению, а не к зависимым процессам. Это гарантирует корректное завершение приложения (без утечки ресурсов).

---

* В Task Manager русской версии Windows XP этот процесс называется «Бездействие системы». — *Прим. перев.*

** На самом деле у этого потока вообще нет никакого приоритета, так как нулевой приоритет может быть только у системного потока обнуления страниц. — *Прим. перев.*

Если вы используете Pmon для анализа выполняемых процессов, обратите внимание на три уникальных процесса:

- **File Cache** — кэш файловой системы, который представляет собой область физической памяти, где хранятся последние использовавшиеся страницы данных для приложений. Когда в файловом кэше происходят изменения, осуществляются операции ввода-вывода. Используемая память показывает общий объем физической памяти, занимаемой файловым кэшем. Ошибки страниц показывают число запрошенных, но не найденных страниц в кэше файловой системы. Такие страницы должны быть считаны либо из другого места памяти, либо с диска. Отслеживая параметр Flts Diff для File Cache, вы можете определить число промахов кэша. Устойчиво высокий уровень промахов кэша указывает на необходимость увеличения объема физической памяти в системе.
- **Idle Process** — в отличие от других процессов Idle (бездействие системы) отслеживает неиспользуемое процессорное время. Таким образом, цифра 99 в столбце CPU для процесса Idle означает, что 99% процессорных ресурсов в настоящий момент не задействовано. Если вы полагаете, что система перегружена, понаблюдайте за процессом Idle. Следите за загруженностью процессора и общим значением процессорного времени. Если у системы устойчиво низкое значение времени простоя (бездействия) (что говорит о значительной нагрузке на процессор), подумайте о замене процессора на более производительный или даже об увеличении числа процессоров.
- **System** — отражает использование ресурсов для локального системного процесса.

Останов процессов

Чтобы остановить процессы в локальной или удаленной системе, применяйте утилиту командной строки Taskkill. Процесс можно остановить по его идентификатору при помощи параметра /Pid или по имени образа, используя параметр /Im. Если вам нужно остановить несколько процессов по их идентификатору или имени образа, укажите несколько параметров /Pid или /Im соответственно. Но будьте осторожны с именами образов, так как Taskkill останавливает все процессы, образы которых имеют данное имя. Так что при выполнении трех экзем-

пляров Helpctr.exe все они будут остановлены, если в качестве параметра было задано имя образа.

Как и в случае с Tasklist, по умолчанию Taskkill выполняется с разрешениями зарегистрированного пользователя, и вы можете указать интересующий вас удаленный компьютер, а также задать разрешения другого пользователя. Для этого предназначен расширенный синтаксис:

```
/s Компьютер /u [Домен\]Пользователь [/p Пароль]
```

где *Компьютер* — имя или IP-адрес удаленного компьютера, *Домен* — необязательное имя домена, где находится учетная запись пользователя, *Пользователь* — имя учетной записи пользователя, разрешения которой вы хотите задействовать, а *Пароль* — необязательный пароль для этой учетной записи. Если домен не указан, подразумевается текущий. А если вы не задали пароль, вам будет предложено ввести его.

Примечание Иногда возникает необходимость в принудительном завершении процесса — обычно, когда процесс перестает отвечать при открытии файла, чтении/записи данных или при выполнении других операций. Для принудительного завершения процесса предназначен параметр /F. Этот параметр применим только для процессов локальной системы. Останов процессов в удаленной системе — операция всегда принудительная.

Совет Исследуя процессы, помните, что одно приложение может запустить несколько процессов. Как правило, эти процессы зависят от основного процесса, начиная с которого формируется дерево зависимых процессов. Иногда нужно остановить все дерево процессов, начиная с родительского приложения и включая все зависимые процессы; для этого применяется параметр /T.

Следующие примеры иллюстрируют применение утилиты Taskkill.

Останов процесса с идентификатором 208:

```
taskkill /pid 208
```

Останов всех процессов с именем образа Cmd.exe:

```
taskkill /im cmd.exe
```

Останов процессов 208, 1346 и 2048 на компьютере MAILER1:

```
taskkill /s Mailer1 /pid 208 /pid 1346 /pid 2048
```

Принудительное завершение локального процесса 1346:

```
taskkill /f /pid 1346
```

Останов дерева процессов, начиная с родительского процесса 1248 и включая все дочерние процессы:

```
taskkill /t /pid 1248
```

Для останова только процессов, удовлетворяющих заданным критериям, используйте фильтры, перечисленные в табл. 7-1, кроме Sessionname. Например, можно применить фильтр, чтобы указать, что должны быть остановлены только те экземпляры Cmd.exe, которые не отвечают, а не все (как при применении параметра /Im).

В Taskkill имеется дополнительный фильтр Modules с операторами EQ и NE, что позволяет исключать или включать указанные DLL. Как вы помните, для определения взаимосвязи между выполняемыми процессами и DLL-модулями используется Tasklist с параметром /m. Применяя Taskkill с фильтром Modules и оператором EQ, можно остановить все процессы, использующие указанную DLL. Применение Taskkill с фильтром Modules и оператором NE, гарантирует, что процессы, работающие с указанной DLL, не будут остановлены.

Совет При применении фильтров не указывайте конкретное имя образа или идентификатор процесса в качестве параметров. То есть в этом случае вы останавливаете процессы, исходя лишь из их соответствия критериям фильтра. Например, можно указать, чтобы были остановлены все процессы, которые перестали отвечать.

Как и в случае с Tasklist, вы можете задать несколько фильтров. Строка фильтра должна быть заключена в двойные кавычки. Следующие примеры иллюстрируют применение фильтров с утилитой Taskkill.

Останов не отвечающих экземпляров Cmd.exe:

```
taskkill /im cmd.exe /fi "status eq not responding"
```

Останов всех процессов, идентификатор которых превышает 4, но при условии, что они не отвечают:

```
taskkill /fi "pid gt 4" /fi "status eq not responding"
```

Останов всех процессов, использующих модуль Winspool.drv:

```
taskkill /fi "modules eq winspool.drv"
```

Внимание! Хотя параметры */Im* и */Pid* в предыдущих примерах не применялись, идентификаторы процессов фильтровались так, что только определенные процессы подвергались воздействию. Будьте осторожны, чтобы случайно не остановить процессы system или idle. Как правило, они выполняются под идентификаторами 4 и 0 соответственно, и, если вы их остановите, система прекратит отвечать или завершится.

Мониторинг для выявления и устранения проблем с производительностью

Хотя Pmon — отличная отправная точка для выявления и устранения проблем с производительностью, часто приходится рыть глубже, чтобы понять, существует ли проблема и, если да, в чем ее причина. В более детальном анализе вам помогут следующие средства управления процессами и мониторинга.

- **Memory Monitor (Memmonitor)** — выводит подробную информацию о памяти, используемой процессом.
- **Page Fault Monitor (Pfmon)** — показывает подробные сведения об ошибках страниц, происходящих в системе.
- **Resource Leak Triage Tool (Memtriage)** — протоколирует использование памяти, в том числе записывает детальную информацию о выделенной и освобожденной памяти применительно к индивидуальным процессам. Служит для выявления утечек памяти и получения подробных сведений о пулах памяти.

Примечание Memmonitor, Pfmon и Memtriage предназначены для администраторов. Они входят в состав Windows Server 2003 Resource Kit. При желании Windows Server 2003 Resource Kit можно установить в Windows XP Professional.

Все эти инструменты рассматриваются в следующих разделах.

Мониторинг подкачки памяти для индивидуальных процессов

Pfmon позволяет получать подробные данные о программных и аппаратных ошибках страниц.

Чтобы увидеть программные ошибки страниц, введите:

```
pfmon /c /p ИдентификаторПроцесса
```

где *ИдентификаторПроцесса* — идентификатор процесса, за которым вам нужно наблюдать (идентификатор определяется через Tasklist), например:

```
pfmon /c /p 1348
```

Чтобы увидеть аппаратные ошибки страниц, введите:

```
pfmon /h /p ИдентификаторПроцесса
```

где *ИдентификаторПроцесса* — идентификатор процесса, за которым вам нужно наблюдать (идентификатор определяется через Tasklist), например:

```
pfmon /h /p 1348
```

Pfmon сообщает о происходящих ошибках страниц в реальном времени. Для остановки Pfmon введите Ctrl+C. Ошибки страниц показываются наряду с источником, вызвавшим их, примерно в таком виде:

```
HARD: HttpSendRequestExA+0xab1 : 00878000
HARD: URLQualifyW+0x356e : URLQualifyW+0x0000356D
HARD: CreateDataCache+0x3608 :
WdtpInterfacePointer_UserFree+0x00007165
HARD: OleSetClipboard+0x8df : OleSetClipboard+0x000008DE
HARD: DoFileDownloadEx+0xba2 : DoFileDownloadEx+0x00000BA1
HARD: RtlImageNtHeaderEx+0x3d : 03730000
HARD: RtlSetThreadErrorMode+0x2a1 : 60c0100c
HARD: GetSysColorBrush+0xa4 : 60c0e5f8
```

Хотя разработчикам может быть интересен источник ошибок страниц, администраторов больше интересует число возникающих ошибок. Как я уже объяснял, большинство процессоров умеет обрабатывать большие количества программных ошибок страниц. Программная ошибка просто означает, что система должна найти запрошенную страницу в другом месте памяти. В случае аппаратной ошибки запрошенная страница памяти должна быть загружена с диска, и при большом числе

таких ошибок вам может понадобиться увеличить объем памяти или уменьшить размер системного кэша.

Мониторинг использования памяти и рабочего набора для индивидуальных процессов

Для наблюдения за использованием памяти индивидуальными процессами во всех деталях предназначена утилита Memmonitor. Синтаксис этой команды:

```
memmonitor /p ИдентификаторПроцесса /nodbg [/int Интервал]
```

где *ИдентификаторПроцесса* — идентификатор отслеживаемого процесса, определяемый при помощи Tasklist, параметр /nodbg указывает утилите не переходить в отладчик, а необязательный параметр *Интервал* задает время ожидания в секундах между проверками памяти. По умолчанию интервал — 60 секунд.

Memmonitor выводит данные в следующем виде:

```
Monitor Process 1284 (Name: SQLAgent.exe)MemMon - 0:00:00
    PageFaults       : 13182
    PeakWSSize       : 22704K     WorkingSetSize: 22252K
    PeakPagedPool    :    58K     PagedPool     :    54K
    PeakNonPagedPool :     8K     NonPagedPool  :     7K
    PeakPagefile     : 13632K     Pagefile      : 13176K
MemMon - 0:00:30
    PageFaults       : 16259
    PeakWSSize       : 24800K     WorkingSetSize: 24352K
    PeakPagedPool    :    58K     PagedPool     :    54K
    PeakNonPagedPool :     8K     NonPagedPool  :     8K
    PeakPagefile     : 16256K     Pagefile      : 15804K
```

Как видно из примера, Memmonitor показывает текущее время работы утилиты, а затем выводит детальную информацию о текущем использовании памяти. Как и Pfmon, Memmonitor не завершается самостоятельно. Однако вывод данных с заданным интервалом выполняется, только когда изменяется использование памяти указанным процессом. Завершить работу с Memmonitor можно в любой момент, нажав Ctrl+C.

Поля, выводимые утилитой, содержат следующую информацию:

- **PageFaults** — число программных и аппаратных ошибок страниц, которые произошли в период выполнения процесса;

- **PeakWSSize** — пиковый объем памяти, выделявшейся процессу;
- **PeakPagedPool** — максимальный объем пула подкачиваемой памяти, использовавшийся процессом;
- **PeakNonPagedPool** — максимальный объем пула неподкачиваемой памяти, использовавшийся процессом;
- **PeakPagefile** — максимальный объем памяти из страничного файла, выделявшийся процессу;
- **WorkingSetSize** — объем памяти, выделенной процессу операционной системой;
- **PagedPool** — объем выделенной памяти, которая может быть сброшена на диск;
- **NonPagedPool** — объем выделенной памяти, которая не может быть сброшена на диск;
- **Pagefile** — размер страничного файла, в который может быть выгружена память.

Работая с Memmonitor, вы глубоко погружаетесь в детали использования памяти указанным процессом. Ключевые точки мониторинга — подкачка страниц памяти и рабочий набор (working set). В идеале, если кэш памяти для процесса (и связанных с ним приложений) выделен правильно, то относительная частота подкачки страниц не должна быть слишком частой. Если же этот показатель становится чрезмерно высоким, имеет смысл увеличить размер резидентного в памяти кэша файлов для данного приложения или объем физической памяти в системе.

Примечание Помните, что сказанное весьма относительно, все зависит от конкретного приложения. Некоторые приложения всегда вызывают интенсивную подкачку страниц. Просто учтите, что доступ к страничному файлу гораздо медленнее, чем к физической памяти.

Размер рабочего набора показывает, сколько памяти выделено процессу операционной системой. Если со временем он увеличивается и никогда не возвращается к базовому значению, в процессе возможна утечка памяти. При такой утечке процесс не освобождает используемую им память, и это ведет к снижению быстродействия всей системы.

Совет Если вы предполагаете утечку памяти, используйте инструменты, которые помогут дополнительно проанализировать проблему. К числу таких инструментов относятся Resource Leak Triage Tool (Memtriage), Memory Snapshot (Memsnap) и Pool Monitor (Poolmon). В большинстве случаев лучшим инструментом является Memtriage, который рассматривается в следующем разделе. Memtriage входит в состав Windows Server 2003 Resource Kit, а Memsnap и Poolmon включены в Support Tools для Windows XP Professional и Windows Server 2003.

Детальный анализ использования памяти и выявление источника утечки памяти

Memtriage помогает определить источник предполагаемой утечки памяти. Синтаксис работы с Memtriage таков:

```
memtriage /mp ИмяЖурнала /t ЧислоСнимков /w Интервал
```

где /Mp указывает Windows сделать снимок текущего состояния системы, процесса и пула ядра, *ИмяЖурнала* — это название используемого журнала, *ЧислоСнимков* — частота записи в журнал информации об использовании памяти, а *Интервал* — период времени между снимками.

Совет Не записывайте в журнал информацию о системе, процессе и пуле ядра одновременно. Вместо /Mp можно использовать /M, чтобы отслеживать только информацию о системе и процессе, или /P, чтобы отслеживать только информацию о пуле ядра.

В большинстве случаев вам потребуется создавать множество снимков и собирать информацию в течение нескольких часов. Следующий пример демонстрирует, как пользоваться утилитой Memtriage:

```
memtriage /mp C:\logs\mem.log /t 8 /w 20
```

Эта команда создает восемь снимков памяти с интервалом в 20 минут и записывает результат в файл C:\logs\mem.log. Memtriage выводит следующие данные:

```
Taking snapshot 1 @ 2003/10/24 19:32:32(Pacific Standard Time)
Creating local pool tag file: c:\localtag.txt ...
Poolsnap: Scan local pool tag file: c:\localtag.txt
```

```
Poolsnap: Scan pool tag file: C:\Program Files\Windows
Resource Kits\Tools\pooltag.txt
Sleeping 20 minutes
Taking snapshot 2 @ 2003/10/24 19:52:32(Pacific Standard Time)
Creating local pool tag file: c:\localtag.txt ...
Poolsnap: Scan local pool tag file: c:\localtag.txt
Poolsnap: Scan pool tag file: C:\Program Files\Windows
Resource Kits\Tools\pooltag.txt
```

Как видите, Memtriage создает рабочий файл Localtag.txt и проверяет файл Pooltag.txt из набора ресурсов. Указав /Mp с именем журнала Mem.log, вы создаете три файла журнала:

- **mem.log.system** — содержит снимок (или снимки) информации о системе;
- **mem.log.process** — содержит снимок (или снимки) информации о процессе;
- **mem.log.pool** — содержит снимок (или снимки) информации о пуле ядра.

Примечание Не волнуйтесь — вам не придется копаться в этих файлах. Просто запомните, что такие файлы создаются, и используйте Memtriage для их обработки по завершении работы.

Когда Memtriage завершит работу, вы захотите проанализировать созданные журналы, чтобы обнаружить возможные утечки памяти. Для этого введите команду:

```
memtriage /a ИмяЖурнала
```

где *ИмяЖурнала* — название журнала, заданное в параметре /Mp, например:

```
memtriage /a c:\logs\mem.log
```

Вывод представляет собой очень детальный анализ памяти. В ходе анализа обнаруживаются не только изменения в использовании памяти, но и тенденции, а также определяется почасовой уровень изменений. Вот пример такого вывода:

```
================================** System ================================
Name         Inc-Trend  Object         Change  Start   End    Percent Rate/hour
System       Always     AvailableKByte -1204  261928  260724       0    -21572
System       Sometime   NpagedPoolKByte   16    8276    8292       0       286
```

```
================================= Per Process ============================
```

Name	Inc-Trend	Object	Change	Start	End	Percent	Rate/hour
sqlgea.exe	Sometime	CommitKByte	32	7536	7568	0	573
sqlgea.exe	Sometime	VirtualKByte	256	47956	48212	0	4586
sqlgea.exe	Sometime	Handles	7	533	540	1	125
sqlgea.exe	Sometime	Threads	1	20	21	5	17
sqlmon.exe	Sometime	PagedPoolKByte	1	77	78	1	17
sqlmon.exe	Sometime	NpagedPoolKByte	1	85	86	1	17
sqlmon.exe	Sometime	CommitKByte	52	29604	29656	0	931
sqlmon.exe	Sometime	VirtualKByte	1040	94280	95320	1	18634
sqlmon.exe	Sometime	Handles	10	994	1004	1	179
sdman.exe	Sometime	Handles	1	141	142	0	17
sdman.exe	Always	Handles	4	868	872	0	71
rrsrvc.exe	Sometime	Handles	2	83	85	2	35
sqldbms.exe	Sometime	Handles	2	2050	2052	0	35
mret.exe	Sometime	CommitKByte	12	13320	13332	0	215
mret.exe	Sometime	VirtualKByte	512	353088	353600	0	9173
mret.exe	Sometime	Threads	1	45	46	2	17
wmiprvse.exe	Sometime	CommitKByte	216	2944	3160	7	3870
wmiprvse.exe	Sometime	VirtualKByte	768	29484	30252	2	13760

В этих данных обратите внимание на следующие элементы.

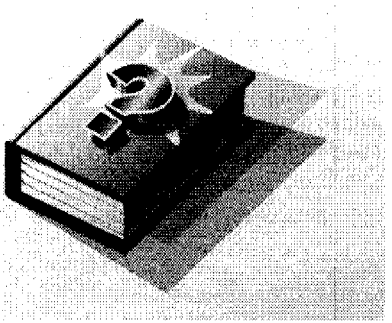
- **Inc-Trend** — указывает, происходит ли увеличение использования памяти и в какой мере. Уделите особое внимание процессам, имеющим тенденцию к увеличению расхода памяти.
- **Change, Start, End** — отмечают начальное и конечное значения объемов используемой памяти и разницу между этими значениями. Отрицательное изменение указывает на освобождение памяти процессом, а положительное — на ее дополнительное выделение.
- **Percent** — определяет процентное изменение между значениями Start и End. Обращайте внимание на значительные изменения, а также на процессы, которые постепенно занимают все больше и больше памяти. Постепенное увеличение занимаемой памяти может указывать на небольшую утечку памяти.
- **Rate/Hour** — указывает примерный ежечасный уровень изменения объема занимаемой памяти, полученный на основании текущего изменения и вычисленной тенденции. Резкие изменения могут указывать на утечку памяти. Однако значения Rate/Hour имеют смысл, если анализ базируется на большом количестве снимков, сделанных в течение нескольких часов.

Часть III

Управление дисками и файловыми системами в Windows

Пользователи хранят на жестких дисках свои документы, электронные таблицы и другие типы данных. Если вы какое-то время уже поработали с Microsoft Windows XP или Windows Server 2003, то, вероятно, обращались к оснастке Disk Management (Управление дисками).

Аналогом этой оснастки является утилита командной строки DiskPart. DiskPart позволяет выполнять большинство задач, связанных с управлением дисками, а также некоторых другие операции, которые нельзя осуществить с помощью GUI-инструмента. Глава 8 представляет собой введение в DiskPart; в ней же рассказывается о таких утилитах, как FSUtil, CHKDSK и CHKNTFS. Глава 9 посвящена базовым дискам и их разметке, а глава 10 — динамическим дискам, их использованию, а также реализации RAID-массивов, управлению ими и устранению возможных проблем.



Глава 8

Конфигурирование и обслуживание жестких дисков

Прочитав эту главу, вы научитесь конфигурировать и обслуживать жесткие диски — а это гораздо важнее, чем многим кажется. В Microsoft Windows Server 2003 и Windows XP Professional жесткие диски могут быть сконфигурированы как базовые (basic disks) или динамические (dynamic disks) и с двумя типами разделов: с основной загрузочной записью (Master Boot Record, MBR) и с таблицей разделов на основе GUID-идентификаторов (GUID Partition Table, GPT). Типы диска и раздела вы выбираете, исходя прежде всего из архитектуры системы. Если вы работаете с системами на основе процессоров x86, вам доступен базовый или динамический тип дисков с разделами MBR. А если вы имеете дело с системами на основе IA64, то можете использовать базовый диск с разделами GPT.

Приступаем к работе с DiskPart

DiskPart — это инструмент для работы с дисками, разделами и томами. При помощи DiskPart вы выполняете такие важные операции, как преобразование типов дисков, создание разделов и томов, конфигурирование RAID-массивов. Помимо этого, DiskPart служит для настройки автоматического монтирования новых дисков в файловой системе, для назначения букв дискам и путей подключенным сетевым дискам. Однако DiskPart не предназначен для форматирования дисков. С этой целью применяется команда FORMAT, которая рассматривается в разделе «Форматирование разделов» главы 9.

Основы DiskPart

В отличие от других утилит командной строки, которые уже были рассмотрены в этой книге, DiskPart — не простая утилита, запускаемая командной строкой с параметрами. Это скорее консольный интерпретатор команд со своей командной строкой и набором внутренних команд. Запускается DiskPart вводом **diskpart** в командной строке.

DiskPart работает с физическими жесткими дисками, установленными в компьютере. CD/DVD-приводы, съемные носители или подключаемые к USB-портам карты флэш-памяти не поддерживаются. Прежде чем использовать команды DiskPart, нужно перечислить, а затем выбрать диск, раздел или том, с которым вы хотите работать, для передачи ему фокуса. Когда диск, раздел или том находится в фокусе, любые команды DiskPart воздействуют именно на этот диск, раздел или том.

Для перечисления доступных дисков, разделов и томов предназначены команды:

- **list disk** — перечисляет все физические жесткие диски компьютера;
- **list volume** — перечисляет все дисковые тома (включая разделы жесткого диска и логические диски);
- **list partition** — перечисляет разделы на диске, который находится в фокусе.

Примечание При перечислении в список включаются CD/DVD-приводы, съемные носители и подключенные к USB-портам карты флэш-памяти. Однако, как я уже говорил, DiskPart не работает с этими устройствами.

При использовании команд list рядом с диском, томом или разделом в фокусе появляется звездочка (*). Вы выбираете диск, том, или раздел по его номеру или букве диска, например disk 0, partition 1, volume 2 или volume D.

Завершив работу с DiskPart, введите **exit** в командной строке DiskPart для возврата в стандартную командную строку.

DiskPart: пример

Чтобы понять, как работать с DiskPart, рассмотрим следующий пример, где запускается DiskPart, перечисляются доступные диски и устанавливается фокус на диск 2.

1. Для запуска DiskPart введите **diskpart** в командной строке.
2. Командная строка примет следующий вид:

```
DISKPART>
```

3. Это говорит о том, что запущен интерпретатор DiskPart. Для перечисления доступных дисков введите **list disk** в его командной строке.
4. Список содержит доступные диски, их состояние, размер и свободное пространство:

```
Disk ###  Status      Size     Free     Dyn  Gpt
--------  ----------  -------  -------  ---  ---
Disk 0    Online      56 GB    0 B
Disk 1    Online      29 GB    0 B
Disk 2    Online      37 GB    33 GB    *
```

5. Так как вы собираетесь работать с диском 2, то должны передать ему фокус, введя команду **select disk 2**.
6. DiskPart сообщит:

```
Disk 2 is now the selected disk.
```

7. По завершении работы с диском введите в командной строке **exit** для выхода из DiskPart.

Что такое фокус

При выборе диска, раздела или тома фокус остается на этом объекте, пока не будет выбран другой. В предыдущем примере фокус был установлен на диск 2, но если вы потом выберете том 2 диска 0, фокус перейдет от диска 2 к тому 2 диска 0. В некоторых случаях фокус передается автоматически, в соответствии с заданной командой. Например, при создании раздела или тома фокус автоматически переходит к новому разделу или тому.

Вы можете передать фокус только разделу выбранного диска. Когда фокус установлен на раздел, связанный с ним том (если он есть) тоже получает фокус. А когда том находится в фокусе, связанный с ним диск и раздел также получают фокус, если данному тому соответствует единственный раздел. Если же том составлен из нескольких разделов, фокус получает только том.

Команды и сценарии DiskPart

LIST и SELECT — лишь две из множества команд, поддерживаемых DiskPart. Полный список команд DiskPart приведен в табл. 8-1. Многие из перечисленных команд принимают Noerr в качестве дополнительного параметра. Параметр Noerr применяется со сценариями DiskPart и указывает, что при возникновении ошибки DiskPart должен продолжать обработку команд сценария. В отсутствие этого параметра DiskPart при ошибке завершит работу, и выполнение сценария прекратится.

- Команды, с которыми применяется Noerr и которые при завершении работы возвращают код ошибки: ADD, ASSIGN, AUTOMOUNT, BREAK, CONVERT, CREATE, DELETE, EXTEND, IMPORT, ONLINE, REMOVE и REPAIR.
- Команды, с которыми Noerr не применяется или которые не возвращают код ошибки: ACTIVE, CLEAN, DETAIL, EXIT, GPT, HELP, INACTIVE, LIST, REM, RESCAN, RETAIN и SELECT.

Табл. 8-1. Сводка команд DiskPart

Команда	Описание	Синтаксис
ACTIVE	На MBR-дисках помечает раздел, имеющий фокус, как активный системный, т. е. этот раздел должен содержать загрузочные файлы операционной системы	active
ADD	Создает зеркальный том на выбранном динамическом диске	add disk=*n* (где *n* — номер диска, который будет содержать зеркало)
ASSIGN	Назначает букву диска или точку монтирования для выбранного раздела, логического диска или тома	assign letter=*x* assign mount=*путь*
AUTOMOUNT	Управляет автоматическим монтированием новых базовых томов, которые добавляются в систему, и назначением им букв дисков (только в Windows Server 2003)	automount enable | disable | scrub
BREAK	Расформировывает зеркальный том. Nokeep указывает, что следует сохранить лишь один том, а другой — удалить	break disk=*n* break disk=*n* nokeep

(см. след. стр.)

Табл. 8-1. (*продолжение*)

Команда	Описание	Синтаксис
CLEAN	Удаляет все разделы или тома на диске, находящемся в фокусе. CLEAN ALL заполняет все секторы диска нулями	clean clean all
CONVERT	Преобразование диска из одного типа в другой	convert basic | dynamic convert gpt | mbr
CREATE	Создает раздел или том указанного типа	create partition efi | extended | logical | msr | primary create volume simple | raid | stripe
DELETE	Удаляет диск, раздел или том, который находится в фокусе	delete disk | partition | volume
DETAIL	Выводит подробные сведения о диске, разделе или томе в фокусе	detail disk | partition | volume
EXIT	Завершает работу DiskPart	exit
EXTEND	Расширяет простой том на выбранном диске или распределяет его по нескольким дискам	extend size=*n* disk=*n*
GPT	Изменяет GPT-атрибуты раздела в фокусе (только в Windows Server 2003)	gpt attributes=*n*
HELP	Выводит список команд	help
IMPORT	Импортирует внешний диск	import
INACTIVE	На MBR-дисках помечает раздел в фокусе как неактивный, т. е. компьютер не удастся загрузить с данного системного раздела и вместо этого будет проверен следующий вариант загрузки в BIOS (только в Windows Server 2003)	inactive
LIST	Выводит список дисков или томов и информацию о них, либо список разделов диска в фокусе	list disk | partition | volume
ONLINE	Подключает выбранный диск или том к RAID-массиву. Проводит повторную синхронизацию зеркального тома или тома RAID-5, который находится в фокусе	online

Табл. 8-1. *(окончание)*

Команда	Описание	Синтаксис
REM	Отмечает начало комментария в сценарии DiskPart	rem *комментарий*
REMOVE	Удаляет букву диска или точку монтирования для выбранного тома. Вы можете добавить необязательные параметры All и Dismount	remove letter=*x* remove mount=*путь*
REPAIR	Восстанавливает том RAID-5, который находится в фокусе, заменяя аварийный том указанным динамическим диском (только в Windows Server 2003)	repair disk=*n*
RESCAN	Поиск новых дисков, которые могут быть включены в систему	rescan
RETAIN	Подготавливает выбранный простой том к использованию в качестве загрузочного или системного	retain
SELECT	Выбирает диск, раздел или том, передавая ему фокус	select disk | partition | volume

Способ использования сценариев с DiskPart слегка отличается от такового для других команд. Причина в том, что DiskPart — это интерпретатор командной строки, а не обычная утилита. Когда вы запускаете DiskPart (вводом **diskpart** в командной строке), интерпретатору нужно сообщить о том, что вы хотите использовать сценарий, добавив параметр /S, как показано ниже:

```
diskpart /s ИмяСценария.txt
```

где *ИмяСценария*.txt — имя текстового файла с нужным сценарием. По умолчанию DiskPart записывает вывод в текущую командную строку. Вы можете перенаправить вывод в файл:

```
diskpart /s ИмяСценария.txt > ФайлЖурнала.log
```

или

```
diskpart /s ИмяСценария.txt >> ФайлЖурнала.log
```

где *ФайлЖурнала*.log — имя текстового файла, в который DiskPart должен записать (или дозаписать) выводимую информацию.

Примечание Помните, что символ > применяется для создания или перезаписи файла при перенаправлении вывода, а >> — для создания или дозаписи существующего файла.

Совет Преимущество использования сценариев перед прямым вводом команд в том, что можно автоматизировать выполнение дисковых операций, причем с их точным воспроизведением в последующем. Создание сценариев для задач управления дисками полезно, если вы устанавливаете Windows в необслуживаемом режиме (unattended setup), с помощью RIS (службы удаленной установки) или утилиты Sysprep, которые не поддерживают создание томов, отличных от загрузочных.

При выполнении сценариев DiskPart проверяйте следующие коды ошибок:

- 0 — ошибок нет, сценарий выполнен полностью;
- 1 — фатальное исключение (возможно, наличие серьезных проблем);
- 2 — некорректные параметры команды;
- 3 — не удалось открыть указанный файл сценария или выходной файл;
- 4 — сервис, используемый DiskPart, вернул код ошибки или сообщил о неудаче;
- 5 — неправильный синтаксис команды (обычно из-за неверного выбора диска, раздела или тома, либо из-за невозможности его использования с данной командой).

DiskPart: пример сценария

Используя сценарии DiskPart, все операции, которые вы хотите выполнить, нужно завершить в рамках одного сеанса. Сценарий должен содержать все необходимые вам команды DiskPart. Включать в сценарий команду EXIT не требуется, так как в конце сценария интерпретатор автоматически завершает свою работу. Рассмотрим пример, показанный на листинге 8-1.

Листинг 8-1. Пример сценария DiskPart

```
rem Выбираем диск 2
select disk 2

rem Создаем основной (primary) раздел на диске
rem и назначаем ему букву диска
create partition primary size=4096assign letter=s

rem Создаем дополнительный (extended) раздел
rem с двумя логическими дисками
create partition extended size=4096
create partition logical size=2048
assign letter=u
create partition logical size=2047
assign letter=v
```

В этом примере создаются основной и дополнительный разделы на диске 2. Размер основного раздела определен в 4096 Мб и ему назначается буква S. Размер дополнительного раздела определен в 4096 Мб; в нем создается два логических раздела: первый — размером 2048 Мб с буквой U, второй — размером 2047 Мб с буквой V. Такие размеры логических разделов заданы из-за некоторых потерь дискового пространства при разбиении на разделы. Вы также могли бы создать единственный логический раздел размером 4096 Мб.

Примечание Создание разделов и назначение букв дискам, как показано в этом примере, еще не означает, что они готовы для использования. Их нужно отформатировать командой FORMAT. О форматировании см. раздел «Форматирование разделов» главы 9.

Совет Так как DiskPart должен выполнить и затем применить изменения, нельзя запускать несколько сценариев подряд. Вместо этого нужно подождать 10–15 секунд до запуска следующего сценария или выполнять все задания в одном сеансе DiskPart. Это гарантирует не только выполнение следующей команды после завершения предыдущей, но и завершение предыдущего сеанса DiskPart до начала следующего.

Сценарий можно запустить командой **diskpart /s *ИмяСценария***, в данном случае — **diskpart /s disk2config.txt**. После запуска этого сценария вывод должен быть таким:

```
Disk 2 is now the selected disk.
DiskPart succeeded in creating the specified partition.
DiskPart successfully assigned the drive letter or mount
point.
DiskPart succeeded in creating the specified partition.
DiskPart succeeded in creating the specified partition.
DiskPart successfully assigned the drive letter or mount
point.
DiskPart succeeded in creating the specified partition.
DiskPart successfully assigned the drive letter or mount
point.
```

Как видите, DiskPart информирует об успешном или неудачном выполнении каждой команды. Имейте в виду, что сценарий не обязательно должен находиться на локальном компьютере. Если сценарий DiskPart хранится на сетевом ресурсе \\corpserver01\scripts, его запуск осуществляется так:

```
diskpart /s \\corpserver01\scripts\disk2config.txt
```

Здесь подразумевается, что сетевой ресурс доступен локальной системе. Вы также можете назначить сетевому ресурсу букву диска командой NET USE. Формат этой команды выглядит следующим образом:

```
net use БуквaДиска \\ИмяКомпьютера\ИмяСетевогоРесурса
```

Применительно к нашему примеру команда получится такой:

```
net use X: \\corpserver01\scripts
```


Примечание Команда NET USE также принимает имя пользователя и пароль в виде /USER:*Домен\Пользователь*. Кроме того, вы можете указать, будет ли подключенный диск постоянным (т.е. сохранится ли связь с сетевым ресурсом после перезагрузки компьютера). Для этого служит параметр /Persistent:Yes. Постоянное подключение сетевого ресурса можно отменить командой **net use *\\ИмяКомпьютера\ИмяСетевогоРесурса*/DELETE**.

Если DiskPart обнаруживает ошибку при выполнении команды, то по умолчанию прекращает выполнение сценария и

показывает код ошибки. Однако, если вы зададите параметр Noerr, DiskPart сообщит об ошибке и продолжит выполнение сценария. Кроме того, не следует вводить команду вызова DiskPart в командной строке. Эта команда должна быть частью общего сценария, который я называю основным (master script). Пример основного сценария приведен на листинге 8-2.

Листинг 8-2. Пример основного сценария

```
@echo off
@if not "%OS%"=="Windows_NT" goto :EXIT
@if "%1"=="" (set INFO=echo && set SEXIT=1) else (set INFO=rem
&& set SEXIT=0)

%INFO% ************************
%INFO% Script: Disk2Setup.bat
%INFO% Creation Date: 6/8/2004
%INFO% Last Modified: 9/23/2004
%INFO% Author: William R. Stanek
%INFO% Email: williamstanek@aol.com
%INFO% ************************
%INFO% Description: Configures the standard partitions
%INFO% on workstations with a third hard drive. The script
%INFO% is configured so that it will only run if you pass
%INFO% in a parameter, which can be any value. This is
%INFO% meant as a safeguard to help prevent accidental
%INFO% formatting of disks.
%INFO% ************************
@if "%SEXIT%"=="1" goto :EXIT

@title "Configuring Disk 2..."
cls
color 07

net use x: \\corpserver01\scripts
diskpart /s x:\disk2config.txt

format s: /fs:ntfs
format u: /fs:ntfs
format v: /fs:ntfs

:EXIT
echo Exiting...
```

Вот и все, что касается введения в DiskPart, — в остальной части главы обсуждаются особенности применения DiskPart и таких команд, как CHKDSK и DEFRAG, для создания, управления и обслуживания дисков, разделов и томов.

Установка жестких дисков и управление ими

Главное предназначение DiskPart — упростить конфигурирование и обслуживание жестких дисков. К основным задачам управления дисками относятся поиск новых дисковых устройств, определение состояния дисков и управление типами таблиц разделов.

Установка и поиск новых устройств

Операционная система Windows поддерживает диски как с «горячей» заменой, так и не допускающие подобную замену. *Горячая замена* (hot swapping) — возможность замены устройства без выключения компьютера. В большинстве случаев диски, допускающие горячую замену, устанавливаются и извлекаются с лицевой части компьютера, и, если компьютер поддерживает горячую замену, вы можете устанавливать диски в компьютер без его выключения. После горячей замены диска запустите DiskPart и введите **rescan** для поиска новых дисков. Найденные новые диски будут добавлены как базовые. Если подключенный диск найти не удалось, перезагрузите компьютер.

Если компьютер не поддерживает горячую замену дисков, вы должны выключить его и затем установить новые диски. Далее, если это необходимо, проверьте наличие нового диска, как описано выше.

Проверка состояния и конфигурации диска

Для проверки состояния дисков служит команда **list disk** командной строки интерпретатора DiskPart. Обычно вывод команды list disk имеет вид:

```
Disk ##   Status    Size      Free      Dyn   Gpt
----      -----     ----      ----      --    --
Disk 0    Online    56 GB     0 B
Disk 1    Online    29 GB     0 B
Disk 2    Offline   37 GB     31 GB
```

Как видите, list disk показывает следующие данные о каждом сконфигурированном в системе диске:

- **Disk ### (Диск ###)** — номер диска;
- **Status (Состояние)** — текущее состояние диска;
- **Size (Размер)** — общая емкость диска;

- **Free (Свободно)** — пространство, доступное для разметки диска (а не объем реально свободного места на диске);
- **Dyn (Дин)** — указывает, что диск относится к динамическому типу, если в этом столбце звездочка, иначе у диска базовый тип;
- **Gpt (Gpt)** — указывает, что тип таблицы разделов на данном диске — GPT, если в этом столбце звездочка. Если звездочки нет, значит, это диск с основной загрузочной записью (Master Boot Record, MBR).

В предыдущем примере у компьютера три базовых диска с разделами типа MBR. Диски 0 и 1 находятся в состоянии Online (Подключен), а состояние диска 2 — Offline (Не подключен). Последний может быть подключен путем передачи ему фокуса (командой **select disk 2**) и выполнением команды **online**.

Знание состояния диска полезно при установке новых дисков, а также при устранении неполадок в работе дисков. Наиболее часто встречающиеся состояния приведены в табл. 8-2.

Табл. 8-2. Некоторые состояния дисков

Состояние	Описание	Решение
Audio CD	Аудио компакт-диск в CD/DVD-приводе	С диском нет никаких известных проблем
Foreign	Динамический диск подключен к компьютеру, но не импортирован для использования. Сбойный диск, вновь подключенный к системе, иногда тоже перечисляется как внешний (foreign)	Добавьте диск в систему командой IMPORT
Initializing	Временное состояние при преобразовании базового диска в динамический	После инициализации состояние автоматически изменится на Online
Missing	Динамический диск поврежден, выключен или не подключен. Это значение появляется как имя диска, а не в столбце состояния	Повторно подключите или включите диск, а затем выполните команду RESCAN для его определения. Если диск не перейдет в нормальное состояние, удалите его из списка дисков командой DELETE DISK

(см. след. стр.)

Табл. 8-2. (*продолжение*)

Состояние	Описание	Решение
No Media	В приводе CD-ROM или дисководе нет носителя. Только CD-ROM и съемные типы дисков могут иметь такое состояние	Вставьте CD-ROM, дискету или съемный диск для перевода диска в состояние Online
Not Initialized	На диске нет сигнатуры. После установки нового диска Windows должна создать MBR или GPT при первом запуске консоли Disk Management (Управление дисками) с помощью мастера, который показывает новые обнаруженные диски. Если прервать работу мастера до записи сигнатуры, у диска будет именно такое состояние	Если вы не запустили консоль Disk Management, сделайте это и при помощи мастера инициализации дисков запишите сигнатуру диска. Другой способ — вызовите контекстное меню диска в Disk Management и выберите команду Initialize Disk (Инициализировать диск)
Offline	Динамический диск временно недоступен или поврежден. Если имя диска изменяется на Missing (Отсутствует), значит, система не может найти его или распознать	Проверьте, все ли в порядке с диском, его контроллером и кабелями. Убедитесь, что диск правильно подключен и на него подано питание. При помощи команды ONLINE переведите диск обратно в состояние Online (если это возможно)
Online	Нормальное состояние диска (динамического или базового). Оно означает, что диск доступен и не имеет проблем	С диском нет никаких известных проблем
Online (Errors)	На динамическом диске обнаружены ошибки ввода-вывода	Временные ошибки можно попытаться исправить командой ONLINE. Она также выполняет повторную синхронизацию зеркальных томов и томов RAID-5

Табл. 8-2. *(окончание)*

Состояние	Описание	Решение
Unreadable	Диск (динамический или базовый) в данный момент недоступен, что бывает при повторном сканировании дисков	Если диски в данный момент не сканируются, то указанный диск поврежден или дает ошибки при вводе-выводе. Команда RESCAN может устранить проблему. Также может потребоваться перезагрузка системы
Unrecognized	Диск неизвестного типа, и система не может его использовать. Это состояние может быть у дисков, используемых системами, отличными от Windows	Данная система не может работать с этим диском, подключите другой

Изменение типа таблицы разделов

Установив диск в компьютер, вы должны сконфигурировать его для использования. Конфигурирование диска заключается в создании разделов и файловых систем на этих разделах в соответствии с вашими потребностями. Разделы могут быть двух типов: с основной загрузочной записью (Master Boot Record, MBR) и таблицей разделов GUID (GUID Partition Table, GPT).

MBR- и GPT-разделы

Компьютеры с процессорами семейства x86 используют разделы типа MBR. MBR содержит таблицу разделов, описывающую их размещение на диске. В этом случае первый сектор жесткого диска включает основную загрузочную запись, в состав которой также входит двоичный код, называемый основным загрузочным кодом, который осуществляет загрузку системы. Этот сектор не входит в состав разделов и скрыт от просмотра для защиты системы.

На MBR-разделах поддерживаются тома размером до 4 терабайтов (Тб), и эти разделы в свою очередь подразделяются на:

- основной (primary);
- дополнительный (extended).

Каждый MBR-диск может содержать до четырех основных разделов или три основных и один дополнительный. Основ-

ные разделы — это части диска, которые непосредственно используются для хранения файлов. После создания файловой системы раздел становится доступным для пользователей. В отличие от основных дополнительные разделы напрямую недоступны. В них нужно сначала создать один или несколько логических дисков, на которых вы сможете хранить файлы. Разметка дополнительного раздела на логические диски позволяет разделить физический диск более чем на четыре части.

Компьютеры на основе процессоров Itanium под управлением 64-разрядных версий Windows используют GPT-разделы. Диски с GPT-разделами содержат два обязательных раздела (required partitions) и минимум один дополнительный (OEM или данные):

- системный раздел EFI (EFI system partition, ESP);
- раздел, зарезервированный Microsoft (Microsoft Reserved partition, MSR);
- по крайней мере один раздел с данными.

Кроме того, GPT-диски поддерживают тома размером до 18 эксабайтов (Эб) и до 128 разделов. Несмотря на различия между GPT- и MBR-разделами, большинство операций, связанных с дисками, выполняется одинаково.

Преобразование типов разделов

Команда CONVERT утилиты DiskPart позволяет преобразовать тип раздела из MBR в GPT и из GPT в MBR. Изменение типа раздела полезно, когда вы:

- перемещаете диск между компьютерами на основе процессоров x86 и IA64;
- получаете новые диски с неподходящей разметкой.

Однако преобразование таблиц разделов возможно лишь на пустых дисках. Это значит, что диск должен быть новым или только что отформатированным. Вы также можете освободить диск, перенеся все существующие разделы или тома на другие диски.

Примечание В DiskPart есть команда CLEAN для стирания всей информации о томах или разделах на диске. Когда вы передаете диску фокус и затем применяете команду CLEAN, такая информация полностью удаляется. На MBR-дисках это приводит к перезаписи таблицы разделов и ин-

формации скрытого сектора. На GPT-дисках перезаписывается информация о GPT-разделах, включая защищенную MBR. Помимо этого, при помощи команды CLEAN ALL можно обнулить все секторы диска.

Внимание! Если у вас нет резервной копии данных, хранящихся на диске, который вы хотите преобразовать, не удаляйте с него разделы или тома. Это приведет к потере всех данных.

Процедура преобразования разделов заключается в следующем.

1. Запустите DiskPart, введя **diskpart** в командной строке.
2. Установите фокус на диск, с которым вы будете работать, например:

```
DISKPART> select disk 2
```

3. Выполните преобразование таким образом:
   - для преобразования из MBR в GPT введите в командной строке **convert gpt**;
   - для преобразования из GPT в MBR введите в командной строке **convert mbr**.

Работа с базовыми и динамическими дисками

Windows Server 2003 и Windows XP поддерживают два типа конфигурации дисков:

- **базовый** — стандартный тип дисков, использовавшийся предыдущими версиями Windows. Базовые диски содержат разделы, и с ними можно работать как в текущей, так и в предыдущих версиях Windows;
- **динамический** — расширенный тип диска, который можно обновлять без перезапуска системы (в большинстве случаев). Динамические диски делятся на один или несколько томов и могут быть объединены в программный RAID-массив.

Примечание Динамические диски не создаются на мобильных компьютерах или сменных носителях. Динамические диски поддерживаются только операционными системами Windows 2000, Windows XP и Windows Server 2003.

Базовые и динамические диски

При обновлении до Windows XP или Windows Server 2003 диски с разделами инициализируются как базовые. При установке Windows XP или Windows Server 2003 на систему с неразмеченным диском предоставляется возможность инициализировать диск либо как базовый, либо как динамический.

Базовые диски поддерживают все отказоустойчивые средства Microsoft Windows NT 4; однако создавать отказоустойчивые дисковые массивы на основе базовых дисков больше нельзя. Если вы хотите настроить программный RAID, то должны преобразовать базовые диски в динамические, а потом создать тома, использующие зеркалирование или чередование. Отказоустойчивые средства и возможность модификации дисков без перезагрузки компьютера — ключевые особенности динамических дисков, отличающие их от базовых.

Хотя на одном компьютере можно использовать и базовые, и динамические диски, операции конфигурирования этих типов дисков различаются. Базовые диски требуют работы с разделами. Это значит, что вы можете:

- форматировать разделы и помечать нужный как активный;
- создавать и удалять основные и дополнительные разделы;
- создавать и удалять логические диски внутри дополнительного раздела;
- преобразовать базовый диск в динамический.

В случае динамических дисков вы работаете с томами. То есть вы можете:

- создавать стандартные и отказоустойчивые тома;
- удалять зеркальный диск из зеркального тома;
- расширять простые или перекрытые (spanned) тома;
- разделять том на два тома;
- восстанавливать зеркальные тома или тома RAID-5;
- повторно активизировать диск, находящийся в состоянии Missing или Offline;
- преобразовать динамический диск обратно в базовый (для чего потребуется предварительное удаление всех существующих томов).

Оба типа дисков позволяют:

- просматривать свойства дисков, разделов и томов;
- назначать дискам буквы;
- настраивать защиту и общий доступ к диску.

Независимо от того, с какими типами дисков вы работаете, помните, что на дисках существуют три особых области.

- **Системная** — системный раздел (том), который содержит аппаратно-зависимые файлы, необходимые для загрузки операционной системы.
- **Загрузочная** — загрузочный раздел (том), содержащий файлы операционной системы. Системный и загрузочный разделы (тома) могут быть совмещены.
- **Активная** — активный раздел (том), с которого начинается загрузка компьютера.

Примечание Существующий динамический том нельзя пометить как активный, но можно преобразовать базовый диск, содержащий активный раздел, в динамический. После этого раздел становится простым активным томом.

Создание активного раздела

На компьютерах с процессорами семейства x86 MBR-раздел можно пометить как активный. Это значит, что с этого раздела будет начинаться загрузка компьютера. Помечать тома динамического диска как активные нельзя. При преобразовании базового диска с активным разделом в динамический диск, этот раздел автоматически становится простым активным томом.

Примечание Прежде чем пометить раздел как активный, удостоверьтесь, что это основной раздел и что на нем есть необходимые загрузочные файлы. Для Windows Server 2003 это Boot.ini, Ntdetect.com, Ntldr и Bootsect.dos. В определенных обстоятельствах может потребоваться и файл Ntbootdd.sys.

Чтобы назначить раздел активным, выполните следующую процедуру.

1. Запустите DiskPart, введя **diskpart** в командной строке.
2. Выберите диск, содержащий раздел, который вы хотите сделать активным, например так:

```
DISKPART> select disk 0
```

3. Перечислите разделы диска командой **list partition**.
4. Выберите необходимый раздел:

```
DISKPART> select partition 0
```

5. Сделайте выбранный раздел активным, введя команду **active**.

Внимание! Номера диска и раздела в этом примере выбраны произвольно, только для демонстрации самой процедуры. Удостоверьтесь, что вы выбрали нужные диск и раздел при выполнении операций по пп. 2 и 4. Если вы выберете в качестве активного раздел, на котором нет загрузочных файлов операционной системы, компьютер не удастся загрузить с жесткого диска.

Изменение типа диска

Windows XP и Windows Server 2003 поддерживают базовые и динамические диски. Иногда возникает необходимость преобразовать диск одного типа в другой, и Windows предоставляет средства для выполнения этой задачи. При преобразовании базового диска в динамический, разделы автоматически преобразовываются в тома соответствующего типа. Однако выполнить обратное преобразование томов в разделы базового диска просто так нельзя. Сначала нужно удалить тома динамического диска и лишь затем преобразовать его обратно в базовый. Удаление томов приведет к потере всей информации на диске.

Преобразование базового диска

Преобразование базового диска в динамический — процесс элементарный, но накладывающий некоторые ограничения. Прежде чем начать эту операцию, примите во внимание следующие соображения.

- С динамическими дисками работают только компьютеры под управлением Windows 2000, Windows XP или Windows Server 2003. Поэтому, если диск, предназначенный для преобразования, содержит более ранние версии Windows, вам не удастся загрузить эти версии после преобразования.
- На дисках с MBR-разделами должно быть не менее 1 Мб свободного места в конце диска. Иначе преобразование не будет выполнено. Консоль Disk Management (Управление

дисками) и DiskPart резервируют это пространство автоматически; однако при применении других утилит для работы с дисками вы должны сами побеспокоиться о доступности этого свободного пространства.

- На дисках с GPT-разделами должны быть непрерывные, распознанные разделы данных. Если GPT-диск содержит разделы, не распознанные Windows, например созданные другой операционной системой, преобразовать такой диск в динамический не удастся.

Помимо сказанного, для любых типов дисков справедливо следующее:

- нельзя преобразовать диски с секторами размером более 512 байтов. Если используются секторы большего размера, диск нужно заново отформатировать;
- динамические диски нельзя создать на портативных компьютерах или сменных носителях. В этом случае диски могут быть только базовыми с основными разделами;
- нельзя выполнить преобразование диска, если системный или загрузочный раздел входит в состав зеркального, перекрытого или чередующегося тома, а также тома RAID-5. Сначала вы должны отменить перекрытие, зеркалирование или чередование;
- однако вы можете преобразовать диски с другими типами разделов, которые входят в состав зеркальных, перекрытых или чередующихся томов, а также томов RAID-5. Эти тома становятся динамическими такого же типа, и вы должны преобразовать все диски набора.

Преобразование базового диска в динамический выполняется в следующей последовательности.

1. Запустите DiskPart, введя **diskpart** в командной строке.
2. Выберите диск, предназначенный для преобразования, например:

```
DISKPART> select disk 0
```

3. Преобразуйте диск, введя команду **convert dynamic**.

Преобразование динамического диска

Для преобразования динамического диска в базовый необходимо удалить все тома диска. Это гарантирует, что диск будет

пуст и все содержащиеся на нем данные перенесены. DiskPart предоставляет команду для стирания всей информации о томе или разделе диска. Это команда CLEAN. Когда вы передаете диску фокус и выполняете команду CLEAN, вся информация о разделах или томах диска удаляется.

Для MBR-дисков это означает, что данные о разделах и скрытый сектор перезаписываются. На GPT-дисках данные о разделах GPT, включая защищенную MBR, перезаписываются. Также можно использовать команду CLEAN ALL для заполнения каждого сектора диска нулевым значением, что полностью удалит все данные на диске.

Преобразование выполняется в следующем порядке.

1. Запустите DiskPart, введя **diskpart** в командной строке.
2. Выберите диск, предназначенный для преобразования, например:

```
DISKPART> select disk 0
```

3. Преобразуйте диск, введя команду **convert basic**.

Динамический диск будет преобразован в базовый, и вы сможете создать на нем новые разделы и логические диски.

Обслуживание жестких дисков

Есть масса утилит командной строки, упрощающих обслуживание жестких дисков. В их число входят FSUtil, ChkDsk и Defrag.

Получение информации о диске и управление файловыми системами с помощью FSUtil

Познакомимся с инструментом, до сих пор не исследованным нами, — с File System Utility (FSUtil).

FSUtil: краткий обзор

FSUtil содержит достаточно сложную структуру команд, но она по сути сводится к тому, что вам необходимо ввести командную строку, содержащую команду и подкоманду, чтобы FSUtil выполнила необходимую задачу. Команды для работы с FSUtil перечислены в табл. 8-3.

Табл. 8-3. Команды FSUtil и их применение

BEHAVIOR	Соответствующие подкоманды позволяют задать или выяснить, как генерируются краткие (MS-DOS) имена файлов и как обновляется последнее время доступа к томам NTFS, насколько часто записываются события квоты (quota events) в системный журнал, каковы уровни внутреннего кэша NTFS пулов подкачиваемой и неподкачиваемой памяти и объем дискового пространства, зарезервированного под Master File Table (MFT)
DIRTY	Соответствующие подкоманды позволяют запросить или установить флаг «данные изменены» (dirty flag) тома. Если флаг установлен, предполагается, что том содержит ошибки и что после перезагрузки компьютера программа AUTOCHK проверит диск, а затем при необходимости запустит Check Disk для исправления ошибок
FILE	Соответствующие подкоманды позволяют искать файл по имени пользователя (только если заданы дисковые квоты), проверять файл на наличие неиспользуемых областей, задавать допустимую длину файла и обнулять части разреженных файлов (sparse files)
FSINFO	Соответствующие подкоманды позволяют перечислять диски компьютера, определять тип диска и получать сведения о томах
HARDLINK	Соответствующие подкоманды позволяют создавать жесткие ссылки (hard links), в результате чего один файл может появиться в нескольких каталогах (или даже в одном каталоге с разными именами). Программы могут открыть любую ссылку для редактирования файла, а сам файл удаляется, только если удалены все ссылки на него
OBJECTID	Соответствующие подкоманды позволяют управлять идентификаторами объектов файлов и каталогов
QUOTA	Соответствующие подкоманды позволяют управлять дисковыми квотами на томах NTFS
REPARSEPOINT	Соответствующие подкоманды позволяют просматривать или удалять точки повторного разбора (reparse points)
SPARSE	Соответствующие подкоманды позволяют управлять разреженными файлами. Разреженным называется файл с одним или несколькими диапазонами, в которых пока нет данных

(см. след. стр.)

Табл. 8-3. *(окончание)*

USN	Соответствующие подкоманды позволяют управлять журналом изменений USN (update sequence number). В этом журнале регистрируются все изменения в файлах на томе
VOLUME	Соответствующие подкоманды позволяют демонтировать тома или запрашивать доступный объем свободного пространства

Применение утилиты FSUtil

Хотя у FSUtil много областей применения, рассчитанных на опытных специалистов (в частности, эта утилита позволяет удалять на дисках точки повторного разбора, управлять дисковыми квотами и создавать разреженные файлы), ее можно использовать и для решения более прозаичных задач, которые будут полезны для получения информации о дисках.

Перечисление дисков. Для этого введите:

```
fsutil fsinfo drives
```

В результате будет показаны доступные диски в алфавитном порядке, например:

```
Drives: A:\ C:\ D:\ F:\ G:\ T:\ U:\
```

Определение типа диска. Зная список дисков в системе, можно определить тип конкретного диска. Для этого введите команду **fsutil fsinfo drivetype** и укажите букву диска с двоеточием, например:

```
C:\>fsutil fsinfo drivetype g:
g: - CD-ROM Drive
```

В данном случае диск G: — привод CD-ROM. Конечно, эта информация может быть получена командой list volume утилиты DiskPart. Тем не менее это еще один способ получения той же информации, который может пригодиться.

Получение детальных сведений о диске. Для этого введите **fsutil fsinfo volumeinfo** и укажите диск. В Windows Server 2003 и Windows XP диск указывается по-разному. В Windows Server 2003 вы вводите букву диска с двоеточием, например:

```
C:\>fsutil fsinfo volumeinfo c:
```

А в Windows XP – букву диска, двоеточие и обратную косую черту, например:

```
C:\>fsutil fsinfo volumeinfo c:\
```

FSUtil перечисляет имя тома, его серийный номер, тип файловой системы и поддерживаемые возможности:

```
Volume Name : Primary
Volume Serial Number : 0x23b36g45
Max Component Length : 255
File System Name : NTFS
Supports Case-sensitive filenames
Preserves Case of filenames
Supports Unicode in filenames
Preserves & Enforces ACL's
Supports file-based Compression
Supports Disk Quotas
Supports Sparse files
Supports Reparse Points
Supports Object Identifiers
Supports Encrypted File System
Supports Named Streams
```

Получение сведений о секторах и кластерах NTFS-диска. Для этого введите **fsutil fsinfo ntfsinfo** и букву диска с двоеточием, например:

```
C:\>fsutil fsinfo ntfsinfo c:
```

Тогда FSUtil покажет подробную информацию о количестве секторов и кластеров, включая сведения об общем числе кластеров, а также количестве свободных и зарезервированных кластеров, примерно в таком виде:

```
NTFS Volume Serial Number :       0x23b36g45
Version : 3.1
Number Sectors :    0x0000000008fcf7c3
Total Clusters :    0x0000000000eb9f38
Free Clusters  :    0x0000000000d12400
Total Reserved :    0x0000000000000000
Bytes Per Sector  :    512
Bytes Per Cluster :    4096
Bytes Per FileRecord Segment  :  1024
Clusters Per FileRecord Segment  :  0
```

Определение свободного дискового пространства. FSUtil позволяет узнать и объем свободного дискового пространства. Введите **fsutil volume diskfree** и укажите диск, например:

```
C:\>fsutil volume diskfree c:
```

FSUtil сообщит общее количество байтов на диске, равно как и общее число свободных и доступных байтов:

```
Total # of free bytes   :   52231667712
Total # of bytes :   60028059648
Total # of avail free bytes   :   52231667712
```

Проверка диска на ошибки и поврежденные сектора

Для проверки диска на ошибки и поврежденные сектора применяется утилита командной строки Check Disk (Chkdsk.exe). Эта утилита проверяет целостность как базовых, так и динамических дисков. Она применяется для проверки и исправления ошибок, обнаруженных на томах FAT, FAT32 и NTFS.

Check Disk способен найти и исправить много видов ошибок. Утилита прежде всего ищет несогласованность в файловой системе и связанных с ней метаданных. Один из способов, при помощи которого Check Disk находит ошибки, — сравнение битовой карты тома с дисковыми секторами, назначенными файлам. Check Disk не позволяет восстановить поврежденные данные внутри файлов, даже если их структура не нарушена.

Анализ диска без исправления ошибок

Вы можете проверить целостность диска, введя имя команды и букву диска с двоеточием. Например, для проверки целостности диска C, введите:

```
chkdsk c:
```

Check Disk выводит отчет по ходу выполнения каждой фазы работы в следующем виде:

```
The type of the file system is NTFS.
Volume label is Primary.

WARNING!  F parameter not specified.
Running CHKDSK in read-only mode.

CHKDSK is verifying files (stage 1 of 3)...
File verification completed.
CHKDSK is verifying indexes (stage 2 of 3)...
Index verification completed.
CHKDSK is recovering lost files.
CHKDSK is verifying security descriptors (stage 3 of 3)...
Security descriptor verification completed.
```

```
CHKDSK discovered free space marked as allocated in the
master file table (MFT) bitmap.
CHKDSK discovered free space marked as allocated in the
volume bitmap.
Windows found problems with the file system.
Run CHKDSK with the /F (fix) option to correct these.

  58621153 KB total disk space.
   7484424 KB in 42596 files.
     13188 KB in 2115 indexes.
        20 KB in bad sectors.
    113933 KB in use by the system.
     65536 KB occupied by the log file.
  51009588 KB available on disk.

      4096 bytes in each allocation unit.
  14655288 total allocation units on disk.
  12752397 allocation units available on disk.
```

Как видите, Check Disk выполняет операции в три этапа. На первом этапе Check Disk проверяет структуру файлов:

```
CHKDSK is verifying files (stage 1 of 3)...
File verification completed.
```

На втором этапе идет проверка индексов:

```
CHKDSK is verifying indexes (stage 2 of 3)...
Index verification completed.
CHKDSK is recovering lost files.
```

Если в результате проверки индексов найдены потерянные файлы, Check Disk восстановит их такими, как они есть. Обычно восстановленные файлы хранятся с расширением .chk в корневом каталоге соответствующего диска.

На третьем этапе Check Disk проверяет дескрипторы защиты:

```
CHKDSK is verifying security descriptors (stage 3 of 3)...
Security descriptor verification completed.
```

В завершение Check Disk выводит отчет, где сообщает, было ли свободное пространство ошибочно отмечено как используемое, и, если да, рекомендует исправить ошибку, запустив Check Disk с ключом/F:

```
CHKDSK discovered free space marked as allocated in the
master file table (MFT) bitmap.
CHKDSK discovered free space marked as allocated in the
```

```
volume bitmap.
Windows found problems with the file system.
Run CHKDSK with the /F (fix) option to correct these.
```

Исправление ошибок диска

Анализируя диск, вы его проверяете, но в действительности ничего не исправляете. Для проверки диска и устранения любых обнаруженных проблем нужно указывать ключ /F, после чего Check Disk будет искать и исправлять ошибки:

```
chkdsk /f C:
```

Check Disk не может восстанавливать тома, которые находятся в использовании. Если том используется, Check Disk запрашивает, хотите ли вы, чтобы том был проверен при следующей загрузке компьютера. Ключ /F подразумевает ключи /R и /X (причем ключ /X применяется только для томов NTFS). Ключ /R задает поиск плохих секторов диска и восстановления читаемой информации, а ключ /X — принудительное отключение NTFS-тома в случае необходимости.

Вы можете сообщить Check Disk выводить более подробную информацию о ходе проверки при помощи ключа /V. Для томов NTFS можно ограничить проверку индексов, задав ключ /I, и пропустить проверку циклов внутри структур папок, указав ключ /C.

Чтобы понять, как применяется Check Disk, рассмотрим несколько примеров.

Поиск и исправление ошибок на диске C:

```
chkdsk /f C:
```

Примечание Помните, что ключ /F подразумевает использование ключей /R и /X.

Поиск и восстановление плохих секторов на диске C:

```
chkdsk /r C:
```

Выполнение минимальной проверки на диске C (NTFS-том):

```
chkdsk /i /c C:
```

Управление автоматической проверкой при загрузке

По умолчанию Windows Server 2003 и Windows XP проверяют при загрузке все диски и, если нужно, запускают Check Disk для исправления любых ошибок. Автоматическую проверку дисков при загрузке контролируют две программы: AUTOCHK и CHKNTFS. Auto Check используется операционной системой для инициации автоматической проверки дисков при загрузке. Напрямую запустить Auto Check нельзя, но можно управлять ее работой с помощью программы Check NTFS, которая позволяет определить, будет ли проверяться диск при следующей загрузке компьютера, и изменить параметры автоматической проверки.

Определение состояния Auto Check

Чтобы выяснить, будет ли диск проверяться при следующей загрузке компьютера, введите:

```
chkntfs Том:
```

где *Том:* — буква проверяемого диска с двоеточием, например:

```
chkntfs c:
```

Вы можете указать несколько дисков. Обозначения дисков разделяются пробелом. В следующем примере определяется состояние дисков C, D и E:

```
chkntfs c: d: e:
```

Check NTFS информирует о типе файловой системы и сообщает, установлен ли для диска флаг «данные изменены» (dirty flag), например:

```
The type of the file system is NTFS.
C: is not dirty.
The type of the file system is NTFS.
D: is dirty.
```

В данном случае флаг «данные изменены» для диска C не установлен, и Auto Check не станет запускать Check Disk для проверки диска. А для диска D такой флаг установлен, что указывает на возможное наличие ошибок, и Auto Check запустит Check Disk для проверки этого диска при следующей загрузке компьютера.

Настройка параметров Auto Check

Check NTFS позволяет настроить параметры работы Auto Check. При перезагрузке компьютера операционная система показывает таймер с обратным отсчетом времени, что дает возможность пользователю отменить автоматическую проверку до ее начала. Указав ключ /T, можно установить длительность отсчета следующим образом:

```
chkntfs /t:ЧислоСекунд
```

где *ЧислоСекунд* — время в секундах для отсчета. Например:

```
chkntfs /t:15
```

Для запрета проверки тома или томов при перезагрузке компьютера, даже если том помечен, как требующий проверки, укажите ключ /X и буквы дисков с двоеточием:

```
chkntfs /x d: e:
```

В данном случае будет пропущена проверка дисков D и E, даже если для них установлен флаг «данные изменены».

Для принудительной проверки томов при перезагрузке компьютера (что для Auto Check является настройкой по умолчанию) укажите ключ /C и буквы дисков с двоеточием, как в следующем примере:

```
chkntfs /c c: d:
```

В данном случае будет выполнена автоматическая проверка дисков C и D при перезагрузке компьютера.

Последний параметр — /D. Он восстанавливает параметры Auto Check по умолчанию, кроме времени обратного отсчета. По умолчанию все диски автоматически проверяются при перезагрузке компьютера.

Дефрагментация дисков

При записи, удалении или перемещении файлов данные на дисках компьютера могут стать фрагментированными. Когда диск фрагментирован, большие файлы могут не поместиться в одну непрерывную область диска. Как результат, файлы записываются в несколько небольших областей диска, и для чтения такого файла с диска потребуется дополнительное время. Для уменьшения степени фрагментации следует периодически анализировать и дефрагментировать диски при помощи утилиты командной строки Defrag.

Использование Defrag

Обычно дефрагментация выполняется в два этапа. Сначала диск анализируется, чтобы определить степень фрагментации и выяснить, нужно ли его дефрагментировать. Оба этапа выполняются одной утилитой Defrag. Для ее запуска введите defrag и укажите букву диска с двоеточием, например:

```
defrag c:
```

Defrag проанализирует диск и, если потребуется его дефрагментация, начнет ее выполнение. Если дефрагментация не нужна, Defrag прекратит работу после выполнения анализа и сообщит, что диск не требует дефрагментации.

Для полной дефрагментации необходимо минимум 15% свободного пространства от общего объема диска. Это пространство используется для сортировки фрагментов файлов. Если на томе меньше 15% свободного пространства, Defrag дефрагментирует его лишь частично. Кроме того, вам не удастся дефрагментировать диск, если для него установлен флаг «данные изменены», что указывает на наличие ошибок на этом диске. Тогда нужно запустить Check Disk и исправить ошибки, прежде чем выполнять дефрагментацию диска.

Defrag принимает несколько параметров. Параметр –a указывает, что нужно выполнить анализ диска без дефрагментации, –v используется для вывода подробной информации, а –f — для принудительной дефрагментации диска даже при нехватке свободного пространства. Однако принудительная дефрагментация таких дисков может оказаться очень длительной или не полной.

Флаг «данные изменены»

Один из способов узнать, установлен ли для диска флаг «данные изменены» (dirty flag), — использовать File System Utility (FSUtil). Введите **fsutil dirty query** и букву диска с двоеточием, например:

```
fsutil dirty query c:
```

Если на диске есть ошибки, которые следует исправить (или был установлен флаг «данные изменены»), FSUtil сообщит:

```
Volume - c: is Dirty
```

Если ошибок на диске нет, FSUtil сообщит:

```
Volume - c: is NOT Dirty
```

Выполнение только анализа диска

Иногда нужно лишь проанализировать диск, чтобы просто определить, надо ли его дефрагментировать. Для анализа диска без дефрагментации введите команду **defrag –a** и букву диска с двоеточием. Defrag выведет отчет, нужна ли дефрагментация диска. Вот пример отчета об анализе диска, не требующего дефрагментации:

```
C:\>defrag -a c:
Windows Disk Defragmenter
Copyright (c) 2001 Microsoft Corp. and Executive Software
International, Inc.

Analysis Report
    28.62 GB Total,  4.78 GB (16%) Free,  2% Fragmented
(5% file fragmentation)

You do not need to defragment this volume.
```

Как видите, диск фрагментирован только на 2%, а степень фрагментации файлов равна 5%. Поскольку фрагментация невысока, диск не требует дефрагментации.

А вот пример отчета об анализе сильно фрагментированного диска:

```
C:\>defrag -a d:
Windows Disk Defragmenter
Copyright (c) 2001 Microsoft Corp. and Executive Software
International, Inc.

Analysis Report
    55.91 GB Total,  48.65 GB (87%) Free,  27% Fragmented
(55% file fragmentation)
You should defragment this volume.
```

В данном случае рекомендуется выполнить дефрагментацию диска, и вы должны запланировать эту операцию.

Глава 9

Разбиение базовых дисков на разделы

При установке нового компьютера или обновлении существующего часто возникает необходимость в разбиении (разметке) дисков на разделы. DiskPart может работать с разделами с основной загрузочной записью (Master Boot Record, MBR) или с таблицей разделов на основе GUID-идентификаторов (GUID Partition Table, GPT). При использовании MBR-разделов на диске может быть до четырех основных разделов или трех основных и одного дополнительного. В случае GPT-дисков в Windows XP Professional или Windows Server 2003 обязательно наличие системного раздела EFI, раздела, зарезервированного Microsoft Reserved, и минимум одного OEM-раздела или раздела данных (всего до 128).

Получение информации о разделах

DiskPart позволяет получить информацию о разделах выбранного диска командой LIST PARTITION. Как видно из следующего примера, LIST PARTITION выводит информацию обо всех разделах диска. Если вы выбрали диск командой **select disk 2** и потом ввели **list partition**, то увидите список разделов диска 2:

```
Partition ###     Type          Size          Offset
-------------     ------        -------       -------
Partition 1       Primary       706 MB        32 KB
Partition 2       Primary       706 MB        706 MB
Partition 3       Primary       706 MB        1412 MB
Partition 4       Extended      1004 MB       2118 MB
Partition 5       Logical       502 MB        2118 MB
Partition 6       Logical       502 MB        2620 MB
```

Итак, LIST PARTITION показывает следующую информацию.

- **Partition ### (Раздел ###)** — номер раздела. Для работы с конкретным разделом применяется команда select partition *N*.
- **Type (Тип)** — тип раздела. Может быть основной (primary), дополнительный (extended) и логический (logical).
- **Size (Размер)** — общий размер раздела.
- **Offset (Смещение)** — смещение раздела в байтах, которое всегда округляется до границ ближайшего цилиндра.

Примечание Цилиндр — это секция диска внутри раздела. Цилиндр в свою очередь состоит из дорожек, которые разбиваются на сектора, а сектор — из группы отдельных байтов. Например, на диске размером 4 Гб может быть 525 цилиндров с 255 дорожками на цилиндр. Каждая дорожка содержит 63 сектора; размер сектора составляет 512 байтов. В данном примере размер цилиндра — 8 Мб, следовательно, смещение раздела всегда будет кратным 8 Мб.

Создание разделов

Порядок создания разделов на базовых дисках зависит от типа таблицы разделов. В связи с тем, что MBR- и GPT-диски имеют разные типы разделов, вопрос создания разделов рассматривается отдельно для каждого типа дисков.

Создание разделов на MBR-дисках

Основные и дополнительные разделы на MBR-дисках создаются при помощи DiskPart. Основной раздел может занимать весь диск или какую-то его часть в соответствии с требуемой конфигурацией. На каждом физическом диске может быть один дополнительный раздел, содержащий один или несколько логических дисков, которые просто являются частями раздела со своими файловыми системами. Хотя вы можете устанавливать размер логического диска по своему усмотрению, следует учитывать, как этот диск будет использоваться на рабочей станции или сервере. Как правило, логические диски предназначены для разбиения больших физических дисков на легко управляемые части. Принимая это во внимание, вы, возможно, захотите разбить 60-гигабайтный дополнительный раздел на три логических диска по 20 Гб каждый.

Создание основных разделов

Перед созданием основного раздела оцените объем свободного места на диске, а также проверьте существующую конфигурацию разделов. Для выполнения этих операций действуйте по следующей схеме.

1. Запустите DiskPart, введя в командной строке **diskpart**.
2. Перечислите диски компьютера командой **list disk** и проверьте свободное место:

```
Disk ###      Status      Size      Free      Dyn   Gpt
--------      ------      ----      ----      ---   ---
Disk 0        Online      56 GB     0 B
Disk 1        Online      29 GB     0 B
Disk 2        Online      37 GB     37 GB
```

3. В данном примере, свободное пространство на диске 2 составляет 37 Гб, доступное для создания разделов. Этот диск не содержит разделов, так как размер всего диска также равен 37 Гб. Если размер диска и объем свободного пространства имеют разные значения, значит, диск уже содержит разделы. Если вы намерены работать с диском 2, выберите его и проверьте разделы на нем, введя следующую последовательность команд: **select disk 2** и **list partition.**

Выбрав диск и передав ему фокус, вы можете создать основной раздел командой:

```
create partition primary size=N
```

где *N* — размер выделяемого пространства в Мб. Если размер не указывается, раздел создается с использованием всего неразмеченного пространства диска.

Примечание В данном примере раздел создается в начале неразмеченного пространства диска. DiskPart делает это, автоматически присваивая параметру offset соответствующее значение. Важно отметить, что смещение округляется до границы ближайшего цилиндра. Такое округление, естественно, влияет на конечный размер раздела или логического диска.

После создания этому разделу автоматически передается фокус, т. е. он становится выбранным. Ему пока не назначена буква или точка монтирования (mount point). Для этой операции нужно выполнить команду ASSIGN. Завершается подготов-

ка раздела его форматированием стандартной внешней командой Windows — FORMAT. Дополнительную информацию см. в разделах «Назначение диску буквы или точки монтирования» и «Форматирование разделов» далее в этой главе.

Создание дополнительных разделов с логическими дисками

На каждом диске может быть один дополнительный раздел. Как и в случае с основными разделами, вы должны оценить объем свободного пространства на диске и проверить существующую конфигурацию разделов перед созданием дополнительного раздела. После этого можно создать дополнительный раздел в неразмеченном пространстве выбранного диска (имеется в виду, что нужный диск уже находится в фокусе).

Дополнительный раздел создается командой:

```
create partition extended size=N
```

где *N* — размер выделяемого пространства в Мб. Если размер не указывается, раздел создается с использованием всего неразмеченного пространства диска.

После создания дополнительному разделу автоматически передается фокус, т. е. он становится выбранным. В отличие от основных дополнительному разделу не назначается буква диска, и он не форматируется непосредственно. Вместо этого вы создаете в нем один или несколько логических дисков, назначаете им буквы, а затем форматируете.

Логический диск в дополнительном разделе создается командой:

```
create partition logical size=N
```

где *N* — размер выделяемого пространства в Мб. Если размер не указывается, логический диск создается с использованием всего неразмеченного пространства дополнительного раздела. Указывая размер логических дисков, помните, что общий объем всех логических дисков дополнительного раздела должен быть меньше размера самого раздела. Вот почему в главе 8 мы создавали дополнительный раздел размером 4096 Мб и логические диски внутри него объемом 2048 и 2047 Мб.

После создания логическому диску автоматически передается фокус, но на этом этапе у него еще нет назначенной буквы или точки монтирования. При помощи команды ASSIGN

вы должны назначить букву диска или точку монтирования, а потом отформатировать логический диск командой FORMAT.

Создание разделов на GPT-дисках

DiskPart позволяет работать с GPT-дисками и создавать системный раздел Extensible Firmware Interface (EFI), а также раздел Microsoft Reserved Partition (MSR) и разделы данных. Создавать разделы EFI и MSR произвольным образом нельзя. На GPT дисках, используемых для запуска 64-разрядных версий Microsoft Windows XP или Windows Server 2003, системный раздел EFI должен быть первым разделом диска, а MSR — вторым. GPT-диски, не являющиеся загрузочными, не содержат системный раздел EFI, и на них раздел MSR должен быть первым. Кроме того, имейте в виду, что Windows не монтирует разделы MSR, поэтому хранить на них данные или удалять их нельзя.

Как и на MBR-дисках, перед разбиением GPT-диска на разделы, оцените объем свободного места на диске, а также проверьте текущую конфигурацию разделов. Выбрав нужный диск и передав ему фокус, вы можете создать раздел следующим образом.

Создание системного раздела EFI:

```
create partition efi size=N
```

Создание раздела MSR:

```
create partition msr size=N
```

Создание основного раздела данных:

```
create partition primary size=N
```

где *N* — объем выделяемого пространства в Мб. Как и на MBR-дисках, DiskPart автоматически присваивает параметру offset соответствующее значение; в большинстве случаев устанавливать это значение самостоятельно не требуется.

Созданный раздел автоматически получает фокус, т. е. становится выбранным. Разделу пока не назначена буква или точка монтирования, что необходимо для раздела EFI и раздела данных. В завершение подготовки разделов данных или EFI их нужно отформатировать через FORMAT, которая является стандартной внешней командой Windows, а не внутренней командой DiskPart.

Управление назначением букв дисков и точками монтирования

После разметки диска каждому разделу можно назначить букву диска или точку монтирования, а затем отформатировать его, чтобы он был готов к хранению данных. Обычно вам доступны буквы от E до Z, буквы от A до D зарезервированы или в большинстве случаев уже задействованы. Во многих системах буква A выделяется дисководу гибких дисков, B зарезервирована за устройством со съемными носителями, C — за первым диском, D — за приводом CD-ROM или DVD.

Если вам нужны дополнительные разделы, вы можете создать их, используя точки монтирования, которые позволяют подключать диски как папки файловой системы, например C:\Data. Но здесь есть одно ограничение — диск можно связывать только с пустой папкой файловой системы NT File System (NTFS).

Назначение букв диска или точек монтирования

Чтобы назначить букву диска или точку монтирования, выполните следующие действия.

1. Запустите DiskPart, введя в командной строке **diskpart**.
2. Перечислите тома и командой **list volume** проверьте, какие буквы дисков им назначены.

Примечание Только команда LIST VOLUME показывает, какие буквы дисков и точки монтирования назначены для всех разделов, логических дисков и томов. Именно поэтому она используется вместо LIST PARTITION. Назначение букв дисков и точек монтирования выполняется одинаково как на базовых, так и на динамических дисках.

3. Назначьте букву диска или точку монтирования (предварительно выбрав нужный раздел):
   - для назначения буквы диска введите **assign letter=x**, где *x* — буква диска, например:

     ```
     DISKPART> assign letter=f
     ```

   - для назначения точки монтирования введите **assign mount=Путь**, где *Путь* — путь к пустой NTFS-папке, которая будет задействована как точка монтирования, например:

     ```
     DISKPART> assign mount=c:\data
     ```

Смена букв диска или точек монтирования

Команда ASSIGN также позволяет сменить существующую букву диска или точку монтирования. Просто выберите нужный раздел и примените команду ASSIGN для установки новой буквы диска или точки монтирования. DiskPart сменит букву диска и сообщит, что для вступления изменений в силу следует перезагрузить компьютер:

```
DiskPart assigned the drive letter, but your computer needs to
be rebooted before the changes take effect.
```

Относительно точки монтирования DiskPart сообщит, что изменения выполнены без необходимости перезагрузки компьютера:

```
DiskPart successfully assigned the drive letter or mount
point.
```

Удаление букв дисков или точек монтирования

Вы можете удалить букву диска или точку монтирования для раздела в фокусе при помощи команды REMOVE. Для этого выполните следующие действия.

1. Запустите DiskPart, введя **diskpart** в командной строке.
2. Перечислите тома командой **list volume** и проверьте текущие назначения. Помните, что лишь команда LIST VOLUME показывает назначенные буквы дисков и точки монтирования для всех разделов, логических дисков и томов.
3. Выберите нужный том командой **select volume** и укажите номер раздела, с которым вы хотите работать. Казалось бы, так делать для тома нелогично, зато это самый простой способ.
4. Удалите назначенную букву диска или текущую точку монтирования для выбранного раздела, введя команду **remove**.

Без параметров эта команда удалит первую букву диска или точку монтирования, которую она обнаружит, и сообщит:

```
DiskPart successfully removed the drive letter or mount point.
```

Этот способ хорош, когда тому присвоена только одна буква диска или точка монтирования. Если тому назначено несколько букв диска или точек монтирования, вам придется указать, какую именно букву диска или точку монтирования следует удалить, введя либо параметр **letter=*x***, либо параметр **mount=*Путь*** соответственно:

```
DISKPART> remove letter=d
```

или

```
DISKPART> remove mount=D:\Data
```

Также можно удалить все буквы и точки монтирования сразу и сообщить DiskPart закрыть все открытые в томе описатели (handles), а затем демонтировать том после удаления букв диска или точек монтирования. Для этого используются параметры All и Dismount, как показано в примерах ниже.

Удаление всех букв диска и точек монтирования:

```
DISKPART> remove all
```

Удаление всех букв диска и точек монтирования с последующим демонтированием соответствующих томов:

```
DISKPART> remove all
```

Удаление тома, смонтированного как диск d:, и его демонтирование:

```
DISKPART> remove letter=d dismount
```

Примечание На MBR-дисках нельзя удалить букву диска для системного, загрузочного раздела или любого раздела, где находится активный страничный файл (файл подкачки) или аварийный дамп памяти. На GPT-дисках нельзя удалить букву диска для EFI-, OEM- или нераспознанного раздела, а также любого раздела, не предназначенного для хранения данных. Однако эта команда позволяет удалить букву диска для устройства со съемными носителями.

Форматирование разделов

При форматировании создается файловая система раздела и все существующие в нем данные безвозвратно теряются*. Windows XP и Windows Server 2003 поддерживают файловые системы FAT, FAT32 и NTFS. FAT — файловая система, также поддерживаемая MS-DOS, Windows 3.1, Windows 95, Windows 98 и Windows Millennium Edition (Windows Me). FAT32 — 32-разрядная версия FAT. NTFS — «родная» файловая система Windows NT, Windows 2000, Windows XP и Windows Server 2003.

* Вообще-то данные после форматирования можно восстановить с помощью сторонних программ вроде Easy Recovery Pro и т. д. — *Прим. перев.*

Примечание Детальное описание каждой файловой системы см. в главе 13 книги «Microsoft Windows Server 2003. Справочник администратора».

Команда Format

Для форматирования раздела служит команда FORMAT. Если в данный момент вы работаете с программой DiskPart, введите **exit** для выхода в стандартную командную строку. Базовый синтаксис команды FORMAT для форматирования жестких дисков выглядит так:

```
format Том /fs:ФайловаяСистема /v:Метка /a:РазмерКластера
```

где *Том* — буква диска или точка монтирования, *ФайловаяСистема* — тип файловой системы, *Метка* — описательное имя (метка), а *РазмерКластера* — размер кластера в байтах. Максимальная длина метки тома составляет 11 символов (включая пробелы). Метка используется с буквами диска, но не с точками монтирования. Если вы не укажете размер кластера, FORMAT выберет стандартный размер в зависимости от размера тома. Допустимые размеры кластеров:

- **512** — 512 байтов на кластер;
- **1024** — 1024 байта на кластер;
- **2048** — 2048 байтов на кластер;
- **4096** — 4096 байтов на кластер;
- **8192** — 8192 байта на кластер;
- **16K** — 16 Кб на кластер;
- **32K** — 32 Кб на кластер;
- **64K** — 64 Кб на кластер.

Чтобы понять, как пользоваться Format, рассмотрим несколько примеров.

Форматирование диска F с созданием файловой системы FAT32 и меткой AppData:

```
format f: /fs:fat32 /v:AppData
```

Форматирование точки монтирования C:\Data с созданием файловой системы NTFS и размером кластера 512 байтов:

```
format c:\data /fs:ntfs /a:512
```

Форматирование диска S с созданием файловой системы NTFS и меткой AppData:

```
format s: /fs:ntfs /v:AppData
```

Примечание Если файловая система уже существует, FORMAT предложит ввести текущую метку в качестве меры предосторожности. После ввода метки вы должны подтвердить необходимость форматирования, которое уничтожит все существующие на диске данные. Обойти эту процедуру нельзя.

Совет В некоторых случаях требуется демонтировать том, прежде чем вы сможете его отформатировать. Для этого предназначен параметр /X. Кроме того, если вы работаете с диском, который уже отформатирован и на нем нет никаких проблем, то можете применить параметр /Q для быстрого форматирования. При быстром форматировании осуществляется подготовка файловой системы к использованию без проверки на возможные ошибки. На больших разделах это обычно позволяет сэкономить несколько минут. Однако при таком форматировании нельзя пометить сектора как сбойные и заблокировать их.

Форматирование: пример

При запуске FORMAT сообщает о типе текущей и вновь создаваемой файловой системы следующим образом:

```
C:\>format e: /fs:ntfs
The type of the file system is RAW.
The new file system is NTFS.
```

Примечание Здесь форматируется неразмеченное пространство, по этой причине тип файловой системы указан как RAW. Некоторые приложения записывают на диск неструктурированные данные (raw data) как битовый поток. В этом случае преимущества разметки диска и файловой системы не используются.

Затем FORMAT предупредит, что любые существующие данные будут потеряны, и запросит подтверждение на продолжение операции:

```
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE E: WILL BE LOST!
Proceed with Format (Y/N)?
```

Если вы продолжите, команда FORMAT проверит раздел и начнет его форматировать, отображая ход операции:

```
Verifying 500M
25 percent completed.
```

После этого создается структура файловой системы и выводится сообщение о завершении форматирования:

```
Creating file system structures.
Format complete.
```

Команда FORMAT также сообщает общий размер дискового пространства и доступное пространство на новом диске:

```
511999 KB total disk space.
507066 KB are available.
```

Совет Для сжатия диска укажите параметр /C. Встроенная поддержка сжатия доступна только для файловой системы NTFS. При этом сжатие осуществляется незаметно для пользователя, и работа со сжатыми файлами не отличается от работы с обычными. Если выбрать этот вариант, файлы и каталоги данного диска будут сжиматься автоматически.

Управление разделами

К наиболее распространенным задачам управления разделами относятся преобразование разделов FAT и FAT32 в NTFS, изменение меток томов, расширение и удаление разделов. Выполнение этих задач рассматривается ниже.

Преобразование разделов или томов в NTFS

Если вы создали раздел или том с файловой системой FAT или FAT32, его можно преобразовать в NTFS без форматирования. Преимущество такого способа в том, что структура файлов и каталогов сохраняется, а данные не теряются. Для преобразования FAT или FAT32 в NTFS служит команда CONVERT.

Преобразование: предварительные проверки

До преобразования нужно выполнить следующее.

- **Проверьте, является ли раздел активным загрузочным или системным, содержащим операционную систему.** В x86-системах активный загрузочный раздел можно преобразовать в NTFS. Однако для этого утилита CONVERT должна получить монопольный доступ к разделу, что возможно лишь при загрузке системы. Таким образом, если вы попытаетесь преобразовать активный загрузочный раздел, то получите запрос на подтверждение преобразования при следующей перезагрузке системы. Ответив Yes и перезагрузив систему, вы инициируете процесс преобразования. Учтите, что для полного преобразования активного загрузочного раздела обычно требуется несколько перезагрузок.
- **Удостоверьтесь, достаточно ли на диске свободного места для выполнения преобразования.** На диске должно быть свободно примерно 25% от общего объема раздела или тома. Например, если размер раздела равен 20 Гб, для его преобразования потребуется около 5 Гб свободного места. CONVERT проверяет наличие свободного места перед преобразованием и в случае его недостатка прерывает работу.

Внимание! Утилиты для преобразования NTFS в FAT нет. Преобразование из NTFS в FAT или в FAT32 возможно только путем удаления раздела или тома и его повторным созданием уже как FAT- или FAT32-тома.

Выполнение базовых преобразований

CONVERT запускается из командной строки. Для преобразования диска используйте следующий синтаксис:

```
convert Том /FS:NTFS
```

где *Том* — буква диска с двоеточием, путь к диску или имя тома. Например, для преобразования файловой системы диска D в NTFS введите:

```
convert D: /FS:NTFS
```

Совет Для томов, преобразованных в NTFS из FAT или FAT32, таблица MFT (Master File Table) размещается в другом месте, что может быть причиной снижения производительности. Для оптимальной производительности вы, вероятно, захотите использовать область, предназначенную для преобразования, как описано в разделе «Использование параметра CvtArea».

При преобразовании загрузочных и системных томов CONVERT устанавливает те же разрешения, что и по умолчанию при установке Windows. Для других томов разрешения задаются так, чтобы обеспечить доступ группе Users (Пользователи) и запретить его группе Everyone (Все). Для предоставления доступа к данным диска группе Everyone укажите параметр /NoSecurity:

```
convert D: /FS:NTFS /nosecurity
```

Внимание! Параметр /NoSecurity удаляет все атрибуты защиты и делает все файлы и каталоги диска доступными группе Everyone.

Утилита CONVERT поддерживает несколько дополнительных параметров. Параметр /V обеспечивает вывод подробной информации при преобразовании, /X демонтирует раздел или том перед преобразованием, если это нужно. Основная причина демонтирования диска перед преобразованием — исключить доступ к диску приложений или процессов на время его преобразования. Однако загрузочный или системный диск демонтировать нельзя. Эти диски будут преобразованы при перезагрузке системы.

Базовая процедура преобразования нормально работает с большинством типов дисков. Но иногда она не дает идеальный результат. Например, преобразованный диск может работать медленнее, чем до преобразования. Для решения этой проблемы служит параметр /CvtArea, который задает имя непрерывного файла в корневом каталоге; на место этого файла помещаются системные файлы NTFS.

Использование параметра CvtArea

В идеале, чем чаще доступ к файлу, тем ближе тот должен быть размещен к началу диска для ускорения поиска и чтения файла. Именно поэтому при форматировании диска определенные системные файлы NTFS помещаются в начало диска. Однако, когда используется базовая процедура преобразования, Windows не может разместить созданные системные файлы NTFS в начале диска, так как это место уже занято другими файлами, которые должны быть сохранены. В результате преобразованный диск может работать несколько медленнее, чем до преобразования.

В Windows Server 2003 и Windows XP для решения этой проблемы можно указать область, предназначенную для преобразования, при помощи параметра CvtArea с последующим заданием имени временного файла. Синтаксис команды таков:

```
convert Том /FS:NTFS /CVTAREA:ИмяФайла
```

где *ИмяФайла* — имя временного файла, заранее создаваемого специально для резервирования пространства. Например:

```
convert C: /FS:NTFS /CVTAREA:temp.txt
```

Здесь указывается, что Master File Table (MFT) и другие файлы метаданных NTFS записываются на место существующего непрерывного временного файла temp.txt. Без параметра CvtArea системные файлы FAT, расположенные в начале диска, не перемещаются. Они просто удаляются, и на их место впоследствии будут записаны обычные файлы. Тогда как при наличии параметра CvtArea утилита CONVERT ищет указанный файл и резервирует соответствующее место в начале диска, предотвращая размещение там обычных файлов. Когда диск конвертируется в NTFS, CONVERT удаляет заданный параметром CvtArea временный файл и помещает вместо него созданные системные файлы NTFS. Таким образом, использование параметра /CvtArea может уменьшить фрагментацию файловой системы после преобразования.

Подстановочный файл создается командой FSUTIL заблаговременно, до начала преобразования. Утилита CONVERT этот файл не создает. Для достижения оптимального результата размер этого файла должен быть установлен из расчета 1 Кб на каждый файл или папку файловой системы. Простейший способ определения количества файлов и каталогов, содержащихся в файловой системе, — проверить свойства каждой папки корневого каталога диска, записать общее число файлов и папок, а затем вычислить общее значение для всех папок корневого каталога. Для этого выполните следующие действия.

1. Запустите Windows Explorer (Проводник), вызовите контекстное меню для папки корневого каталога и выберите команду Properties (Свойства).
2. Отметьте общее число файлов и папок в строке Contains (Содержит) и нажмите OK. (Примите во внимание, что са-

ма папка, свойства которой проверяются, при этом не учитывается.)

3. Повторите эту процедуру для каждой папки корневого каталога, а затем найдите сумму.*

Зная общее число файлов и папок, умножьте это число на 1 Кб для определения размера подстановочного файла. Например, если у вас 1000 файлов и папок, размер подстановочного файла должен быть 1000 Кб. Для создания этого файла введите:

```
fsutil file createnew ИмяФайла Размер
```

где *ИмяФайла* — имя создаваемого файла, а *Размер* — размер файла в байтах. Для создания файла Temp.txt размером 1000 Кб нужно указать размер файла равным 1 024 000 байтов (в каждом Кб содержится 1024 байта):

```
fsutil file createnew temp.txt 1024000
```

Примечание Учтите, что этот файл будет перезаписан метаданными NTFS. Неиспользуемое пространство этого файла после преобразования будет освобождено.

Изменение или удаление метки тома

Метка тома — описательное имя тома, которому назначена буква диска. Ее размер не может превышать 11 символов (включая пробелы) и отображается при доступе к диску различными утилитами, такими как Windows Explorer. Вы можете изменить или удалить метку тома командой LABEL.

Синтаксис изменения метки тома:

```
label диск: метка
```

где *диск:* — буква диска с двоеточием, а *метка* — назначаемое описание, например:

```
label f: AppData
```

---

* Для начала нужно настроить свойства папки для отображения скрытых файлов и папок, которые по умолчанию не отображаются; кроме того, нужно учесть файлы в корневом каталоге. И вообще, гораздо проще в корневом каталоге нажать клавиши Ctrl+A, щелкнуть правой кнопкой мыши и выбрать команду Properties. Так вы сразу увидите суммарный объем, занимаемый всеми файлами и папками на диске, без всяких вычислений. — *Прим. перев.*

Примечание Для работы с метками томов полезна команда VOL. Она выводит метку текущего диска (если таковая есть).

Расширение разделов

Если созданный вами раздел оказался слишком мал, иногда приходится его расширять. Раньше вы могли расширять разделы, используя утилиты сторонних поставщиков. Сейчас DiskPart позволяет расширять существующие разделы так: когда вам нужно расширить последний раздел диска, это можно сделать при помощи команды EXTEND. Последний раздел — единственный, который можно расширить независимо от того, является он основным, дополнительным или логическим диском. Но расширить загрузочный или системный разделы нельзя; кроме того, поддерживается расширение только NTFS-разделов.

Для расширения последнего раздела диска выполните следующие действия.

1. Запустите DiskPart, введя **diskpart** в командной строке.
2. Перечислите диски командой **list disk** и проверьте наличие неразмеченного пространства.
3. Выберите нужный диск, например диск 2, для чего введите **select disk 2**.
4. Перечислите разделы выбранного диска, введя **list partition**.
5. Выберите последний раздел в списке. Например, введите **select partition 6**.
6. Расширьте раздел командой **extend size=*N***, где *N* — размер добавляемого пространства в Мб, например:

```
DISKPART> extend size=1000
```


Примечание Размер округляется до ближайшей границы цилиндра, что обычно немного отличается от указываемого размера. В данном случае размер добавленного пространства составит 1004 Мб. Если размер не указан, раздел расширяется с использованием всего неразмеченного пространства диска.

Совет Вы также можете расширять логические диски внутри дополнительных разделов. В данном случае вы расширили логический диск, а DiskPart автоматически увеличивает размер дополнительного раздела, увеличивая затем размер выбранного логического диска.

Удаление разделов

Для изменения конфигурации полностью размеченного диска, может потребоваться удаление существующих разделов. Удаляя раздел, вы удаляете существующую на нем файловую систему и все данные. Следовательно, перед удалением раздела нужно сделать резервную копию всех файлов и каталогов, которые содержит данный раздел.

На базовых дисках можно удалить раздел, находящийся в фокусе, командой DELETE PARTITION. Но эта команда не годится для удаления системного, загрузочного раздела или любого раздела, содержащего активный страничный файл (файл подкачки) или аварийный дамп памяти. Следующий пример демонстрирует применение команды DELETE PARTITION.

1. Запустите DiskPart, введя **diskpart** в командной строке.
2. Перечислите диски командой **list disk.** Выберите необходимый для работы базовый диск, введя **select disk** и номер диска.
3. Перечислите разделы выбранного диска, введя **list partition.**
4. Выберите раздел для удаления, введя **select partition** с номером раздела, а затем удалите его, введя **delete partition.**

Примечание DiskPart позволяет удалять только известные разделы данных. Вы можете изменить это поведение, если хорошо знаете, что делаете. Для этого к команде DELETE PARTITION добавьте параметр Override.

Глава 10

Управление томами и RAID на динамических дисках

При работе с динамическими дисками вы создаете тома, а не разделы. *Том* (volume) — это просто часть диска, которую можно напрямую использовать для хранения данных. Тома создаются во многом аналогично разделам, но предоставляют массу дополнительных возможностей и позволяют создать:

- *простой том* (simple volume) — том на одном диске;
- *расширенный том* (extended volume) — существующий том расширяется с добавлением в него свободного дискового пространства;
- *перекрытый* том (spanned volume), хранящийся на нескольких дисках;
- RAID-массив — Microsoft Windows Server 2003 и Windows XP Professional поддерживают RAID-0, RAID-1 и RAID-5.

Такие тома и RAID-массивы создаются на динамических дисках, поэтому они доступны только в Windows 2000, Windows XP и Windows Server 2003. Если ваш компьютер может загружать в качестве второй операционной системы предыдущую версию Windows, динамические диски будут в ней недоступны. Однако через сеть динамические диски доступны так же, как и любые другие. То есть компьютеры под управлением предыдущих версий Windows могут обращаться к динамическим дискам через сеть.

Получение сведений о томах и их состоянии

Если при работе с DiskPart требуется выяснить состояние разделов и томов, используйте команду LIST VOLUME. Как показано в примере ниже, LIST VOLUME выводит текущую статистику по всем томам, разделам и логическим дискам компьютера:

```
DISKPART> list volume

Volume ###  Ltr  Label       Fs     Type        Size     Status     Info
-------  ---  -----       ---    ----        ----     ---------  --------
Volume 0     T   Data        NTFS   Simple      502 MB   Healthy
Volume 1     U   Data2       NTFS   Simple      500 MB   Healthy
Volume 2     F               NTFS   DVD-ROM     0 B
Volume 3     G               NTFS   CD-ROM      0 B
Volume 4     C   Primary     NTFS   Partition   56 GB    Healthy    System
Volume 5     D   Secondary   NTFS   Partition   29 GB    Healthy
```

Как видите, LIST VOLUME показывает следующую информацию.

- **Volume ### (Том ###)** Номер тома. Для работы с томом *N* введите команду select volume *N*.
- **Ltr (Имя)** Буква диска тома.
- **Label (Метка)** Метка тома.
- **Fs (ФС)** Тип файловой системы: FAT, FAT32 или NTFS.
- **Type (Тип)** Тип структуры тома. Для динамических дисков возможны следующие типы структуры: simple (простой), spanned (перекрытый), mirrored (зеркальный), striped (чередующийся) или RAID-5.
- **Size (Размер)** Общий размер тома.
- **Status (Состояние)** Состояние тома, например Healthy (Исправен).
- **Info (Сведения)** Дополнительные сведения о томе.

Важные данные в статистике тома — сведения о его состоянии. Знать состояние тома полезно при установке новых томов или при устранении проблем. В табл. 10-1 дано краткое описание состояний, относящихся в основном к динамическим томам.

Табл. 10-1. Проблемы, на которые указывает состояние тома, и их устранение

Состояние	Описание	Как устранить проблему
Data Incomplete	Перекрытый том на внешнем диске является неполным. Скорее всего вы забыли добавить диски в перекрытый том	Добавьте диски, содержащие остальные данные перекрытого тома, затем за один прием импортируйте все диски
Data Not Redundant	Отказоустойчивый том на внешнем диске неполный (не обеспечивает избыточность). Скорее всего вы забыли добавить диски в зеркальный набор или набор RAID-5	Добавьте недостающие диски, затем за один прием импортируйте все диски
Failed	Свидетельствует об ошибке диска. Диск недоступен или поврежден	Убедитесь, что соответствующий динамический диск подключен. При необходимости заново выполните сканирование томов или подключите том командой ONLINE
Failed Redundancy	Свидетельствует об ошибке диска. Один из дисков зеркального набора или набора RAID-5 отключен	Убедитесь, что соответствующий динамический диск подключен. Попытайтесь подключить том. Если это не удается, возможно, требуется заменить сбойный зеркальный диск или восстановить сбойный том RAID-5
Formatting	Временное состояние, указывающее, что диск форматируется	Показывается, на сколько процентов выполнено форматирование. При успешном завершении форматирования состояние должно смениться на Healthy
Healthy	Том находится в нормальном состоянии	Том исправен

Табл. 10-1. *(окончание)*

Состояние	Описание	Как устранить проблему
Regenerating	Временное состояние, указывающее на добавление или импорт зеркального тома или на то, что заново генерируются данные и информация о четности для тома RAID-5	Показывается, на сколько процентов выполнена операция. По завершении состояние должно смениться на Healthy
Resynching	Временное состояние, указывающее на выполнение повторной синхронизации зеркального набора	Показывается, на сколько процентов выполнена операция. По завершении состояние должно смениться на Healthy
Stale Data	Нарушена синхронность данных на внешних отказоустойчивых дисках	Заново выполните сканирование или перезагрузите компьютер, затем проверьте состояние. Оно должно смениться на другое, например на Failed Redundancy
Unknown	Вероятно, поврежден загрузочный сектор тома, и к данным на нем нельзя обратиться	Возможно, диск не инициализирован. Запустите Disk Management. Если после этого мастер Initialize Disk Wizard не запустится автоматически, щелкните диск правой кнопкой мыши и выберите Initialize Disk (Инициализировать диск)

Создание простых томов и управление ими

Работая с динамическими дисками, вы можете создавать простые тома с помощью DiskPart. Это базовый тип динамических томов. В отличие от разделов простой том может занимать весь диск или нужную его часть.

Создание простых томов

Перед добавлением простого тома диска следует определить количество свободного места на диске и посмотреть текущую конфигурацию томов. Для этого выполните следующие операции.

1. Введите **diskpart** в командной строке, чтобы запустить DiskPart.
2. Выведите список дисков и проверьте свободное пространство на них:

```
DISKPART> list disk

Disk ###      Status      Size      Free      Dyn    Gpt
--------      ------      ----      ----      ---    ---
Disk   0      Online      72 GB     0 B
Disk   1      Online      29 GB     20 GB     *
Disk   2      Online      37 GB     37 GB     *
```

В этом примере диски 1 и 2 отформатированы как динамические (на что указывает звездочка в столбце Dyn) с использованием разбиения на MBR-разделы (об этом сообщает пустой элемент в столбце Gpt). На диске 1 имеется 20 Гб свободного пространства, а на диске 2 — 37 Гб.

Выбрав нужный диск, вы можете создать простой том командой:

```
create volume simple size=N disk=N
```

где *size=N* — размер тома в Мб, а *disk=N* указывает, на каком диске создается том.

После создания том автоматически получает фокус, т. е. становится выбранным. У тома пока нет буквы диска или точки монтирования. Чтобы их назначить, нужно выполнить команду ASSIGN. Затем, чтобы завершить подготовку тома, отформатируйте его командой FORMAT (стандартной внешней командой Windows, а не подкомандой DiskPart). Эти операции для томов и разделов выполняются одинаково. См. разделы «Назначение букв диска или точек монтирования» и «Форматирование разделов» главы 9.

Расширение простых томов

Если вы обнаружили, что вам не хватает пространства простого тома, увеличьте его размер одним из двух способов. Первый — расширение простого тома внутри того же диска, что приводит к созданию расширенного тома. А второй — расширение простого тома с распространением на другие диски; тогда вы получите перекрытый том. В любом случае том должен быть отформатирован под NTFS.

Для расширения простого тома выполните следующие операции.

1. Введите **diskpart** в командной строке, чтобы запустить DiskPart.
2. Перечислите диски и посмотрите объем свободного пространства на них:

```
DISKPART> list disk
```

3. Выведите список томов:

```
DISKPART> list volume
```

4. Выберите том, который вы собираетесь расширить, например том 5:

```
DISKPART> select volume 5
```

5. Расширьте том.
   - Для расширения тома за счет свободного пространства на текущем диске используйте команду вида:

```
DISKPART> extend size=N disk=N
```

     где *size=N* — объем добавляемого пространства в Мб, а *disk=N* — диск, на котором в настоящий момент располагается том. Например, если том размещается на диске 2, для увеличения его размера на 1004 Мб введите:

```
DISKPART> extend size=1004 disk=2
```

Примечание Размер округляется до ближайшего значения, кратного размеру цилиндра, поэтому обычно в результате добавляется чуть больше или чуть меньше дискового пространства.

   - Для расширения тома за счет свободного пространства на другом динамическом диске используйте команду вида:

```
DISKPART> extend size=N disk=N
```

     где *size=N* — объем добавляемого пространства в Мб, а *disk=N* — диск, которому принадлежит добавляемое пространство. Например, если том находится на диске 0 и требуется расширить том за счет пространства на диске 1, можно выполнить команду:

```
DISKPART> extend size=2008 disk=1
```

В данном случае вы добавляете в том на диске 0 пространство диска 1. Размер добавляемого пространства диска 1 составляет 2008 Мб.

Внимание! При расширении наборов томов (volume sets) действует ряд ограничений. Нельзя расширять загрузочные или системные тома, равно как и зеркальные или чередующиеся. Нельзя создать перекрытый том, охватывающий более 32 дисков. Более того, нельзя расширять тома с файловой системой FAT или FAT32; файловую систему таких томов нужно сначала преобразовать в NTFS.

Подключение динамических дисков

Динамические диски гибче, чем базовые. Не составляет труда устранить ошибки и вернуть в систему диски, которые были отключены для обслуживания. Кроме того, можно проверить изменения в конфигурации дисков и импортировать диски, переносимые с одного компьютера на другой.

Как рассказывалось в главе 8, команда LIST DISK сообщает состояние каждого диска, доступного в системе. Если для динамического диска показывается состояние Online (Errors) [Подключен (Ошибки)] или Offline (Отключен), во многих случаях можно устранить проблему командой ONLINE. В DiskPart просто выберите нужный диск, например командой **select disk 0**, а затем введите **online**. Если состояние диска не изменилось, возможно, требуется перезагрузить компьютер. Если перезагрузка не решила проблему, проверьте диск, его контроллер, шлейф и питание, чтобы убедиться, что диск подключен правильно. Кроме того, ONLINE заново синхронизирует зеркальные тома или тома RAID-5.

Если изменилась конфигурация диска или диск только что установлен в компьютер, с помощью команды RESCAN можно заново выполнить сканирование всех дисков компьютера и проверить изменения в их конфигурации. Повторное сканирование иногда устраняет проблемы в работе дисков, для которых показывается состояние Unreadable (Нечитаемый).

Если вы перенесли динамический диск с одного компьютера на другой, не исключено, что этот диск будет помечен как Foreign (Внешний). Кроме того, диск помечается как Foreign, если он дал сбой, а затем вновь был подключен. Чтобы с по-

мощью DiskPart подключить диск, выберите его, например командой **select disk 0**, а затем введите **import**.

Удаление томов

При работе с динамическими дисками не следует использовать команду DELETE PARTITION, так как она может удалить все динамические тома на диске. Если вы собираетесь удалить том динамического диска в фокусе, применяйте команду DELETE VOLUME. Как и в случае DELETE PARTITION, эта команда не позволяет удалять системные или загрузочные тома, а также тома, содержащие активный страничный файл (файл подкачки) или аварийный дамп памяти.

Чтобы понять, как работает DELETE VOLUME, рассмотрим следующий пример.

1. Введите **diskpart** в командной строке, чтобы запустить DiskPart.
2. Выведите список томов компьютера:

```
DISKPART> list volume
```

3. Выберите удаляемый том, а затем удалите его:

```
DISKPART> select volume 5

DISKPART> delete volume
```

Примечание По умолчанию DiskPart позволяет удалять только известные тома с данными. Как и в случае разделов, это поведение можно переопределить, добавив в команду DELETE VOLUME параметр Override.

Создание RAID-массивов на динамических дисках

RAID-массивы позволяют лучше защищать важные данные от сбоев дисков. RAID можно реализовать на аппаратном или программном уровне. Аппаратные RAID реализуются и управляются средствами, предоставляемыми производителем аппаратного обеспечения. Программные RAID реализуются и управляются средствами операционной системы.

Windows XP Professional и Windows Server 2003 поддерживают три уровня RAID на динамических дисках.

- **RAID-0** Чередование дисков (disk striping). Два или более тома, каждый из которых находится на отдельном диске, конфигурируются как чередующийся набор (stripe set). Данные разбиваются на блоки, называемые чередующимися областями (stripes); эти блоки последовательно записываются на все диски чередующегося набора. RAID-0 повышает производительность дисковой подсистемы, но не обеспечивает отказоустойчивость.
- **RAID-1** Зеркальные или дуплексные диски. Два тома или два диска конфигурируются идентичным образом. Данные записываются на оба диска. Если один диск дает сбой, данные не пропадают, так как они содержатся еще и на другом диске. RAID-1 обеспечивает избыточность и обладает более высокой производительностью при записи, чем чередующийся набор с записью четности (striped set with parity) (RAID-5).
- **RAID-5** Чередование дисков с записью четности. Три или более тома, каждый на отдельном диске, используются для создания чередующегося набора с контролем четности. При сбое данные можно восстановить. RAID-5 обеспечивает отказоустойчивость с меньшими издержками и обладает более высокой производительностью по чтению, чем при зеркалировании дисков.

Примечание Реализовать программный RAID на базовых дисках нельзя. Windows XP Professional и Windows Server 2003 поддерживают лишь минимальный набор средств управления, позволяющих после обновления системы работать с базовыми дисками, использующими RAID. В этой главе рассматриваются исключительно динамические диски.

Реализация RAID-0

В RAID-0 (чередование дисков) два или более тома, каждый из которых хранится на отдельном диске, конфигурируются как чередующийся набор. Данные, записываемые в чередующийся набор, разбиваются на блоки, называемые *чередующимися областями*. Эти области последовательно записываются на все диски чередующегося набора. Хотя количество дисков в чередующемся наборе может достигать 32, в большинстве случае наилучшую производительность обеспечивают наборы, содержащие от двух до пяти томов, а дальнейший рост числа томов заметно снижает производительность.

Использование RAID-0

Одна из основных причин применения RAID-0 — повышение скорости. Поскольку при обращении к данным происходит доступ к нескольким дискам одновременно, производительность чтения значительно возрастает. Однако возрастает и вероятность катастрофического сбоя. Если любой жесткий диск чередующегося набора дает сбой, набор больше неработоспособен и все его данные теряются. Для восстановления придется заново создать чередующийся набор и восстановить данные с резервных копий. Резервирование и восстановление данных рассматриваются в главе 15 книги «Microsoft Windows Server 2003. Справочник администратора».

При создании чередующихся наборов имейте в виду следующее.

- Загрузочные и системные тома не могут входить в чередующийся набор. Поэтому не используйте эти тома при чередовании дисков.
- Общий размер чередующегося набора определяется объемом наименьшего тома. Поэтому следует использовать тома приблизительно одинакового размера.
- Для большей производительности подключайте диски через разные дисковые контроллеры. Это позволит системе одновременно обращаться к дискам.

Команда LIST DISK или DETAIL DISK показывает для дисков массива RAID-0 тип тома STRIPED. Если выполнить для чередующегося тома команду DETAIL VOLUME, то DiskPart покажет все простые тома, образующие чередующийся набор.

Когда чередующийся том поврежден, его состояние показывается как Missing (Отсутствует). Команда DETAIL DISK, выполненная для одного из оставшихся дисков, должна показать состояние Failed, свидетельствующее о неработоспособности набора. Если вы видите состояние Failed, но не знаете, какие еще диски входят в чередующийся набор, то можете найти проблемный диск, выполнив DETAIL DISK для всех остальных дисков компьютера. Состояние проблемного диска будет показано как Missing.

Для восстановления чередующегося набора обычно удаляют сбойный диск, заменяют его новым и конфигурируют новый диск как входящий в новый чередующийся набор. Для

этого запускают DiskPart, выбирают новый диск и выполняют CONVERT DYNAMIC, чтобы преобразовать тип диска. После этого форматируют новый диск и задают букву диска. Затем в DiskPart удаляют тома на дисках, входящих в поврежденный чередующийся набор, и заново создают чередующийся набор командой CREATE VOLUME STRIPE. По завершении этих операций, если вы выберете чередующийся набор и введете команду LIST VOLUME, состояние должно измениться на Healthy (Исправен).

Внимание! При удалении томов все их данные удаляются с дисков. Вам придется восстановить данные из резервной копии. Если резервной копии дисков нет, не записывайте на них никакой информации. Возможно, часть данных удастся восстановить с помощью утилит восстановления, предлагаемых сторонними поставщиками.

Конфигурирование чередующегося набора

Чтобы реализовать RAID-0, выполните следующие операции.

1. Введите **diskpart** в командной строке, чтобы запустить DiskPart.
2. Выведите список дисков компьютера и объем свободного пространства и убедитесь, что диски, с которыми вы собираетесь работать, сконфигурированы как динамические:

   ```
   DISKPART> list disk
   ```

3. Создайте чередующийся набор командой вида:

   ```
   DISKPART> create volume stripe size=N disk=N,N,N,...
   ```

   где *size=N* — объем пространства в Мб, которое том будет использовать на каждом диске. Если размер не указан, DiskPart задействует все свободное пространство на диске наименьшего размера и столько же места на каждом из остальных дисков. Наконец, *disk=N,N,N,...* задает диски, на которых располагается чередующийся том. Вы должны указать минимум два динамических диска.

   Рассмотрим несколько примеров.

Создание чередующегося тома на дисках 0, 1 и 2 с использованием всего доступного пространства на диске наименьшего размера и такого же объема на каждом из остальных дисков:

```
create volume stripe disk=0,1,2
```

Создание чередующегося тома на дисках 0, 1 и 2 с использованием по 4 Гб (4096 Мб) на каждом диске:

```
create volume stripe size=4096 disk=0,1,2
```

Реализация RAID-1

В RAID-1 (зеркалирование дисков) для создания избыточного набора данных используются тома одинакового размера, хранящиеся на двух разных дисках. В этом случае зеркальные диски содержат идентичную информацию, т. е. данные считываются только с основного зеркального диска, но записываются на оба диска. Поскольку необходима двукратная запись данных, каждый зеркальный диск часто подключается к отдельному дисковому контроллеру, что позволяет одновременно записывать данные на оба диска. Когда используются два дисковых контроллера, говорят, что диски дуплексные. Таким образом, зеркальные и дуплексные диски отличаются тем, сколько дисковых контроллеров ими управляют — один или два (в дальнейшем я не буду проводить различие между зеркальными и дуплексными дисками).

Использование RAID-1

Одна из главных причин для зеркалирования дисков заключается в том, что если один из дисков дает сбой, для чтения и записи данных автоматически используется другой диск. Кроме того, на основе работоспособного диска можно заново сгенерировать данные сбойного диска на том же или другом диске. Чтобы восстановить зеркальный набор, его нужно расформировать (см. раздел «Управление RAID и восстановление после сбоя» далее в этой главе).

Наверно, вы уже поняли, что за зеркалирование диска приходится расплачиваться: зеркалирование в два раза уменьшает объем внешней памяти. Например, для зеркалирования диска размером 80 Гб потребуется еще один диск такого же объема. Следовательно, для хранения 80 Гб информации нужно 160 Гб дискового пространства.

Примечание В отличие от чередования дисков зеркалирование применимо к любым типам простых томов. Следовательно, при необходимости можно зеркалировать загрузочные и системные тома.

При выполнении команды LIST DISK или DETAIL DISK для диска RAID-1 выводится тип тома Mirrored. Если выполнить для зеркального тома команду DETAIL VOLUME, DiskPart покажет сведения обо всех томах зеркального набора.

При любом повреждении зеркального тома выводится состояние Missing. В этом случае, если для одного из остальных дисков выполнить команду DETAIL DISK, сообщается Failed Redundancy (избыточность нарушена). Если вы видите такое состояние, но не знаете, какой еще диск входит в зеркальный набор, найдите проблемный диск, выполнив команду DETAIL DISK для всех остальных дисков компьютера. Состояние проблемного диска будет показано как Missing.

Для восстановления зеркального набора обычно удаляют сбойный диск, заменяют его новым, а затем конфигурируют новый диск как часть зеркального набора. С этой целью сначала запускают DiskPart, выбирают новый диск и выполняют CONVERT DYNAMIC, чтобы преобразовать тип диска. Далее командой BREAK DISK расформировывают существующий зеркальный набор и командой ADD DISK добавляют новый диск в новый зеркальный набор. По завершении этих операций, если вы выберете зеркальный набор и введете команду LIST VOLUME, состояние должно измениться на Healthy (Исправен).

Конфигурирование зеркальных или дуплексных дисков

Чтобы создать зеркальный набор, выберите простой том, для которого требуется создать зеркало, и добавьте диск, используемый в качестве второго диска зеркального набора. На втором диске должно быть доступно свободное пространство, не меньшее объема выбранного тома. Выполните следующие операции.

1. Введите **diskpart** в командной строке, чтобы запустить DiskPart.
2. Выведите список дисков компьютера и объем свободного пространства и убедитесь, что диски, с которыми вы собираетесь работать, сконфигурированы как динамические:

```
DISKPART> list disk
```

3. Выберите диск, для которого требуется создать зеркало. В данном примере выбирается диск 0:

```
DISKPART> select disk 0
```

4. Добавьте диск, который будет вторым диском зеркального набора. В данном примере добавляется диск 1:

```
DISKPART> add disk=1
```

После выполнения этих команд операционная система начнет создание зеркального тома и для обоих томов будет показываться состояние Resynching.

Реализация RAID-5

В RAID-5 (чередование дисков с записью четности) для обеспечения отказоустойчивости используются минимум три жестких диска, на которых создаются тома одинакового размера. Одна из главных причин для применения RAID-5 заключается в том, что он защищает компьютер от сбоя одного диска. Если дают сбой два диска, информации о четности не хватает для восстановления данных, и тогда придется восстанавливать чередующийся набор с резервной копии.

Использование RAID-5

RAID-5 можно рассматривать как усовершенствованную версию RAID-0, в которой также используется чередование для повышения производительности. В отличие от RAID-0 сбой одного диска не приводит к потере работоспособности набора дисков в целом. Набор продолжает функционировать: дисковые операции выполняются над остальными томами набора. Кроме того, по остальным томам можно восстановить чередующийся набор, добавив новый или восстановленный диск. О восстановлении дисков см. в разделе «Управление RAID и восстановление после сбоя» далее в этой главе.

Внимание! Загрузочные и системные тома не могут входить в чередующиеся наборы. Не используйте эти тома при чередовании дисков с записью четности.

При выполнении команды LIST DISK или DETAIL DISK для диска из набора RAID-5 показывается тип тома RAID-5. Если выполнить команду DETAIL VOLUME для тома RAID-5, то DiskPart перечислит все тома, входящие в набор.

Состояние Missing указывает на то, что том поврежден. Если выполнить команду DETAIL DISK для одного из оставшихся дисков, она должна показать состояние Failed Redundancy, свидетельствующее о нарушении избыточности. Если вы видите состояние Failed Redudancy, но не знаете, какие еще диски входят в набор RAID-5, найдите проблемный диск, выполнив DETAIL DISK для всех дисков, которые могут входить в набор RAID-5. Для проблемного диска будет показано состояние Missing (Отсутствует).

Для восстановления набора RAID-5 обычно удаляют сбойный диск, заменяют его новым и конфигурируют новый диск как входящий в набор RAID-5. Для этого запускают DiskPart, выбирают новый диск и выполняют команду CONVERT DYNAMIC, чтобы преобразовать тип диска. Затем командой SELECT DISK выбирают том RAID-5 и выполняют команду REPAIR DISK, указывая, что данный диск будет задействован как новый член массива. Таким образом, заново создается набор RAID-5, членом которого становится новый диск. По завершении этих операций, если вы выберете том RAID-5 и введете команду LIST VOLUME, состояние должно измениться на Healthy (Исправен).

Конфигурирование чередования дисков с записью четности

Для реализации RAID-5 выберите три динамических диска, на которых достаточно свободного места для создания набора RAID требуемого размера. Выполните следующие операции.

1. Введите **diskpart** в командной строке, чтобы запустить DiskPart.
2. Выведите список дисков компьютера и объем свободного пространства и убедитесь, что диски, с которыми вы собираетесь работать, сконфигурированы как динамические:

   ```
   DISKPART> list disk
   ```

3. Создайте набор RAID-5 командой вида:

   ```
   DISKPART> create volume raid size=N disk=N,N,N,...
   ```

   где *size=N* — объем пространства в Мб, которое набор томов будет использовать на каждом диске. Если размер не указан, DiskPart задействует все свободное пространство на диске наименьшего размера и столько же пространства на каждом из остальных дисков. Наконец, *disk=N,N,N,...* зада-

ет диски набора RAID-5. Необходимо указать минимум три динамических диска.

Рассмотрим несколько примеров.

Создание набора RAID-5 на дисках 2, 3 и 4 с использованием всего доступного пространства на диске наименьшего размера и такого же объема на каждом из остальных дисков:

```
create volume raid disk=2,3,4
```

Создание тома RAID-5 на дисках 2, 3 и 4 с использованием по 8 Гб (8192 Мб) пространства на каждом диске:

```
create volume raid size=8192 disk=2,3,4
```

Примечание После создания набор RAID-5 нельзя расширять. Поэтому перед созданием набора следует тщательно продумать его конфигурацию.

Управление RAID и восстановление после сбоя

Управление зеркальными дисками и чередующимися наборами отличается от управления томами других типов. Сбойный зеркальный диск или чередующийся набор нужно восстановить, выполнив определенные операции. Если вы хотите прекратить зеркалирование дисков, то должны расформировать зеркальный набор. А чтобы прекратить использование RAID-5, следует удалить весь набор томов.

Расформирование зеркального набора

Расформирование зеркального набора — стандартная процедура, выполняемая, когда требуется отменить зеркалирование дисков или заново создать зеркальный набор. Если зеркалирование дисков больше не нужно, можно расформировать зеркальный набор и работать с данными только на одном диске. Это позволяет использовать пространство второго диска в других целях. Если один из зеркальных дисков набора дает сбой, дисковые операции продолжают выполняться с использованием другого диска. Чтобы восстановить зеркальный набор, сначала расформируйте его, а потом сформируйте заново.

Совет При расформировании зеркального набора его данные не удаляются, тем не менее перед этой операцией всегда следует резервировать данные. Это гарантирует, что в случае проблем данные можно будет восстановить.

Для расформирования зеркального набора выполните следующие операции.

1. Введите **diskpart** в командной строке, чтобы запустить DiskPart.
2. Выведите список дисков компьютера, чтобы определить, какие диски входят в зеркальный набор:

```
DISKPART> list disk
```

3. Расформируйте зеркальный набор, высвободив заданный диск. У диска, который вы укажете при расформировании зеркального набора, не будет буквы диска и точки монтирования. Например, если зеркалируются диски 0 и 1 и требуется, чтобы пользователи продолжали работать с диском 0, можно расформировать зеркальный набор командой:

```
DISKPART> break disk=1
```

После расформирования зеркального набора у вас будет два диска, содержащих одинаковую информацию. Однако только у диска 0 будет буква диска или точка монтирования. Если вы хотите расформировать зеркальный набор и уничтожить копию информации на втором диске, добавьте параметр nokeep, например:

```
DISKPART> break disk=1 nokeep
```

Повторная синхронизация и восстановление зеркального набора

Когда один из дисков зеркального набора дает сбой, для поддержки зеркалирования нужно восстановить зеркальный набор. Для этого расформируйте зеркальный набор и заново сформируйте его, включив в него новый или восстановленный диск (если сбойный диск удалось восстановить). Бывают случаи, когда происходит не полная утрата работоспособности сбойного диска, а нарушение синхронизации данных. В таких случаях один из дисков по каким-то причинам скорее всего перешел в состояние Offline (Отключен), и из-за этого данные записывались только на один диск.

Для восстановления зеркального набора подключите оба диска зеркального набора и в зависимости от состояния сбойного тома выполните корректирующие операции.

- При состоянии Missing (Отсутствует) или Offline (Отключен) убедитесь в правильности подключения шлейфа и кабеля питания. Запустите DiskPart и попытайтесь обнаружить том командой RESCAN. Затем командой ONLINE заново синхронизируйте зеркальный том. Состояние диска должно смениться на Regenerating, а затем на Healthy. Если состояние не сменилось на Healthy, попробуйте расформировать зеркальный набор, а затем сформировать его заново, добавив восстановленный диск.
- При состоянии Online (Errors) заново синхронизируйте зеркальный том командой ONLINE. Состояние диска должно смениться на Regenerating, а затем на Healthy. Если этого не произошло, отмените зеркалирование командой BREAK, а затем командой ADD заново сформируйте зеркальный набор, добавив восстановленный или новый диск.
- Если для одного из дисков показывается состояние Unreadable, выполните повторное сканирование всех дисков системы командой RESCAN. Если состояние диска не изменилось, попробуйте перезагрузить компьютер.
- Если один из дисков не переходит в подключенное состояние, расформируйте зеркальный набор, удалив из него сбойный диск. Замените или отремонтируйте диск, затем добавьте его командой ADD, чтобы заново создать зеркальный набор.

Примечание Сбой зеркального диска может привести к невозможности загрузки системы. Обычно это происходит при зеркалировании системного или загрузочного тома, когда дает сбой основной зеркальный диск. В таких случаях отредактируйте файл boot.ini, чтобы система запускалась с дополнительного диска зеркального набора (см. главу 12 в книге «Microsoft Windows Server 2003. Справочник администратора».

Восстановление RAID-0

Как уже говорилось, RAID-0 не обеспечивает отказоустойчивость. Если диск, входящий в набор RAID-0, дает сбой, весь чередующийся набор становится непригодным. Перед восста-

новлением чередующегося набора следует отремонтировать или заменить сбойный диск. После этого нужно заново создать массив RAID-0 и восстановить содержавшиеся в нем данные с резервной копии.

Восстановление RAID-5

RAID-5 позволяет восстановить чередующийся набор при сбое одного из дисков. О сбое одного из дисков свидетельствует изменение состояния набора на Failed Redundancy. В зависимости от состояния сбойного тома следует выполнить соответствующую корректирующую операцию.

- При состоянии Missing (Отсутствует) или Offline (Отключен) убедитесь в правильности подключения шлейфа и кабеля питания. Затем запустите DiskPart и командой ONLINE заново синхронизируйте набор томов. Состояние диска должно смениться на Regenerating, а затем на Healthy. Если этого не произошло, выполните команду REPAIR.
- При состоянии Online (Errors) заново синхронизируйте том RAID-5 командой ONLINE. Состояние диска должно смениться на Regenerating, а затем на Healthy. Если этого не произошло, выполните команду REPAIR.
- Если для одного из дисков показывается состояние Unreadable, выполните повторное сканирование всех дисков системы командой RESCAN. Если состояние диска не изменилось, попробуйте перезагрузить компьютер.
- Если один из дисков так и не перешел в подключенное состояние, выполните команду REPAIR.

Для восстановления набора RAID-5 служит команда REPAIR. По возможности перед этой командой следует создать резервную копию данных. Это гарантирует, что в случае проблем вы сможете восстановить данные. Для устранения неполадок с RAID-5 выполните следующие операции.

1. Введите **diskpart** в командной строке, чтобы запустить DiskPart.
2. Выведите список дисков компьютера, чтобы убедиться в том, что один из дисков набора RAID-5 дал сбой:

```
DISKPART> list disk
```

3. Удалите и замените сбойный диск, если это необходимо и возможно. Затем укажите новый диск, который должен войти в набор RAID-5, следующей командой REPAIR:

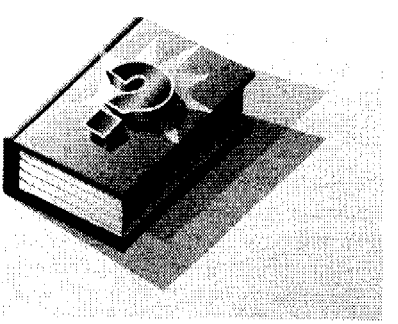
```
DISKPART> repair disk=N
```

где *N* — динамический диск, замещающий сбойный диск набора RAID-5. Учтите, что объем свободного пространства на этом диске должен быть больше или равен размеру сбойного RAID-диска.

Часть IV

Администрирование сетей и управление Active Directory

В этой части книги рассматриваются основные команды, используемые для конфигурирования и администрирования Active Directory, служб печати и TCP/IP-сетей, а также для устранения проблем в их работе. В главе 11 описываются основные средства администрирования службы каталогов, в частности средства запроса информации, хранящейся в каталогах. Тема главы 12 — средства создания учетных записей компьютеров в Active Directory и управления ими. Кроме того, вы познакомитесь с конфигурированием контроллеров домена как глобальных каталогов и координаторов операций (operations masters). В главе 13 завершается рассмотрение службы каталогов. В ней говорится о том, как создавать и конфигурировать учетные записи пользователей и групп в Active Directory. В главе 14 рассказывается о сетевой печати и службах печати. В главе 15 рассматривается применение средств командной строки для конфигурирования, обслуживания и устранения проблем TCP/IP-сетей.



Глава 11

Основы управления Active Directory

Active Directory — одна из самых важных областей управления сетями в Windows. Active Directory — расширяемая и масштабируемая служба каталогов, которая поддерживает охватывающую всю сеть базу данных для хранения учетных записей и информации о ресурсах. При работе с Active Directory вы имеете дело с целостной системой именования, описания, поиска, изменения и защиты информации о ресурсах. Поэтому Active Directory отлично подходит как для управления учетными записями пользователей, групп и компьютеров, так и для работы с приложениями, файлами, принтерами и другими типами ресурсов. Active Directory применяется для управления сетевой инфраструктурой, системного администрирования и управления пользовательскими средами.

Active Directory доступна только в доменах Windows с контроллерами домена под управлением Windows 2000 или более поздней версии. *Контроллер домена* (domain controller) — это сервер под управлением серверной версии операционной системы Windows. Active Directory можно рассматривать как усовершенствование доменной архитектуры Windows NT, в которой на смену базе данных SAM (Security Accounts Manager) пришла более гибкая, расширяемая и масштабируемая база данных. Как и SAM, Active Directory используется в качестве централизованного хранилища информации защиты; однако в отличие от SAM служба Active Directory является еще и средством интеграции гетерогенных систем. Она позволяет выполнять все операции управления ресурсами с помощью единого набора GUI-средств администрирования, работающих в Windows. В этой главе рассматриваются эквивалентные средства командной строки, управляющие службой каталогов.

Управление Active Directory из командной строки

Чтобы использовать многочисленные средства командной строки, управляющие Active Directory, нужны базовые знания об Active Directory и ее структурах. В соответствии с замыслом Microsoft эта служба использует в качестве системы именования DNS (Domain Name System). В DNS информация о сетевых ресурсах хранится в виде иерархической структуры, соответствующей схеме управления ресурсами. Эта иерархия доменов, или дерево доменов, лежит в основе среды Active Directory и во многом аналогична структуре каталогов файловой системы. Еще одно название этой иерархии — *пространство имен* (namespace). В каждой организации, использующей домены Active Directory, существует своя иерархия (или пространство имен) Active Directory.

Домены, контейнеры и объекты

Первый домен Active Directory, который вы создаете, — *корень* дерева и *родитель* всех доменов более низкого уровня. Домены ниже корневого называются *дочерними*. Например, корневым доменом является cpandl.com. Вы можете разделить свою организацию на домены по географическому или функциональному признаку. В первом случае можно определить, скажем, такие дочерние домены: seattle.cpandl.com, ny.cpandl.com и la.cpandl.com. При функциональном подходе дочерними доменами могут быть, например, sales.cpandl.com, support.cpandl.com и tech.cpandl.com. Главное, чтобы имя дочернего домена наследовало имя родительского. Иначе домен относится к другому пространству имен. Так, microsoft.com, msn.com и hotmail.com — пространства имен, отличные от cpandl.com.

При необходимости в пространстве имен можно создать дополнительные уровни. Например, в дочернем домене la.cpandl.com можно создать домены sales.la.cpandl.com, tech.la.cpandl.com и support.la.cpandl.com. Если этого недостаточно, добавьте в пространство имен еще один уровнь, а затем еще и еще. Active Directory управляет отношениями между узлами этого дерева и устанавливает доверительные отношения между доменами.

В Active Directory доверительные отношения являются двусторонними и транзитивными. При создании дочернего домена, например tech.la.cpandl.com, устанавливается двустороннее доверительное отношение между tech.la.cpandl.com и

la.cpandl.com. Доверие распространяется вверх и вниз по дереву. Поскольку la.cpandl.com доверяет cpandl.com, то и tech.la.cpandl.com автоматически доверяет cpandl.com, как и другим доменам того же пространства имен.

Для представления сетевых ресурсов, таких как пользователи, группы и компьютеры, в Active Directory используются *объекты*. Кроме того, специализированные объекты, называемые *контейнерами*, служат для упорядочения сетевых ресурсов по территориальным, функциональным или бизнес-признакам. Обычно контейнеры используются для группирования объектов, имеющих одинаковые атрибуты. Например, если требуется применить определенный набор разрешений ко всем инженерам, это проще сделать, поместив всех этих пользователей в один контейнер.

Каждый контейнер отражает группу объектов, а каждый ресурс представляется уникальным объектом Active Directory. Самый общий тип контейнера Active Directory — организационная единица, или OU (organizational unit). Объекты, помещаемые в OU, принадлежат только домену, к которому она относится. Например, OU, относящиеся к tech.la.cpandl.com, содержат объекты только этого домена. Следовательно, в эти контейнеры нельзя добавить объекты из support.la.cpandl.com, la.cpandl.com или tech.ny.cpandl.com.

У каждого класса объекта Active Directory, такого как контейнер, пользователь, группа или принтер, есть набор атрибутов, описывающий ресурс. Например, у объекта пользователя (user object) имеются атрибуты, описывающие учетную запись пользователя и, в частности, содержащие информацию о контактах, разрешения и привилегии. То есть к атрибутам объекта пользователя относятся имя, фамилия, отображаемое имя, телефонный номер, адрес электронной почты, пароль и т. д.

Поскольку каждый объект Active Directory, в сущности, является записью базы данных, набор атрибутов можно расширить в соответствии с потребностями конкретной организации, в том числе вводить собственные атрибуты, позволяющие лучше описать объект (например, добавить атрибут, содержащий идентификационный код сотрудника).

Логические и физические структуры Active Directory

Пока что мы рассматривали структуры Active Directory, используемые для логической организации данных каталога: домены, поддомены и OU. Вы можете использовать эти струк-

туры для организации Active Directory в соответствии с бизнес-требованиями или требованиями к функциональности. Кроме того, можно использовать их для группирования по территориальному признаку, например создать домены ny.cpandl.com, la.cpandl.com и seattle.cpandl.com.

Домены, поддомены и OU в Active Directory никак не связаны с реальным миром, даже если вы используете территориальное группирование доменов и OU. Это просто области каталога, где хранятся соответствующие данные. В Active Directory они физически находятся в одном месте, пока вы не укажете, какие физические структуры сопоставляются вашим логическим структурам доменов, поддоменов и OU.

На практике каждая из этих логических структур может охватывать более одного участка. И не важно, что представляют собой эти участки, — разные этажи одного здания, разные здания или даже города. Главное, что это различные физические участки. Чтобы сообщить Active Directory об этих участках, вы должны определить подсети и сайты. *Подсеть* (subnet) — часть сети со специфическим диапазоном IP-адресов и сетевой маской. *Сайт* (site) — группа, содержащая одну или несколько подсетей и сопоставленная физической структуре вашей сети. Поскольку сопоставления сайтов не зависят от логической структуры доменов, физическая структура сети и логическая структура доменов не обязательно должны быть связаны между собой.

Вы можете создать несколько сайтов в одном домене или один сайт, охватывающий несколько доменов. Например, если поддомены группируются по территориальному признаку, вы могли бы использовать дочерние домены seattle.cpandl.com, ny.cpandl.com и la.cpandl.com и сопоставить им сайты с именами Seattle-Site, NY-Site и LA-Site. Но если у организации только один офис и поддомены группируются по функциональности, у вас может быть, например, несколько дочерних доменов sales.cpandl.com, support.cpandl.com и tech.cpandl.com и один сайт, скажем, Main-Site.

Составные имена

У каждого объекта Active Directory имеется составное имя (distinguished name, DN). DN уникально идентифицирует объект по его простому имени и местонахождению в пространстве имен. Простое имя (common name) — это имя на английском языке, которое присваивается объекту при его создании.

Простое имя объекта задается как CN=*Имя*, где *Имя* — простое имя объекта, например:

```
CN=William Stanek
```

Простое имя объекта также называется относительным DN объекта (relative DN, RDN). Такое название отражает тот факт, что это часть полного имени объекта, в которое включается еще и местонахождение объекта в Active Directory. Местонахождение объекта определяется по именам объектов-контейнеров и доменов, содержащих объект. OU-контейнеры идентифицируются конструкцией OU=, а компоненты домена — конструкцией DC=. Каждый уровень дерева доменов является отдельным компонентом домена. Рассмотрим пример:

```
OU=Engineering, DC=ny, DC=tech, DC=cpandl, DC=com
```

В данном случае указывается DN для OU Engineering в домене ny.tech.cpandl.com. Компоненты имени отделяются друг от друга запятыми и указываются от самого низкого уровня дерева до самого высокого, т. е. от OU, содержащей собственно объект, и до корневого домена.

DN очень важны, так как они задают точное местонахождение объекта; Active Directory использует DN для поиска, считывания и изменения объектов базы данных. Зная DN объекта, можно выполнить все эти операции.

С каждым из объектов связаны контейнеры и компоненты домена. Обычно контейнером для пользователей, компьютеров, групп и объектов других типов являются OU, но это не всегда так, поскольку в Active Directory есть несколько контейнеров по умолчанию, где также могут храниться объекты. Эти контейнеры идентифицируются по обычному имени (CN=) и включают:

- **Builtin** — контейнер для встроенных групп безопасности;
- **Computers** — контейнер по умолчанию для рядовых серверов (member servers) и рабочих станций, входящих в домен;
- **ForeignSecurityPrincipals** — контейнер для объектов из доверяемого внешнего домена;
- **Users** — контейнер по умолчанию для пользователей.

Примечание Контейнер Domain Controllers создается как OU. Значит, для его идентификации нужно использовать OU=Domain Controllers.

Зная это, можно идентифицировать объект в любом из перечисленных контейнеров. Например, чтобы указать объект в контейнере Users домена tech.cpandl.com, напишите:

```
CN=Users,DC=tech,DC=cpandl,DC=com
```

Если объектом является учетная запись для пользователя William Stanek, полное DN будет иметь вид:

```
CN=William Stanek,CN=Users,DC=tech,DC=cpandl,DC=com
```

Если учетная запись этого пользователя будет впоследствии перемещена в OU Engineering, DN примет вид:

```
CN=William Stanek,OU=Engineering,DC=tech,DC=cpandl,DC=com
```

Средства командной строки, работающие с Active Directory

Освоив базовые структуры Active Directory и научившись идентифицировать используемые объекты по DN, вы готовы управлять Active Directory из командной строки. Применение командной строки дает важное преимущество — дополнительную гибкость. Из командной строки легко выполняются многие операции, осуществить которые GUI-средствами гораздо сложнее или просто невозможно. Например, вы можете найти все учетные записи компьютеров, неактивные более недели, и отключить эти записи. Или одной командой изменить свойства нескольких учетных записей пользователей.

Для работы с Windows-доменами Windows Server 2003 и Windows XP предоставляют набор средств командной строки, управляющих Active Directory. К ним относятся:

- **DSADD** — добавляет объекты в Active Directory;
- **DSGET** — показывает свойства объектов, зарегистрированных в Active Directory;
- **DSMOD** — изменяет свойства объектов, существующих в Active Directory;
- **DSMOVE** — перемещает один объект в новое место в том же домене или переименовывает объект, не перемещая его;
- **DSQUERY** — ищет объекты Active Directory по определенному критерию;
- **DSRM** — удаляет объекты из Active Directory.

Каждая из утилит командной строки предназначена для работы с определенным набором объектов Active Directory (AD). Эти утилиты и объекты, с которыми они работают, перечислены в табл. 11-1.

Табл. 11-1. Утилиты командной строки для работы с Active Directory

Объект	Dsquery	Dsget	Dsadd	Dsmod
Computer (Компьютер)	Да	Да	Да	Да
Contact (Контакт)	Да	Да	Да	Да
Group (Группа)	Да	Да	Да	Да
Partition (Раздел)	Да	Да	Нет	Да
Quota (Квота)	Да	Да	Да	Да
Server (Сервер)	Да	Да	Нет	Да
Site (Сайт)	Да	Да	Нет	Нет
Subnet (Подсеть)	Да	Да	Нет	Нет
User (Пользователь)	Да	Да	Да	Да
OU (OU)	Да	Да	Да	Да

В большинстве случаев при операциях с AD-объектами указывается набор параметров, специфичный для типа объекта, с которым вы работаете, и совпадающее с именем объекта имя подкоманды, используемой для обращения к этим параметрам. Например, если вы хотите добавить компьютер в домен, введите команду DSADD COMPUTER с соответствующими параметрами. А если вам нужно добавить в домен учетную запись пользователя — команду DSADD USER с соответствующими параметрами.

Примечание DSMOVE и DSRM отсутствуют в этой таблице, поскольку они работают с любым объектом каталога. Для перемещения или удаления объекта нужно указать его DN. Кроме того, подставляя в команде DSQUERY звездочку (*) вместо имени типа объекта, вы можете найти все объекты каталога, отвечающие критерию запроса.

Запросы к каталогам командой DSQUERY

Команда DSQUERY позволяет искать в Active Directory объекты, соответствующие определенному набору критериев. Например, вы можете запросить все учетные записи компьютеров, имена которых начинаются с «D», или все отключенные

учетные записи пользователей — DSQUERY возвратит список объектов, отвечающих критерию.

Подкоманды и синтаксис DSQUERY

Запросы к каталогам выполняются по следующему синтаксису подкоманд.

- **DSQUERY COMPUTER** — ищет учетные записи компьютеров, соответствующие критерию:

```
dsquery computer [{НачальныйУзел | forestroot | domainroot}]
[-o {dn | rdn | samid}] [-scope {subtree | onelevel | base}]
[-name Имя] [-desc Описание] [-samid SAM-имя] [-inactive
ЧислоНедель] [-stalepwd ЧислоДней] [-disabled] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r]
[-gc] [-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

- **DSQUERY CONTACT** — ищет контакты, соответствующие критерию:

```
dsquery contact [{НачальныйУзел | forestroot | domainroot}]
[-o {dn | rdn}] [-scope {subtree | onelevel | base}] [-name
Имя] [-desc Описание] [{-s Сервер | -d Домен}] [-u ИмяПоль-
зователя] [-p {Пароль | *}] [-q] [-r] [-gc] [-limit
ЧислоОбъектов] [{-uc | -uco | -uci}]
```

- **DSQUERY GROUP** — ищет учетные записи групп, соответствующие критерию:

```
dsquery group [{НачальныйУзел | forestroot | domainroot}]
[-o {dn | rdn | samid}] [-scope {subtree | onelevel | base}]
[-name Имя] [-desc Описание] [-samid SAM-имя] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r]
[-gc] [-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

- **DSQUERY OU** — ищет организационные единицы, соответствующие критерию:

```
dsquery ou [{НачальныйУзел | forestroot | domainroot}]
[-o {dn | rdn }] [-scope {subtree | onelevel | base}] [-name
Имя] [-desc Описание] [{-s Server | -d Domain}] [-u
ИмяПользователя] [-p {Пароль | *}] [-q] [-r] [-gc] [-limit
ЧислоОбъектов] [{-uc | -uco | -uci}]
```

- **DSQUERY PARTITION** — ищет разделы Active Directory, соответствующие критерию:

```
dsquery partition [-o {dn | rdn}] [-part Фильтр] [{-s Сервер
| -d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q]
[-r] [-limit Число_Объектов] [{-uc | -uco | -uci}]
```

- **DSQUERY QUOTA** — ищет квоты, соответствующие критерию:

```
dsquery quota {domainroot | DNОбъекта } [-o {dn | rdn}] [-
acct Имя] [-qlimit Фильтр] [-desc Описание] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r]
[-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

- **DSQUERY SERVER** — ищет контроллеры домена, соответствующие критерию:

```
dsquery server [-o {dn | rdn}] [-forest] [-domain ИмяДомена]
[-site ИмяСайта] [-name Имя] [-desc Описание] [-hasfsmo
{schema | name | infr | pdc | rid}] [-isgc] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r]
[-gc] [-limit Число_Объектов] [{-uc | -uco | -uci}]
```

- **DSQUERY SITE** — ищет сайты Active Directory, соответствующие критерию:

```
dsquery site [-o {dn | rdn}] [-name Имя] [-desc Описание] [{-s
Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль | *}]
[-q] [-r] [-gc] [-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

- **DSQUERY SUBNET** — ищет объекты подсетей, соответствующие критерию:

```
dsquery subnet [-o {dn | rdn}] [-name Имя] [-desc Описание]
[-loc Местонахождение] [-site ИмяСайта] [{-s Сервер | -d
Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r]
[-gc] [-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

- **DSQUERY USER** — ищет учетные записи пользователей, соответствующие критерию:

```
dsquery user [{НачальныйУзел | forestroot | domainroot}]
[-o {dn | rdn | upn | samid}] [-scope {subtree | onelevel |
base}] [-name Имя] [-desc Описание] [-upn UPN] [-samid SAM-
имя] [-inactive ЧислоНедель] [-stalepwd ЧислоДней] [-disabled]
[{-s Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль |
*}] [-q] [-r] [-gc] [-limit ЧислоОбъектов] [{-uc | -uco |
-uci}]
```

- **DSQUERY *** — ищет объекты Active Directory любого типа, соответствующие критерию:

```
dsquery * [{НачальныйУзел | forestroot | domainroot}]
[-scope {subtree | onelevel | base}] [-filter LDAP-фильтр]
[-attr {СписокАтрибутов | *}] [-attrsonly] [-l] [{-s Сервер
| -d Домен}] [-u ИмяПользователя] [-p {Пароль| *}] [-q] [-r]
[-gc] [-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

На первый взгляд, синтаксис непомерно сложен. Но пусть это не отпугивает вас от DSQUERY. Большинство подкоманд DSQUERY используют один и тот же стандартный синтаксис и содержат лишь несколько расширений стандартного синтаксиса, специфичных для типа объекта, с которым они работают. Самый лучший способ изучить команды DSQUERY — посмотреть их в действии. Давайте так и поступим.

Поиск по именам, описаниям и именам учетных записей в SAM

Какие бы другие параметры вы ни использовали, параметры поиска обязательно должны включать имя, описание или имя учетной записи в SAM, по которым выполняется поиск. Когда вы задаете параметр **–name**, проводится поиск объектов заданного типа, имя которых совпадает со значением параметра. Для поиска частично совпадающих имен подставьте звездочку, например **–name Will***. Простой поиск по имени выполняется командой:

```
dsquery user -name Will*
```

Этот запрос выводит DN любых учетных записей пользователей, отвечающих критерию, например:

```
"CN=William R. Stanek,CN=Users,DC=cpandl,DC=com"
```

Вот и все, что нужно при базовом поиске. Для получения требуемых результатов достаточно указать один параметр.

Примечание Имейте в виду, что в случае пользователей параметр –Name задает имя, которому должно соответствовать содержимое поля Display Name (Полное имя) диалогового окна свойств пользователя. В данном случае отображаемым именем учетной записи является William R. Stanek. Для других типов объектов это значение, отображаемое в поле Name (Имя) на вкладке General (Общие) диалогового окна Properties (Свойства) для данного объекта.

Звездочку можно указать в любой части критерия поиска. Например, если вы знаете фамилию пользователя, а имя — нет, ведите поиск по фамилии:

```
dsquery user -name *Stanek
```

Кроме того, возможен поиск по началу и концу имени:

```
dsquery user -name W*Stanek
```

Если указать параметр **–desc**, запрашиваются объекты заданного типа с подходящим описанием. В качестве знака подстановки служит звездочка, например, описание Engineering Workstation будет соответствовать параметру **–desc Eng***. Рассмотрим пример:

```
dsquery computer -desc Server*
```

При выполнении этого запроса выводятся DN учетных записей компьютеров, описания которых соответствуют запросу, например:

```
"CN=CORPSVR02,OU=Domain Controllers,DC=cpandl,DC=com"
```

Примечание При указании параметра –Desc выполняется поиск по полю Description (Описание), отображаемому в диалоговом окне свойств объекта. В предыдущем примере предполагается, что описание учетной записи компьютера, возвращенной командой, начинается со слова «Server».

Параметр –Samid позволяет искать объекты заданного типа, у которых имя учетной записи в SAM соответствует определенному значению. Для запроса объектов, у которых имя учетной записи в SAM частично совпадает со строкой поиска, можно использовать звездочку, например, имя *wrstanek* соответствует параметру **–samid wr***.

Примечание В диалоговом окне свойств пользователя имя учетной записи в SAM указывается на вкладке Account в поле User Logon Name. В случае компьютеров и групп имя учетной записи в SAM совпадает с именем учетной записи.

Задание при поиске разрешений Logon и Run As

По умолчанию при выполнении команды DSQUERY вы устанавливаете соединение с контроллером домена в вашем домене входа (logon domain). Можно установить соединение с определенным контроллером домена в любом домене леса, указав параметр –S. После параметра –S должно идти DNS-имя сервера, скажем:

```
-s corpdc01.cpandl.com
```

В данном случае вы устанавливаете соединение с контроллером домена corpdc01 в домене cpandl.com.

Примечание С технической точки зрения, необязательно указывать полностью определенное доменное имя (т. е. DNS-имя) сервера. В принципе, достаточно и просто имени сервера. Но это замедляет поиск, так как Active Directory придется вести поиск в DNS (DNS lookup), чтобы получить полное имя, и только потом выполнять запрос, заданный в команде.

Вместо подключения к определенному контроллеру в заданном домене, можно установить соединение с любым доступным контроллером домена. Для этого предназначен параметр –D, после которого указывается DNS-имя домена, например:

```
-d tech.cpandl.com
```

Здесь устанавливается соединение с любым доступным контроллером домена в домене tech.cpandl.com. Учтите, что параметры –S и –D несовместимы. То есть вы можете установить соединение либо с определенным контроллером домена, либо с любым доступным контроллером в данном домене.

Как и в случае многих других типов команд, при необходимости можно пройти аутентификацию, указав имя пользователя и пароль. Для этого предназначены параметры:

```
-u [Домен\]Пользователь [-p Пароль]
```

где *Домен* — необязательное имя домена, в котором находится учетная запись пользователя, *Пользователь* — имя учетной записи пользователя, разрешения которой вы собираетесь задействовать, а *Пароль* — необязательный пароль для этой учетной записи пользователя. Если домен не указан, подразумевается текущий. Если пароль не указан, вам будет предложено ввести его.

Чтобы понять, как используются все эти параметры, рассмотрим несколько примеров.

Соединение с контроллером домена corpsvr02 в домене tech.cpandl.com под учетной записью пользователя WRSTANEK в домене входа CPANDL и поиск учетных записей пользователей, отображаемое имя которых оканчивается на Stanek:

```
dsquery user -name *Stanek -s corpsvr02.tech.cpandl.com -u
cpandl\wrstanek
```

Соединение с любым контроллером домена в домене tech.cpandl.com под учетной записью пользователя Wrstanek в домене входа cpandl и поиск учетных записей пользователей, отображаемое имя которых начинается с Will:

```
dsquery user -name Will* -d tech.cpandl.com -u cpandl\wrstanek
```

Задание начального узла, области поиска и максимального числа объектов

В синтаксисе команды начальный узел обозначается как {*НачальныйУзел* | **forestroot** | **domainroot**} или же используется *DN-Объекта*. Он указывает узел, с которого начинается поиск. Вы можете указать корень леса (**forestroot**), корень домена (**domainroot**) или DN узла (*НачальныйУзел*), например "**CN=Users, DC=cpandl,DC=com**". Если вы ввели **forestroot**, выполняется поиск по глобальному каталогу. По умолчанию используется значение **domainroot**. Это означает, что поиск начинается с контейнера верхнего уровня домена входа для вашей пользовательской учетной записи. Некоторые подкоманды принимают DN объекта, с которым вы собираетесь работать (*DNОбъекта*), например "**CN=William Stanek,CN=Users,DC=cpandl, DC=com**".

Примечание Наверно, вы заметили, что я заключил DN обоих объектов в двойные кавычки. Это разумный подход, поскольку кавычки нужны, когда DN содержит пробел, как в DN второго объекта.

Если вам нужен всеобъемлющий поиск, укажите DN начального узла. Насколько полезна эта возможность, станет ясно, когда вам понадобится получить полный набор объектов. Например, можно получить список всех объектов заданного типа, принадлежащих заданному контейнеру, просто указав начальный узел без параметров –Name, –Desc или –Samid.

Чтобы понять, как используются начальные узлы, рассмотрим несколько примеров.

Получение списка всех учетных записей компьютеров в домене:

```
dsquery computer "DC=cpandl,DC=com"
```

Получение списка всех учетных записей компьютеров в контейнере Computers:

```
dsquery computer "CN=Computers,DC=cpandl,DC=com"
```

Получение списка всех компьютеров в OU «Domain Controllers»:

```
dsquery computer "OU=Domain Controllers,DC=cpandl,DC=com"
```

Получение списка всех пользователей домена:

```
dsquery user "DC=cpandl,DC=com"
```

Получение списка всех пользователей в контейнере Users:

```
dsquery user "CN=Users,DC=cpandl,DC=com"
```

Получение списка всех пользователей в OU «Tech»:

```
dsquery user "OU=Tech,DC=cpandl,DC=com"
```

Можно указать не только начальный узел, но и область поиска. Она обозначается в синтаксисе команды как {-scope **subtree** | **onelevel** | **base**}. По умолчанию используется область поиска **subtree**, т. е. поддерево, корнем которого является начальный узел. В случае **domainroot** областью поиска будет весь домен, а при указании **forestroot** — весь лес. При задании конкретного контейнера областью поиска будут этот контейнер и все его дочерние контейнеры. Например, если задан начальный узел "OU=Tech, DC=cpandl,DC=com", Active Directory будет искать в OU «Tech» и во всех OU, принадлежащих ему.

Значение **onelevel** устанавливает в качестве области поиска заданный начальный узел и его прямые потомки. Так, в случае **domainroot** областью поиска будут домен и его контейнеры, а также OU верхнего уровня. Однако, если какой-либо OU содержит дополнительные (дочерние) OU, поиск по этим OU не выполняется.

В случае **base** областью поиска является единственный объект, задаваемый начальным узлом. Например, выполняется поиск только по заданному OU, но не по его дочерним OU.

Примечание Если задан начальный узел **forestroot**, то в качестве области поиска допускается только **subtree**.

Чтобы понять, как используются области поиска, рассмотрим несколько примеров:

Поиск учетных записей компьютеров в OU «Tech» и в любых OU, принадлежащих ему:

```
dsquery computer "OU=Tech,DC=cpandl,DC=com"
```

Примечание По умолчанию используется область поиска subtree, т. е. автоматически подразумевается **–scope subtree**.

Поиск учетных записей компьютеров только в OU «Tech»:

```
dsquery computer "OU=Tech,DC=cpandl,DC=com" -scope base
```

Поиск учетных записей компьютеров в OU «Tech» и в OU, непосредственно под ней:

```
dsquery computer "OU=Tech,DC=cpandl,DC=com" -scope onelevel
```

Еще один необязательный параметр, который вы, возможно, будете использовать, — параметр –Limit. Он задает максимальное число объектов, возвращаемое в результатах поиска. По умолчанию (если этот параметр не указан) возвращаются первые 100 объектов. Если требуется задать другое количество, укажите его в этом параметре. Например, если вас интересуют только первые 10 результов, введите **–limit 10**. Нулевое значение (**–limit 0**) снимает ограничение и позволяет увидеть все объекты, соответствующие критерию.

Совет В крупных организациях, где могут существовать тысячи объектов, не следует снимать это ограничение. Указывайте конкретное число объектов или просто принимайте значение по умолчанию. Это гарантирует, что ваш запрос не создаст лишней нагрузки на контроллер домена, с которым вы работаете.

Формат вывода имен

При работе с DSQUERY можно задать формат вывода возвращаемых значений имен, а также формат отдельных символов. В синтаксисе команды формат вывода имен определяется параметром **–o**, за которым идет один из следующих элементов: {dn | rdn | upn | samid}. По умолчанию выходным форматом является DN (**–o dn**), например "**CN=William R.Stanek, CN=Users, DC=cpandl, DC=com**". Кроме того, в качестве выходного формата можно указать относительное DN (**–o rdn**), имя пользователя как участника безопасности (user principal name, UPN) (**–o upn**) или имя учетной записи в SAM (**–o samid**).

RDN — простое имя объекта, которое берется из самой низкоуровневой части DN. В случае пользователей RDN — то же, что

Display Name (Полное имя), отображаемое в диалоговом окне свойств пользователя. Для объектов других типов значение RDN показывается в поле Name (Имя) на вкладке General (Общие) диалогового окна свойств объекта. Вот некоторые примеры RDN:

- "William R. Stanek";
- "CORPSVR01";
- "Administrators".

UPN применимы только к учетным записям пользователей. В Active Directory есть поле с этим именем, используемое для входа в систему и аутентификации. В диалоговом окне свойств пользователя имя для входа и домен входа показываются на вкладке Account. Примером UPN является wrstanek@cpandl.com, где wrstanek — имя для входа, а @cpandl.com — информация о домене входа.

Имя учетной записи в SAM применимо к пользователям, компьютерам и группам. Как и в случае UPN, в Active Directory имеется поле с этим именем, и это имя можно посмотреть в диалоговом окне свойств. Для пользователей оно равнозначно имени учетной записи в операционных системах, предшествовавших Windows 2000, и показывается на вкладке Account соответствующего диалогового окна свойств. Для групп имя учетной записи в SAM — это значение в поле Name (Имя) на вкладке General (Общие), а для компьютеров — значение в поле Name на вкладке General со знаком доллара ($) в качестве суффикса.

Примечание Знак доллара ($) является частью имени учетной записи компьютера, но обычно он скрыт и не показывается. Active Directory использует этот знак для поддержки учетных записей пользователя и компьютера с одинаковыми именами. Благодаря этому пользователь JAMESW может работать на компьютере с именем JAMESW, что запрещалось в операционных системах, предшествовавших Windows 2000.

Чтобы лучше понять форматы имен, рассмотрим следующие примеры.

Получение RDN компьютеров, отвечающих критерию поиска:

```
dsquery computer -name corp* -o rdn
```

Получение имен учетных записей в SAM для пользователей в соответствии с критерием поиска:

```
dsquery user -name Wi* -o samid
```

Получение UPN пользователей, отвечающих критерию поиска:

```
dsquery user "OU=Tech,DC=cpandl,DC=com" -o upn
```

Получение DN пользователей, отвечающих критерию поиска:

```
dsquery user "CN=Users,DC=cpandl,DC=com"
```

Примечание По умолчанию используется формат DN, т. е. автоматически подразумевается параметр **-o dn**.

Использование DSQUERY совместно с другими средствами командной строки

Поскольку DSQUERY возвращает DN объектов, соответствующих критерию поиска, можно применять конвейеризацию и указывать возвращаемый DSQUERY набор результатов в качестве входных данных для других утилит командной строки, работающих с Active Directory. Рассмотрим пример, где запрашиваются все учетные записи пользователей с именами, начинающимися с *Willia*:

```
dsquery user -name Willia*
```

Этот запрос возвратит DN учетных записей, соответствующих критерию, скажем:

```
"CN=William R. Stanek,CN=Users,DC=cpandl,DC=com"
```

С помощью символа конвейеризации можно указать, что набор результатов этой команды является вводом команды DSGET USER, и вывести список групп, в которые входит пользователь:

```
dsquery user -name Willia* | dsget user -memberof -expand
```

Эта команда выведет DN групп, в которые входит пользователь, например:

```
"CN=Domain Admins,CN=Users,DC=cpandl,DC=com"
"CN=Enterprise Admins,CN=Users,DC=cpandl,DC=com"
"CN=Administrators,CN=Builtin,DC=cpandl,DC=com"
"CN=Domain Users,CN=Users,DC=cpandl,DC=com"
"CN=Users,CN=Builtin,DC=cpandl,DC=com"
```

Поиск проблемных учетных записей пользователей и компьютеров

У команд DSQUERY USER и DSQUERY COMPUTER несколько синтаксических расширений, предназначенных для поиска проблемных учетных записей. Параметр –Disabled служит для поиска отключенных учетных записей. Например, для поиска отключенных учетных записей во всем домене введите **dsquery user –disabled**.

Эта команда выведет DN отключенных учетных записей пользователей, например:

```
"CN=Guest,CN=Users,DC=cpandl,DC=com"
"CN=SUPPORT_456945a0,CN=Users,DC=cpandl,DC=com"
"CN=krbtgt,CN=Users,DC=cpandl,DC=com"
```

Еще одним очень полезным параметром команды является –Stalepwd. Он позволяет искать учетные записи, у которых пароль не менялся по крайней мере в течение заданного числа дней. Например, можно найти учетные записи всех пользователей, пароль которых не менялся минимум 15 дней, введя команду **dsquery user –stalepwd 15**.

Команда выведет список DN пользователей:

```
"CN=Administrator,CN=Users,DC=cpandl,DC=com"
"CN=Guest,CN=Users,DC=cpandl,DC=com"
"CN=SUPPORT_456945a0,CN=Users,DC=cpandl,DC=com"
"CN=krbtgt,CN=Users,DC=cpandl,DC=com"
"CN=William R. Stanek,CN=Users,DC=cpandl,DC=com"
"CN=Howard Smith,CN=Users,DC=cpandl,DC=com"
```

Примечание Вы можете ввести в действие политику управления паролями, требующую регулярной смены паролей. Об этом см. в главе 9 книги «Microsoft Windows Server 2003. Справочник администратора». Эти политики применяются, только когда пользователи входят в домен. Если пользователь в отпуске или отсутствует по какой-то другой причине, время последней смены пароля окажется слишком давним (обычно пользователь должен сменить свой пароль при следующем входе в систему). Большинство отключенных учетных записей будут присутствовать и в списке записей с устаревшими паролями.

Наконец, вам может потребоваться найти учетные записи компьютеров или пользователей, которые были неактивны по

крайней мере в течение заданного числа недель. Неактивной считается учетная запись, которая не использовалась для входа в течение определенного периода. Например, если нужно найти всех пользователей, которые не входили в домен минимум две недели, наберите команду **dsquery user –inactive 2.**

Обычно пользователи не входят в домен из-за отсутствия в офисе, например находятся в отпуске, болеют или работают вне офиса. Учетные записи компьютеров могут быть неактивными из-за того, что компьютер выключен или не подключен к сети. Например, если пользователь ушел в отпуск, забрал с собой свой ноутбук и во время отпуска не подключается к сети удаленно, то соответствующая учетная запись компьютера будет неактивна в течение этого периода.

Переименование и перемещение объектов

Переименование и перемещение объектов в домене выполняются командой DSMOVE. Почему одной командой, а не двумя? Потому что при переименовании объекта вы, по сути, перемещаете его, изменяя текущее DN на новое. Вспомните, что DN состоит из двух частей — обычного имени или RDN и местонахождения.

Синтаксис DSMOVE имеет следующий вид:

```
dsmove DNОбъекта [-newname НовоеИмя] [-newparent DNРодителя]
[{-s Сервер | -d Домен }] [-u ИмяПользователя]
[-p {Пароль | *}] [-q] [{-uc | -uco | -uci}]
```

Для переименования пользователя, компьютера, группы или другого объекта Active Directory вы должны указать DN объекта, а затем в параметре –Newname сообщить новое относительное имя объекта. Так, для переименования объекта пользователя William Stanek в William R. Stanek выполняется команда **dsmove "CN= William Stanek,OU=Tech,DC=cpandl,DC=Com"–newname "WilliamR. Stanek"**.

Для перемещения пользователя, компьютера, группы или другого объекта Active Directory в пределах домена вы должны указать текущее DN объекта, а затем в параметре –Newparent сообщить новое местонахождение или DN родителя объекта. Допустим, вам нужно переместить учетную запись пользователя из OU «Tech» в OU «Engineering». Для этого укажите DN объекта, например **"CN=William Stanek,OU=Tech,DC=cpandl, DC=com"**, и DN нового местонахождения объекта вроде **"OU= Engineering, DC=cpandl,DC=com"**. Соответствующая команда имеет вид:

```
dsmove "CN=William Stanek,OU=Tech,DC=cpandl,DC=com" -newparent
OU=Engineering,DC=cpandl,DC=com
```

Чтобы переименовать объект и в то же время переместить его, просто добавьте параметр –Newname для присвоения объекту нового имени. Рассмотрим следующий пример:

```
dsmove "CN=William Stanek,OU=Tech,DC=cpandl,DC=com" -newparent
OU=Engineering,DC=cpandl,DC=com -newname "William R. Stanek"
```

Эта команда перемещает учетную запись пользователя William Stanek в OU «Engineering» и переименовывает ее в William R. Stanek.

В каждом из этих примеров можно было бы сначала получить DN объекта командой DSQUERY. Для этого просто передайте вывод DSQUERY команде DSMOVE по механизму конвейеризации, например:

```
dsquery user -name "William Stanek" | dsmove -newname "William
R. Stanek"
```

Здесь DN объекта, "CN=William Stanek,OU=Tech,DC=cpandl, DC=Com", является выводом команды DSQUERY USER и вводом команды DSMOVE. В результате выполняется переименование объекта User.

Совет Вас интересует, как перемещать объекты из одного домена в другой? Командой MOVETREE из Windows Support Tools. Как и в DSMOVE, в этой команде указываются исходные и целевые DN перемещаемых объектов. Кроме того, вы должны установить соединения с соответствующими контроллерами в исходном и целевом доменах.

Удаление объектов из Active Directory

Если вам больше не нужно, чтобы объект хранился в Active Directory, удалите его командой DSRM, синтаксис которой выглядит так:

```
dsrm DNОбъекта ... [-subtree [-exclude]] [-noprompt] [{-s
Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль | *}]
[-c] [-q] [{-uc | -uco | -uci}]
```

Внимание! Не используйте команду DSRM, предварительно не поэкспериментировав с ней в изолированном тестовом домене. Это очень мощная команда. Она удалит любой объект, который вы в ней укажете, в том числе и контейнер.

При использовании DSRM лучше всего указывать конкретный удаляемый объект. В следующем примере учетная запись компьютера engcomp18 удаляется из OU «Eng» в домене cpandl.com:

```
dsrm "CN=engcomp18,OU=Eng,DC=cpandl,DC=com"
```

По умолчанию DSRM предлагает подтвердить удаление:

```
Are you sure you wish to delete
CN=engcomp18,OU=Eng,DC=cpandl,DC=com (Y/N)?
```

Чтобы отключить запрос на подтверждение, укажите ключ *–noprompt*, например:

```
dsrm "CN=engcomp18,OU=Eng,DC=cpandl,DC=com" -noprompt
```

Однако так следует поступать, только когда вы полностью уверены, что DSRM удалит именно тот объект, который вы собираетесь удалить.

DSRM годится как для удаления объектов из контейнеров или OU, так и для удаления самих контейнеров и OU. Если контейнер или OU пустой, его можно удалить, указав его DN, например:

```
dsrm "OU=Eng,DC=cpandl,DC=com"
```

Но если контейнер или OU не пустой, его нельзя удалить таким способом. DSRM сообщит:

```
Failed: The operation cannot be performed because child objects
exist. This operation can only be performed on a leaf object.
```

Чтобы удалить контейнер и все объекты, которые он содержит, укажите параметр –Subtree. Рассмотрим пример:

```
dsrm "OU=Eng,DC=cpandl,DC=com" -subtree
```

Ключ –Subtree указывает, что удаляются все содержащиеся в OU «Eng» объекты (независимо от типа) и сам контейнер. Чтобы удалить все объекты в контейнере, но оставить сам контейнер, используются параметры –Subtree и –Exclude, например:

```
dsrm "OU=Eng,DC=cpandl,DC=com" -subtree -exclude
```

В данной команде параметр –Subtree сообщает, что удаляются все содержащиеся в OU «Eng» объекты (независимо от типа), а параметр –Exclude исключает OU «Eng» из списка удаляемых объектов.

Глава 12

Управление учетными записями компьютеров и контроллерами домена

Основное внимание в этой главе уделяется управлению доменными учетными записями компьютеров, которые контролируют доступ к сети и ее ресурсам. Как и пользовательские учетные записи, доменные учетные записи компьютеров имеют атрибуты, включая имена и членство в группах, которыми вы можете управлять. Вы также можете добавлять учетные записи компьютеров в любой контейнер или организационную единицу (OU) в службе Active Directory. Но лучше использовать Computers (Компьютеры), Domain Controllers (Контроллеры домена) и любые OU, созданные вами. Стандартный инструмент Microsoft Windows для работы с учетными записями компьютеров — оснастка Active Directory Users And Computers (Active Directory - Пользователи и компьютеры). В командной строке доступно множество команд, каждая из которых предназначена для определенных целей. Независимо от того, какая у вас система — Windows XP Professional или Windows Server 2003, вы можете использовать методики, описанные в этой главе, для управления учетными записями компьютеров и контроллеров домена.

Общие сведения об управлении учетными записями компьютеров из командной строки

Для управления учетными записями компьютеров домена есть два набора утилит командной строки. Первый набор может быть использован с любым типом учетной записи компьютера, включая рабочие станции, рядовые серверы (member servers) и контроллеры домена. Второй набор команд предназна-

чен только для управления дополнительными возможностями контроллеров домена.

Помимо команды DSQUERY computer, которая обсуждалась в предыдущей главе, общий набор команд управления учетными записями компьютеров включает в себя следующие команды.

- **DSADD computer** — создает учетную запись компьютера в Active Directory:

```
dsadd computer DNКомпьютера [-samid SAM-имя] [-desc
Описание] [-loc Местонахождение] [-memberof DNГруппы ...]
[{-s Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль |
*}] [-q] [{-uc | -uco | -uci}]
```

- **DSGET computer** — выводит на экран свойства учетной записи компьютера по синтаксису одного из двух видов. Вот синтаксис для просмотра свойств нескольких компьютеров:

```
dsget computer DNКомпьютера ... [-dn] [-samid] [-sid]
[-desc] [-loc] [-disabled] [{-s Сервер | -d Домен}]
[-u ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [-l]
[{-uc | -uco | -uci}] [-part DNРаздела [-qlimit] [-qused]]
```

И синтаксис для просмотра информации о членстве в группах для одного компьютера:

```
dsget computer DNКомпьютера [-memberof [-expand]] [{-s
Сервер | -d Домен }] [-u ИмяПользователя] [-p {Пароль | *}]
[-c] [-q] [-l] [{-uc | -uco | -uci}]
```

- **DSMOD computer** — изменяет атрибуты одной или нескольких учетных записей компьютеров в каталоге:

```
dsmod computer DNКомпьютера ... [-desc Описание] [-loc
Местонахождение] [-disabled {yes | no}] [-reset] [{-s
Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль | *}]
[-c] [-q] [{-uc | -uco | -uci}]
```

Совет DSQUERY можно использовать совместно с любой из команд, работающих с сервером или компьютером; это позволяет указать нужный объект (или объекты). Если вы хотите ввести составное имя (DN) для каждого интересующего вас объекта, то можете так и поступить. Просто отделяйте каждое DN пробелом.

Кроме DSQUERY server, описанной в предыдущей главе, к командам, позволяющим управлять дополнительными свойствами контроллеров домена, относятся следующие.

- **DSGET server** — отображает различные свойства контроллеров домена по синтаксису одного из трех видов. Вот синтаксис для вывода основных свойств выбранного контроллера домена:

```
dsget server DNСервера ... [-dn] [-desc] [-dnsname] [-site]
[-isgc] [{-s Сервер | -d Домен}] [-u ИмяПользователя]
[-p {Пароль | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
```

  Синтаксис для получения списка пользователей, которые владеют наибольшим числом объектов каталога в указанном контроллере домена, выглядит так:

```
dsget server DNСервера ... [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [-l] [{-uc |
-uco | -uci}] [-topobjowner Число_выводимых_объектов]
```

  Наконец, синтаксис для отображения DN разделов каталога на выбранном сервере:

```
dsget server DNСервера ... [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [-l] [{-uc |
-uco | -uci}] [-part]
```

- **DSMOD server** — изменяет свойства контроллера домена:

```
dsmod server DNСервера ... [-desc Описание] [-isgc {yes |
no}] [{-s Сервер | -d Домен}] [-u ИмяПользователя] [-p
{Пароль | *}] [-c] [-q] [{-uc | -uco | -uci}]
```


Примечание Еще одна полезная команда для работы с контроллерами домена и Active Directory — NTDSUTIL. Это командный интерпретатор текстового режима, запустив который вы можете управлять сервисами каталога, используя отдельную командную строку и внутренние команды. Чтобы запустить интерпретатор NTDSUtil, введите **ntdsutil**.

Создание учетных записей компьютеров в доменах Active Directory

Вы можете создать учетную запись компьютера для рабочей станции или сервера, который нужно добавить в домен, при помощи команды DSADD computer. Сделав это, вы заранее

создаете учетную запись данного компьютера так, что она становится доступна при присоединении компьютера к домену. Для создания учетной записи компьютера требуются соответствующие права. Большинство пользователей могут создавать учетную запись компьютера в своем домене входа (logon domain). Однако на этот процесс влияют политики групп и другие ограничения доступа.

Создание учетной записи компьютера

При создании учетной записи компьютера единственной необходимой информацией является составное имя (DN). Как вы помните из предыдущей главы, DN определяет полное имя объекта в Active Directory и включает в себя путь к местонахождению объекта. Поэтому, предоставляя DN для учетной записи, вы определяете имя учетной записи компьютера и контейнер, где эта запись должна быть создана. Например:

```
dsadd computer "CN=CORPSERVER05,OU=Domain
Controllers,DC=cpandl, DC=com"
```


Совет Составное имя указывает место создания учетной записи в иерархии домена. Вы можете создавать учетные записи компьютеров в любом домене леса, для которого у вас есть необходимые права доступа. В некоторых случаях вам придется непосредственно войти на контроллер домена, с которым вы хотите работать. Для подключения к конкретному контроллеру в любом домене леса используйте параметр –S *Сервер*, а для подключения к любому доступному контроллеру домена в конкретном домене — параметр –D *Домен*.

В данном примере вы создаете учетную запись компьютера с именем CORPSERVER05 в контейнере Domain Controllers (Контроллеры домена) службы Active Directory. Если создание учетной записи прошло успешно, команда DSADD computer сообщит:

```
dsadd succeeded:CN=CORPSERVER05,OU=Domain
Controllers,DC=cpandl, DC=com
```

Для запуска операции с привилегиями другого пользователя предназначены параметры –U *ИмяПользователя* и –P *Пароль*.

Однако создание учетных записей не всегда проходит удачно. Наиболее распространенная причина неудач — некорректное составное имя. Так, если бы вы использовали команду:

```
dsadd computer "CN=CORPSERVER05,CN=Domain
Controllers,DC=cpandl, DC=com"
```

DSADD computer сообщила бы:

```
dsadd failed:CN=CORPSERVER05,CN=Domain Controllers,DC=cpandl,
DC=com:Directory object not found.
```

Причина данной ошибки в том, что объект Domain Controllers создан как организационная единица (OU), а не как универсальный контейнер. Так что мы неверно использовали CN=Domain Controllers вместо OU=Domain Controllers.

Другой распространенной причиной неудач является ситуация, когда уже существует некий объект с именем, которое вы пытаетесь присвоить учетной записи. В таком случае выберите другое имя для учетной записи компьютера.

Настройка атрибутов учетных записей компьютеров и членства в группах

Когда вы предоставляете только составное имя, некоторые параметры устанавливаются автоматически. Участие в группах настраивается так, чтобы компьютер был членом группы Domain Computers (Компьютеры домена). Учетная запись в SAM (Security Account Manager) создается из атрибута простого имени (common name), используемого в составном имени компьютера. Обычно команда DSADD computer добавляет знак доллара ($) как суффикс к этому имени. В предыдущем примере простое имя — это CORPSERVER05, значит, имя учетной записи в SAM будет CORPSERVER05$.

Если вы хотите настроить атрибуты учетной записи компьютера при ее создании, то можете сделать это с помощью дополнительных параметров:

- **Samid** — задает имя учетной записи в SAM, которое должно оканчиваться знаком доллара, например **–samid** CORPSERVER05$;
- **Desc** — задает описание добавляемого компьютера, например **–desc "CNMember Server"**;
- **Loc** — указывает текстовое описание физического расположения добавляемого компьютера. Обычно это офис и зда-

ние, где находится компьютер. Так, если компьютер расположен в офисе 110 здания E, вы могли бы написать **–loc "E/110"**.

Вы можете настроить членство в группах для новой учетной записи компьютера, используя параметр –Memberof. Этот параметр принимает список разделенных пробелами DN, представляющий группы, в которые вы хотите включить компьютер. Например, если вы хотите, чтобы новая учетная запись компьютера вошла в группу Engineering, и DN для этой группы – CN=Engineering,OU=Eng,DC=cpandl,DC=com, то можете ввести команду, подобную этой:

```
dsadd computer "CN=CORPSERVER05,OU=Domain
Controllers,DC=cpandl,DC=com" -memberof
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
```

Если вам нужно, чтобы новая учетная запись компьютера была членом групп Engineering и Tech, DN которых CN=Engineering,OU=Eng,DC=cpandl,DC=com и CN=Tech,CN=Users, DC=cpandl,DC=com соответственно, используйте команду наподобие такой:

```
dsadd computer "CN=CORPSERVER05,OU=Domain
Controllers,DC=cpandl,DC=com" -memberof
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Tech,CN=Users,DC=cpandl,DC=com"
```

Примечание Указывать группу Domain Computers (Компьютеры домена) необязательно. Новые учетные записи компьютеров автоматически включаются в эту группу.

Управление свойствами учетных записей компьютеров

Управление учетными записями компьютеров из командной строки несколько отличается от управления ими в оснастке Active Directory Users And Computers (Active Directory - Пользователи и компьютеры) главным образом потому, что у вас появляется больше вариантов действий, особенно когда приходится работать с несколькими учетными записями компьютеров одновременно.

Просмотр и поиск учетных записей компьютеров

Как говорилось в главе 11, команда DSQUERY computer позволяет искать компьютеры. Поиск возможен не только по имени учетной записи в Active Directory или в SAM, но и с применением символов подстановки. Вывод команды DSQUERY computer содержит DN компьютеров, отвечающих критериям поиска. Этот вывод может быть перенаправлен в качестве ввода для других команд, включая DSGET computer, которая позволяет отображать свойства учетной записи компьютера.

Команду DSGET computer лучше всего использовать совместно с DSQUERY computer. Команда DSQUERY computer сообщает DN одного или нескольких компьютеров, а DSGET computer отображает свойства соответствующих учетных записей. Отображаемые свойства задаются следующими параметрами.

- **Dn** — выводит DN компьютера, удовлетворяющего критериям поиска.
- **Samid** — выводит имя учетной записи компьютера в SAM, отвечающей критериям поиска.
- **Sid** — выводит идентификатор защиты для учетной записи компьютера, отвечающей критериям поиска.
- **Desc** — выводит описание учетной записи компьютера, удовлетворяющей критериям поиска.
- **Loc** — выводит атрибут местонахождения учетной записи компьютера, удовлетворяющей критериям поиска.
- **Disabled** — выводит значение Yes/No, которое указывает, отключена ли данная учетная запись компьютера.

DSGET computer выводит результаты в форме таблицы. Вообще говоря, вы всегда будете использовать –Dn, –Samid или –Sid как параметры, чтобы определить конкретные компьютеры в выводе. Например, если вы хотите найти все компьютеры инженеров с именами, начинающимися на engcomp, введите:

```
dsquery computer -name engcomp* | dsget computer -dn -disabled
```

Вот результаты, показывающие DN и состояние отключения:

```
dn disabled
CN=engcomp18,OU=Eng,DC=cpandl,DC=comyes
CN=engcomp19,OU=Eng,DC=cpandl,DC=comyes
```

```
CN=engcomp20,OU=Eng,DC=cpandl,DC=comno
CN=engcomp21,OU=Eng,DC=cpandl,DC=comno
CN=engcomp22,OU=Eng,DC=cpandl,DC=comno
dsget succeeded
```

Вы также можете вывести имя учетной записи в SAM:

```
dsquery computer -name engcomp* | dsget computer -samid -disabled

samid        disabled
ENGCOMP18$   yes
ENGCOMP19$   yes
ENGCOMP20$   no
ENGCOMP21$   no
ENGCOMP22$   no
dsget succeeded
```

Или идентификатор защиты:

```
dsquery computer -name engcomp* | dsget computer -sid -disabled

sid disabled
S-1-5-21-4087030303-3274042965-2323426166-1119    yes
S-1-5-21-4087030303-3274042965-2323426166-1120    yes
S-1-5-21-4087030303-3274042965-2323426166-1122    no
S-1-5-21-4087030303-3274042965-2323426166-1123    no
S-1-5-21-4087030303-3274042965-2323426166-1124    no
dsget succeeded
```

В любом случае у вас есть идентификатор, по которому легче различать элементы в списке учетных записей компьютеров. Вы можете использовать второй синтаксис команды DSGET computer, чтобы получить информацию о членстве компьютеров в группах. Например, если вы хотите увидеть, в какие группы входит компьютер с именем ENGCOMP18, введите:

```
dsquery computer -name engcomp18 | dsget computer -memberof
```

или

```
dsget computer "CN=engcomp18,OU=Eng,DC=cpandl,DC=com" -memberof
```

Обе команды работают одинаково. В первом примере DSQUERY computer сообщает DN учетной записи компьютера, а во втором — вы делаете это сами. В любом случае в выводе будут показаны сведения о членстве в группах, например:

```
"CN=Tech,CN=Users,DC=cpandl,DC=com"
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"
```

Здесь компьютер является членом групп Tech, Engineering и Domain Computers.

Хотя этот прием годится и для получения сведений о членстве в группах нескольких компьютеров, способа вывода DN или имен учетных записей в SAM соответствующих компьютеров нет. Вы получите список с информацией о членстве в группах, и единственный индикатор того, что блоки данных относятся к разным компьютерам, — пустые строки, разделяющие результаты. Например, если вы воспользуетесь запросом:

```
dsquery computer -name engcomp* | dsget computer -memberof
```

вывод может быть таким:

```
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"

"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"

"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"

"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"

"CN=Tech,CN=Users,DC=cpandl,DC=com"
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Domain Computers,CN=Users,DC=cpandl,DC=com"
```

Здесь на экран выведена информация о пяти учетных записях компьютеров (вы можете определить это по разделению блоков записей в списке пустыми строками), но у вас нет сведений о том, к каким учетным записям компьютеров относятся конкретные элементы списка.

Примечание Не упускайте из виду возможность применения команды DSQUERY computer для документирования текущей конфигурации учетных записей компьютеров. Образец соответствующей командной строки выглядит так:

```
dsquery computer "DC=cpandl,DC=com" | dsget computer -dn
-samid -sid -desc -loc -disabled > domaincomputers.txt
```

Здесь перечисляются все учетные записи компьютеров и их свойства в домене cpandl.com, и эта информация сохраняется в файле.

Установка или изменение атрибутов местонахождения и описания

Команда DSMOD computer позволяет легко задать или сменить расположение учетной записи компьютера и описание. Вы можете задать эти атрибуты для 1, 10, 100 или более компьютеров одновременно. Допустим, вы хотите, чтобы все 500 компьютеров в OU «Engineering» содержали в описании «Engineering Computer», а в атрибуте местонахождения — «Engineering Dept.». Это можно сделать одной командной строкой, например:

```
dsquery computer "OU=Engineering,DC=cpandl,DC=com" | dsmod
computer -loc "Engineering Dept." -desc "Engineering Computer"
```

Команда DSMOD computer сообщит об удачном или неудачном изменении для каждого компьютера:

```
dsmod succeeded:CN=Engineeringcomp01,OU=Engineering,
   DC=cpandl,DC=com
dsmod succeeded:CN=Engineeringcomp02,OU=Engineering,
   DC=cpandl,DC=com
dsmod succeeded:CN=Engineeringcomp03,OU=Engineering,
   DC=cpandl,DC=com
...
dsmod succeeded:CN=Engineeringcomp499,OU=Engineering,
   DC=cpandl,DC=com
dsmod succeeded:CN=Engineeringcomp500,OU=Engineering,
   DC=cpandl,DC=com
```

Изменение этих значений через GUI заняло бы у вас часы, а в командной строке весь процесс потребовал лишь нескольких минут. Вы просто набрали командную строку и дали DSMOD computer сделать всю работу за вас.

Отключение и включение учетных записей компьютеров

Вы можете включать или отключать учетные записи компьютеров командой DSMOD computer с параметром –Disabled. Наберите **–disabled yes** для отключения учетной записи компьютера и **–disabled no** для ее включения.

В следующем примере отключаются учетные записи всех компьютеров в OU «TestLab»:

```
dsquery computer "OU=TestLab,DC=cpandl,DC=com" | dsmod
computer -disabled yes
```

Команда DSMOD computer сообщит о каждом удачном или неудачном изменении:

```
dsmod succeeded:CN=TestLabcomp01,OU=TestLab,DC=cpandl,DC=com
dsmod succeeded:CN=TestLabcomp02,OU=TestLab,DC=cpandl,DC=com
dsmod succeeded:CN=TestLabcomp03,OU=TestLab,DC=cpandl,DC=com
```

Восстановление заблокированных учетных записей компьютеров

Как и учетные записи пользователей, учетные записи компьютеров имеют пароли. Однако в отличие от пользовательских учетных записей, пароли учетных записей компьютеров управляются и поддерживаются автоматически. В учетных записях компьютеров два пароля: стандартный, который по умолчанию меняется каждые 30 дней, и закрытый ключ-пароль для установления защищенных соединений с контроллерами домена, который по умолчанию тоже меняется каждые 30 дней.

Оба пароля должны быть синхронизированы. Если синхронизация закрытого ключа-пароля и пароля учетной записи компьютера будет просрочена, компьютеру не удастся войти в домен, а для службы Netlogon в журнале событий появится сообщение об ошибке подключения с идентификатором события 3210 или 5722. Если вы увидите такое сообщение, значит, ваш пароль просрочен и нужно восстановить учетную запись для синхронизации паролей.

Для восстановления рассинхронизированного пароля служит команда DSMOD computer с параметром –Reset. Вот пример:

```
dsmod computer
"CN=Engineeringcomp01,OU=Engineering,DC=cpandl,DC=com" -reset
```

Здесь вы восстанавливаете пароль для компьютера Engineeringcomp01 в OU «Engineering» домена cpandl.com.

Вы можете восстановить сразу все учетные записи в OU «Engineering». Для этого задействуйте команду DSQUERY computer, чтобы получить список всех компьютеров в домене, и DSMOD computer, чтобы восстановить их пароли, например:

```
dsquery computer "OU=Engineering,DC=cpandl,DC=com" | dsmod
computer -reset
```

Примечание Один из способов определить, что пароль учетной записи компьютера просрочен, — воспользоваться командой DSQUERY computer с параметром –Stalepwd. Если вы приняли стандартное значение (30 дней) для паролей учетных записей компьютеров, то найдете просроченные пароли, указав –Stalepwd 30. Например:

```
dsquery computer -stalepwd 30
```

Результат покажет список компьютеров с паролями старше 30 дней, что означает, что пароль просрочен или компьютер в этот период не использовался.

Перемещение учетных записей компьютеров

Учетные записи компьютеров обычно располагаются в Computers (Компьютеры), Domain Controllers (Контроллеры домена) или в дополнительно созданных контейнерах/OU. Вы можете переместить учетную запись компьютера в другой контейнер или OU внутри текущего домена командой DSMOVE. Укажите текущее DN учетной записи компьютера и воспользуйтесь параметром –Newparent, чтобы задать новое местонахождение или DN нового родителя учетной записи компьютера. Если вы хотите переместить учетную запись компьютера CORPSVR03 из OU «Tech» в OU «Engineering», то должны указать DN учетной записи компьютера в виде «CN=CORPSVR03,OU=Tech,DC=cpandl,DC=com» и предоставить DN родителя для нового местонахождения, например «OU=Engineering,DC=cpandl,DC=com». Такая команда будет выглядеть следующим образом:

```
dsmove "CN=CORPSVR03,OU=Tech,DC=cpandl,DC=com" -newparent
"OU=Engineering,DC=cpandl,DC=com"
```

Также можно было бы использовать учетную запись компьютера, полученную командой DSQUERY computer. Чтобы сделать это, нужно просто перенаправить вывод DSQUERY computer на вход команды DSMOVE:

```
dsquery computer -name "CORPSVR03" | dsmove -newparent
"OU=Engineering,DC=cpandl,DC=com"
```

Здесь DN учетной записи компьютера «CN=CORPSVR03, OU=Tech,DC=cpandl,DC=com» получено командой DSQUERY computer и перенаправлено на вход команды DSMOVE.

Этот пример работает независимо от того, является ли данный компьютер рабочей станцией, рядовым сервером или контроллером домена.

Удаление учетных записей компьютеров

Если учетная запись компьютера больше не нужна, ее можно удалить из Active Directory командой DSR. В большинстве случаев вы будете удалять одну конкретную учетную запись компьютера, например Corpserver03. Тогда, чтобы удалить запись, вы передадите команде DSRM составное имя (DN) учетной записи компьютера:

```
dsrm "CN=corpserver03,OU=Eng,DC=cpandl,DC=com"
```

По умолчанию DSRM запросит подтверждение на удаление. Если вы не хотите видеть этот запрос, включите параметр –Noprompt, например:

```
dsrm "CN=corpserver03,OU=Eng,DC=cpandl,DC=com" -noprompt
```

Работа с контроллерами домена

Компьютеры под управлением Windows Server 2003 могут выступать в роли рядового сервера (member server) или контроллера домена (domain controller, DC). Хотя все, что обсуждалось в предыдущих разделах этой главы, применимо к любому типу учетной записи компьютера, содержание данного раздела относится только к контроллерам домена.

Установка и удаление контроллеров домена

Контроллеры домена выполняют массу важных задач в доменах Active Directory. Рядовой сервер можно сделать контроллером домена через команду DCPROMO, которая устанавливает службы каталогов и назначает рядовой сервер на роль контроллера домена. Если вы запустите DCPROMO второй раз, то вернете контроллеру домена роль рядового сервера.

Примечание Команда DCPROMO запускает GUI-утилиту. Однако она принимает некоторые параметры из командной строки вроде /Answer:*ИмяФайла* и /Adv. Параметр /Answer позволяет предоставить имя файла ответов, который описывает установку служб каталогов. Если вы автоматизируете установку всего сервера, добавьте запись GUIRunOnce в файл Unattend.txt для автоматического запуска DCPROMO

по окончании необслуживаемой установки. Параметр /Adv сообщает DCPROMO, что она должна работать в расширенном режиме, — это позволяет создать контроллер домена, восстановив его из резервных копий файлов. Для этого на контроллере домена, работающем под управлением Windows Server 2003 и находящемся в том же домене, что и рядовой сервер, который вы хотите повысить, делается резервное копирование состояния системы. Затем вы должны восстановить файлы состояния системы (информацию о домене) на рядовом сервере.

Поиск контроллеров домена в Active Directory

Если вы хотите работать исключительно с учетными записями контроллеров домена, а не всех компьютеров, используйте команды DSQUERY server и DSGET server. По умолчанию DSQUERY server ведет поиск в вашем домене входа. Фактически, если вы просто наберете в командной строке **dsquery server** и нажмете Enter, то получите список всех контроллеров домена, к которому вы подключены. Если нужно, укажите домен для поиска через параметр –Domain, например:

```
dsquery server -domain tech.cpandl.com
```

Здесь вы получаете список все контроллеров домена tech.cpandl.com. Если вам требуется список всех контроллеров доменов в целом лесу, введите **dsquery server –forest**.

Во всех этих примерах вы получаете список DN контроллеров домена. В отличие от DN, с которыми мы имели дело ранее, эти DN включают сведения о конфигурации сайта, скажем:

```
"CN=CORPSVR02,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com"
```

Эти дополнительные сведения предоставляются командой DSQUERY server и описывают сайт, где находится сервер. Помните, что домены могут охватывать несколько физических мест, и вы сообщаете Active Directory о них через сайты и подсети. В этом примере показывается сайт Default-First-Site-Name в контейнере Sites().

Примечание DSQUERY server поддерживает дополнительные параметры, которые помогут вам искать глобальные каталоги и координаторы операций (operations masters). Данные параметры рассматриваются в разделах «Поиск глобальных каталогов» и «Поиск координаторов операций» далее в этой главе.

Как и в случае команд, ориентированных на операции с рядовыми компьютерами, DSQUERY server и DSGET server лучше использовать совместно. DSQUERY server позволяет получить DN одного или нескольких контроллеров домена, а затем передать вывод в DSGET server для отображения свойств соответствующих учетных записей. Выводимые свойства определяются следующими параметрами.

- **Dn** — выводит DN (составные имена) контроллеров домена, отвечающих критериям поиска.
- **Desc** — выводит описание контроллеров, удовлетворяющих критериям поиска.
- **Dnsname** — выводит полное доменное имя (FQDN) контроллера домена.
- **Isgc** — выводит значение Yes или No, указывающее, является ли контроллер домена еще и сервером глобального каталога.

Например, если вам нужна детальная сводка обо всех контроллерах доменов в лесу, введите команду:

```
dsquery server -forest | dsget server -desc -dnsname -isgc
```

Для сохранения этой информации прямо в файл измените команду так:

```
dsquery server -forest | dsget server -desc -dnsname -isgc >
forest-dcs.txt
```

Серверы глобального каталога

Контроллер домена, назначенный на роль глобального каталога, хранит полную копию всех объектов Active Directory для своего домена и частичную копию для остальных доменов леса. Глобальные каталоги используются в процессах входа и поиска информации. Фактически, если глобальный каталог недоступен, обычные пользователи не смогут войти в домен. Единственный способ изменить такое поведение — кэшировать ин-

формацию об универсальном членстве в группах на локальных контроллерах домена. По умолчанию первый контроллер, установленный в домене, назначается глобальным каталогом. Вы также можете добавить глобальные каталоги в домен для ускорения отклика при подключении и запросах на поиск. Рекомендуется один глобальный каталог на каждый сайт внутри домена.

Любой контроллер домена, обслуживающий глобальный каталог, должен иметь широкополосное соединение с сетью и другими контроллерами домена, выступающими в роли координаторов инфраструктуры (infrastructure masters). Координатор инфраструктуры — одна из пяти ролей координатора операций, которые можно назначить доменному контроллеру. Этот координатор отвечает за обновление ссылок на объекты. Для этого координаторы инфраструктуры сравнивают свои данные с данными глобального каталога. Если координатор инфраструктуры обнаруживает устаревшие данные, он запрашивает обновленные данные из глобального каталога. Затем координатор инфраструктуры копирует изменения на другие контроллеры домена.

Совет Когда в домене только один контроллер, вы можете назначить роли координатора инфраструктуры и глобального каталога одному и тому же контроллеру домена. Однако, когда в домене два или более контроллера домена, глобальный каталог и координатор инфраструктуры не должны работать на одном контроллере, так как это может повлиять на способность координатора инфраструктуры распознавать устаревшие данные в каталоге.

Поиск серверов глобального каталога

Хотите узнать, где находится глобальный каталог? Для вашего текущего (входного) домена просто наберите **dsquery server –isgc**. В результате вы получите список DN серверов глобального каталога, например:

```
"CN=CORPSVR02,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com"
```

Команда DSQUERY server также годится для поиска глобальных каталогов в конкретном домене. Чтобы сделать это, используйте параметр –Domain:

```
dsquery server -domain tech.cpandl.com -isgc
```

Здесь вы ищете серверы глобального каталога в домене tech.cpandl.com. Для поиска серверов во всем лесу наберите:

```
dsquery server -forest -isgc
```

Вы также можете искать серверы глобального каталога по сайту, но для этого нужно знать полное имя сайта и нельзя использовать символы подстановки. Например, чтобы найти все серверы глобального каталога для сайта Default-First-Site-Name, введите **dsquery server –site Default-First-Site-Name**.

Примечание «Посайтовый» поиск важен потому, что обычно у вас будет минимум один сервер глобального каталога на сайт. Если вы проверили сайт и не нашли глобальный каталог, подумайте: может, стоит его добавить?

Добавление или удаление глобального каталога

Вы можете назначить контроллер домена на роль глобального каталога командой DSMOD server. Укажите DN нужного сервера и параметр **–isgc yes**, чтобы этот сервер обслуживал глобальный каталог, например:

```
dsmod server "CN=corpdc05,OU=Eng,DC=cpandl,DC=com" -isgc yes
```

Другой способ решения той же задачи — применить DSQUERY server для получения списка серверов, с которыми вы хотите работать. Допустим, в домене tech.cpandl.com три контроллера домена, и вы хотите, чтобы все они стали серверами глобального каталога. Тогда введите примерно такую командную строку:

```
dsquery server -domain tech.cpandl.com | dsmod server -isgc yes
```

Здесь вы используете DSQUERY server для получения DN всех контроллеров в домене tech.cpandl.com и передачи этой информации на вход команды DSMOD server, которая сделает каждый контроллер домена сервером глобального каталога.

Если вы в дальнейшем захотите, чтобы сервер прекратил выступать в роли глобального каталога, наберите **–isgc no**. В следующем примере отменяется обслуживание глобального каталога сервером corpdc04 в домене tech.cpandl.com:

```
dsmod server "CN=corpdc04,OU=Tech,DC=cpandl,DC=com" -isgc no
```

Проверка параметров кэширования и настроек глобального каталога

В зависимости от конфигурации сети вам доступны различные уровни функциональности доменов и леса. Если все контроллеры домена в вашем домене или лесу работают под управлением хотя бы Windows 2000 Server и уровень функциональности установлен как режим Windows 2000 Native, ваша организация сможет задействовать многие дополнительные средства Active Directory, но использовать первичные (primary domain controllers, PDC) и резервные (backup domain controllers, BDC) контроллеры домена, работающие под управлением Windows NT, уже не удастся. Одна из особенностей этого режима — возможность кэширования информации о членстве в универсальных группах.

Если при попытке входа пользователя нет доступа к глобальному каталогу, кэширование информации о членстве в универсальных группах все же позволяет регистрироваться обычным пользователям. Кэширование включается или отключается для каждого сайта индивидуально, и вы можете определить, включено ли оно, через команду DSGET site. Для этого укажите DN нужного сайта и параметр –Cachegroups, как показано в следующем примере:

```
dsget site "CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com" -cachegroups
```

Если кэширование включено, вывод будет таким:

```
cachegroups
yes
dsget succeeded
```

В ином случае вы получите:

```
cachegroups
no
dsget succeeded
```

Другой способ такой проверки — применение команды DSQUERY site. Если вы просто наберете **dsquery site**, команда вернет список всех сайтов в лесу. Чтобы ограничить набор результатов, используйте параметр –Name и укажите простое имя сайта — либо полностью, либо с символами подстановки, например:

```
dsquery site -name *First*
```

Здесь вы ищете любой сайт, в простом имени которого есть комбинация букв «First».

Кроме того, вы можете воспользоваться конвейеризацией этих команд, чтобы выяснить настройки кэширования на всех сайтах леса:

```
dsquery site | dsget site -cachegroups
```

Вы получите список «yes» и «no» примерно в таком виде:

```
cachegroups
yes
yes
no
no
yes
dsget succeeded
```

Чтобы получить более осмысленную информацию, добавьте параметр –Dn для отображения DN соответствующих сайтов:

```
dn                                                          cachegroups
CN=Seattle-Site-
Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com                  yes
CN=LA-Site-Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com      yes
CN=NY-Site-Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com      yes
CN=Chicago-Site-
Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com                 yes
CN=Detroit-Site-
Name,CN=Sites,CN=Configuration,DC=cpandl,DC=com                 yes
dsget succeeded
```

Если кэширование информации о членстве в универсальных группах включено, в домене со множеством глобальных каталогов для каждого сайта может быть основной глобальный каталог (preferred global catalog), который используется для обновления кэша с информацией о членстве в универсальных группах для соответствующих контроллеров домена. Определить основной глобальный каталог позволяет параметр –Prefgcsite. Например, можно набрать **dsquery site | dsget site –cachegroups –prefgcsite**, чтобы получить полную конфигурацию кэширования для всех глобальных каталогов в лесу. Если основные глобальные каталоги настроены, вы увидите значения «yes» или «no», а если такие каталоги не сконфигурированы — строку «Not Configured».

Назначение ролей координаторам операций

В Active Directory определено пять ролей для координаторов операций (operations masters), каждая из которых крайне важна в обеспечении сетевых операций. Некоторые роли могут быть назначены только один раз в доменном лесу, а другие — по одному разу в каждом домене.

В пределах всего леса обязательно должны быть назначены роли schema master и domain naming master. Координатор схемы (schema master) контролирует обновления и изменения в схеме каталогов, а координатор именования доменов (domain naming master) — добавление или удаление доменов в лесу. Эти роли должны быть уникальны для леса, поэтому вы можете назначить только один координатор схемы и один координатор именования доменов во всем лесу.

В качестве доменных ролей могут быть назначены relative ID master, PDC emulator master и infrastructure master. Как следует из их названий, координатор относительных идентификаторов (relative ID master) выделяет относительные идентификаторы контроллерам домена. Всякий раз, когда вы создаете объект пользователя, группы или компьютера, контроллер домена присваивает объекту уникальный идентификатор защиты. Этот идентификатор состоит из доменного префикса идентификатора защиты и уникального относительного идентификатора, который назначается координатором относительных идентификаторов. Координатор эмулятора PDC (PDC emulator master) работает как PDC под управлением Windows NT, если сеть использует смешанный или промежуточный режим операций. Он аутентифицирует подключения Windows NT, обрабатывает смену паролей и реплицирует обновления на BDC (резервные контроллеры домена). Координатор инфраструктуры (infrastructure master) обновляет ссылки на объекты, сравнивая данные из своего каталога с данными глобального каталога. Если данные устарели, координатор инфраструктуры запрашивает обновленные данные из глобального каталога и копирует изменения на другие контроллеры домена. Эти роли должны быть уникальны в пределах домена. Поэтому вы можете назначить лишь по одному координатору относительных идентификаторов, координатору эмулятора PDC и координатора инфраструктуры в каждом домене.

Поиск координаторов операций

Когда вы устанавливаете новую сеть, первый контроллер домена в первом домене получает все роли координатора операций. Если потом вы создадите новый дочерний или корневой домен в новом дереве, первый контроллер в новом домене также будет автоматически назначен координатором операций. В новом доменном лесу контроллер домена имеет все роли координатора. Если в том же лесу появляется новый домен, контроллеру назначаются роли relative ID master, PDC emulator master и infrastructure master. Роли schema master и domain naming master остаются в первом домене этого леса. При необходимости администратор может передавать роли координаторов операций другим контроллерам.

Чтобы определить, какие роли назначены контроллерам домена в лесу или в домене, используйте параметр –Hasfsmo команды DSQUERY server. Допустимы следующие значения этого параметра:

- **schema** – возвращает DN координатора схемы леса;
- **name** – возвращает DN координатора именования доменов в лесу;
- **infr** – возвращает DN координатора инфраструктуры домена. Если домен не указан через параметр –Domain, будет использован текущий;
- **pdc** – возвращает DN координатора эмулятора PDC домена. Если домен не указан через параметр –Domain, будет использован текущий;
- **rid** – возвращает DN координатора относительных идентификаторов домена. Если домен не указан через параметр –Domain, будет использован текущий.

Координаторы схемы и именования доменов — роли, действующие в пределах леса. Введя **dsquery server –hasfsmo schema** или **dsquery server –hasfsmo name**, вы всегда получаете DN координаторов операций в лесу Active Directory.

Координаторы инфраструктуры, эмулятора PDC и относительных идентификаторов — роли, действующие в пределах домена. Введя **dsquery server –hasfsmo infr**, **dsquery server –hasfsmo pdc** или **dsquery server –hasfsmo rid**, вы всегда получаете DN соответствующего координатора операций в своем домене входа. Если вы хотите получить DN координатора

операций в другом домене, используйте параметр –Domain, например:

```
dsquery server -hasfsmo rid -domain tech.cpandl.com
```

Здесь вы получаете DN координатора относительных идентификаторов в домене tech.cpandl.com. Если в лесу несколько доменов, вы можете получить и список всех контроллеров каждого домена, которым назначены определенные роли. Для этого предназначен параметр –Forest:

```
dsquery server -hasfsmo rid -forest
```

Настройка ролей координаторов операций из командной строки

Хотя команды служб каталогов позволяют проверять местонахождение координаторов операций, они не годятся для конфигурирования ролей координаторов операций. Для этого вы должны воспользоваться NTDSUtil. NTDSUtil — это командный интерпретатор текстового режима, запустив который можно управлять службами каталогов из отдельной командной строки. Вы можете запустить интерпретатор NTDSUtil, набрав в командном окне **ntdsutil** и нажав Enter.

NTDSUtil позволяет передавать роли координаторов операций от одного контроллера домена другому и присваивать роли, когда их нельзя передать обычным образом. Например, на контроллере домена, выступающем в роли координатора инфраструктуры, может выйти из строя жесткий диск, что приведет к останову всего сервера. Если вы не можете вновь запустить сервер, вам, вероятно, понадобится передать роль координатора инфраструктуры другому контроллеру домена. Но никогда не делайте этого, если вы планируете со временем вновь задействовать аварийный сервер. Однажды перехватив роль, вы полностью выведете старый сервер из рабочей схемы, и единственным способом вернуть ему исходную роль будет форматирование загрузочного диска и переустановка Windows Server 2003.

Чтобы передать роль с помощью командной строки, выполните следующие действия.

1. Войдите на сервер, которому вы хотите назначить новую роль координатора операций, и запустите командную оболочку.

2. В командной строке введите **ntdsutil** для запуска NTDSUtil.
3. В командной строке *ntdsutil* наберите **roles**. Это переведет утилиту в режим Operations Master Maintenance, и приглашение командной строки сменится на:

```
fsmo maintenance:
```

4. В командной строке *fsmo maintenance* введите **connections** для появления приглашения командной строки в виде *server connections*. Теперь наберите **connect to server** и добавьте полное доменное имя текущего координатора схемы, например:

```
connect to server corpdc01.eng.cpandl.com
```

5. В случае удачного соединения введите **quit**, чтобы выйти из *server connections* в *fsmo maintenance*, затем введите **transfer** и укажите идентификатор передаваемой роли. Существуют идентификаторы:
   - *pdc* — координатор эмулятора PDC;
   - *rid master* — координатор относительных идентификаторов;
   - *infrastructure master* — координатор инфраструктуры;
   - *schema master* — координатор схемы;
   - *domain naming master* — координатор именования доменов.
6. Роль передана. Введите **quit** в *fsmo maintenance* и наберите **quit** в *ntdsutil*.

Если вам не удается обычным образом передать роль из-за того, что сервер — владелец роли выключен или недоступен по другим причинам, можно перехватить роль иначе.

1. Убедитесь, что контроллер домена, в данный момент владеющий ролью, отключен окончательно. Если сервер может быть восстановлен, не начинайте эту процедуру, пока вы не будете готовы полностью переустановить данный сервер.
2. Войдите на сервер, которому вы хотите назначить новую роль координатора операций и запустите командную оболочку.
3. В командной строке наберите **ntdsutil** для запуска командного интерпретатора NTDSUtil текстового режима.

4. В командной строке *ntdsutil* введите **roles**. Это переведет утилиту в режим Operations Master Maintenance, и приглашение командной строки сменится на:

```
fsmo maintenance:
```

5. В командной строке *fsmo maintenance* введите **connections**, чтобы появилось приглашение командной строки *server connections*. Затем наберите **connect to server** и добавьте полное доменное имя текущего координатора схемы, например:

```
connect to server corpdc01.eng.cpandl.com
```

6. В случае удачного соединения наберите **quit**, чтобы выйти из *server connections* в *fsmo maintenance*, введите **seize** и укажите идентификатор перехватываемой роли. Существуют идентификаторы:
   - *pdc* — координатор эмулятора PDC;
   - *rid master* — координатор относительных идентификаторов;
   - *infrastructure master* — координатор инфраструктуры;
   - *schema master* — координатор схемы;
   - *domain naming master* — координатор именования доменов.
7. Роль перехвачена. Введите **quit** в *fsmo maintenance* и наберите **quit** в *ntdsutil*.

Глава 13

Управление пользователями и группами Active Directory

Главная задача администратора — создание учетных записей пользователей и групп и управление ими. В этой главе вы научитесь создавать учетные записи пользователей и управлять ими из командной строки. Затем вы увидите, как делать то же самое применительно к группам. Основное внимание в данной главе уделяется работе с пользователями и группами в Active Directory.

Обзор управления учетными записями пользователей из командной строки

В Microsoft Windows Server 2003 определено два типа учетных записей пользователей.

- **Доменные учетные записи пользователей** — это учетные записи пользователей, определенные в Active Directory и способные получать доступ ко всем ресурсам домена. Вы можете создавать такие учетные записи и управлять ими, используя команды служб каталогов.
- **Локальные учетные записи пользователей** — это учетные записи пользователей, определенные на локальном компьютере и требующие аутентификации, прежде чем они получат доступ к сетевым ресурсам. Вы можете создавать такие учетные записи и управлять ими с помощью команд сетевых сервисов.

Примечание Локальные учетные записи компьютера в основном используются в конфигурациях с рабочими группами, а не в доменах Windows. Однако до сих пор каждый компьютер в сети имеет одну или более локальных учетных записей. Единственное исключение — контроллеры домена, у которых нет локальных учетных записей. Чтобы работать с локальными учетными записями компьютеров, используйте команды сетевых сервисов.

Ниже перечислены команды служб каталогов, применяемые для управления доменными учетными записями пользователей.

- **DSADD USER** — создает учетную запись пользователя в Active Directory. Ее синтаксис:

```
dsadd user DNПользователя [-samid SAM-имя] [-upn UPN] [-fn
Имя] [-mi Отчество] [-ln Фамилия] [-display Отображаемое
Имя] [-empid ИдентификаторСотрудника] [-pwd {Пароль | *}]
[-desc Описание] [-memberof Группа ...] [-office Офис]
[-tel Телефон] [-email E-mail] [-hometel ДомашнийТелефон]
[-pager Пейджер] [-mobile МобильныйТелефон] [-fax Факс]
[-iptel IP-телефон] [-webpg Web-страница] [-title
Должность] [-dept Отдел] [-company Компания] [-mgr Менед-
жер] [-hmdir ОсновнойКаталог] [-hmdrv БуквaДиска:]
[-profile Путь] [-loscr Путь] [-mustchpwd {yes | no}]
[-canchpwd {yes | no}] [-reversiblepwd {yes | no}]
[-pwdneverexpires {yes | no}] [-acctexpires ЧислоДней]
[-disabled {yes | no}] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-q] [{-uc | -uco | -uci}]
```

- **DSGET USER** — выводит свойства учетных записей пользователей по синтаксису одного из двух видов. Для просмотра свойств множества учетных записей пользователей предназначен синтаксис:

```
dsget user DNПользователя ... [-dn] [-samid] [-sid] [-upn]
[-fn] [-mi] [-ln] [-display] [-empid] [-desc] [-office]
[-tel] [-email] [-hometel] [-pager] [-mobile] [-fax]
[-iptel] [-webpg] [-title] [-dept] [-company] [-mgr]
[-hmdir] [-hmdrv] [-profile] [-loscr] [-mustchpwd]
[-canchpwd] [-pwdneverexpires] [-disabled] [-acctexpires]
[-reversiblepwd] [{-uc | -uco | -uci}] [-part DNРаздела
[-qlimit] [-qused]] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [-l]
```

Синтаксис для просмотра информации о членстве пользователей в группах:

```
dsget user DNПользователя [-memberof [-expand]] [{-s Сервер
| -d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-c]
[-q] [-l] [{-uc | -uco | -uci}]
```

- **DSMOD USER** — изменяет атрибуты одной или более учетных записей в каталоге:

```
dsmod user DNПользователя ... [-upn UPN] [-fn Имя] [-mi
Отчество] [-ln Фамилия] [-display ОтображаемоеИмя] [-empid
ИдентификаторСотрудника] [-pwd {Пароль | *}] [-desc
Описание] [-office Офис] [-tel Телефон] [-email E-mail]
[-hometel ДомашнийТелефон] [-pager Пейджер] [-mobile
МобильныйТелефон] [-fax Факс] [-iptel IP-телефон] [-webpg
Web-страница] [-title Должность] [-dept Отдел] [-company
Компания] [-mgr Менеджер] [-hmdir ОсновнойКаталог] [-hmdrv
БуквaДиска:] [-profile Путь] [-loscr Путь] [-mustchpwd {yes
| no}] [-canchpwd {yes | no}] [-reversiblepwd {yes | no}]
[-pwdneverexpires {yes | no}] [-acctexpires ЧислоДней]
[-disabled {yes | no}] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [{-uc | -uco
| -uci}]
```

Совет Эти команды принимают в качестве ввода результаты работы команды DSQUERY USER, что позволяет указывать составные имена (DN) нужных учетных записей пользователей. DN каждого пользователя можно вводить и самостоятельно. Помните, что каждое DN отделяется пробелом.

На первый взгляд команды управления пользователями выглядят крайне сложными. Но на самом деле они не столько сложны, сколько многофункциональны. Они позволяют добавлять, просматривать или изменять учетные записи пользователей и включают обширный набор свойств учетных записей, с которыми вы можете работать. Параметры для работы с конкретными свойствами совпадают с используемыми для добавления, просмотра или изменения учетной записи. Например, создавая учетную запись, вы можете указать телефонный номер офиса пользователя через параметр –Tel. А чтобы выяснить этот телефон, воспользуйтесь параметром –Tel команды DSGET USER. Если вам нужно изменить телефонный номер, вы опять же применяете параметр –Tel, но на этот раз в команде DSMOD USER.

Для управления локальными учетными записями служит команда NET USER — одна из команд сетевых сервисов. В частности, NET USER поддерживает несколько видов синтаксиса, и вы используете тот из них, который подходит для вашей ситуации. Рассмотрим несколько примеров.

Вывод локальных учетных записей пользователей:

```
net user [ИмяПользователя [Пароль | *] [/active:{no | yes}]
[/comment: "Описание"] [/countrycode: NNN] [/expires:{{ММ/ДД/
ГГГГ | ДД/ММ/ГГГГ | ммм,дд,ГГГГ} | never}] [/fullname:"Имя"]
[/homedir:Путь] [/passwordchg:{yes | no}] [/passwordreq:{yes |
no}] [/profilepath:[Путь]] [/scriptpath:Путь] [/times:{День
[-День] [,День[-День]] ,Время [-Время] [,Время[-Время]] [;...]
| all}] [/usercomment:"Текст"] [/workstations:{ИмяКомпьютера
[,...] | *}]
```

Создание или изменение локальных учетных записей пользователей:

```
net user [ИмяПользователя {Пароль | *} /add [/active:{no |
yes}] [/comment:"Описание"] [/countrycode:NNN] [/expires:
{{ММ/ДД/ГГГГ | ДД/ММ/ГГГГ | ммм,дд,ГГГГ} | never}] [/
fullname:"Имя"] [/homedir:Путь] [/passwordchg:{yes | no}]
[/passwordreq:{yes | no}] [/profilepath:[Путь]] [/
scriptpath:Путь] [/times:{День[-День] [,День[-День]] ,Время
[-Время] [,Время[-Время]] [;...] | all}] [/sercomment:"Текст"]
[/workstations:{ИмяКомпьютера [,...] | *}]]
```

Удаление локальных учетных записей пользователей:

```
net user ИмяПользователя /delete
```

Как видите, NET USER позволяет работать с довольно ограниченным набором свойств учетных записей пользователей. Эти свойства больше подходят для управления локальными учетными записями.

Примечание Команда NET USER также позволяет работать с доменными учетными записями в вашем домене входа. Но в отличие от команд служб каталогов, которые дают возможность создавать и управлять доменными учетными записями пользователей в любом домене леса Active Directory, вы не получаете доступа за пределы текущего домена.

Добавление учетных записей пользователей

У каждого пользователя, которому нужен доступ к ресурсам сети, должна быть учетная запись. Ее тип зависит от конфигурации сети. В случае доменов Active Directory вы имеете дело с доменными учетными записями пользователей, а в слу-

чае рабочих групп — с локальными, которые относятся к конкретным машинам.

Создание доменных учетных записей пользователей

Создавая доменную учетную запись пользователя, вы передаете команде DSADD USER составное имя (DN) пользователя. Простое имя (common name) является частью составного. Остальная часть составного имени указывает, где в Active Directory должна располагаться учетная запись; для этого нужно указать контейнер, в котором следует создать запись, и соответствующий домен. Например, вы можете создать учетную запись для Mary Baker в OU «Sales» домена cpandl.com командой: **dsadd user "CN=Mary Baker,OU=Sales,DC=cpandl, DC=com"**. Это приведет к созданию учетной записи с именем Mary Baker, используемым для входа, но поскольку другие свойства не указаны, она будет автоматически отключена из соображений безопасности.

Имена пользователей не чувствительны к регистру букв и могут быть длиной до 64 символов. Обычно в дополнение к DN учетной записи пользователя задаются следующие свойства.

- Имя, указываемое параметром –Fn.
- Отчество, указываемое параметром –Mi.
- Фамилия, указываемая параметром –Ln.
- Отображаемое имя, указываемое параметром –Display.

Примечание В большинстве случаев отображаемому имени должно быть присвоено то же значение, что и простому имени учетной записи. Так вам будет легче управлять учетной записью, поскольку, зная отображаемое имя, вы знаете и простое имя (как часть составного).

- Имя учетной записи в SAM (также называемое логином), указываемое параметром –Samid.
- Пароль задается параметром –Pwd. Он должен соответствовать требованиям к паролям, установленным политиками групп (если таковые есть).

Первые 20 символов простого имени используются при задании имени учетной записи в SAM; они также называются логином в операционных системах до Windows 2000. Имя учетной записи в SAM должно быть уникальным в рамках домена, и в случае совпадений вам придется сделать его отличным от

отображаемого. Тогда-то вы и зададите имя учетной записи в SAM через параметр –Samid.

В отличие от учетных записей, созданных в оснастке Active Directory Users And Computers (Active Directory – Пользователи и компьютеры), значения полей для имени пользователя, отчества и фамилии не применяются для формирования отображаемого имени. Вы должны задать его через параметр –Display. Отображаемым считается имя, которое Windows показывает в диалогах. Простое имя, являющееся частью имени учетной записи пользователя и частью доменного имени в DN, служит для формирования полного логина (полного входного имени). Такое имя предназначено для входа и аутентификации. Например, если входной домен пользователя — cpandl, а логин — marybaker, то полное входное имя пользователя — cpandl\marybaker.

Для создания учетной записи для Mary A. Baker, которая использует такие параметры, можно ввести команду вида:

```
dsadd user "CN=Mary Baker,OU=Sales,DC=cpandl,DC=com" -fn Mary
-mi A -ln Baker -samid "marybaker" -display "Mary Baker" -pwd
dg56$2#
```

Примечание Обратите внимание на двойные кавычки в этом примере. Если в значении параметра есть пробелы, вы должны заключить его в двойные кавычки. Советую всегда пользоваться двойными кавычками при задании DN, имени учетной записи в SAM и отображаемого имени. Тогда команды будут выполнены и в тех случаях, если эти значения содержат пробелы. Иначе вы забудете о кавычках, и создать учетную запись не удастся.

Если возникают проблемы при создании учетных записей, вы увидите предупреждение, и вам понадобится проверить синтаксис и убедиться, что все значения заданы правильно и составное имя верно. Если все в порядке, команда DSADD USER вернет DSADD SUCCEEDED.

Примечание Что больше всего запутывает при создании учетных записей пользователей или групп из командной строки, — так это слишком большое количество компонентов имен. Давайте немного проясним этот вопрос. Простое имя (common name) учетной записи, также называемое *относительным составным именем* (relative distinguished name), —

это имя, присваиваемое в первой части (CN=) составного имени, например CN=Mary Baker. Учетные записи пользователей также имеют отображаемое имя. Обычно отображаемым (display name) является полное имя пользователя, и вы, наверное, часто встречаете упоминания как раз о полном имени пользователя, а не о его отображаемом имени. Для учетных записей пользователей и групп также предусмотрены имена, применяемые в операционных системах до Windows 2000 (pre-Windows 2000 names). В случае пользователей такое имя служит для входа в домен и аутентификации, поэтому его также часто называют логином для операционных систем, предшествовавших Windows 2000 (pre-Windows 2000 logon name).

Настройка атрибутов доменных учетных записей пользователей и участия в группах

Все новые пользователи домена становятся членами группы Domain Users (Пользователи домена); это их основная группа. Вы можете указывать членство в дополнительных группах через параметр –Memberof (к нему нужно добавить DN группы). Если DN группы содержит пробелы, оно должно быть заключено в кавычки, например:

```
dsadd user "CN=Mary Baker,OU=Sales,DC=cpandl,DC=com" -memberof
"CN=Backup Operators,CN=Builtin,DC=cpandl,DC=com" "CN=DHCP
Administrators,CN=Builtin,DC=cpandl,DC=com"
```

Примечание Обратите особое внимание на пробел между DN групп. Если вы не поставите такой пробел, членство в группах будет сконфигурировано неправильно и возникнет ошибка.

Здесь создается учетная запись пользователя, а затем добавляется в группы Backup Operators и DHCP Administrators. Это двухэтапный процесс: сначала создается учетная запись, а затем конфигурируется ее участие в группах. Если происходит ошибка при включении в группы, DSADD USER сообщит, что объект создан, но после его создания возникла ошибка. Проверьте синтаксис задания DN группы, потом воспользуйтесь командой DSMOD USER для правильной настройки членства пользователя в группах.

Из соображений безопасности при создании учетной записи пользователя вам также могут понадобиться следующие параметры.

- **–Mustchpwd {yes | no}** — по умолчанию пользователю не надо менять свой пароль при первом входе, т. е. подразумевается параметр **–mustchpwd no**. Если вы укажете **–mustchpwd yes**, пользователю придется сменить свой пароль при первом входе.
- **–Canchpwd {yes | no}** — по умолчанию пользователь может сменить свой пароль, т. е. подразумевается параметр **–canchpwd yes**. При **–canchpwd no** пользователь не сможет сменить свой пароль.
- **–Pwdneverexpires {yes | no}** — по умолчанию предполагается **–pwdneverexpires no**, и пароль пользователя устаревает в соответствии с настройками политик групп. Если же вы установите **–pwdneverexpires yes**, пароль для этой учетной записи никогда не устареет.

Примечание Параметр –pwdneverexpires yes переопределяет соответствующий параметр в доменной политике учетных записей. Никогда не устаревающий пароль — это, как правило, довольно плохо. Такой подход не отвечает одной из главных целей, для которых предназначены пароли.

- **–Disabled {yes | no}** — по умолчанию, как только вы создаете учетную запись с паролем, она становится доступна для использования (подразумевается значение **–disabled no**). Если вы укажете **–disabled yes**, учетная запись отключается. Это временно запретит использование данной учетной записи.

Рассмотрим несколько примеров, чтобы лучше понять команду DSADD USER.

Создание учетной записи для Scott L. Bishop в контейнере Users домена cpandl.com и обязательная смена пароля при первом входе в систему:

```
dsadd user "CN=Scott L. Bishop,CN=Users,DC=cpandl,DC=com" -fn
Scott -mi L -ln Bishop -samid "scottb" -display "Scott L.
Bishop" -pwd acornTree -mustchpwd yes
```

Создание учетной записи для Bob Gage в OU «Engineering» домена ny.cpandl.com с заданием пароля, срок действия которого никогда не истекает, но учетная запись отключается:

```
dsadd user "CN=Bob
Gage,OU=Engingeering,DC=ny,DC=cpandl,DC=com" -fn Bob -ln Gage
```

```
-samid "bgage" -display "Bob Gage" -pwd dazedOne
-pwdneverexpires yes -disabled
```

Создание учетной записи для Eric F. Lang в OU «Marketing» домена cpandl.com с заданием пароля, который нельзя сменить:

```
dsadd user "CN=Eric F. Lang,OU=Marketing,DC=cpandl,DC=com" -fn
Eric -mi F -ln Lang -samid "eflang" -display "Eric F. Lang"
-pwd albErt -canchpwd no
```

Совет Вы можете создавать учетные записи в любом домене леса, для доступа к которому у вас есть соответствующие права. В некоторых случаях может понадобиться вход непосредственно на контроллер нужного домена. Для подключения к конкретному контроллеру в любом домене леса укажите параметр –S *Сервер*, а для подключения к любому доступному контроллеру в указанном домене — параметр –D *Домен*.

Параметры, рассмотренные в этом разделе, относятся к числу наиболее часто используемых при создании учетных записей. Но, как вы видели по синтаксису команды DSADD USER, есть масса других параметров для учетных записей пользователей. Как задавать соответствующие свойства, вы узнаете далее в этой главе.

Примечание Сотрудники, которые обращаются к Windows Server 2003 через службы для Macintosh, используют первичные группы. Когда пользователь Macintosh создает файлы или каталоги в системе Windows Server 2003, этим файлам или каталогам назначаются первичные группы. По умолчанию все учетные записи пользователей имеют в качестве первичной группу Domain Users (Пользователи домена). Сменить первичную группу можно в оснастке Active Directory Users And Computers (Active Directory – Пользователи и компьютеры). Но сделать это из командной строки нельзя.

Создание локальных учетных записей пользователей

Локальные учетные записи пользователей создаются на индивидуальных компьютерах. Если вы хотите создать локальную учетную запись на конкретном компьютере, вы должны подключиться локально или удаленно для получения доступа к

локальной командной строке. Войдя в нужную систему, вы можете создать необходимую учетную запись командой NET USER. Иногда политики локальных компьютеров позволяют создать учетную запись просто по ее имени и с параметром /Add, например:

```
net user wrstanek /add
```


Примечание Создать локальную учетную запись пользователя на контроллере домена нельзя. Контроллеры домена не поддерживают локальные учетные записи.

Сейчас вы создали локальную учетную запись с именем **wrstanek** и пустым паролем. Хотя пустые пароли допустимы, они создают риск для безопасности компьютера и, возможно, сети. Поэтому, советую указывать имя пользователя и пароль для новых локальных учетных записей. Пароль должен следовать за именем учетной записи:

```
net user wrstanek dg56$2# /add
```

Здесь вы создаете локальную учетную запись для **wrstanek** и устанавливаете пароль **dg56$2#**.

Если создание учетной записи прошло удачно, NET USER вернет «Command Completed Successfully» («Команда выполнена успешно»). Однако, если возникли проблемы при создании учетной записи, NET USER не сообщит об ошибке. Вместо этого появится подсказка по синтаксису команды. В таком случае проверьте синтаксис своей команды и убедитесь, что все значения заданы верно.

Вот список других значений и параметров, которые могут понадобиться при работе с локальными учетными записями пользователей:

- **/comment:"*Описание*"** — задает описание учетной записи пользователя. Обычно указывается должность или отдел;
- **/fullname:"*Имя*"** — задает полное имя пользовательской учетной записи. Полное имя также является отображаемым именем;
- **/passwordchg {yes | no}** — по умолчанию пользователи могут менять свои пароли, т. е. подразумевается параметр **/password yes**. Если вы укажете **/passwordchg no**, пользователи не смогут менять свои пароли;

- **/passwordreq {yes | no}** — по умолчанию в учетных записях должны быть пароли. Это подразумевается параметром **/passwordreq yes**, т. е. пароль не может быть пустым;
- **/active {yes | no}** — по умолчанию учетные записи пользователей включаются при создании, что подразумевает параметр **/active yes**. Если вы укажете **/active no**, учетная запись будет отключена. Используйте этот параметр для временного запрета использования данной учетной записи.

Рассмотрим несколько примеров, чтобы лучше понять применение команды NET USER:

Создание локальной учетной записи для группы Desktop Support с полным именем и описанием:

```
net user dsupport squ5 /fullname:"Desktop Support"
/comment:"Desktop Support Account" /add
```

Создание локальной учетной записи для Phil Spencer с полным именем, описанием и обязательным паролем:

```
net user pspencer magma2 /fullname:"Phil Spencer"
/comment:"Offsite Sales Manager" /passwordreq yes /add
```

Создание локальной учетной записи для Chris Preston с полным именем и описанием. Пароль задается, но его смена пользователем не разрешается:

```
net user chrisp apples /fullname:"Chris Preston" /comment:
"PR Manager" /passwordchg no /add
```

Управление учетными записями пользователей

Управление учетными записями из командной строки отличается от управления ими в оснастке Active Directory Users And Computers (Active Directory – Пользователи и компьютеры) в основном тем, что у вас появляется больше возможностей и удобства в операциях над несколькими учетными записями одновременно.

Просмотр и поиск учетных записей пользователей

Для поиска пользователей служит команда DSQUERY USER. Вы можете искать не только по простому имени, имени учетной записи в SAM и описанию, но и указывать символы под-

становки в любом из этих полей. Вывод DSQUERY USER содержит DN пользователей, отвечающих критериям поиска, и этот вывод можно перенаправить на вход другим командам, в том числе DSGET USER для отображения свойств учетных записей пользователей.

Команды DSQUERY USER и DSGET USER лучше применять совместно. Так, вы можете задействовать DSQUERY USER для получения DN для одного или более пользователей, а затем с помощью DSGET USER вывести на экран свойства соответствующих учетных записей. DSGET USER позволяет показывать свойства в соответствии с параметрами:

- **–display** — выводит полные имена найденных учетных записей;
- **–desc** — выводит описания найденных учетных записей;
- **–dn** — выводит DN найденных учетных записей;
- **–empid** — выводит идентификаторы сотрудников из найденных учетных записей;
- **–fn** — выводит имена пользователей из найденных учетных записей;
- **–mi** — выводит отчества из найденных учетных записей;
- **–samid** — выводит имена учетных записей в SAM для найденных учетных записей;
- **–sid** — выводит идентификаторы защиты из найденных учетных записей;
- **–disabled** — выводит значение Yes/No (Да/Нет), указывающее, отключена ли данная учетная запись.

DSGET USER показывает результаты в виде таблицы. В общем случае вы всегда будете указывать параметр –Dn, –Samid или –Display, чтобы идентифицировать пользователей в выводе команды. Например, если вы хотите найти всех пользователей — инженеров, учетные записи которых были отключены, введите командную строку наподобие:

```
dsquery user "OU=Eng,DC=cpandl,DC=com" | dsget user -dn
-disabled
```

Здесь вы выводите список отключенных учетных записей пользователей в OU «Engineering» домена cpandl.com:

```
dn                                              disabled
CN=edwardh,OU=Eng,DC=cpandl,DC=com              yes
CN=jacobl,OU=Eng,DC=cpandl,DC=com               yes
CN=maryk,OU=Eng,DC=cpandl,DC=com                yes
CN=ellene,OU=Eng,DC=cpandl,DC=com               yes
CN=williams,OU=Eng,DC=cpandl,DC=com             yes
dsget succeeded
```

Вы также могли бы вывести имена учетных записей в SAM, как показано в следующем примере:

```
dsquery user -name william* | dsget user -samid -disabled
  samid                                                 disabled
  williamb                                              yes
  williamd                                              yes
  williams                                              no
dsget succeeded
```

Вы находите все учетные записи, простые имена которых начинаются с William, затем выводите имена учетных записей в SAM и состояние каждой записи (включена или отключена).

Определение членства в группах для индивидуальных учетных записей пользователей

Для получения информации о членстве отдельных пользователей в группах предназначен второй синтаксис команды DSGET USER. Например, если вы хотите увидеть, членом каких групп является учетная запись WilliamS, введите команду:

```
dsquery user -name williams | dsget user -memberof
```

или

```
dsget user "CN=William Stanek,OU=Eng,DC=cpandl,DC=com" -memberof
```

Обе команды работают одинаково. В первом примере вы получаете DN учетной записи пользователя с помощью DSQUERY USER. Во втором — прямо указываете DN. В любом случае вы получите информацию о членстве данной учетной записи в группах, например:

```
"CN=Tech,CN=Users,DC=cpandl,DC=com"
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Domain Users,CN=Users,DC=cpandl,DC=com"
```

Здесь пользователь является членом групп Tech, Engineering и Domain Users.

Хотя эта методика годится для вывода информации о членстве в группах нескольких пользователей, способа отобразить DN или имена учетных записей в SAM нет. То есть вы получаете список групп, и единственное, что указывает на наличие разных пользователей, — пустые строки в соответствующих местах списка. Так, если вы введете запрос:

```
dsquery user -name bill* | dsget user -memberof
```

результат может выглядеть следующим образом:

```
"CN=Tech,CN=Users,DC=cpandl,DC=com"
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Domain Users,CN=Users,DC=cpandl,DC=com"

"CN=Domain Users,CN=Users,DC=cpandl,DC=com"

"CN=Tech,CN=Users,DC=cpandl,DC=com"
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Domain Users,CN=Users,DC=cpandl,DC=com"

"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Domain Users,CN=Users,DC=cpandl,DC=com"

"CN=Tech,CN=Users,DC=cpandl,DC=com"
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Domain Users,CN=Users,DC=cpandl,DC=com"

"CN=Domain Users,CN=Users,DC=cpandl,DC=com"

"CN=Domain Users,CN=Users,DC=cpandl,DC=com"
```

Здесь вы видите вывод для семи учетных записей пользователей. Это можно определить по пустым строкам, отделяющим каждый набор строк в списке. Но никаких указаний на то, к каким учетным записям эти строки относятся, нет.

Настройка атрибутов учетных записей пользователей

Команда DSMOD USER позволяет легко задать или изменить атрибуты учетных записей пользователей из командной строки. Фактически вы можете задавать атрибуты для одного или нескольких пользователей одновременно. Допустим, вы хотите, чтобы у всех пользователей в OU «Sales» в атрибуте отдела было задано значение «Sales & Marketing», в атрибуте компании — «City Power and Light», а в атрибуте должности — «Customer Sales». Все это можно сделать одной командной строкой:

```
dsquery user "OU=Sales,DC=cpandl,DC=com" | dsmod user -dept
"Sales & Marketing" -company "City Power and Light" -title
"Customer Sales"
```

Команда DSMOD USER должна вернуть информацию об удачном или неудачном изменении для каждой учетной записи:

```
dsmod succeeded:CN=edwardh,OU=Sales,DC=cpandl,DC=com    no
dsmod succeeded:CN=erinp,OU=Sales,DC=cpandl,DC=com      no
dsmod succeeded:CN=jayo,OU=Sales,DC=cpandl,DC=com       no
dsmod succeeded:CN=johng,OU=Sales,DC=cpandl,DC=com      yes
...
dsmod succeeded:CN=williams,OU=Sales,DC=cpandl,DC=com   yes
```

Изменение этих параметров через GUI заняло бы часы, а весь процесс из командной строки потребует лишь нескольких минут. Вы просто набираете команду, и DSMOD USER делает работу за вас.

Другие параметры, которые могут пригодиться:

- **–webpg** – задает интранет- или Интернет-адрес, который появится в списке каталогов для данного пользователя, например \\Intranet\Sales;
- **–profile** – устанавливает путь к пользовательскому профилю, который определяет переменные окружения для учетных записей, например \\Gamma\Profiles\wrstanek;
- **–hmdrv** – задает букву диска, с которой будет сопоставлен основной каталог пользователя, например X:. Основной каталог подключается как сетевой диск с этот буквой;
- **–hmdir** – задает основной каталог пользователя, например \\Gamma\Users\wrstanek.

Внимание! Не рекомендуется изменять пути к профилям пользователей, основным дискам или каталогам, когда пользователи вошли в свои системы, так как это может вызвать проблемы. Лучше изменить эту информацию позже или попросить пользователя выйти из системы на несколько минут, а затем вновь войти.

Совет По умолчанию, если в ходе изменений произойдет ошибка, DSMOD USER прекратит работу и сообщит об ошибке. Как правило, такое поведение наиболее эффективно, поскольку вы не захотите внесения неправильных изменений. Но через параметр –C можно указать команде DSMOD USER, чтобы она, сообщив об ошибке, продолжала работу.

Эти параметры принимают особое значение *$username$*. Оно позволяет назначать пути и имена файлов на основе имен индивидуальных пользователей. Например, если вы хотите задать путь к основному каталогу как \\Gama\Users\$username$\ или C:\Home\$username$, Windows заменит значение *$username$* реальным именем пользователя и сделает это для каждого пользователя, которым вы управляете. То есть, если вы работаете с учетными записями erinb, sandyr, miked и kyler, всем им будут присвоены уникальные основные каталоги: либо \\Gama\Users\erinb, \\Gama\Users\sandyr, \\Gama\Users\miked и \\Gama\Users\kyler, либо C:\Home\erinb, C:\Home\sandyr, C:\Home\miked и C:\Home\kyler. Здесь \\Gama\Users — путь к общему сетевому каталогу, а C:\Home представляет каталог на компьютере пользователя.

В соответствии с этим вы могли бы задать Web-страницу, профиль, основные диск и каталог для всех пользователей OU «Sales», введя:

```
dsquery user "OU=Sales,DC=cpandl,DC=com" | dsmod user -webpg
\\Intranet\Sales\$username$ -profile
"\\corpdc02\sales\$username$" -hmdrv "X:" -hmdir
"\\corpserver01\users\$username$"
```

Примечание В оснастке Active Directory Users And Computers (Active Directory – Пользователи и компьютеры), чтобы получить путь и имя файла на основе имени пользователя, вы вводите значение *%username%*. Не используйте его со специальными параметрами, описываемыми здесь. Дело в том, что *%username%* — это переменная окружения, и GUI известно, чем заменить переменную окружения для каждого из пользователей. Однако командная строка интерпретирует эту и другие переменные окружения на основе текущего зарегистрированного пользователя. В последнем случае переменная *%username%* содержит имя учетной записи в SAM для той учетной записи пользователя, под которой вы вызываете команду.

Включение и отключение учетных записей

Вы можете включить или отключить учетную запись пользователя командой DSMOD USER с параметром –Disabled. Укажите **–disabled yes** для отключения учетной записи пользователя или **–disabled no** для ее включения.

В следующем примере вы отключаете всех пользователей в OU «OffsiteUsers»:

```
dsquery user "OU=OffsiteUsers,DC=cpandl,DC=com" | dsmod user
-disabled yes
```

Команда DSMOD USER сообщит об удачном или неудачном отключении для каждой учетной записи.

Восстановление просроченных учетных записей

Доменные учетные записи пользователей могут иметь конкретную дату окончания действия. Вы можете проверить эту дату командой DSGET USER с параметром –Acctexpires. Например, если вы хотите проверить дату окончания действия всех учетных записей пользователей в OU «Sales», введите:

```
dsquery user "OU=Sales,DC=cpandl,DC=com" | dsget user -dn
-acctexpires
```

В результате будут показаны даты окончания действия каждой учетной записи в OU «Sales» и соответствующие DN учетных записей, например:

```
dn                                                acctexpires
  CN=Mary Baker,OU=Sales,DC=cpandl,DC=com         never
  CN=Bradley Beck,OU=Sales,DC=cpandl,DC=com       11/15/2006
  CN=Ann Bebbe,OU=Sales,DC=cpandl,DC=com          never
  CN=Max Benson,OU=Sales,DC=cpandl,DC=com         12/31/2006
dsget succeeded
```

Здесь учетные записи без ограничения срока действия имеют значение «never», а для остальных учетных записей указана конкретная дата, например 11/15/2006.

Если вам нужно изменить дату окончания срока действия учетной записи, вы можете сделать это командой DSMOD USER. Задайте параметр –Acctexpires с указанием числа дней, в течение которых учетная запись должна быть доступна. Например, если учетная запись должна быть доступна в течение следующих 60 дней, введите **–acctexpires 60**, например:

```
dsquery user -name johnw | dsmod user -acctexpires 60
```

или

```
dsmod user "CN=John Wood,OU=Sales,DC=cpandl,DC=com"
-acctexpires 60
```

Здесь вы изменяете срок действия для учетной записи John Wood.

Если вы хотите удалить дату окончания срока действия учетной записи, используйте 0 в качестве значения, чтобы указать, что данная учетная запись никогда не устаревает, например:

```
dsquery user -name johnw | dsmod user -acctexpires 0
```

Примечание Чтобы задать уже прошедшую дату окончания срока действия, введите отрицательное значение, например **–acctexpires -1**.

Управление и восстановление паролей пользователей

Команда DSGET USER позволяет проверить параметры паролей в учетных записях пользователей. Обычно вам нужно знать, есть ли у пользователя право менять свой пароль, истекает ли срок его действия и применяется ли для пароля обратимое шифрование. Чтобы проверить эти настройки, используйте соответственно параметры –Canchpwd, –Pwdneverexpires и –Reversiblepwd. Также вам может понадобиться узнать, должен ли пользователь сменить свой пароль при следующем входе. Для этого укажите параметр –Mustchpwd. Например, если вы хотите проверить эти значения для всех учетных записей пользователей в контейнере Users, введите:

```
dsquery user "CN=Users,DC=cpandl,DC=com" | dsget user -samid
-canchpwd -pwdneverexpires -reversiblepwd -mustchpwd
```

В результате будут выведены параметры пароля для каждой учетной записи в контейнере Users и имена их учетных записей в SAM, например:

```
samid      mustchpwd  canchpwd  reversiblepwd  pwdneverexpires
  andya    no         yes       no             no
  billg    no         yes       no             no
  bobh     yes        yes       no             no
  brianw   no         yes       no             no
  conniej  no         yes       yes            yes
dsget succeeded
```

DSMOD USER предоставляет несколько параметров для управления этими и другими настройками пароля. Вы можете задать пароль для конкретной учетной записи через пара-

метр –Pwd, а затем настроить, как этот пароль должен использоваться в дальнейшем:

- укажите **–mustchpwd yes**, чтобы потребовать смены пароля при следующем входе;
- укажите **–canchpwd no**, чтобы пользователи не могли сменить пароль для своих учетных записей;
- укажите **–pwdneverexpires no**, чтобы у пароля не было ограничений по сроку действия. Этот параметр переопределяет соответствующие параметры в групповых политиках.

Самое удобное в командной строке — возможность управления паролями множества пользователей так же легко, как и паролем одного пользователя. Например, если бы вы решили сменить пароль для каждого пользователя в OU «TempEmployee» на Time2ChangeMe и потребовать его смены при следующем входе, то могли бы сделать это командой вроде:

```
dsquery user "OU=TempEmployee,DC=cpandl,DC=com" | dsmod user
-pwd Time2ChangeMe -mustchpwd yes
```

Перемещение учетных записей пользователей

Обычно учетные записи пользователей размещаются в контейнере Users или в OU. Вы можете переместить учетную запись в другой контейнер или OU внутри се текущего домена командой DSMOVE. Укажите текущее DN учетной записи пользователя и задайте параметр –Newparent, чтобы определить новое размещение или DN родителя для учетной записи пользователя. Например, если бы вы захотели переместить учетную запись пользователя William Stanek из OU «Tech» в OU «Engineering», то могли бы указать DN учетной записи как "**CN=William Stanek,OU=Tech,DC=cpandl,DC=com**" и предоставить DN родителя для нового расположения как "**OU=Engineering,DC=cpandl,DC=com**". Такая команда выглядела бы примерно так:

```
dsmove "CN=William Stanek,OU=Tech,DC=cpandl,DC=com" -newparent
"OU=Engineering,DC=cpandl,DC=com"
```

Вы также могли бы получить DN учетной записи командой DSQUERY USER. Для этого просто перенаправьте вывод DSQUERY USER в DSMOVE, как показано ниже:

```
dsquery user -name "William Stanek" | dsmove -newparent
"OU=Engineering,DC=cpandl,DC=com"
```

Здесь DN учетной записи, "CN=William Stanek,OU=Tech, DC=cpandl,DC=com", получено командой DSQUERY USER и передано на вход команды DSMOVE.

Переименование учетных записей пользователей

Хотя переименовывать учетные записи пользователей довольно легко, не стоит делать этого бездумно. Переименовывая учетную запись, вы присваиваете ей новое простое имя. Это может понадобиться в случае свадьбы, развода или принятия новой фамилии пользователем. Например, если Nancy Anderson (nancya) выходит замуж, она может сменить свое имя пользователя на Nancy Buchanan (nancyb). Когда вы переименуете ее учетную запись, соответствующие привилегии и разрешения перейдут к этой записи. Поэтому файл, который был доступен nancya, теперь доступен nancyb (а у nancya такого доступа больше нет).

Для переименования учетных записей пользователей служит команда DSMOVE. Укажите DN учетной записи пользователя и через параметр –Newname задайте новое простое имя. Вы можете изменить имя пользователя с Nancy Anderson на Nancy Buchanan, введя:

```
dsmove "CN=Nancy Anderson,OU=Marketing,DC=cpandl,DC=com"
-newname "Nancy Buchanan"
```

Вы также можете получить DN пользователя командой DSQUERY USER. Обратите внимание на следующий пример:

```
dsquery user -name N*Anderson | dsmove -newname "Nancy
Buchanan"
```

Здесь вы используете DSQUERY USER для поиска учетной записи, которая начинается с буквы «N» и заканчивается на «Anderson». Затем команда DSMOVE переименовывает найденную учетную запись.

Переименование не изменяет другие свойства учетной записи. Так как некоторые свойства могут содержать старую фамилию, вам придется обновить их с помощью команды DSMOD USER, чтобы отразить изменение имени. В вашем распоряжении следующие параметры.

- **–Ln** — позволяет изменить фамилию в учетной записи пользователя;
- **–Display** — позволяет изменить отображаемое имя учетной записи;

- **–Samid** — позволяет изменить имя учетной записи в SAM;
- **–Profile** — позволяет изменить путь к профилю учетной записи. Впоследствии вам понадобится переименовать соответствующий каталог на диске;
- **–Loscr** — если вы используете отдельные сценарии регистрации для каждого пользователя, параметр –Loscr позволяет изменить имя сценария. Впоследствии не забудьте переименовать сценарий на диске;
- **–Hmdir** — позволяет изменить путь к основному каталогу. Впоследствии не забудьте переименовать соответствующий каталог на диске.

Примечание В большинстве случаев не рекомендуется изменять эту информацию, пока пользователь находится в системе, так как иначе могут возникнуть проблемы. Лучше изменить эту информацию позже или попросить пользователя выйти из системы на несколько минут, а затем вновь войти.

Обратите внимание на следующий пример:

```
dsquery user -name N*Buchanan | dsmod -samid nancyb -ln
Buchanan -display Nancy Buchanan
```

Здесь вы изменяете имя учетной записи в SAM, фамилию и отображаемое имя, чтобы они совпадали с предыдущим изменением имени для учетной записи Nancy Buchanan.

Примечание Имена пользователей упрощают управление и работу с учетными записями. Но Windows Server 2003 на самом деле использует идентификаторы защиты (SID) для определения, отслеживания и управления учетными записями независимо от имени пользователя. SID — это уникальные идентификаторы, генерируемые при создании учетной записи. Благодаря тому, что SID сопоставляются с именами пользователей на внутреннем уровне, вам не приходится изменять привилегии или разрешения в переименованных учетных записях.

Удаление учетных записей пользователей

Если учетная запись пользователя больше не нужна, ее можно удалить из Active Directory командой DSRM. В большинстве случаев нужно удалить лишь определенную учетную за-

пись, например учетную запись для Mary Baker. В таком случае вы указываете команде DSRM составное имя (DN) учетной записи пользователя, например:

```
dsrm "CN=Mary Baker,OU=Sales,DC=cpandl,DC=com"
```

По умолчанию DSRM требует подтверждать удаление. Если этот запрос вам не нужен, используйте параметр –Noprompt, например:

```
dsrm "CN=Mary Baker,OU=Sales,DC=cpandl,DC=com" -noprompt
```

Примечание Даже когда вы удаляете учетную запись, Windows Server 2003 не удаляет профиль пользователя, а также персональные файлы или основной каталог. Если вы хотите удалить эти файлы и каталоги, вам придется сделать это вручную. Если такая операция выполняется вами регулярно, лучше создать сценарий, который будет выполнять необходимые задачи за вас. Учтите, что сначала вам может понадобиться создать резервные копии файлов или данных, поскольку они, вполне вероятно, пригодятся в будущем.

Обзор управления учетными записями групп из командной строки

Учетные записи групп помогают управлять привилегиями множества пользователей. В Windows Server 2003 существует три типа групп.

- **Группы безопасности (security groups)** Группы, с которыми сопоставлены дескрипторы защиты; упрощают управление правами доступа. Вы можете создавать такие группы и управлять ими с помощью команд служб каталогов.
- **Группы распространения (distribution groups)** Группы, используемые как списки для рассылки электронной почты; с ними не сопоставлены дескрипторы защиты. Вы можете создавать такие группы и управлять ими с помощью команд служб каталогов.
- **Локальные группы (local groups)** Группы, используемые только на локальном компьютере. Вы можете создавать такие группы и управлять или с помощью команд сетевых сервисов.

Группы безопасности и распространения используются в доменах, что делает их доступными в любом месте каталога.

Локальные группы, напротив, доступны только на компьютере, где они были созданы. Общие команды для управления учетными записями групп в доменах рассматриваются ниже.

- **DSADD GROUP** — создает учетную запись группы в Active Directory. Синтаксис выглядит так:

```
dsadd group DNГруппы [-secgrp {yes | no}] [-scope {l | g |
u}] [-samid SAM-имя] [-desc Описание] [-memberof Группа
...] [-members Участник ...] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-q] [{-uc | -uco | -uci}]
```

- **DSGET GROUP** — отображает свойства учетных записей групп по синтаксису одного из двух видов. Синтаксис для просмотра свойств нескольких групп:

```
dsget group DNГруппы ... [-dn] [-samid] [-sid] [-desc]
[-secgrp] [-scope] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [-l] [{-uc |
-uco | -uci}] [-part DNРаздела [-qlimit] [-qused]]
```

  Синтаксис для просмотра информации о членстве в группах для отдельной группы:

```
dsget group DNГруппы [{-memberof | -members} [-expand]]
[{-s Server | -d Домен}] [-u ИмяПользователя] [-p {Пароль |
*}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
```

- **DSMOD GROUP** — изменяет атрибуты одной или нескольких учетных записей групп в каталоге. Синтаксис выглядит так:

```
dsmod group DNГруппы ... [-samid SAM-имя] [-desc Описание]
[-secgrp {yes | no}] [-scope {l | g | u}] [{-addmbr |
-rmmbr | -chmbr} DNУчастника ...] [{-s Сервер | -d Домен}]
[-u ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [{-uc |
-uco | -uci}]
```

Совет Для задания DN учетных записей нужных групп либо используйте вывод DSQUERY GROUP, либо вводите их самостоятельно. В последнем случае отделяйте каждое DN пробелом.

Для управления учетными записями локальных групп служит команда NET LOCALGROUP. Она поддерживает несколько видов синтаксиса. Применяемый синтаксис зависит от того, что вы хотите сделать, например:

- **Создание учетных записей локальной группы:** net localgroup [*ИмяГруппы* {/add [/comment:"*Текст*"]}
- **Модификация учетных записей локальной группы:** net localgroup [*ИмяГруппы Имя* [...] {/add | /delete}]
- **Удаление учетных записей группы:** net localgroup [*ИмяГруппы* {/delete [/comment:"*Текст*"]}

Примечание Команда NET LOCALGROUP позволяет добавить локальную группу к группе в текущем домене (домене входа). Такое может понадобиться лишь в очень редких случаях, но, как правило, необходимости выдавать таким способом права на доступ обычным пользователям нет. Например, если вы создаете группу DevTesters, то можете добавить ее к группе Developers домена. Это даст локальным пользователям компьютера, которые являются членами группы DevTesters, те же привилегии, что и другим членам группы Developers. В ней разработчикам, тестирующим конфигурации локальных систем, нужен доступ к домену.

Добавление учетных записей групп

Тип нужной вам группы зависит от конфигурации сети. В доменах вы обычно будете работать с группами безопасности и распространения, а в рабочих группах — в основном с локальными группами, которые относятся только к конкретным компьютерам.

Создание групп безопасности и распространения

Как я уже говорил, группы безопасности предназначены для управления правами доступа групп пользователей, а группы распространения служат как списки рассылки. Независимо от типа создаваемой группы принцип работы с ней зависит от области (scope), которая определяет границы действия этой группы. Существуют следующие области.

- **Локальные группы домена.** Предназначены для выдачи разрешений в рамках одного домена. Их членами могут быть только учетные записи (как компьютеров, так и пользователей) и группы из домена, в котором они определены.
- **Глобальные группы.** Предназначены для задания прав доступа к объектам в любом домене в дереве доменов или лесу. Их членами могут быть только учетные записи и группы из домена, в котором они определены.

- **Универсальные группы.** Предназначены для выдачи разрешений в пределах всего дерева доменов или леса. Их членами могут быть учетные записи, глобальные группы и другие универсальные группы из любого домена в дереве доменов или лесу.

Примечание Универсальные группы безопасности доступны, только если Active Directory работает в «родном» режиме Windows 2000 или Windows Server 2003; кроме того, они удобнее в больших сетях. Это вызвано в основном тем, что они вводят еще один уровень в иерархию групп, т. е. их преимущества очевиднее в крупных системах, где нужен больший контроль над группами.

Возможности групп с учетом областей и режима работы суммированы в табл. 13-1. Как показано в таблице, оба фактора влияют на то, что вы можете и чего не можете делать с группами.

Табл. 13-1. Возможности групп с учетом функционального уровня и области действия

Возможности группы	Локальная область домена	Глобальная область	Универсальная область
«Родной» функциональный уровень Windows Server 2003 и Windows 2000	Членами группы могут быть учетные записи пользователей, локальные и глобальные группы из того же домена, а также универсальные группы из любого домена	Членами группы могут быть только учетные записи пользователей и глобальные группы из того же домена	Членами группы могут быть учетные записи пользователей и группы из любого домена независимо от области
Смешанный функциональный уровень Windows 2000	Членами группы могут быть учетные записи пользователей и глобальные группы из любого домена	Членами группы могут быть учетные записи пользователей и группы из того же домена	Универсальные группы безопасности не создаются в доменах смешанного режима
Член группы	Может быть помещен в локальные группы другого домена; получает разрешения в том же домене	Может быть помещен в другой домен; получает разрешения в любом домене	Может быть помещен в другие группы; получает разрешения в любом домене

Создавая группы, вы передаете команде DSADD GROUP составное имя (DN) группы. Простое имя, являющееся частью DN, задает отображаемое имя группы. Остальная часть DN указывает, где в Active Directory должна находиться группа, в том числе определяя контейнер, в котором будет создана группа, и соответствующий домен. По умолчанию создается глобальная группа безопасности. Например, вы можете создать глобальную группу безопасности Sales в OU «Sales» домена cpandl.com командой **dsadd group "CN=Sales,OU=Sales,DC=cpandl,DC=com"**. В этом случае Sales считается как отображаемым именем группы, так и именем учетной записи в SAM. Однако остальные свойства не будут заданы.

Имена групп не чувствительны к регистру букв и могут быть длиной до 64 символов. В большинстве случаев нужно указывать тип группы и область ее действия. Параметр –Secgrp позволяет задать тип группы — безопасности или распространения:

- введите **–secgrp yes**, чтобы создать группу безопасности;
- введите **–secgrp no**, чтобы создать группу распространения.

Для указания области действия группы используйте параметр –Scope:

- **–scope l** — создает локальную группу домена;
- **–scope g** — создает глобальную группу;
- **–scope u** — создает универсальную группу. В случае групп безопасности это возможно только на функциональном уровне Windows Server 2003.

Примечание По умолчанию группы создаются как группы безопасности с глобальной областью (действия). Поэтому, даже если вы создаете группу безопасности с другой областью, указывать параметр **–secgrp yes** не нужно, так как он предполагается по умолчанию.

Первые 20 символов имени группы определяют имя учетной записи в SAM для группы, которое также является именем группы для операционных систем версий до Windows 2000. Имя учетной записи в SAM должно быть уникально в рамках домена, поэтому вам может понадобиться указать имя учетной записи в SAM для группы отличным от отображаемого имени. В таком случае вы можете задать имя учетной записи в SAM через параметр –Samid.

При создании группы можно указать и ее членство в других группах. Если создаваемая группа должна быть членом существующей группы, используйте –Memberof для задания DN этих групп. Если группа должна включать в себя пользователей или другие группы, можно указать участников группы с помощью параметра –Members. Но гораздо легче использовать команду DSMOD GROUP для настройки членства в группах. Почему? Вы можете передать этой команде список DN из команды DSQUERY USER. Это избавит вас от набора десятков, а иногда и сотен DN.

Рассмотрим несколько примеров, показывающих, как создавать группы.

Создание локальной группы безопасности домена с именем Engineering и ее добавление в OU «Engineering» домена tech.cpandl.com:

```
dsadd group
"CN=Engineering,OU=Engineering,DC=tech,DC=cpandl,DC=com" -scope l
```

Создание глобальной группы безопасности Engineering Global в контейнере Users домена cpandl.com с присвоением gEngineering в качестве имени учетной записи в SAM:

```
dsadd group "CN=Engineering Global,CN=Users,DC=cpandl,DC=com"
-samid "gEngineering"
```

Создание универсальной группы распространения Engineering All в OU «Engineering» домена cpandl.com с присвоением allEngineering в качестве имени учетной записи в SAM:

```
dsadd group "CN=Engineering
All,OU=Engineering,DC=cpandl,DC=com" -samid "allEngineering"
-secgrp no -scope u
```

Если при создании группы возникли проблемы, вы увидите предупреждение и вам придется проверить синтаксис команды, чтобы убедиться, что все значения заданы правильно и DN корректны. В ином случае DSADD GROUP сообщит DSADD SUCCEEDED. После создания группы в нее можно добавить участников и задать дополнительные свойства.

Создание локальных групп и включение в них участников

Локальные группы создаются на индивидуальных компьютерах, чтобы упростить управление правами пользователей, которые подключаются локально, а не к домену. Для создания локальной группы вам нужно войти на нужный компьютер локально или удаленно. После этого вы можете создавать необходимые локальные группы командой NET LOCALGROUP.

Вы можете создать локальную группу, просто введя имя команды, имя группы и параметр /Add:

```
net localgroup localDevs /add
```

Примечание Создать локальную учетную запись группы на контроллере домена нельзя. Контроллеры домена не поддерживают локальные учетные записи.

Здесь вы создаете группу localDevs на локальном компьютере. При желании можно воспользоваться параметром /Comment для добавления описания группы, например:

```
net localgroup localDevs /comment:"Local Developers and
Testers" /add
```

Если создание учетной записи группы прошло успешно, команда NET LOCALGROUP сообщит "Command Completed Successfully". А если возникнут проблемы с созданием учетной записи, NET LOCALGROUP просто выведет подсказку по синтаксису команды. В таком случае проверьте правильность команды.

Создавая локальную группу, вы можете указать и список локальных учетных записей пользователей, которые должны быть членами этой группы. Список имен должен следовать за именем группы, например:

```
net localgroup localDevs williams johng edwardh /add
```

В данном случае вы создаете группу localDevs и добавляете в нее пользователей WilliamS, JohnG и EdwardH.

Вы можете добавить членов локальной группы и после ее создания. Синтаксис команды совпадает с тем, которым вы пользовались при создании группы. Например, вы создали группу custSupport командой:

```
net localgroup custSupport /add
```

Тогда впоследствии можете добавить пользователей в группу так:

```
net localgroup custSupport williams johng edwardh /add
```

Здесь вы добавляете пользователей WilliamS, JohnG и EdwardH в группу custSupport.

Управление учетными записями групп

Управление учетными записями групп из командной строки отличается от управления ими из оснастки Active Directory Groups and Computers (Active Directory – Группы и компьютеры) в основном тем, что командная строка предоставляет больше возможностей и упрощает одновременную работу со множеством учетных записей групп.

Просмотр и поиск учетных записей групп

Если вам нужна информация об учетных записях групп, используйте команду DSQUERY GROUP. Эта команда позволяет вести поиск по простому имени, имени учетной записи в SAM и по описанию. Она также поддерживает символы подстановки в любых из этих полей. Команда DSQUERY GROUP выводит DN групп, удовлетворяющих условиям поиска; ее вывод может быть перенаправлен на вход других команд, в том числе DSGET GROUP.

Обычно вы будете использовать команды DSQUERY GROUP и DSGET GROUP совместно. Сначала с помощью DSQUERY GROUP вы получите DN одной или более групп, а затем выведете свойства соответствующих учетных записей через DSGET GROUP. Вот список наиболее полезных параметров DSGET GROUP:

- **–Desc** – отображает описание найденных учетных записей групп;
- **–Dn** – выводит DN найденных учетных записей групп;
- **–Samid** – выводит имена учетных записей в SAM для найденных учетных записей групп;
- **–Scope** – отображает область действия найденных учетных записей групп (локальная, глобальная или универсальная);
- **–Secgrp** – выводит yes (да), если группа является группой безопасности и no (нет), если это группа распространения;

- **–Sid** – отображает идентификаторы защиты для найденных учетных записей групп.

Как и другие команды DSGET, DSGET GROUP выводит результат в виде таблицы, и обычно нужно добавлять параметр –Dn или –Samid, чтобы идентифицировать группы в выходных данных. Например, если вы хотите найти все группы с именем, начинающимся с «marketing», воспользуйтесь командной строкой вида:

```
dsquery group -name marketing* | dsget group -dn -scope -secgrp
```

В результате будет выведен список DN, области и информация о группе безопасности:

```
dn                                                  scope          secgrp
  CN=MarketingAll,OU=Sales,DC=cpandl,DC=com         universal      no
  CN=Marketing Global,OU=Sales,DC=cpandl,DC=com     global         no
  CN=Marketing Local,OU=Sales,DC=cpandl,DC=com      domain local   no
dsget succeeded
```

Определение членства в группах

Определить членство в группах позволяет другой синтаксис команды DSGET GROUP, который включает два дополнительных параметра: –Members и –Memberof. Параметр –Members сообщает, какие пользователи и группы принадлежат к данной группе, а параметр –Memberof – к каким группам принадлежит данная группа. Как работают эти параметры? Допустим, вы хотите перечислить текущих участников группы AllUsers. Для этого вы вводите:

```
dsquery group -name AllUsers | dsget group -members
```

или сами указываете DN группы, например:

```
dsget group "CN=AllUsers,CN=Users,DC=cpandl,DC=com" -members
```

Здесь группа содержится в контейнере Users домена cpandl.com. В любом случае появится список DN членов этой группы, например:

```
"CN=Tech,OU=Tech,DC=cpandl,DC=com"
"CN=Engineering,OU=Eng,DC=cpandl,DC=com"
"CN=Sales,OU=Sales,DC=cpandl,DC=com"
"CN=Domain Users,CN=Users,DC=cpandl,DC=com"
```

Как показано в листинге, группа AllUsers включает группы Tech, Engineering, Sales и Domain Users. Группа AllUsers также может включать учетные записи пользователей.

Если вы хотите определить, к каким группам относится данная группа, используйте параметр –Memberof. Например, группа DevUsers может входить в группы Domain Administrators и Developers, и, чтобы вывести информацию о членстве в группах, наберите:

```
dsquery group -name devusers | dsget group -memberof
```

или

```
dsget group "CN=devusers,OU=Dev,DC=cpandl,DC=com" -memberof
```

Обе команды работают одинаково. В первом случае вы используете DSQUERY GROUP для получения DN учетной записи группы, а во втором — сами указываете DN. Так или иначе результатом будет список групп, в которые входит DevUsers.

Примечание Оба приема позволяют получать сведения о членстве в нескольких группах. Однако нет способа вывести DN или имена учетных записей в SAM соответствующих групп, так как второй синтаксис вызова DSGET GROUP это не поддерживает.

Изменение типа или области групп

Иногда после создания группы нужно изменить ее тип или область (действия). Это не так-то просто, как кажется поначалу, потому что множество правил препятствует произвольным изменениям, которые могут повлиять на доступ к ресурсам в рамках всей организации. Во-первых, тип группы или область нельзя изменять при использовании функционального уровня (режима) Mixed в Windows 2000 или Interim в Windows Server 2003. В режиме Native в Windows 2000 или Windows Server 2003 действуют следующие правила.

- **Локальные группы домена** Область может быть преобразована в универсальную, если в них не входят другие группы с локальной областью.
- **Глобальные группы** Область может быть преобразована в универсальную, если эти группы не входят в другие группы с глобальной областью.
- **Универсальные группы** Область может быть преобразована в любую другую. Учтите, что в глобальные группы не могут входить универсальные и что локальные группы могут быть членами только других локальных групп.

С этими ограничениями вы можете использовать команду DSMOD GROUP с параметром –Secgrp, чтобы преобразовать:

- группу распространения в группу безопасности (**–secgrp yes**);
- группу безопасности в группу распространения (**–secgrp no**).

Рассмотрим пару примеров.

Преобразование группы безопасности Engineering в группу распространения:

```
dsquery group -name Engineering | dsmod group =secgrp no
```

Преобразование группы распространения AllMarketing в группу безопасности:

```
dsmod group "CN=AllMarketing,OU=Marketing,DC=cpandl,DC=com"
=secgrp yes
```

Вы также можете использовать команду DSMOD GROUP с параметром –Scope, чтобы установить область как:

- локальную для домена (**–scope l**);
- глобальную (**–scope g**);
- универсальную (**–scope u**).

Рассмотрим пару примеров.

Установить область группы Marketing как локальную для домена:

```
dsquery group -name Marketing | dsmod group -scope l
```

Установить область группы Sales как глобальную:

```
dsmod group "CN=Sales,CN=Users,DC=cpandl,DC=com" -scope g
```

Добавление, удаление или замена членов групп

Используя командную строку, легко управлять членством в группах для любой группы. Как и через GUI, вы можете добавлять или удалять пользователей, группы или компьютеры как члены нужной группы. Но утилиты командной строки предоставляют более «продвинутые» возможности, позволяя выполнять такие операции сразу над несколькими членами. Вы даже можете полностью заменить существующий список членов.

Добавление членов к группе

Единственной командной строкой можно, например, добавить всех 100 пользователей OU «Sales» в группу AllSales. Для этого вы должны получить список нужных учетных записей пользователей командой DSQUERY USER, а затем передать этот список на вход команде DSMOD GROUP. Для добавления членов в группу служит параметр –Addmbr, поэтому команда может выглядеть так:

```
dsquery user "OU=Sales,DC=ny,DC=cpandl,DC=com" | dsmod group
"CN=AllSales,OU=Sales,DC=ny,DC=cpandl,DC=com" -addmbr
```

Здесь вы получаете список всех учетных записей пользователей в OU домена ny.cpandl.com и передаете их на вход команды DSMOD GROUP. Затем команда DSMOD GROUP добавляет этих пользователей как членов в группу AllSales, которая находится в контейнере Sales домена ny.cpandl.com.

Другой способ применения параметра –Addmbr — самостоятельный ввод DN объектов, которые вы хотите добавить. Так, если вы хотите добавить группы SalesLocal и SalesGlobal в группу AllSales, воспользуйтесь командой:

```
dsquery group -name AllSales | dsmod group -addmbr
"CN=SalesLocal,OU=Sales,DC=ny,DC=cpandl,DC=com"
"CN=SalesGlobal,OU=Sales,DC=ny,DC=cpandl,DC=com"
```

Примечание Помните, что DN объектов могут включать имена учетных записей пользователей, групп и компьютеров.

Удаление членов из группы

Для удаления членов из групп предназначен параметр –Rmmbr. Как и –Addmbr, –Rmmbr принимает DN объектов из вывода DSQUERY USER или из списка, разделенного пробелами. Поэтому, если вы хотите удалить всех сотрудников отделов маркетинга и поддержки пользователей, один из способов сделать это выглядит так:

```
dsquery user "OU=Marketing,DC=ny,DC=cpandl,DC=com" | dsmod
group "CN=AllSales,OU=Sales,DC=ny,DC=cpandl,DC=com" -rmmbr

dsquery user "OU=CustSupport,DC=ny,DC=cpandl,DC=com" | dsmod
group "CN=AllSales,OU=Sales,DC=ny,DC=cpandl,DC=com" -rmmbr
```

В данном случае первая команда получает список всех пользователей в OU «Marketing», а затем передает его на вход

команды DSMOD GROUP, которая удаляет эти учетные записи из группы AllSales. Вторая команда получает список пользователей в OU «CustSupport» и передает их на вход команды DSMOD GROUP, которая удаляет эти учетные записи из группы AllSales.

Примечание Когда эти два списка пользователей не совпадают с текущим списком членов группы AllSales, возникает проблема. Например, если новые сотрудники отдела маркетинга приступили к работе и были добавлены в OU «Marketing», но не получили права на доступ к информации по продажам, они не станут членами группы AllSales. Когда команда DSMOD GROUP обнаружит первое несовпадение, она завершит работу и сообщит об ошибке. Но, так как вы не хотите, чтобы это произошло из-за мелкого расхождения, добавьте параметр –C (тогда команда продолжит свою работу). Этот параметр указывает выводить сообщения об ошибках, но продолжать обработку изменений.

Как и при использовании параметра –Addmbr, вы можете сами указать DN удаляемых объектов. Например, чтобы удалить группы SalesLocal и SalesGlobal из группы AllSales:

```
dsquery group -name AllSales | dsmod group -rmmbr
"CN=SalesLocal,OU=Sales,DC=ny,DC=cpandl,DC=com"
"CN=SalesGlobal,OU=Sales,DC=ny,DC=cpandl,DC=com"
```

Примечание Из-за специфики оформления страниц этой книги можно не заметить, что между DN групп ставится пробел. Этот пробел необходим для того, чтобы каждое DN группы было правильно интерпретировано.

Замена всех членов в группе

Командная строка опережает GUI в том плане, что позволяет заменить весь список членов группы одной командой. Например, если состав группы AllUsers устарел и добавлять/удалять пользователей вручную слишком нудно, вы можете заменить текущий список членов.

Для замены текущего списка членов другим списком предназначен параметр –Chmbr команды DSMOD GROUP. Этот параметр принимает на входе как результат работы DSQUERY USER, так и список DN, отделенных пробелами. Одним из способов замены текущего списка членов группы и добавле-

ния всех пользователей организации в группу AllUsers является применение следующей команды:

```
dsquery user -name * | dsmod group
"CN=AllUsers,CN=Users,DC=seattle,DC=cpandl,DC=com" -chmbr
```

Здесь DSMOD GROUP сначала удаляет все существующие объекты, являющиеся членами группы, а затем добавляет объекты, переданные на вход команды. В случае ошибки команда завершится, и изменения не вступят в силу.

Примечание Параметр –C позволяет сообщить, что операция должна продолжаться даже при наличии ошибок. Это может привести к тому, что список членов группы окажется пустым. Такое произойдет, если команда DSMOD GROUP удалит текущий список пользователей без проблем, но не сумеет добавить членов. Удаление членов группы требует лишь наличия прав администратора. Однако добавление пользователей зависит от предоставленных входных данных.

Перемещение учетных записей групп

Как и учетные записи пользователей, учетные записи групп можно легко перемещать в другой контейнер или OU внутри текущего домена. Для этого укажите текущее DN группы командой DSMOVE, а затем задайте DN нового местонахождения учетной записи группы через параметр –Newparent. Например, если вы хотите переместить группу ProdDev из контейнера Users в OU «Developers», то должны указать DN учетной записи группы как "**CN=ProdDev,CN=Users,DC=cpandl, DC=com**" и DN родителя для нового местонахождения как "**OU=Developers,DC=cpandl,DC=com**". Такая команда должна выглядеть так:

```
dsmove "CN=ProdDev,CN=Users,DC=cpandl,DC=com" -newparent
"OU=Developers,DC=cpandl,DC=com"
```

DSQUERY GROUP избавит вас от ввода DN групп в команде DSMOVE:

```
dsquery group -name "ProdDev" | dsmove -newparent
"OU=Developers,DC=cpandl,DC=com"
```

Здесь DSMOVE получает DN учетной записи группы ("CN=ProdDev,CN=Users,DC=cpandl,DC=com") от команды DSQUERY GROUP.

Переименование учетных записей групп

Как и с учетными записями пользователей, с учетными записями групп сопоставляются идентификаторы защиты. Это позволяет изменять имя группы без модификации прав доступа к индивидуальным ресурсам, таким как файлы и папки. Когда вы переименовываете группу, вы изменяете лишь ее простое имя.

Переименовать группу позволяет команда DSMOVE. Указав DN группы и параметр -Newname для задания нового простого имени, вы можете, например, переименовать группу из ProdDevs в TechDevs:

```
dsmove "CN=ProdDevs,OU=Developers,DC=cpandl,DC=com" -newname
"TechDevs"
```

Как и при перемещении групп, DN группы можно получить от команды DSQUERY GROUP. Рассмотрим следующий пример:

```
dsquery group -name ProdDevs | dsmove -newname "TechDevs"
```

В данном случае DSQUERY GROUP определяет DN группы ProdDevs и передает эту информацию команде DSMOVE для переименования группы.

Так как переименование группы не меняет ее имя для операционных систем до Windows 2000 или описание, связанное с группой, вам нужно изменить эти свойства. Для этого воспользуйтесь командой DSMOD GROUP. Параметр -Samid устанавливает имя группы для операционных систем до Windows 2000, а параметр -Desc задает описание. Рассмотрим следующий пример:

```
dsquery group -name TechDevs | dsmod -samid techdevs -desc
"Technical Developers Group"
```

Здесь вы меняете имя группы для операционных систем до Windows 2000 на **techdevs**, а описание — на «Technical Developers Group».

Удаление учетных записей групп

Чтобы удалить группу из Active Directory, применяйте команду DSRM. В большинстве случаев требуется удалить лишь конкретную группу, а не несколько групп, например с именами, начинающимися на «М». Вы удаляете группу, передав команде DSRM составное имя (DN) учетной записи группы:

```
dsrm "CN=AllSales,OU=Sales,DC=chicago,DC=cpandl,DC=com"
```

По умолчанию DSRM запросит подтверждение на удаление. Если вы не хотите получать такой запрос, воспользуйтесь параметром –Noprompt, например:

```
dsrm "CN=AllSales,OU=Sales,DC=chicago,DC=cpandl,DC=com"
-noprompt
```

В некоторых ситуациях может понадобиться удалить несколько групп сразу. Например, если при глобальной реорганизации компании отдел маркетинга передается другой компании, вам больше не потребуются группы, связанные с этим отделом. И если имена групп начинаются со слова «Marketing», вы можете удалить их командой:

```
dsquery group -name Marketing* | dsrm -c
```

Здесь вы передаете команде DSRM составные имена (DN) всех групп, которые начинаются со слова «Marketing». Добавление параметра –C позволяет продолжать работу даже при возникновении ошибки.

Внимание! Если на вход DSRM передаются DN, полученные командой DSQUERY GROUP, использовать команду DSRM саму по себе нельзя. Например, команда **dsquery group –name Marketing* | dsrm** недопустима, так как вы должны указать какой-нибудь параметр. Наиболее безопасен параметр –C, поскольку он лишь указывает DSRM продолжить работу при ошибке. С другой стороны, –Noprompt сообщает команде DSRM удалять все без запроса на подтверждение, и это может привести к тому, что будет удалено гораздо больше групп, чем ожидалось, а способа отменить операцию нет.

Глава 14

Управление сетевыми принтерами и службами печати

В большинстве организаций установлены принтеры для печати как больших объемов (высокопроизводительные дорогие принтеры), так и малых (недорогие принтеры). Обычно принтеры, предназначенные для печати больших объемов, выдерживают тяжелую ежедневную нагрузку, создаваемую множеством пользователей, а недорогие принтеры, рассчитанные на печать малых объемов, эксплуатируются небольшими группами или индивидуальными пользователями. Независимо от способа использования принтеров у сервера печати должно быть достаточно памяти и процессорных мощностей для управления службами печати. В случае больших объемов печати или при постоянной необходимости печатать большие и сложные документы может потребоваться специальная конфигурация сервера и выделение отдельной машины, которая будет обслуживать только службы печати. В остальных случаях серверы печати обычно не являются выделенными серверами. На самом деле большинство серверов печати — стандартные настольные компьютеры, которые выполняют в сети и другие функции. Просто учтите, что операционные системы Microsoft Windows Server 2003 и Windows XP Professional отдают приоритет доступу к файлам, а не к сетевым принтерам, поэтому, если система выполняет функции и файл-сервера, и сервера печати, печать может замедляться, чтобы избежать снижения производительности при доступе к файлам.

Кроме того, на серверах печати должен быть достаточный объем дискового пространства для того, чтобы управлять заданиями на печать (print jobs). Требуемый объем дискового пространства зависит от размеров таких заданий и от длины оче-

реди печати. Для большей производительности каталог буферных файлов принтера должен располагаться на выделенном диске, не применяемом в других целях. Основной смысл администрирования служб печати — их обслуживание. Для правильного обслуживания и поддержки служб печати вы должны отлеживать информацию о спулере принтера (подсистеме буферизации) и статистику использования. Эти сведения помогут определить, как работают службы. Хотя основное внимание уделяется производительности, полезно задействовать несколько утилит командной строки, помогающих поддерживать службы печати и устранять неполадки в работе принтеров. Эти инструменты тоже будут рассмотрены в данной главе.

Получение технических сведений и информации о проблемах в работе принтеров

Принтеры часто покупаются и размещаются без учета того, как они будут использоваться в дальнейшем. Кто-то видит, что на данном участке нужен принтер, и принтер заказывается и устанавливается. Иногда заказом и установкой занимается не администратор, поэтому, когда дело доходит до управления принтером, администратору приходится действовать вслепую. Независимо от того, как был получен принтер, администратор или служба поддержки должна иметь информацию о конфигурации принтера, в том числе сведения, какие драйверы доступны и какие драйверы реально используются. Вам нужно периодически проверять, насколько загружен принтер и как он справляется с нагрузкой. Вам также потребуется отслеживать состояние принтера, число заданий в очереди и другую важную информацию, которая поможет вовремя выявить проблемы. Во многих случаях эта информация пригодится и при планировании.

Отслеживание информации о принтерах и их драйверах

Получить детальную информацию о принтерах, установленных в системе, позволяет команда PRINTDRIVERINFO. Она включена в Windows Server 2003 Resource Kit и предназначена для работы как с локальными, так и с удаленными системами.

Получение информации о драйверах

По умолчанию PRINTDRIVERINFO возвращает сведения обо всех драйверах принтеров, установленных в вашей локальной

системе. То есть, набрав в командной строке **printdriverinfo**, вы получите список драйверов для локального компьютера. Как показывает листинг 14-1, информация о драйверах очень подробна.

Листинг 14-1. Вывод команды PRINTDRIVERINFO

```
-------Report by driver name-------
Driver Name: hp business inkjet 1100 series
Environment: Windows NT x86
Kernel Mode Driver: FALSE
Using inf : C:\WINDOWS\inf\ntprint.inf
Inbox Driver: FALSE
Driver Technology: Monolithic
Driver Stack File: hpz2ku08.dll , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpf0uk08.dat , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzpm308.dll , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpfmom08.hlp , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzrer08.dll , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzl3208.dll , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzcfg08.exe , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzeng08.exe , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzflt08.dll , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzime08.dll , Date: 27/06/2003 , Version: 3.02
Driver Stack File: hpzjui08.dll , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzpre08.exe , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzres08.dll , Date: 27/06/2003 , Version: 0.00
Driver Stack File: hpzvip08.dll , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpzrm308.dll , Date: 27/06/2003 , Version: 2.224
Driver Stack File: hpwhlmn.dll , Date: 27/06/2003 , Version: 2.213
Driver Stack File: hpwhsvb.dll , Date: 27/06/2003 , Version: 2.213
-=Misc Driver Info=-
DriverPath: C:\WINDOWS\System32\spool\DRIVERS\W32X86\3\hpz2ku08.dll
Data File: C:\WINDOWS\System32\spool\DRIVERS\W32X86\3\hpf0uk08.dat
Config File: C:\WINDOWS\System32\spool\DRIVERS\W32X86\3\hpzpm308.dll
Help file: C:\WINDOWS\System32\spool\DRIVERS\W32X86\3\hpfmom08.hlp
Monitor Name: HPWHLMN
Default Data Type: RAW
```

Если вы внимательно посмотрите на эту информацию, то заметите, что она включает следующее.

- **Имя драйвера принтера, например «hp business inkjet 1100 series»** Имя драйвера принтера используется операционной системой Windows для отслеживания этого драйвера. Драйвер должен подходить к конкретному подключенному

принтеру. В данном примере принтер — Hewlett-Packard (HP) Business InkJet 1100 series. Когда вы командуете приложению напечатать документ, оно использует драйвер принтера для преобразования документа в формат, понятный физическому устройству печати. Если у вас возникают проблемы с печатью и вы подозреваете, что причиной является неподходящий драйвер, то ответ вы найдете в сведениях об имени драйвера.

- **Режим драйвера принтера** Драйвер печати может работать в режиме ядра или в пользовательском режиме. В режиме ядра драйвер функционирует как программы, запущенные самой операционной системой, а в пользовательском режиме — как программа, запущенная пользователем. Стоп-код, появляющийся из-за ошибки драйвера в режиме ядра, как правило, более информативен, чем сообщение, выводимое при ошибке в пользовательском режиме. Однако драйверы принтеров, работающие в режиме ядра, чаще приводят к нестабильности системы, если в них есть какие-то проблемы. Разрешается ли устанавливать драйверы принтера режима ядра, зависит от политик групп. В Windows Server 2003 установка драйверов режима ядра заблокирована по умолчанию, а в Windows XP Professional, напротив, разрешена по умолчанию. Соответствующий флажок — Disallow Installation Of Printers Using Kernel-Mode Drivers — находится в Policy\Computer Configuration\Administrative Templates\Printers.
- **INF-файл драйвера принтера** С каждым драйвером, установленным в системе, сопоставлен свой INF-файл. Этот файл служит для настройки драйвера принтера.
- **DLL спулера печати и соответствующие файлы данных** Конкретная DLL спулера печати определяется путем к драйверу. Со спулером связаны файлы данных, а также конфигурационные и справочные файлы. Спулер принтера — компонент, который передает процессору печати документы, которые хочет напечатать пользователь. В свою очередь процессор печати преобразует данные в формат, понятный физическому устройству печати. Затем эти данные возвращаются спулеру для пересылки на устройство.
- **Файлы стека драйвера принтера** В этой части информации сообщаются все файлы стека, связанные с данным драйвером принтера. Для каждого файла выводятся дата

создания и версия. Документы пересылаются (с использованием маршрутизатора печати) из спулера в стек принтера, который также называется очередью печати. Документ, попадая в очередь печати, становится заданием на печать, т. е. документы фактически являются заданиями, которыми должен управлять спулер.

- **Монитор печати** С каждым устройством печати сопоставляется монитор печати. С принтерами, которые поддерживают двунаправленную печать, сопоставляются языковый монитор (language monitor), который управляет двухсторонней передачей данных между принтером и спулером печати, а также монитор порта, который контролирует порт ввода-вывода принтера. В совокупности они называются *монитором печати для устройства печати* (print monitor for a print device). Если у принтера есть языковой монитор, имя этого монитора отображается в списке (оно совпадает с именем его файла, но без расширения .dll). Если у принтера нет языкового монитора, выводится значение (Null). Когда документ достигает вершины стека драйвера принтера, монитор печати отвечает за его отправку устройству печати. Устройство печати — это физическое устройство, на котором будет отпечатан документ. У большинства устройств печати свои мониторы печати, создаваемые производителями этих устройств. Если монитор печати поврежден или отсутствует, вам придется переустановить его.
- **Тип данных по умолчанию** К числу типов данных, используемых принтером по умолчанию, обычно относятся EMF (enhanced metafile) и RAW. EMF опирается на PCL (printer control language) — язык описания страниц, поддерживаемый принтером. Документы в формате EMF передаются серверу печати с минимальной предварительной обработкой, что требует больше работы от сервера печати. RAW обычно используется с PostScript-принтерами. Документы в формате RAW обрабатываются полностью на клиентской машине и не изменяются сервером печати.

Проверка конкретных принтеров, драйверов и систем

Чтобы получить информацию об удаленных серверах печати и сетевых принтерах через PRINTDRIVERINFO, используйте параметр /S: и укажите имя сервера в домене, например:

```
printdriverinfo /s:corpserver01
```

Здесь вы получаете информацию о драйвере печати на сервере CorpServer01.

Команда PRINTDRIVERINFO также позволяет искать конкретный драйвер по имени. Для этого служит параметр /D:

```
printdriverinfo /d:"hp business inkjet 1100 series"
```

Здесь вы ищете информацию о драйвере HP Business InkJet 1100 series.

Примечание К сожалению, символы подстановки в имени драйвера не поддерживаются. Вы должны указать полное имя драйвера.

Если вы знаете имя принтера на основе содержимого папки Printer And Faxes (Принтеры и факсы) в Control Panel (Панель управления), то можете искать по имени и конкретный принтер. В следующем примере используется параметр /P: и указано имя принтера:

```
printdriverinfo /p:"magicolor 2300 dl"
```

Результатом будет отчет, полученный от команды PRINTDRIVERINFO по имени принтера. Этот отчет включает дополнительную информацию, в том числе имя принтера и раздел сведений о принтере, выглядящий как:

```
-=Misc Printer Info=-
Share Name: centralprinter
Port Name: IP_192.168.1.100
Print Processor Name: MIMFPR_B
Data type Name: IMF
```

В дополнительных сведениях сообщается следующее.

- **Имя принтера в сети** Если принтер используется как общий сетевой ресурс, в этой строке выводится соответствующее имя. По умолчанию такое имя формируется из первых восьми символов имени принтера (пробелы игнорируются).
- **Порт принтера** Принтеры напрямую подключаются к серверу печати через порт LPT, COM или USB. Подключаемые к сети принтеры обычно имеют TCP/IP-порты. Имя TCP/IP-порта по умолчанию показывается как IP-адрес порта. Но имя такого порта может быть сформировано на основе имени принтера (что и происходит, когда к сети подключаются принтеры HP).

- **Процессор печати** С каждым принтером сопоставлен процессор печати, который отвечает за создание RAW-данных, необходимых для печати на принтере.

Отслеживание информации о спулере печати и статистики

Отслеживание информации о спулере печати и статистики использования помогает ответить на важные вопросы о службах печати в организации.

- Насколько в среднем загружен сервер печати?
- Каков средний размер заданий на печать?
- Сколько заданий находится в очереди?
- Каково текущее состояние принтера?
- Сколько времени работает спулер печати?
- Сколько времени используется принтер?
- Сколько времени используется сервер печати?

Почему так важны ответы на эти вопросы? Дело в том, что, зная эти ответы, вы сможете эффективнее поддерживать и управлять службами печати в организации, избегая авральных ситуаций. Кроме того, это позволит планировать, что именно вам потребуется в будущем.

Получение сводной информации о спулере печати

Основной инструмент, с помощью которого вы будете получать информацию о спулере печати и статистику, — команда SPLINFO из Windows Server 2003 Resource Kit. У этой команды два режима отображения: сводный (summary) и подробный (verbose). Сводный режим действует по умолчанию; в нем показывается общая статистика по какому-то серверу печати. Подробный режим активируется параметром /V.

Чтобы увидеть сводную информацию по серверу печати, к которому вы подключены, наберите в командной строке **splinfo**. Для удаленного сервера печати добавьте его UNC-имя. Например, если вам нужно увидеть сводную статистику для сервера CorpServer03, введите:

```
splinfo \\corpserver03
```

Независимо от того, работаете вы с локальным или удаленным сервером печати, сводная информация выглядит примерно так, как показано на листинге 14-2.

Листинг 14-2. Стандартный вывод команды SPLINFO

```
Number Local Printers          3
Windows Version                5.1 Build 2600 (Service Pack 1)
Number of Processors           1 PROCESSOR_INTEL Level 15
Total Jobs Spooled             258
Total Bytes Printed            14,512,067,850
Average Bytes/Job              56,248,325
Browse List Requested          0
Browse Printer Added           0
Spooler Up Time                04:17:18
Server Up Time                 04:17:52
```

Примечание Если в системе, которую вы проверяете, нет установленных принтеров, SPLINFO вернет «No local printers installed».

Здесь в однопроцессорной системе под управлением Windows XP Professional установлено три принтера. Хотя эти принтеры рассматриваются как локальные, они не обязательно должны быть подключены к компьютеру физически. Они могли бы быть подключены и через сеть (при наличии у них собственных сетевых плат). Если вы изучите статистику, то обнаружите, что:

- в среднем сервер печатает одно задание в минуту. В спулере печати, который работает четыре часа и 17 минут (257 минут), всего 258 заданий;
- средний размер задания на печать около 50 Мб. Чтобы получить это значение, разделите среднее число байтов на 1024 для получение размера в Кб, а потом еще раз поделите на 1024 для получения размера в Мб;
- за четыре часа и 17 минут через спулер прошло около 12 Гб данных.

В этом примере сервер печати перегружен, особенно если учесть, что службы печати работают под управлением Windows XP Professional (т. е. скорее всего на обычном настольном компьютере). Если вы проверили статистику несколько раз и после нескольких перезапусков принтера/спулера обнаружили такую же нагрузку, у вас есть повод для беспокойства, потому что принтер сильно нагружен и средний размер заданий на печать довольно велик (около 50 Мб). При таком уровне нагрузки вам следует детальнее исследовать производительность и использование сервера, как обсуждалось в главе 7. При по-

иске информации о конфигурации каждого принтера вам скорее всего потребуется глубже разобраться в статистике. Не исключено, что вам придется решать, будет ли производительность выше при использовании Windows Server 2003 или лучше перенести часть дополнительных служб на другую машину в сети. После мониторинга производительности системы и интенсивности ее использования в течение достаточно длительного времени, возможно, вы обнаружите, что требуется:

- дополнительная память в связи с большим средним размером заданий на печать;
- дополнительные процессорные мощности из-за большого среднего числа заданий;
- дополнительное дисковое пространство или выделенный диск под папку спулера печати.

Также вы можете обнаружить, что эту систему просто необходимо регулярно проверять, чтобы убедиться в нормальной работе служб печати. Здесь вы могли бы легко автоматизировать мониторинг, создав сценарий, который записывает статистику в файл, и запланировав этот сценарий на периодический запуск, как описано в главе 4.

Получение детальной информации о спулере печати

Параметр /V команды SPLINFO позволяет получить детальную информацию по каждому сконфигурированному принтеру. Как она выглядит, см. на листинге 14-3. Учтите, что обычно нужно отслеживать использование принтера в течение нескольких дней для выявления изменений, которые могут понадобиться в конфигурации подсистемы печати. В течение этого времени вам, возможно, потребуется перезапускать спулер печати для сброса статистики.

Листинг 14-3. Подробный вывод команды SPLINFO

```
Number Local Printers          3
Windows Version                5.1 Build 2600 (Service Pack 1)
Number of Processors           1 PROCESSOR_INTEL Level 15
Total Jobs Spooled             258
Total Bytes Printed            14,512,067,850
Average Bytes/Job              56,248,325
Browse List Requested          0
Browse Printer Added           0
Spooler Up Time                04:17:18
Server Up Time                 04:17:52
```

```
Printer Name                    magicolor 2300 DL
Total Printer Jobs:             96
Total Printed Bytes:            8,451,245,024
Printer Up Time                 04:17:21
Number of Jobs in Queue         3
cRef                            2
cRefIC                          0
Max cRef                        5
Number spooling                 2
Max Number spooling             3
Printer Started                 11/15/2004  16:57 (UTC)
Average Bytes/Job               88,033,802
Printer Change Count ID         28633
Printer Status                  0

Printer Name                    hp businessjet 1100 series
Total Printer Jobs:             162
Total Printed Bytes:            6,060,822,826
Printer Up Time                 04:17:21
Number of Jobs in Queue         24
cRef                            0
cRefIC                          0
Max cRef                        1
Number spooling                 1
Max Number spooling             2
Printer Started                 11/15/2004  16:57 (UTC)
Average Bytes/Job               37,412,486
Printer Change Count ID         43c3c1
Printer Status                  0

Printer Name                    hp officejet 5500 series
Total Printer Jobs:             0
Total Printed Bytes:            0
Printer Up Time                 04:17:21
Number of Jobs in Queue         0
cRef                            8
cRefIC                          0
Max cRef                        10
Number spooling                 0
Max Number spooling             0
Printer Started                 11/15/2004  16:57 (UTC)
Average Bytes/Job               0
Printer Change Count ID         54619b
Printer Status                  0
```

Для каждого принтера вы можете изучить следующую статистику.

- **Total printer jobs (общее число заданий на печать)** Показывает, сколько заданий на печать обработано с момента включения принтера. Сравните общее число заданий со временем работы принтера — это хороший показатель того, насколько принтер загружен на самом деле. Из ранее приведенного листинга видно, что цветной лазерный принтер Minolta QMS magicolor обрабатывает около 22 заданий на печать в час, так что он загружен весьма значительно.
- **Total printed bytes (общее число байтов, отправленных на печать)** Показывает общее число байтов, переданных принтеру для печати с момента его последнего включения. Это дает представление о том, сколько данных обрабатывает принтер. Сравните это значение с временем работы принтера и вы получить хороший показатель того, сколько данных проходит через принтер в течение часа или дня. Из ранее приведенного листинга видно, что цветной лазерный принтер Minolta QMS magicolor обрабатывает около 1,84 Гб данных в час.

Примечание В некоторых конфигурациях задания на печать сохраняются после того, как поставлены в очередь. Это дает возможность пользователям повторно отправлять документ на принтер из очереди печати, а не из приложения. Если вы настроили принтер на сохранение заданий, вам может потребоваться более точно узнать, сколько данных в среднем печатается за час. Это поможет определить, сколько дискового пространства нужно для поддержки служб печати, и послужит хорошим показателем, который позволит определить, как часто следует удалять старые задания из очереди.

- **Number of jobs in queue (число заданий в очереди)** Показывает число заданий, стоящих в очереди и ожидающих печати. У нормально загруженных принтеров обычно несколько заданий в очереди, особенно в моменты пиковой нагрузки. Но если вы часто видите много заданий, ожидающих печати, принтер может быть перегружен. Как видно из предыдущего листинга, настроено три принтера, но используются только два из них: Minolta QMS magicolor и HP Business InkJet. Из этих двух принтеров Business InkJet ис-

пользуется чаще, и при наличии 24 заданий в очереди он может оказаться перегружен. Здесь вы можете улучшить ситуацию, сообщив пользователям о доступности более быстрого принтера magicolor или установив это устройство печати для некоторых пользователей как принтер по умолчанию. Вы также можете сообщить какой-то части сотрудников о принтере HP OfficeJet. Однако он больше похож на чей-то персональный принтер и, возможно, просто не используется в данный момент.

- **Average bytes per printer job (среднее число байтов в задании на печать)** Показывает средний размер печатаемых заданий. У большинства принтеров есть внутренняя память, и в идеале ее объем должен быть таким, чтобы в нее помещалось все задание. Средний размер заданий, печатаемых на принтере Minolta QMS magicolor, составляет около 84 Мб, и так как у принтера всего 32 Мб памяти, у вас есть повод для беспокойства. Возможно, вы захотите увеличить память на этом устройстве. Заметьте, что для определения объема установленной памяти вам может понадобиться страница конфигурации принтера (она нередко печатается самим принтером).
- **Printer up time (время работы принтера)** Показывает, сколько времени работает принтер, но самом деле имеется в виду время работы спулера и очереди печати.

Управление принтерами

Вы можете устанавливать и управлять принтерами из командной строки, используя утилиту Prnmngr. Prnmngr конфигурируется как сценарий. Поэтому, если вы впервые работаете с Windows-сценариями из командной строки или если в качестве основного хоста сценариев настроен WScript, вам придется настроить CScript как хост сценариев по умолчанию. Для этого введите **cscript //h:cscript //s** в командной строке. Затем вы сможете работать с хостом сценариев командной строки, а не с графической средой WScript. Учтите, что настройка основного хоста сценариев производится для каждого пользователя индивидуально. То есть, если вам нужно запускать сценарий от имени конкретного пользователя, не исключено, что для него CScript не является хостом сценариев по умолчанию, и тогда следует включать в сценарии строку **cscript //h:cscript //s**.

Основы управления принтерами

Команда Prnmngr позволяет работать как с *локальными устройствами печати* (физически подключенными к компьютеру и используемыми только тем сотрудником, который работает на данном компьютере), так и *сетевыми устройствами печати* (настроенными для удаленного доступа через сеть). Ключевое различие между локальным принтером и сетевым в том, что локальный принтер не является общим ресурсом. Открывая сетевой доступ к принтеру, вы используете компьютер для работы служб печати. Такой компьютер называется *сервером печати*.

Основная задача сервера печати — обеспечение сетевого доступа к устройству печати для совместного использования в сети и управление очередью заданий на печать. Сервер печати предоставляет централизованную очередь печати, которой легко управлять, и позволяет не устанавливать драйверы принтера на клиентских системах. Однако вы не обязаны использовать сервер печати. Пользователи могут подключаться к сетевому принтеру напрямую, и тогда сетевые принтеры управляются как локальные, подключенные к клиентскому компьютеру. В таком случае пользователи подключаются к принтеру самостоятельно, и на каждом компьютере создается своя, индивидуально управляемая очередь печати.

Устанавливая принтер на компьютере, вы на самом деле конфигурируете очередь печати так, чтобы через нее можно было передавать задания физическому устройству печати. Для установки или настройки принтеров нужны соответствующие привилегии администратора. То есть вы должны относиться к одной из групп: Administrators (Администраторы), Print Operators (Операторы печати) или Server Operators (Операторы сервера). Но подключение к принтеру и его использование не требует привилегий администратора — достаточно иметь подходящие права доступа.

Установка физически подключенных принтеров

Физически подключенными устройствами печати считаются принтеры, напрямую соединенные с компьютером; они могут быть настроены как локальные или сетевые устройства печати. Хотя локальные принтеры доступны только пользователям, зарегистрированным на данном компьютере, сетевые устройства доступны любым пользователям как общие ресурсы. Для

начала соедините принтер с компьютером параллельным, последовательным или USB-кабелем, а затем включите принтер. Если вы настраиваете сетевой принтер, этот компьютер должен выступать в роли сервера печати. Если принтер соответствует спецификации Plug-and-Play (PnP), простое его подключение инициирует процесс автоматической установки и конфигурирования при условии, что в системе кто-то зарегистрировался.

Вы можете установить локальный принтер вручную, используя команду Prnmngr и следующие параметры.

- **–A AddPrinter** — указывает, что вы хотите добавить или установить локальный принтер.
- **–P PrinterName** — назначает имя принтеру. Это имя, которое вы увидите в папке Printers And Faxes (Принтеры и факсы) или в командной строке.
- **–M PrinterModel** — задает модель принтера. Название модели определяет, какой драйвер принтера будет использоваться.
- **–R PrinterPort** — задает порт, к которому подключен принтер; таким портом может быть параллельный (LPT1:, LPT2: или LPT3:), последовательный (COM1:, COM2: или COM3:) или USB-порт (например USB001).

Примечание Если вы задаете названия принтера и модели, они будут отображаться в командной строке и диалоговых окнах. Хотя эти названия стоит набирать с учетом регистра букв, сами имена не чувствительны к нему, т. е. Windows воспринимает «centralcolorlaser» так же, как и «CentralColorLaser».

Для настройки физически подключенных принтеров не обязательно локально входить на компьютер. Такие принтеры можно настраивать и удаленно. Для этого используйте параметр –S, позволяющий указать имя удаленного компьютера, к которому вы хотите подключить локальный принтер. При необходимости включайте параметры –U и –W, чтобы задать имя пользователя и пароль для подключения к удаленному компьютеру.

Примечание При работе с локальной командной оболочкой нельзя указать имя пользователя и пароль, если вы подключились к этому компьютеру удаленно. Иначе вы получите ошибку. В Windows Server 2003 сообщение об этой ошибке может ввести в заблуждение: «User credentials cannot be used for connections». В Windows XP Professional то же сообщение выглядит более внятно: «User credentials cannot be used for local connections».

Чтобы понять, как используется команда Prnmngr, рассмотрим несколько примеров.

Настройка принтера HP 5500 Series InkJet, подключенного к USB001:

```
prnmngr -a -p "OfficeJetPrinter" -m "hp officejet 5500 series"
-r USB001
```

Настройка принтера HP 1100 DN Series InkJet, подключенного к LPT1:

```
prnmngr -a -p "BusinessJetPrinter" -m "hp businessjet 1100
series DN" -r LPT1
```

Настройка принтера Epson Stylus Photo на cdesign09 с использованием USB001:

```
prnmngr -a -p "PhotoPrinter" -m "epson stylus photo 1270 esc/p 2"
-r USB001 -s cdesign09
```

Настройка принтера Epson Stylus Color на mteam06 с использованием LPT1:

```
prnmngr -a -p "ColorPrinter" -m "epson stylus color esc/p 2"
-r LPT1: -s mteam06 -u wrstanek -w goldfish
```

Если принтер установлен успешно, Prnmngr сообщит «Added printer». В ином случае вы получите сообщение «Unable to add printer» с описанием ошибки. Наиболее частая ошибка — неверно указанная или неизвестная модель устройства, в результате чего Prnmngr сообщает, что драйвер принтера неизвестен. Убедитесь, что вы правильно ввели название модели.

Примечание Первый принтер, установленный на компьютер, считается принтером по умолчанию. Однако общий доступ к нему автоматически не открывается. Если вы хотите предоставить общий доступ к принтеру, чтобы его могли видеть и другие, см. раздел «Совместное использование принтеров» далее в этой главе.

Совет Вы можете создавать дополнительные принтеры для одного и того же устройства печати. Единственное требование заключается в том, что имя принтера и имя общего ресурса должны быть уникальны. Наличие дополнительных принтеров для одного и того же устройства печати позволит задавать разные значения свойств для разных потребностей. Например, вы сможете создать одну конфигурацию для низкоприоритетных заданий на печать и другую — для высокоприоритетных.

Установка сетевых принтеров

Сетевые принтеры подключаются непосредственно к сети через плату сетевого адаптера и обычно настраиваются как сетевые устройства печати, так что они доступны пользователям сети как общие ресурсы. Для начала подключите принтер к сети и присвойте ему подходящий IP-адрес или настройте его на получение IP-адреса от DHCP-сервера. При этом действуйте в соответствии с документацией на принтер.

Настроив TCP/IP на принтере, вам понадобится создать TCP/IP-порт на компьютере, который будет выполнять роль сервера печати для данного принтера. Через этот порт устанавливаются сетевые соединения с принтером. Затем вы можете добавить принтер так же, как и физически подключенное устройство печати. Единственное отличие — использование параметра –R для задания созданного вами TCP/IP-порта вместо LPT-, COM- или USB-порта. Например, если вы создали TCP/IP-порт с именем IP_192.168.10.15, то можете добавить принтер, использующий этот порт, следующей командной строкой:

```
prnmngr -a -p "CentralColorLaser" -m "magicolor 2300 dl"
-r IP_192.168.10.15 -s corpsvr03
```

Здесь вы устанавливаете принтер Minolta QMS magicolor так, чтобы он использовал TCP/IP-порт. Поскольку принтер настроен на сервере CorpSvr03, этот компьютер будет высту-

пать в роли сервера печати для данного устройства. Принтер не будет установлен как принтер по умолчанию для пользователей и не будет открыт для общего доступа. Если вы хотите предоставить принтер для общего доступа, см. раздел «Совместное использование принтеров» далее в этой главе.

Перечисление принтеров, настроенных на компьютере

Вы можете получить список всех принтеров, настроенных на локальном компьютере, введя **prnmngr –l**. Если вы хотите увидеть эту информацию об удаленном компьютере, добавьте параметр –S с указанием имени компьютера, например: **prnmngr –l –s corpsvr03**. При необходимости можно использовать и параметры –U и –W, чтобы задать имя пользователя и пароль учетной записи.

В выводе отображаются имя сервера печати (или пустая строка, если вы работаете на локальном компьютере) и другая важная информация о каждом настроенном принтере, например:

```
Server name corpsvr03
Printer name magicolor 2300 DL
Share name magicolo
Driver name magicolor 2300 DL
Port name hpbusinessinkjet1100
Comment Main printer for the fifth floor.
Location 5/ne
Print processor MIMFPR_B
Data type IMF
Parameters
Attributes 2633
Priority 1
Default priority 0
Status Idle
Average pages per minute 0

Number of printers enumerated 1
```

Имена принтера, драйвера и порта были заданы, когда принтер устанавливался. Принтер доступен для общего доступа всех пользователей домена. Если вы хотите переместить принтер на новый сервер печати, единственное, что вам следует запомнить, – имя драйвера, обычно совпадающее с названием модели принтера.

Просмотр и установка принтера по умолчанию

Вы можете определить принтер по умолчанию для текущего пользователя командой **prnmngr –g**. Чтобы сменить для пользователя принтер по умолчанию, наберите **prnmngr –t –p** с указанием имени принтера, который должен стать принтером по умолчанию, например:

```
prnmngr -t -p "magicolor 2300 DL"
```

Если смена пройдет успешно, Prnmngr сообщит, что принтер теперь установлен как принтер по умолчанию. В ином случае Prnmngr сообщит об ошибке. Ошибка «Not Found» обычно означает, что вы ввели неверное имя принтера.

Переименование принтеров

Переименование — одна из задач, связанных с принтерами, которую нельзя решить с помощью Prnmngr. В таких случаях используется Prncnfg — утилита из Windows Server 2003 Resource Kit. Синтаксис для переименования принтера:

```
prncnfg -x -p ТекущееИмяПринтера -z НовоеИмяПринтера
```

Здесь вы используете Prncnfg с параметром –X, указывая, что вам нужно переименовать принтер. Текущее имя принтера задается через параметр –P, а новое имя принтера — через параметр –Z, например:

```
prncnfg -x -p "CentralColorLaser" -z "EngineeringPrinter"
```

Если принтер существует, Prncnfg сообщит, что принтер переименован и присвоит ему новое имя.

Вы также можете переименовать принтер на удаленном компьютере. Параметр –S позволяет указать имя удаленного компьютера, например:

```
prncnfg -x -s corpsvr03 -p "CentralColorLaser"
-z "EngineeringPrinter"
```

Здесь вы переименовываете принтер на компьютере CorpSvr03. Однако эта команда не позволит выбрать другую учетную запись для входа.

Удаление принтеров

Prnmngr предоставляет два способа для удаления принтеров, которые больше не должны быть доступны на конкретном компьютере. Чтобы удалить принтер, используйте команду:

```
prnmngr -d -p ИмяПринтера
```

например, так:

```
prnmngr -d -p "magicolor 2300 DL"
```

Если вы указали неверное имя принтера, Prnmngr сообщит, что удалить принтер нельзя, потому что такой принтер не найден. Если у вас нет прав на удаление данного принтера, Prnmngr сообщит, что не может перечислить принтеры из-за недостаточных прав пользователя. Тогда войдите в систему под другой учетной записью, где есть соответствующие привилегии администратора. Заметьте, что это не относится к работе с удаленным компьютером. В этом случае вы можете указать учетную запись через параметры –U и –W, например:

```
prnmngr -d -p "magicolor 2300 DL" -s corpsvr03 -u wrstanek -p
goldfish
```

Вы можете удалить все принтеры на компьютере через параметр –X:

```
prnmngr -x
```

Prnmngr не запросит подтверждения, но сообщит о каждом удаляемом принтере, например:

```
Deleted printer OfficeJet
Deleted printer CentralPrinter

Number of local printers and connections enumerated 2
Number of local printers and connections deleted 2
```

Управление TCP/IP-портами для сетевых принтеров

TCP/IP-порты используются для установления соединений с сетевыми принтерами. Вы можете создавать TCP/IP-порты и управлять ими командой Prnport. Как и Prnmngr, Prnport — это Windows-сценарий, выполняемый с использованием хоста сценариев командной строки.

Создание и изменение TCP/IP-портов для принтеров

Создать TCP/IP-порт позволяет команда Prnport с параметром –A. В дальнейшем через параметр –R вы указываете имя порта, а через параметр –H — IP-адрес принтера. Обычно за основу имени берется IP-адрес принтера, к которому вы подключаетесь. Так, если вы настраиваете порт для принтера с IP-адресом 192.168.10.15, то можете присвоить порту имя IP_192.168.10.15.

Вы также должны указать протокол вывода для данного порта. Протокол вывода задается параметром –O и может быть **raw** или **lpr**. Большинство принтеров использует протокол Raw. В этом случае данные передаются принтеру без изменений через порт, указываемый по его номеру. По умолчанию это порт 9100. Вы можете назначить другой номер порта параметром –N. В случае протокола LPR порт используется в сочетании с LPD (Line Printer Daemon) очереди печати. Имя этой очереди задается параметром –Q.

Как и большинство команд конфигурирования принтеров, Prnport не требует локального входа на компьютер для настройки принтера. Если вы хотите сконфигурировать порт на удаленном компьютере, используйте параметр –S, чтобы задать нужный удаленный компьютер. При необходимости указывайте через параметры –U и –W имя пользователя и пароль для подключения к удаленному компьютеру. Имя пользователя может быть введено в виде домен\имя_пользователя, если домен входа отличается от текущего.

Рассмотрим несколько примеров.

Настройка порта на протокол TCP Raw и подключение к 192.168.10.15 через порт 9100:

```
prnport -a -r IP_192.168.10.15 -h 192.168.10.15 -o raw
```

Настройка порта на протокол Raw и подключение к 10.10.1.50 через порт 9500:

```
prnport -a -r IP_192.168.10.15 -h 10.10.1.50 -o raw -n 9500
```

Настройка порта на протокол LPR и подключение к 172.20.18.2. Указываем имя очереди LPRQUEUE:

```
prnport -a -r IP_192.168.10.15 -h 172.20.18.2 -o lpr -q
lprqueue
```

Настройка порта на CORPSVR03 на протокол TCP Raw и подключение к 192.168.10.15 через порт 9100:

```
prnport -a -r IP_192.168.10.15 -h 192.168.10.15 -o raw -s
corpsvr03
```

Если порт создан успешно, Prnport сообщит «Created/updated port». В ином случае команда вернет «Unable to create/update port» и сообщит, какая ошибка произошла.

Большинство сетевых принтеров также поддерживает Simple Network Management Protocol (SNMP). Чтобы разрешить принтеру использовать этот протокол, вы должны включить SNMP параметром –Me, а затем задать имя SNMP-сообщества через параметр –Y и SNMP-индекс устройства через параметр –I. Обычно имя сообщества устанавливается в *public*; это означает, что принтер доступен любому пользователю в сети. Индекс устройства определяет конкретное устройство в SNMP-сообществе. Первое устройство имеет индекс 1, второй — 2 и т. д.

Рассмотрим пример:

```
prnport -a -r IP_192.168.10.15 -h 192.168.10.15 -o raw -me -y
public -i 1
```

Примечание Вы можете отключить SNMP через параметр –Md.

В этом примере вы настраиваете порт на TCP Raw и подключение к 192.168.10.15 через порт 9100. Вы также включаете SNMP, настраиваете имя SNMP-сообщества как public и задаете индекс устройства равным 1.

Если вы впоследствии захотите изменить конфигурацию TCP/IP-порта, то можете сделать это командой Prnport с параметром –T. Параметр –R позволяет указать нужный порт, а прочие параметры — связанные с ним значения. Вот пример:

```
prnport -a -r MainPrinter -h 10.10.12.50 -o raw -md
```

Здесь вы указываете, что хотите изменить TCP/IP-порт с именем MainPrinter. Вы задаете IP-адрес равным 10.10.12.50, устанавливаете протокол вывода Raw и отключаете SNMP.

Вывод информации о TCP/IP-портах, используемых принтерами

Вы можете перечислить все TCP/IP-порты принтеров, настроенных на локальном компьютере, командой **prnport –l**. Если вы хотите просмотреть информацию для удаленного компьютера, добавьте параметр –S с указанием имени компьютера, например **prnport –l –s corpsvr03**. При необходимости укажите через параметры –U и –W имя пользователя и пароль учетной записи, применяемой для подключения.

В результате будет показано имя сервера печати (или пустая строка, если вы работаете с локальным компьютером) и другая информация о каждом настроенном порте. В следующем примере выводится информация для порта, использующего протокол RAW:

```
Server name
Port name IP_192.168.1.101
Host address 192.168.1.101
Protocol RAW
Port number 9100
SNMP Enabled
Community public
Device index 1
```

А вот пример, где выводится информация для порта, использующего протокол LPR:

```
Server name
Port name IP_192.168.1.101
Host address 192.168.1.101
Protocol LPR
Queue crownnet
Byte Count Enabled
SNMP Enabled
Community public
Device index 1
```

Примечание Информация о LPR-порте может отображаться некорректно, если включен подсчет байтов. При включении этого режима компьютер подсчитывает число байтов в документе, прежде чем отправить его на принтер. Большинство принтеров не требует подсчета байтов, и вообще этот процесс может заметно снизить скорость распечатки.

Удаление TCP/IP-портов, используемых принтерами

Вы можете удалить порты, используемые принтерами, по следующему синтаксису:

```
prnport -d -r ИмяПорта
```

например:

```
prnport -d -r IP_192.168.1.101
```

Если вы вводите неверное имя принтера, Prnport сообщает, что удалить порт нельзя, так как он не найден. Если у вас нет прав на удаление принтера, Prnport сообщит, что не может перечислить принтеры из-за недостаточных прав пользователя. Вам придется войти в систему под другой учетной записью, у которой есть необходимые привилегии администратора. Заметьте, что это не относится к работе с удаленным компьютером. В последнем случае вы можете указать учетную запись для подключения через параметры –U и –W, например:

```
prnport -d -r IP_192.168.1.101 -s corpsvr03 -u wrstanek -p
goldfish
```

Настройка свойств принтера

Вы можете просмотреть и настроить свойства принтера, используя Windows-сценарий Prncnfg с параметром –T. Независимо от того, с каким свойством вы работаете, Prncnfg ожидает, что вы укажете имя нужного принтера через параметр –P. Как и большинство команд конфигурирования принтеров, Prncnfg не требует локального входа на компьютер для настройки свойств принтера. Если вы хотите изменить свойства принтера на удаленном компьютере (но не переименовать его), то можете указать имя удаленного компьютера через параметр –S. При необходимости задайте через параметры –U и –W имя пользователя и пароль для подключения к удаленному компьютеру. Имя пользователя можно вводить в виде *домен\имя_пользователя*, если домен входа отличается от текущего.

Добавление комментариев и информации о местонахождении

Вы можете упростить пользователям выбор принтера, добавив комментарии и информацию о местонахождении принтеров. Комментарии дают общие сведения о принтере, например о типе устройства печати и о том, кто отвечает за него. Место-

нахождение описывает реальное физическое расположение устройства печати. После определения эти значения отображаются на вкладке General (Общие) в окне свойств принтера.

Синтаксис добавления комментариев и информации о расположении принтера таков:

```
prncnfg -t -p ИмяПринтера -m "Комментарий" -l "Местонахождение"
```

Здесь вы используете Prncnfg с параметром –T, который указывает, что нужно изменить свойства принтера. Далее задается текст комментария в параметре –M и информация о местонахождении принтера в параметре –L, например:

```
prncnfg -t -s corpsrv03 -p "CentralColorLaser" -m "Main
Engineering Printer" -l "5th Floor SE"
```

Команда Prncnfg должна сообщить о том, что принтер сконфигурирован. Если этого не произойдет, то, возможно, вы забыли двойные кавычки или один из параметров. Не обязательно указывать и комментарий, и местонахождение. Все это можно задать по отдельности.

Совместное использование принтеров

Принтеры, которые вы добавляете из командной строки, не предоставляются для совместного использования автоматически. Если вы хотите обеспечить совместный доступ к такому принтеру, то должны специально сконфигурировать его с помощью Prncnfg. Через параметр –T укажите, что вы устанавливаете или изменяете свойства принтера, а через параметр –P задайте нужный принтер. Затем воспользуйтесь параметром –H для определения имени общего ресурса и параметром +Shared для включения общего доступа. Для совместимости с компьютерами под управлением операционных систем версий до Windows 2000 имя общего ресурса должно быть длиной не более восьми символов и без пробелов.

В крупных организациях имя общего ресурса должно подсказывать, где находится принтер, чтобы пользователям не приходилось изучать его свойства. Например, если принтер расположен в юго-восточном (SE) углу пятого этажа, присвойте ему имя вроде FifthSE. Рассмотрим пример:

```
prncnfg -t -s corpsrv03 -p "CentralColorLaser" -h "FifthSE"
+shared
```

Здесь вы настраиваете лазерный принтер CentralColorLaser на сервере CorpSrv03, чтобы этот принтер был общим ресурсом с именем FifthSE.

Для отмены совместного использования служит параметр –Shared. В следующем примере отменяется совместное использование принтера, включенное в предыдущем примере:

```
prncnfg -t -s corpsrv03 -p "CentralColorLaser" -shared
```

Публикация принтеров в Active Directory

Для упрощения пользователям поиска доступных принтеров опубликуйте информацию о них в Active Directory. После этого пользователи смогут искать принтер, исходя из его местонахождения и возможностей. Настроить публикацию принтера позволяет команда Prncnfg. Если вы хотите опубликовать принтер в Active Directory, укажите через параметр –T, что вы устанавливаете или изменяете свойства принтера; параметр –P позволяет задать нужный принтер. Параметр +Published сообщает, что принтер должен быть опубликован, а параметр –Published — что информация о принтере должна быть удалена из каталога.

Рассмотрим пару примеров.

Публикация принтера CentralColorLaser на сервере CorpSrv03 в Active Directory:

```
Prncnfg -t -s corpsrv03 -p "CentralColorLaser" +published
```

Удаление локального принтера OfficeJet из Active Directory:

```
Prncnfg -t-p "OfficeJet" -published
```

В любом случае Prncnfg должна сообщить, что принтер сконфигурирован. Однако команда не сообщает об ошибке, если принтер уже был опубликован или удален.

Настройка страниц-разделителей и изменение режима печати

Страницы-разделители (separator pages) можно вставлять в начало заданий на печать, чтобы облегчить поиск документов на загруженном устройстве печати. С их помощью также можно изменять режим работы принтера, чтобы, например, пере-

ключить его на использование языка PostScript или Printer Control Language (PCL).

Страницы-разделители хранятся в папке *%SystemRoot%*-\System32. В Windows по умолчанию определены три страницы-разделителя:

- **pcl.sep** — переключает принтер в режим PCL и печатает страницу-разделитель перед каждым документом;
- **pscript.sep** — переключает принтер в режим PostScript, но не печатает страницу-разделитель;
- **sysprint.sep** — переключает принтер в режим PostScript и печатает страницу-разделитель перед каждым документом.

Команда Prncnfg позволяет указать, что принтер должен использовать одну из этих страниц-разделителей или любую другую, которая находится в папке *%SystemRoot%*\System32. Параметр –T сообщает, что вы устанавливаете или изменяете какое-то свойство принтера, а параметр –P определяет принтер, с которым вы работаете. Наконец, параметр –F задает страницу-разделитель.

Рассмотрим следующий пример:

```
Prncnfg -t -s corpsrv03 -p "CentralColorLaser" -f sysprint.sep
```

Здесь вы настраиваете принтер CentralColorLaser на сервере CorpSrv03 для использования страницы-разделителя sysprint.sep.

Чтобы прекратить использование страницы-разделителя, укажите параметр –F со значением " ", например:

```
Prncnfg -t -s corpsrv03 -p "CentralColorLaser" -f " "
```

Планирование заданий на печать и установка приоритетов

Prncnfg позволяет управлять приоритетами заданий на печать и планировать эти задания из командной строки. Задания всегда печатаются в порядке приоритета, где 1 — минимальный приоритет, а 99 — максимальный. Задания с более высоким приоритетом печатаются раньше заданий с более низким. Если с физическим устройством печати связано несколько принтеров (очередей печати), приоритет распространяется и на принтеры. Параметр –T указывает, что вы устанавливаете или изменяете какое-то свойство принтера, а параметр –P определя-

ет принтер, с которым вы работаете. Наконец, параметр –O определяет приоритет, например:

```
prncnfg -t -p "EngineeringPrinter" -o 50
```

Здесь вы указываете, что задания, использующие принтер EngineeringPrinter, имеют приоритет, равный 50. Например, если принтер MarketingPrinter сопоставлен с тем же физическим устройством печати, но имеет меньший приоритет, задания на печать, отправляемые через EngineeringPrinter, всегда будут печататься в первую очередь.

Принтеры могут быть доступны постоянно или только в определенные часы. Параметр –St указывает время, когда принтер становится доступен, а параметр –Ut — момент, после которого принтер становится недоступным. Время устанавливается в 24-часовом формате, например:

```
prncnfg -t -p "EngineeringPrinter" -st 0530 -ut 1930
```

Здесь вы сообщаете, что принтер доступен каждый день с 05:30 утра до 7:30 вечера.

Настройка спулинга и других дополнительных параметров принтера

В случае устройств печати, подключенных к сети, принтеры обычно буферизуют файлы, а не печатают их напрямую. Спулинг (буферизация и постановка в очередь) позволяет использовать принтер (очередь печати) для управления заданиями на печать. Спулинг настраивается следующими ключами команды Prncnfg.

- **+Direct** Документы буферизуются, и программа заканчивает печать быстрее, чем при прямой печати. Этот ключ действует по умолчанию.
- **–Direct** Печать напрямую, а не через буфер. Используйте –Direct, если вы не можете печатать, используя спулинг.
- **+Queued** Печать начинается после того, как закончен спулинг последней страницы. Выберите этот ключ, если хотите, чтобы весь документ попал в буфер до начала печати. Такой вариант гарантирует, что весь документ будет поставлен в очередь. Если по какой-то причине печать из программы отменяется или не заканчивается, задание на физическом устройстве не распечатывается.

- **–Queued** Печать начинается, как только начинается спулинг документа (при условии, что принтер не занят). Этот вариант рекомендуется, если вы хотите, чтобы задания на печать выполнялись быстрее или чтобы приложение как можно быстрее возвращало управление пользователю. Этот ключ действует по умолчанию.
- **+Enabledevq** Спулер проверяет настройки принтера и их соответствие параметрам документа, прежде чем отправить его на устройство печати. Если обнаруживается несоответствие, буфер приостанавливает это задание, но остальные документы (параметры которых соответствуют настройкам принтера) продолжают печататься. Этот ключ полезен, если вы часто изменяете формат страниц или тип бумаги.
- **–Enabledevq** Спулер не проверяет настройки принтера перед отправкой документа устройству печати. Если обнаруживается несовпадение, принтер обычно останавливает печать и ожидает от пользователя отмены задания, изменения формата страниц или вставки в лоток необходимого типа бумаги. Этот ключ действует по умолчанию.

Ниже рассматриваются дополнительные ключи Prncnfg.

- **+Keepprintedjobs** Задания не удаляются из очереди по окончании печати. Используйте этот ключ для документов, очень сложных в подготовке к печати.
- **–Keepprintedjobs** Задания удаляются из очереди по окончании печати. Это освобождает дисковое пространство, выделенное под задания, но лишает вас возможности повторно распечатать задание из очереди печати. Этот ключ действует по умолчанию.
- **+Docompletefirst** Задания, спулинг которых завершен, печатаются до заданий, спулинг которых еще не завершен (независимо от приоритета). Этот ключ действует по умолчанию.
- **–Docompletefirst** Задания с более высоким приоритетом, даже если их спулинг еще не завершен, печатаются до заданий с более низким приоритетом.
- **+Enablebidi** Включает спулинг метафайлов и дополнительные возможности печати (если они поддерживаются), например определение порядка страниц, печать буклетов или задание числа страниц на лист. Если в этом случае воз-

никают какие-то проблемы, отключите эту функцию. Данный ключ действует по умолчанию.

- **–Enablebidi** Отключает спулинг метафайлов и дополнительные возможности печати. Используйте этот ключ, если у вас возникают проблемы с данным принтером.

Чтобы лучше понять, как применяются эти ключи, рассмотрим несколько примеров.

Настройка SalesPrinter на сервере sales06 для печати напрямую и сохранения распечатанных заданий:

```
prncnfg -t -s sales06 -p "SalesPrinter" +direct
+keepprintedjobs
```

Настройка MainPrinter на локальном компьютере для начала печати после спулинга последней страницы:

```
prncnfg -t -p "MainPrinter" -queued
```

Настройка HPLaserJet на сервере corpsvr09 для приостановки обработки документов с неверными параметрами и отключения спулинга метафайлов:

```
prncnfg -t -s corpsvr09 -p "HPLaserJet" +enabledevq -enablebidi
```

Устранение проблем со спулингом

Windows использует службу Print Spooler (Диспетчер очереди печати) для управления спулингом заданий на печать. Если эта служба не запущена, спулинг заданий на печать невозможен.

Проверка службы Print Spooler

Вы можете проверить статус службы Print Spooler на локальном компьютере командой:

```
sc query type= service | find /v "x0"
```

Для удаленного компьютера нужно указать UNC-имя, например:

```
sc \\Engsvr04 query type= service | find /v "x0"
```

В любом случае в выводе должен появиться блок, относящийся к спулеру:

```
SERVICE_NAME: Spooler
DISPLAY_NAME: Print Spooler
```

```
TYPE               : 110 WIN32_OWN_PROCESS (interactive)
STATE              : 4 RUNNING
(STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
```

Это говорит о том, что служба Print Spooler работает. Если она остановлена, вам, видимо, стоит проверить ее конфигурацию. Для этого введите:

```
sc qc spooler
```

или

```
sc \\Engsvr04 qc spooler
```

Вы увидите стартовые параметры службы Print Spooler (Диспетчер очереди печати), как показано ниже:

```
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: spooler
    TYPE               : 110 WIN32_OWN_PROCESS (interactive)
    START_TYPE         : 2  AUTO_START
    ERROR_CONTROL      : 1  NORMAL
    BINARY_PATH_NAME   : C:\WINDOWS\system32\spoolsv.exe
    LOAD_ORDER_GROUP   : SpoolerGroup
    TAG                : 0
    DISPLAY_NAME       : Print Spooler
    DEPENDENCIES       : RPCSS
    SERVICE_START_NAME : LocalSystem
```

Параметр start_type должен иметь значение AUTO_START, указывающее, что Print Spooler запускается автоматически.

Коррекция поврежденного спулера

Спулеры иногда повреждаются. В этом случае вы увидите, что принтер остановился или что задания не передаются устройству печати. Бывает и так, что принтер печатает, но выдает страницы с какой-то мешаниной. Как правило, перезапуск службы Print Spooler решает эту проблему. Вы можете остановить Print Spooler, набрав:

```
sc stop spooler
```

После останова спулера перезапустите его, введя:

```
sc start spooler
```

Если вы работаете с удаленным компьютером, делайте то же самое, но указывайте UNC-имя этого компьютера, например:

```
sc \\Engsvr04 stop spooler

sc \\Engsvr04 start spooler
```

Другие службы, требующие проверки

Если это не решило проблему, проверьте зависимые службы, а также следующие службы печати (если они установлены):

- TCP/IP Print Server;
- Print Server for Macintosh;
- Print Server for Unix.

Примечание Иногда проблемы со спулингом могут быть вызваны неверными правами доступа. Так что проверьте и права доступа к принтеру.

Управление очередями печати и индивидуальными заданиями

Есть несколько Windows-сценариев, предназначенных для работы с очередями печати и содержащимися в них заданиями на печать. Утилита Prnqctl позволяет запускать, останавливать или приостанавливать печать всех документов в очереди. Для работы с заданиями на печать служит утилита Prnjobs.

Просмотр заданий в очереди

Вы можете просматривать задания в очередях командой Prnjobs. Если вы хотите увидеть все задания для всех принтеров на локальном компьютере, наберите **prnjobs –l**. Чтобы просмотреть задания для конкретного принтера, воспользуйтесь параметром –P и укажите имя принтера. Для удаленного компьютера задействуйте параметр –S, чтобы указать нужный удаленный компьютер, и при необходимости задайте через параметры –U и –W имя пользователя и пароль для доступа к этому компьютеру.

Рассмотрим пару примеров.

Просмотр всех заданий на печать на CorpSrv03:

```
prnjobs -l -s corpsrv03
```

Просмотр всех заданий на печать для принтера MainPrinter на локальном компьютере:

```
prnjobs -l -p MainPrinter
```

В выводе для индивидуального задания вы увидите:

- **Job ID** — идентификационный номер задания, необходимый, если вы хотите работать с отдельным заданием на печать;
- **Printer** — имя принтера;
- **Document** — имя файла документа, которое также может включать имя приложения, откуда было выдано задание на его печать;
- **Data Type** — тип данных принтера;
- **Driver Name** — имя драйвера печати, указывающее модель принтера;
- **Description** — описание принтера;
- **Elapsed Time** — время, в течение которого печатается документ;
- **Job Status** — состояние задания на печать (возможны состояния Printing, Spooling, Paused, Deleting и Restarting);
- **Notify** — лицо, которому будет сообщено об окончании печати (если настроена система оповещения);
- **Owner** — владелец документа;
- **Pages Printed** — число напечатанных страниц (если таковые есть);
- **Size** — размер документа в байтах;
- **Time Submitted** — время и дата отправки задания;
- **Total Pages** — общее число страниц в документе.

Приостановка принтера и возобновление печати

Иногда нужно приостановить принтер для работы с физическим устройством печати или устранения каких-либо проблем. Когда вы приостанавливаете принтер, он завершает текущее задание и приостанавливает остальные задания. Приостановить работу принтера позволяет команда Prnqctl. Для локального принтера введите **prnqctl -z**, используя параметр -P, чтобы задать имя нужного принтера. Для удаленного компьютера используйте параметр -S, чтобы указать нужный удален-

ный компьютер, и при необходимости задайте через параметры –U и –W имя пользователя и пароль для доступа к этому компьютеру.

Для возобновления печати всех документов в очереди замените параметр –Z параметром –M.

Рассмотрим несколько примеров.

Приостановка печати для EngineeringPrinter на сервере CorpSrv03:

```
prnqctl -z -s corpsrv03 -p EngineeringPrinter
```

Приостановка заданий на печать для 5thfloorPrinter на локальном компьютере:

```
prnqctl -z -p 5thfloorPrinter
```

Возобновление печати для EngineeringPrinter на сервере CorpSrv03:

```
prnqctl -m -s corpsrv03 -p EngineeringPrinter
```

Очистка очереди печати

Команда Prnqctl позволяет очистить очередь печати и удалить все ее содержимое. На локальном принтере наберите **prnqctl –x** и используйте параметр –P для указания имени принтера, очередь которого вы хотите очистить. Для удаленного компьютера задействуйте параметр –S, чтобы указать нужный удаленный компьютер, и при необходимости задайте через параметры –U и –W имя пользователя и пароль для доступа к этому компьютеру.

Рассмотрим пару примеров.

Очистка очереди печати для SalesPrinter на salespc06:

```
prnqctl -x -s salespc06 -p SalesPrinter
```

Очистка очереди печати для TempPrinter на локальном компьютере:

```
prnqctl -x -p TempPrinter
```

При удачном выполнении Prnqctl сообщит, что документы удалены из очереди печати. Такое сообщение появится, даже если очередь печати была пуста.

Приостановка, возобновление и перезапуск печати отдельных документов

Вы также можете приостановить или возобновить печать отдельных заданий. Приостанавливая печать задания, вы останавливаете печать данного документа и даете возможность печатать остальные. Возобновляя печать задания, вы сообщаете принтеру продолжить печать документа с той точки, на которой он остановился.

Для приостановки задания используйте следующий синтаксис:

```
prnjobs -z -p ИмяПринтера -j IDЗадания
```

где *ИмяПринтера* — это имя принтера, с которым вы работаете, а *IDЗадания* — идентификационный номер задания, которое нужно приостановить.

Для возобновления печати предназначен синтаксис:

```
prnjobs -m -p ИмяПринтера -j IDЗадания
```

где *ИмяПринтера* — это имя принтера, с которым вы работаете, а *IDЗадания* — идентификационный номер задания, печать которого должна быть продолжена.

В любом случае по умолчанию вы работаете с принтерами на локальном компьютере. Для очередей печати на удаленных компьютерах параметр –S позволяет указать нужный удаленный компьютер, а параметры –U и –W — имя пользователя и пароль для доступа к этому компьютеру.

Рассмотрим несколько примеров.

Приостановка задания номер 6 для EngineeringPrinter на сервере CorpSrv03:

```
prnjobs -z -s corpsrv03 -p EngineeringPrinter -j 6
```

Приостановка задания номер 17 для 5thfloorPrinter на локальном компьютере:

```
prnjobs -z -p 5thfloorPrinter -j 17
```

Возобновление печати задания номер 6 для EngineeringPrinter на сервере CorpSrv03:

```
prnjobs -m -s corpsrv03 -p EngineeringPrinter -j 6
```

Prnjobs должна сообщить об успешной приостановке или возобновлении печати задания. Если вы указали неверный номер задания, Prnjobs сообщит «unable to set the print job».

Удаление документа и отмена задания на печать

Prnjobs позволяет отменить отдельное задание на печать и удалить его из очереди. Для локального принтера введите **prnjobs –x** и через параметр –P задайте имя принтера, а через параметр **–j** — номер удаляемого документа. Для удаленного компьютера используйте параметр –S и при необходимости укажите через параметры –U и –W имя пользователя и пароль для доступа к этому компьютеру.

Отмена задания номер 12 для MainPrinter на локальном компьютере:

```
prnjobs -x -p MainPrinter -j 12
```

Отмена задания номер 9 для EngineeringPrinter на CorpSrv03:

```
prnjobs -x -s corpsrv03 -p EngineeringPrinter -j 9
```

Prnjobs должна сообщить об успешной отмене задания. Если вы указали неверный номер задания, Prnjobs сообщит «unable to set the print job».

Примечание Если документ уже печатается в момент отмены, устройство печати может продолжить печать всего документа или его части. Это вызвано тем, что у большинства принтеров имеется внутренний буфер, и принтер не останавливается, пока не распечатает содержимого своего кэша.

Глава 15

Настройка, поддержка и анализ проблем в TCP/IP-сетях

Настройка и поддержка сетей TCP/IP (Transmission Control Protocol/Internet Protocol), а также устранение неполадок в них — крайне важная часть работы любого администратора. Начнем с обсуждения средств командной строки, способных решать перечисленные задачи, а потом перейдем к углубленному рассмотрению каждой из этих задач, что даст вам знания и методики, нужные для успешного управления и поддержки TCP/IP-сетей в операционных системах Windows XP Professional и Windows Server 2003.

Использование оболочки сетевых сервисов

Оболочка сетевых сервисов (Netsh) — это утилита-сценарий командной строки, позволяющая управлять конфигурацией различных сетевых служб на локальном и удаленных компьютерах. Netsh предоставляет свою командную строку, которую можно использовать в интерактивном или неинтерактивном режиме.

Контексты Netsh

В интерактивном режиме вы входите в оболочку, набирая **netsh** и указывая имя контекста нужной сетевой службы. Имена контекстов и их смысл рассмотрены ниже.

- **aaaa** — аутентификация (authentication), авторизация (authorization), управление учетными записями (accounting) и аудит (auditing). Контекст для просмотра и работы с базой данных AAAA, которая используется Internet Authentication Service (IAS) (Служба проверки подлинности в Интер-

нете) и Routing And Remote Access Service (Служба маршрутизации и удаленного доступа).

- **dhcp** – Dynamic Host Configuration Protocol (DHCP). Контекст для просмотра и управления DHCP-серверами. Вы обычно используете контекст DHCP для динамического присваивания конфигурационных данных TCP/IP сетевым клиентам.
- **diag** – диагностика сети. Контекст для просмотра и устранения неполадок в параметрах сервиса сети.
- **interface ip** – IP-интерфейс. Контекст для просмотра и управления сетевой TCP/IP-конфигурацией компьютера. С появлением Windows XP Service Pack 2 вы сможете в этом контексте управлять IPv4 Internet Connection Firewall (ICF) (Брандмауэр Интернета IPv4).

Внимание! Если IPv4 ICF включен в Windows XP SP2 или более поздней версии, также включается сервис защиты периода загрузки (boot-time security). Этот сервис делает так, что компьютер в период загрузки может выполнять в сети лишь базовые задачи, связанные с DNS, DHCP и взаимодействием с контроллерами домена. Как только ICF запускается, эта служба загружает и применяет политики ICF периода выполнения, а затем отключает фильтры периода загрузки. Хотя управлять политикой безопасности периода загрузки нельзя, вы можете настроить то, как будет использоваться ICF. Для этого служит команда Netsh.

- **interface ipv6** – IP-интерфейс версии 6. Контекст для просмотра и управления сетевой конфигурацией IPv6 компьютера. При наличии Windows XP Advanced Networking Pack или Service Pack 2 (или более поздней версии) через этот контекст можно управлять и IPv6 Internet Connection Firewall.
- **interface portproxy** – Interface Port Proxy. Контекст для управления прокси между сетями IPv4 и IPv6.
- **ipsec** – Internet Protocol Security (IPsec) (IP-безопасность). Контекст для просмотра и настройки IPsec.
- **bridge** – сетевой мост. Контекст для включения или выключения режима совместимости транспортного уровня (уровень 3 в модели OSI) для сетевых мостов. Также при-

меняется для просмотра конфигурационных параметров сетевых мостов.

- **ras** — Remote Access Server (RAS) (Сервер удаленного доступа). Контекст для управления конфигурацией сервера удаленного доступа.
- **routing** — маршрутизация. Контекст для управления серверами маршрутизации. Используется со службой маршрутизации и удаленного доступа.
- **rpc** — Remote Procedure Call (RPC) Helper. Контекст для просмотра и управления параметрами IP-адресации, а также IP-адресами подсети, настроенными на компьютере.
- **wins** — Windows Internet Name Service (WINS). Контекст для просмотра и управления настройками WINS-сервера. WINS разрешает NetBIOS-имена в IP-адреса на компьютерах под управлением операционных систем Windows, предшествовавших Windows 2000.

Примечание Некоторые контексты и подконтексты доступны, только когда вы используете Netsh на локальном компьютере. Это прежде всего относится к RPC, доступному только локально. Кроме того, некоторые контексты Netsh и подкоманды требуют наличия настроенной службы маршрутизации и удаленного доступа, даже если вы работаете с локальным компьютером из командной строки. В этом случае вы должны установить политику удаленного доступа Connections To Other Access Servers, чтобы выдать разрешения на удаленный доступ; убедитесь, что служба удаленного доступа работает.

Имя контекста сообщает Netsh, какую вспомогательную DLL надо загрузить. Вспомогательная DLL (helper DLL) предоставляет команды, специфичные для контекста. Например, если вы набрали **netsh** для интерактивной работы с Netsh, а затем ввели **rpc**, то войдете в контекст RPC. После этого вы могли бы набрать **show interfaces** для просмотра IP-интерфейсов, настроенных на этом компьютере. Вот как выглядит эта процедура.

1. Введите **netsh**. Командная строка сменится на **Netsh>**.
2. Введите **rpc**. Командная строка сменится на: **Netsh rpc>**.
3. Введите **show interfaces**. Будут выведены сведения об IP-интерфейсах, настроенных на компьютере, например:

```
Subnet          Interface     Status    Description

127.0.0.0       127.0.0.1     Enabled   MS TCP Loopback interface

192.168.1.0     192.168.1.56  Enabled   Intel(R) PRO/100 VE
                                        Network Connection
```

Каждый контекст имеет свой набор доступных команд, и некоторые из этих команд вызывают подконтексты, у которых тоже есть собственные команды. Учтите, что для работы с каким-либо контекстом в домене должны быть сконфигурированы соответствующие службы. В любом контексте можно просмотреть список доступных команд, набрав **help**. То же самое относится к команде **quit**, введя которую вы покинете оболочку сетевых сервисов и вернетесь в командную строку Windows. Быстро работать с Netsh в интерактивном режиме не удается. Но это даже хорошо для начинающих или для изучения доступных команд.

Когда вы привыкнете к Netsh, вам захочется использовать ее в неинтерактивном режиме. Этот режим позволит вводить полную последовательность команд в командной строке или в сценарии. Например, предыдущая процедура, выполняемая в три этапа, может быть выполнена одной командной строкой:

```
netsh rpc show interfaces
```

Независимо от того, вставите вы эту строку в сценарий или введете ее прямо в командной строке, результат будет один: список интерфейсов компьютера. Как видите, неинтерактивный режим гораздо быстрее.

Работа с удаленными компьютерами

Netsh позволяет работать с удаленными компьютерами. Для операций в интерактивном режиме нужно запустить netsh с параметром -R и указать IP-адрес или доменное имя компьютера, к которому вы хотите подключиться, например:

```
netsh -r 192.168.10.15
```

или

```
netsh -r corpsvr02
```

Если вы работаете с удаленным компьютером, Netsh добавляет в приглашение командной строки IP-адрес или имя этого компьютера:

```
[corpsvr02] netsh>
```

Здесь вы используете Netsh для удаленной работы с CorpSvr02.

Для работы в неинтерактивном режиме с удаленным компьютером предназначен следующий синтаксис:

```
netsh -c Контекст -r УдаленныйКомпьютер Команда
```

где *Контекст* — это идентификатор нужного контекста, *УдаленныйКомпьютер* — имя или IP-адрес удаленного компьютера, а *Команда* — команда, которую нужно выполнить. Рассмотрим пример:

```
netsh -c "interface ip" -r corpsvr02 show ipaddress
```

В этом примере вы получаете список IP-адресов, настроенных на CorpSvr02, с применением контекста IP-интерфейса. Здесь контекст RPC не годится, так как он доступен только на локальном компьютере.

Примечание Netsh требует наличия в сети сконфигурированной службы маршрутизации и удаленного доступа. А точнее, вы должны настроить политику удаленного доступа Connections To Other Access Servers, чтобы предоставить разрешение на удаленный доступ. Также убедитесь, что служба удаленного доступа работает.

Работа с файлами сценариев

Как обсуждалось ранее, вы можете ввести полную последовательность команд Netsh в командной строке или в сценарии. Подвох в том, что вы должны знать полную командную строку, которую собираетесь использовать, и не можете положиться на подсказку Netsh. Некоторые командные строки могут быть очень длинными и сложными. Например, следующие команды выполняют подключение к DHCP-серверу, настраивают DHCP-область, а затем активизируют эту область:

```
netsh dhcp server \\corpsvr02 add scope 192.168.1.0
255.255.255.0 MainScope PrimaryScope

netsh dhcp server \\corpsvr02 scope 192.168.1.0 add iprange
192.168.1.1 192.168.1.254

netsh dhcp server \\corpsvr02 scope 192.168.1.0 add
excluderange 192.168.1.1 192.168.1.25
```

```
netsh dhcp server \\corpsvr02 scope 192.168.1.0 set state 1
```

Если вы сохраните эти команды в сценарий, то сможете запустить его, как любой другой сценарий. Например, присвоив сценарию имя dhcpconfig.bat, просто введите **dhcpconfig** для запуска этого сценария.

При работе с удаленным компьютером сценарий можно поместить в общую сетевую папку, доступную с удаленного компьютера, а затем удаленно подключиться к этому компьютеру для запуска сценария. Или скопировать сценарий непосредственно на удаленный компьютер и, подключившись к нему, удаленно запустить сценарий. Годятся оба способа, но они требуют дополнительных усилий. К счастью, есть более быстрый способ запустить сценарий на удаленном компьютере. Для этого вы должны немного изменить сценарий и использовать синтаксис:

```
netsh -c Контекст -r УдаленныйКомпьютер -f Сценарий
```

где *Контекст* — это идентификатор нужного контекста, *УдаленныйКомпьютер* — имя или IP-адрес удаленного компьютера, а *Сценарий* — файл или сетевой путь к запускаемому сценарию. Рассмотрим следующий пример:

```
netsh -c "dhcp server" -r corpsvr02 -f dhcpconfig.bat
```

В данном случае вы запускаете Netsh-сценарий dhcpconfig.bat на компьютере CorpSvr02, используя контекст DHCP Server. Заметьте, что Server — это подконтекст контекста DHCP. Сценарий содержит команды:

```
add scope 192.168.1.0 255.255.255.0 MainScope PrimaryScope
scope 192.168.1.0 add iprange 192.168.1.1 192.168.1.254
scope 192.168.1.0 add excluderange 192.168.1.1 192.168.1.25
scope 192.168.1.0 set state 1
```

Эти команды создают, настраивают и активизируют DHCP-область на указанном DHCP-сервере (CorpSvr02). Так как вы уже используете контекст DHCP Server на CorpSvr02, необходимости вводить **netsh dhcp server \\corpsvr02** в начале каждой команды нет.

Управление параметрами TCP/IP

Компьютеры используют IP-адреса для взаимодействия через TCP/IP. IP-адресацию можно настраивать вручную или дина-

мически из командной строки. При настройке вручную вы назначаете компьютеру статический IP-адрес. Статический IP-адрес фиксирован и не меняется, пока вы не смените его. При динамической настройке вы конфигурируете компьютер так, чтобы он получал свой IP-адрес от DHCP-сервера в сети. Этот IP-адрес назначается при запуске компьютера и может меняться. В Windows-доменах серверы используют статические IP-адреса, а рабочие станции — динамические.

Статический IP-адрес

Назначая статический IP-адрес, вы сообщаете системе, что она должна использовать именно этот IP-адрес, а также указываете маску подсети для этого IP-адреса и при необходимости основной шлюз (шлюз по умолчанию), используемый для межсетевых соединений. Настроив эти параметры IP, вы должны настроить и параметры разрешения имен через DNS (Domain Name System) и, возможно, через WINS.

Статический IP-адрес назначается в контексте Interface IP командной оболочки Netsh. Для этого используется команда SET ADDRESS с синтаксисом:

```
set address [name=]ИмяИнтерфейса source=static addr=IP-адрес
mask=МаскаПодсети [gateway={none | СтандартныйШлюз
[[gwmetric=]МетрикаШлюза]}
```

В большинстве случаев вы имеете дело с интерфейсом Local Area Connection (Подключение по локальной сети). Вы можете перечислить доступные интерфейсы, введя **netsh interface ip show interface** или просто **show interface** (если вы уже находитесь в контексте Interface IP). IP-адрес, присваиваемый компьютеру, должен быть уникальным в рамках вашей сети. Поле маски подсети гарантирует корректность работы компьютера в сети. Если сеть разбита на подсети, это значение может отличаться в каждом сегменте сети вашей организации. Если компьютеру нужен доступ к другим TCP/IP-сетям, Интернету или к другим подсетям, укажите основной шлюз. Используйте IP-адрес основного маршрутизатора сети.

Метрика шлюза сообщает относительную цену использования этого шлюза. Если к определенному IP-адресу ведет несколько маршрутов, первым используется шлюз с минимальной ценой. Если компьютер не может связаться с первым шлюзом, Windows Server 2003 попытается задействовать шлюз со

следующей наименьшей метрикой. Windows Server 2003 не назначает автоматически метрику для шлюза. Вы должны сделать это сами.

Рассмотрим следующий пример:

```
set address name="Local Area Connection" source=static
addr=192.168.1.50 mask=255.255.255.0 gateway=192.168.1.1
gwmetric=1
```

Здесь вы указываете, что работаете с интерфейсом Local Area Connection, и задаете статический IP-адрес 192.168.1.50 с сетевой маской 255.255.255.0. Основной шлюз — 192.168.1.1 с метрикой, равной 1.

Совет Чтобы проверить, сохранены ли только что настроенные вами параметры, введите в командной строке **netsh interface ip show address** или просто **show address** (если вы уже находитесь в контексте Interface IP).

Динамический IP-адрес

Вы можете назначить динамический IP-адрес любому из сетевых адаптеров компьютера при условии, что в сети доступен DHCP-сервер. После этого IP-адрес будет назначаться DHCP-сервером. Так как динамический IP-адрес может меняться, он не годится для серверов под управлением Windows Server 2003.

Вы назначаете динамический IP-адрес, используя контекст Interface IP в командной оболочке Netsh. Для этого предназначена команда SET ADDRESS с синтаксисом:

```
set address name=ИмяИнтерфейса source=dhcp
```

Примечание Если у компьютера уже настроен IP-адрес, применение SET ADDRESS приводит к замене существующих настроек. Если вы хотите что-то добавить, а не заменить, воспользуйтесь командой ADD ADDRESS.

Рассмотрим следующий пример:

```
set address name="Local Area Connection" source=dhcp
```

Здесь вы имеете дело с контекстом Interface IP и указываете, что хотите установить динамический IP-адрес для интерфейса Local Area Connection.

Добавление IP-адресов и шлюзов

В системах Windows XP Professional и Windows Server 2003 может быть несколько IP-адресов, даже если у компьютера только один сетевой адаптер. Несколько IP-адресов полезны, когда вы хотите, чтобы один компьютер выглядел в сети как несколько компьютеров, или когда ваша сеть разделена на подсети и компьютеру нужен доступ к этим подсетям для маршрутизации или предоставления других межсетевых сервисов.

Примечание Учтите, что при использовании одного сетевого адаптера IP-адреса должны принадлежать одному сегменту или сегментам, которые являются частью одной логической сети. Если ваша сеть состоит из нескольких физических сетей, вы должны использовать несколько сетевых адаптеров, где каждому адаптеру будет назначен IP-адрес в своем физическом сегменте.

Назначить несколько IP-адресов и шлюзов одному сетевому адаптеру позволяет команда ADD ADDRESS в контексте Interface IP командной оболочки Netsh. Синтаксис этой команды похож на синтаксис SET ADDRESS:

```
add address [name=]ИмяИнтерфейса addr=IP-адрес
mask=МаскаПодсети [[gateway=]СтандартныйШлюз
[gwmetric=]МетрикаШлюза]
```

Вот пример:

```
add address name="Local Area Connection" addr=192.168.2.12
mask=255.255.255.0 gateway=192.168.2.1 gwmetric=1
```

Примечание Указывая шлюз, укажите и метрику шлюза. Как и раньше, вы можете проверить настройки, введя **show address**.

Здесь вы указываете, что работаете с интерфейсом Local Area Connection (Подключение по локальной сети) и добавляете IP-адрес 192.168.2.12 с сетевой маской 255.255.255.0. Основной шлюз для этого IP-адреса — 192.168.2.1, а метрика равна 1.

Настройка DNS-серверов

DNS позволяет определять IP-адрес компьютера по его хостимени и наоборот. Для компьютеров со статическими IP-адресами нужно указать, к какому DNS-серверу они должны обращаться; вы можете сделать это, используя контекст Interface

IP в Netsh. Синтаксис для задания конкретного DNS-сервера таков:

```
set dns name=ИмяИнтерфейса source=static addr=АдресDNS
```

Рассмотрим следующий пример:

```
set dns name="Local Area Connection" source=static
addr=192.168.1.56
```

Здесь вы указываете, что работаете с интерфейсом Local Area Connection (Подключение по локальной сети) и задаете адрес DNS-сервера — 192.168.1.56.

Если компьютер использует DHCP и вы хотите, чтобы он получал адрес DNS-сервера через DHCP, введите команду вида:

```
set dns name=ИмяИнтерфейса source=dhcp
```

Вот пример:

```
set dns name="Local Area Connection" source=dhcp
```

Здесь вы указываете, что интерфейс Local Area Connection должен получать адрес DNS-сервера через DHCP.

Примечание Если компьютеру уже заданы IP-адреса DNS-серверов, SET DNS заменяет существующие значения. Чтобы добавить IP-адреса DNS-серверов, а не заменить их, пользуйтесь командой ADD DNS. Для проверки параметров DNS введите **show dns**.

Ниже рассматриваются другие, необязательные параметры:

- **ddns= enabled | disabled** — по умолчанию все IP-адреса для интерфейсов регистрируются в DNS с полным доменным именем компьютера. При такой автоматической регистрации используется протокол динамического обновления DNS. Если вы хотите отключить такую функцию, добавьте параметр **ddns=disabled**;
- **suffix= interface | primary** — по умолчанию полное имя компьютера регистрируется только в его первичном домене. При использовании динамического DNS вы также можете указать, что в DNS должно регистрироваться DNS-имя, специфичное для интерфейса. Введите параметр **suffix= interface**. Если у компьютера несколько сетевых адаптеров, это позволит подключаться к нескольким доменам.

Задание дополнительных DNS-серверов

В большинстве сетей несколько DNS-серверов, применяемых для разрешения доменных имен. Благодаря этому процесс разрешения имен не нарушается, даже если один из DNS-серверов становится недоступен. Когда вы указываете DNS-серверы через DHCP, он автоматически сообщает компьютерам о других DNS-серверах. Однако этого не происходит, если вы указываете DNS-сервер вручную.

Чтобы сообщить компьютеру о других DNS-серверах (в дополнение к основному DNS-серверу, указанному ранее), можно задействовать контекст Interface IP в Netsh и команду ADD DNS. Ее синтаксис выглядит так:

```
add dns name=ИмяИнтерфейса addr=АдресDNS
```

Вот пример:

```
add dns name="Local Area Connection" addr=192.168.1.75
```

Здесь вы указываете, что работаете с интерфейсом Local Area Connection (Подключение по локальной сети), и назначаете альтернативный DNS-сервер с IP-адресом 192.168.1.75.

По умолчанию DNS-сервер добавляется в конец списка таких серверов в конфигурации TCP/IP. Если вы хотите поместить DNS-сервер в определенную позицию списка, задайте параметр Index=. Например, если вам нужно, чтобы дополнительный сервер был первым в списке (т. е. стал основным), присвойте ему индекс, равный 1, например:

```
add dns name="Local Area Connection" addr=192.168.1.75 index=1
```

Задание WINS-сервера

WINS обеспечивает разрешение NetBIOS-имен в IP-адреса. Вы можете задействовать WINS, чтобы компьютеры в сети определяли адреса машин с операционными системами версий до Windows 2000. Хотя WINS поддерживается всеми версиями Windows, Windows Server 2003 в основном использует WINS для обратной совместимости.

Для компьютеров со статическими IP-адресами вы должны сами указать, к каким WINS-серверам им следует обращаться. В контексте Interface IP командной оболочки Netsh для задания WINS-сервера применяется следующий синтаксис:

```
set wins name=ИмяИнтерфейса source=static addr=АдресWINS
```

Рассмотрим пример:

```
set wins name="Local Area Connection" source=static
addr=192.168.1.64
```

Здесь вы указываете, что работаете с интерфейсом Local Area Connection, и задаете адрес WINS-сервера как 192.168.1.64.

Если компьютер использует DHCP и вы хотите, чтобы он получал адрес WINS-сервера через DHCP, введите команду вида:

```
set wins name=ИмяИнтерфейса source=dhcp
```

Вот пример:

```
set wins name="Local Area Connection" source=dhcp
```

Здесь вы указываете, что интерфейс Local Area Connection должен получать адреса WINS-серверов через DHCP.

Примечание Если IP-адреса WINS-серверов уже заданы, команда SET WINS заменит существующие значения. Для добавления, а не замены IP-адреса WINS-сервера используйте команду ADD WINS. Вы можете проверить настройки WINS, набрав **show wins**.

Задание дополнительных WINS-серверов

В большинстве сетей имеются первичный и резервный WINS-серверы. Это обеспечивает разрешение имен, когда один из WINS-серверов становится недоступен. Если вы указываете WINS-серверы через DHCP, он автоматически сообщает компьютеру о других WINS-серверах. Но это не распространяется на тот случай, когда вы задаете WINS-сервер вручную.

Чтобы сообщить компьютеру о других WINS-серверах в дополнение к основному, указанному ранее, можно использовать контекст Interface IP командной оболочки Netsh и команду ADD WINS. Ее синтаксис выглядит так:

```
add wins name=ИмяИнтерфейса addr=АдресWINS
```

Пример:

```
add wins name="Local Area Connection" addr=192.168.1.155
```

Здесь вы работаете с интерфейсом Local Area Connection (Подключение по локальной сети) и определяете для него альтернативный WINS-сервер с IP-адресом 192.168.1.155.

По умолчанию WINS-сервер добавляется в конец списка таких серверов в конфигурации TCP/IP. Если вы хотите поместить WINS-сервер в определенную позицию списка, используйте параметр Index=. Например, чтобы дополнительный сервер оказался в списке первым (т. е. стал основным), присвойте ему индекс 1, например:

```
add wins name="Local Area Connection" addr=192.168.1.155 index=1
```

Удаление кэша ARP

Когда компьютеры обращаются к информации DNS, найденные сопоставления имен и адресов временно сохраняются в кэше ARP (Address Resolution Protocol), чтобы в следующий раз при обращении к той же информации не выполнять поиск заново. Эта информация устаревает в соответствии со значением TTL (Time-To-Live), устанавливаемом при ее получении, и по окончании срока жизни такая информация должна быть обновлена. После получения новой информации устанавливается новое значение TTL. В общем случае эта автоматическая система получения, очистки и обновления информации о сопоставлениях имен и адресов работает хорошо. Но иногда устаревшая информация успевает вызвать проблемы до того, как она будет сброшена. Так, если на каком-то компьютере изменяется DNS-имя, а значение TTL еще не обнулилось, вы временно лишаетесь возможности найти этот компьютер.

У администраторов DNS есть в запасе несколько трюков, с помощью которых можно уменьшить негативное влияние смены имен, например задание меньшего значения TTL перед сменой имени, чтобы старая информация удалялась быстрее и не вызывала проблем. Однако вы можете обнаружить, что легче просто избавиться от старых данных и заставить компьютер вновь просматривать информацию в DNS. Для этого введите **netsh interface ip delete arpcache** в командной строке или **delete arpcache** (если контекст Interface IP в Netsh уже установлен). В результате будет удалена информация о сопоставлениях имен и адресов для всех интерфейсов, настроенных на данном компьютере. Если у вас несколько интерфейсов и вы хотите сбросить информацию лишь для одного интерфейса, ука-

жите нужный интерфейс через **name=ИмяИнтерфейса**, например:

```
delete arpcache name="Local Area Connection"
```

Удаление параметров TCP/IP

В контексте Interface IP также можно удалить конфигурационные параметры TCP/IP. Соответствующие команды перечислены в табл. 15-1.

Табл. 15-1. Команды Interface IP в Netsh для удаления параметров TCP/IP

Операция	Синтаксис	Пример
Удаление назначенного IP-адреса в заданном интерфейсе	delete address name=*ИмяИнтерфейса* addr=*IP-адрес*	delete address name="Local Area Network" address=192.168.1.56
Удаление статического IP-адреса шлюза в указанном интерфейсе	delete address name=*ИмяИнтерфейса* gateway=*АдресШлюза*	delete address name="Local Area Network" gateway=192.168.1.1
Удаление всех статических IP-адресов шлюзов в указанном интерфейсе	delete address name=*ИмяИнтерфейса* gateway=all	delete address name="Local Area Network" gateway=all
Удаление DNS-сервера в указанном интерфейсе	delete dns name=*ИмяИнтерфейса* addr=*IP-адрес*	delete dns name="Local Area Network" address=192.168.1.56
Удаление всех DNS-серверов в указанном интерфейсе	delete dns name=*ИмяИнтерфейса* addr=all	delete dns name="Local Area Network" address=all
Удаление WINS-сервера в указанном интерфейсе	delete wins name=*ИмяИнтерфейса* addr=*IP-адрес*	delete wins name="Local Area Network" address=192.168.1.56
Удаление всех WINS-серверов в указанном интерфейсе	delete wins name=*ИмяИнтерфейса* addr=all	delete wins name="Local Area Network" address=all

Поддержка TCP/IP-сетей

Оболочка Netsh предоставляет два контекста для работы с TCP/IP. Для просмотра статистики TCP/IP и изменения параметров служит контекст Interface IP, а для диагностики проблем TCP/IP — контекст Diag. Применение этих контекстов предполагает, что необходимые сетевые компоненты TCP/IP уже установлены на данном компьютере. Если сетевые компоненты TCP/IP не установлены, установите их, как показано в главе 16 книги «Microsoft Windows Server 2003. Справочник администратора».

Получение и сохранение конфигурации TCP/IP

Если вы уже какое-то время работаете с Windows, то, наверное, знаете, что ввод команды **ipconfig** в командной строке позволяет получить базовые сведения о конфигурации IP в Windows, например:

```
Windows IP Configuration

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix:
   IP Address:                192.168.1.50
   Subnet Mask:               255.255.255.0
   Default Gateway:           192.168.1.1
```

Как видите, здесь сообщается IP-адрес, маска подсети и основной шлюз (шлюз по умолчанию), используемый Ethernet-адаптером Local Area Connection (Подключение по локальной сети). Если вас интересуют детали, наберите **ipconfig /all** для вывода дополнительной информации, включающей физический (MAC) адрес адаптера, состояние DHCP, задействованные DNS-серверы и сведения о хосте, например:

```
Windows IP Configuration

   Host Name:                 salespc09
   Primary Dns Suffix:        cpandl.com
   Node Type:                 Unknown
   IP Routing Enabled:        No
   WINS Proxy Enabled:        No

Ethernet adapter Local Area Connection:
```

```
Connection-specific DNS Suffix:
Description:                    Linksys LNE100TX Fast Ethernet
Physical Address:               EA-BF-C2-D4-EF-12
Dhcp Enabled:                   Yes
Autoconfiguration Enabled:      Yes
IP Address:                     192.168.1.35
Subnet Mask:                    255.255.255.0
Default Gateway:                192.168.1.1
DHCP Server:                    192.168.1.50
Lease Obtained:                 Sunday, January 18, 2006 1:25 PM
Lease Expires:                  Monday, January 26, 2006 1:25 PM
```

Здесь компьютер с полным DNS-именем salespc09.cpandl.com настроен на использование DHCP и имеет IP-адрес 192.168.1.35 с маской подсети 255.255.255.0. Так как IP-адрес назначен динамически, у него есть специфические параметры — Lease Obtained и Lease Expiration.

Набрав **netsh interface ip show config** в командной строке, вы получите похожую, хотя и сокращенную, информацию о конфигурации, например:

```
Configuration for interface "Local Area Connection"
    DHCP enabled:                         No
    IP Address:                           192.168.1.50
    SubnetMask:                           255.255.255.0
    Default Gateway:                      192.168.1.50
    GatewayMetric:                        1
    InterfaceMetric:                      0
    Statically Configured DNS Servers:    192.168.1.56
    Statically Configured WINS Servers:   None
    Register with which suffix:           Primary only
```

Вот и все способы получения сходной информации о параметрах TCP/IP.

Netsh также позволяет сохранить конфигурацию IP, чтобы впоследствии ее можно было восстановить простым запуском Netsh-сценария. Если вы хотите сохранить настройки IP в файл, наберите:

```
netsh interface ip dump > ИмяФайла
```

где *ИмяФайла* — это имя файла, в который вы хотите сохранить информацию о конфигурации IP. Содержимое этого файла будет аналогично приведенному на листинге 15-1.

Листинг 15-1. Сценарий настройки IP

```
# ------------------
# Interface IP Configuration
# ------------------
pushd interface ip

# Interface IP Configuration for "Local Area Connection"

set address name="Local Area Connection" source=static
addr = 192.168.1.50 mask=255.255.255.0
set address name="Local Area Connection" gateway=192.168.1.1
gwmetric=1
set dns name="Local Area Connection" source=static
addr=192.168.1.56 register = PRIMARY
set wins name ="Local Area Connection" source=static addr=none

popd
# End of interface IP configuration
```

Листинг 15-1 — это Netsh-сценарий, запускаемый так:

```
netsh -c "interface ip" -f ИмяФайла
```

Вот пример:

```
netsh -c "interface ip" -f corpsvr02-ipconfig.txt
```

Здесь вы запускаете Netsh-сценарий corpsvr02-ipconfig.txt, используя контекст Interface IP для применения конфигурации IP, определенной в данном сценарии. Одна из основных причин создания такого дампа — резервное копирование конфигурации IP. Если конфигурация будет некорректно изменена в будущем, вы всегда сможете восстановить исходную конфигурацию из сценария.

Проверка IP-адресов и конфигурации интерфейсов

Контекст Interface IP в Netsh предоставляет несколько команд для просмотра IP-адресов и конфигураций интерфейсов. Здесь термин «интерфейс» относится к сетевому адаптеру, используемому компьютером для передачи данных через TCP/IP. На большинстве компьютеров два интерфейса: замкнутый на себя (local loopback interface) и Local Area Connection (Подключение по локальной сети).

Замкнутый на себя интерфейс — это псевдоинтерфейс с адресом 127.0.0.1 и сетевой маской 255.0.0.0. Все IP-сообще-

ния, посылаемые через этот интерфейс, возвращаются обратно компьютеру и не передаются по сети.

Интерфейс Local Area Connection создается автоматически при установке поддержки TCP/IP-сетей. Для каждого сетевого адаптера создается один такой интерфейс. По умолчанию первый интерфейс называется Local Area Connection, второй — Local Area Connection 2 и т. д.

В командной строке Windows можно увидеть информацию о конфигурации IP-адресов, введя **netsh interface ip show ipaddress**. Вывод должен быть похож на:

```
MIB-II IP Address Entry
IP Address      Mask            BC Fmt  Reasm Sz    Interface
--------        ----            ------  --------    ---------
127.0.0.1       255.0.0.0       1       65535       Loopback
192.168.1.50    255.255.255.0   1       65535       Local Area Connection
```

Примечание Отключенные интерфейсы не перечисляются, поскольку они недоступны и их нельзя настроить.

Здесь выводятся IP-адрес и маска для каждого интерфейса, настроенного на компьютере. Значение в столбце Reasm Sz указывает размер восстановленных IP-дейтаграмм — 65 535 байтов. То есть IP-дейтаграммы, отправляемые или получаемые через этот интерфейс, имеют размер 65 535 байтов. Однако блоки данных обычно не передаются порциями по 65 535 байтов. Вместо этого они делятся на фрагменты, которые восстанавливаются получателем в IP-дейтаграмму. Мы рассмотрим фрагментацию IP-дейтаграмм чуть позже.

Для просмотра более детальных сведений об интерфейсах, введите **netsh interface ip show interface**. В результате вы увидите подробную конфигурацию каждого интерфейса, например:

```
MIB-II Interface Information
---------------------------
Index:                  65539
User-friendly Name:     Local Area Connection
GUID Name:              {BG333345-F234-4335-25FB-43D3456B4464}
Type:                   Ethernet
MTU:                    1500
Speed:                  100000000
Physical Address:       EA-BF-C2-D4-EF-12
Admin Status:           Up
Operational Status:     Operational
```

```
Last Change:              583112798
In Octets:                396173
In Unicast Packets:       1323
In Non-unicast Packets:377
In Packets Discarded:     0
In Erroneous Packets:     0
In Unknown Protocol
Packets:                  0
Out Octets:               926667
Out Unicast Packets:      1351
Out Non-unicast
Packets:                  134
Out Packets Discarded:    0
Out Erroneous Packets:    0
Output Queue Length:      0
Description:              Intel(R) PRO/100 VE Network Connection
```


Примечание Как и раньше, отключенные интерфейсы не перечисляются, поскольку они недоступны и их нельзя настроить.

Совет Все команды SHOW контекста Interface IP принимают параметр Rr=, задающий интервал обновления в секундах. Например, чтобы статистика интерфейса автоматически обновлялась каждые 30 секунд, введите **netsh interface ip show interface rr=30**. Установив частоту обновления, вы можете нажать Ctrl+C для выхода из командной строки, и тогда никаких обновлений больше не будет.

Здесь интерфейс Local Area Connection настроен для работы с сетью Ethernet. Максимальный размер передаваемого пакета (MTU) в Ethernet составляет 1500 байтов при инкапсуляции Ethernet II. То есть размер каждого передаваемого блока составляет 1500 байтов, где 20 байт используются для IP-заголовка. Оставшиеся 1480 байтов являются собственно данными (полезной нагрузкой). Таким образом, передача IP-дейтаграммы длиной 65 535 байтов требует ее фрагментации на множество блоков данных меньшего размера. Эти фрагменты затем должны быть восстановлены на узле-получателе.

Вы можете воспользоваться и несколькими другими важными параметрами.

- **Speed** — скорость передачи, используемая интерфейсом. В нашем случае это 100 000 000 (100 Мбит/с).

- **Physical Address** — физический (MAC) адрес, «зашитый» в сетевой адаптер. MAC-адрес служит для отслеживания данных, передаваемых конкретным устройством в IP-сетях.
- **Operational Status** — состояние интерфейса. К сожалению, он обычно обозначается как Operational, даже если в интерфейсе возникли проблемы. Например, если сетевой кабель отключен, интерфейс не сообщит об этом как об ошибке.

В детальной информации об интерфейсе есть полезные сведения о типе полученных (in packets) и отправленных пакетов (out packets). Как правило, пакеты либо одноадресные (unicast), либо нет. Одноадресные пакеты передаются или принимаются по конкретному IP-адресу. Пакеты, отличные от одноадресных, передаются или принимаются по нескольким IP-адресам и являются либо групповыми (многоадресными) (multicast), либо широковещательными (broadcast). При возникновении ошибок маршрутизации или доставки на пути передачи или в точке назначения компьютер отбрасывает проблемные дейтаграммы и протоколирует это действие. Входящие пакеты, созданные неизвестным сетевым протоколом, помечаются как Unknown Protocol Packets, а пакеты, в которых обнаружены ошибки общего характера, помечаются как erroneous.

Протокол ICMP и соответствующие сообщения

Каждый пакет, переданный по IP, является дейтаграммой, т. е. сообщение пересылается маршрутизаторами на IP-адрес получателя без подтверждения и строгой последовательности. Каждый маршрутизатор, получающий дейтаграмму, сам решает, как лучше переслать ее. Это означает, что разные дейтаграммы могут идти разными путями между IP-адресом отправителя (узлом-источником) и IP-адресом получателя (узлом-приемником). А значит, маршрут возврата дейтаграмм тоже может быть разным.

Хотя IP обеспечивает доставку IP-дейтаграмм от отправителя получателю, в него не заложено никаких механизмов, которые сообщали бы о возникших ошибках маршрутизации или доставки. Ошибки и управляющие сообщения отслеживаются протоколом ICMP (Internet Control Message Protocol). Вы можете просмотреть статистику ICMP, набрав **netsh interface ip show icmp**. Вывод этой команды выглядит примерно так:

```
MIB-II ICMP Statistics
--------------------------
INPUT
Messages:                   20302
Errors:                     120
Destination Unreachable:    45
Time Exceeded:              88
Parameter Problems:         0
Source Quench:              4
Redirects:                  6
Echo Requests:              966
Echo Replies:               966
Time Stamp Requests:        0
Time Stamp Replies:         0
Address Mask Requests:      0
Address Mask Replies:       0

OUTPUT
Messages:                   20302
Errors:                     120
Destination Unreachable:    45
Time Exceeded:              88
Parameter Problems:         0
Source Quench:              4
Redirects:                  6
Echo Requests:              966
Echo Replies:               966
Time Stamp Requests:        0
Time Stamp Replies:         0
Address Mask Requests:      0
Address Mask Replies:       0
```

Совет Эту статистику можно обновлять автоматически. Добавьте параметр **Rr=*ЧастотаОбновления***, где *ЧастотаОбновления* — интервал обновления в секундах.

Здесь вы видите детальную статистику по полученным IP-дейтаграммам (входящим сообщениям) и переданным (исходящим сообщениям). Разбор этой статистики несложен, если вы знаете, что искать. Самый элементарный тип IP-дейтаграмм — Echo Request (Эхо-запрос) и Echo Reply (Эхо-ответ). Он используется для передачи простого сообщения IP-узлу; приняв сообщение Echo Request, получатель возвращает отправителю Echo Reply. Многие команды, связанные с TCP/IP, используют Echo Request и Echo Reply, чтобы предоставить информацию о доступности IP-узла назначения и пути к нему.

IP обеспечивает доставку дейтаграмм к IP-узлу назначения по принципу «максимум возможного» (best effort)*. Если возникает ошибка маршрутизации или доставки, маршрутизатор или узел назначения отбрасывает проблемную дейтаграмму и пытается сообщить об ошибке, посылая сообщение Destination Unreachable (Адресат недоступен).

Значение TTL задается в IP-дейтаграмме перед отправкой. Оно отражает максимальное число переходов (hops) на пути между источником и конечным получателем. Когда значение TTL в дейтаграмме обнуляется, она отбрасывается, а отправителю посылается сообщение Time exceeded. Обычно это означает, что между отправителем и получателем больше переходов, чем предполагалось. Тогда для успешной передачи трафика между узлами отправителя и получателя нужно увеличить значение TTL. Нулевое значение TTL также может указывать на наличие в сети петель маршрутизации (routing loops). Такие петли появляются, когда информация на маршрутизаторах некорректна и IP-дейтаграммы пересылаются так, что никогда не достигают адресата.

Если возникает проблема при обработке IP-заголовка в IP-дейтаграмме, маршрутизатор или узел-получатель отправляет сообщение Parameter Problem. Ошибка в IP-заголовке приводит к тому, что IP-дейтаграмма отбрасывается, и, если нет более подходящего ICMP-сообщения для описания возникшей ошибки, отправителю посылается сообщение Parameter Problem. Обычно это указывает на некорректный формат IP-заголовка или неправильные аргументы в полях.

При перегрузке маршрутизатора из-за неожиданного роста трафика, замедления канала связи или неадекватных ресурсов он будет отбрасывать входящие IP-дейтаграммы. При этом маршрутизатор может посылать отправителю сообщения Source Quench (Замедление источника), указывающие, что дейтаграммы прибывают быстрее, чем могут быть обработаны. Узел-получатель может посылать отправителю сообщения Source Quench по аналогичным причинам. Однако этого не делается для каждой отбрасываемой дейтаграммы. В RFC 1812 рекомендуется вообще не посылать сообщения Source Quench, так как их передача создает дополнительный трафик в уже перегружен-

* Стандартный уровень обслуживания в большинстве IP-сетей. Пакеты передаются без гарантии малых задержек или адекватной полосы пропускания. — *Прим. перев.*

ных соединениях. Однако, если таковое сообщение поступает, отправитель повторяет передачу соответствующего TCP-сегмента с меньшей скоростью, чтобы избежать перегрузки.

Когда используются подсети, первая часть IP-адреса не годится для определения маски подсети. Чтобы выяснить маску подсети, IP-узел отправляет сообщение Address Request известному маршрутизатору или использует широковещательную рассылку по всем подсетям либо ограниченную широковещательную IP-рассылку. Маршрутизатор отвечает сообщением Address Reply с маской подсети для сетевого сегмента, в котором было получено сообщение Address Request. Если IP-узел не знает свой IP-адрес, он может указать в сообщении Address Request адрес отправителя как 0.0.0.0; тогда получатель этого сообщения отвечает широковещательным сообщением.

Перед отправкой данных по TCP получатель сообщает, какой объем данных он может принять единовременно. Это значение называется размером TCP-окна. При передаче данных размер этого окна определяет, сколько данных будет передано до того, как отправитель станет ждать подтверждения приема от получателя. Размер TCP-окна — это 16-битное поле, т. е. максимальный размер TCP-окна может быть 65 535 байтов. Это означает, что узлу-отправителю не удастся отправить больший объем данных единовременно. При использовании механизма масштабирования TCP-окна получатель может указать размер окна примерно до 1 Гб.

Для расчета таймаута повторной передачи (retransmission time-out, RTO) TCP отслеживает время полного цикла передачи (round-trip time, RTT) между TCP-сегментами. Обычно значение RTO рассчитывается однократно для каждого отправляемого окна данных. В большинстве сетевых сред этот подход работает хорошо и предотвращает лишние повторные передачи данных. Однако в широкополосных сетях или из-за длительных задержек в любой среде такой подход теряет свою эффективность. Однократного расчета для каждого окна недостаточно для корректного определения текущего значения RTO и предотвращения лишних передач данных.

Чтобы иметь возможность расчета RTT и RTO в любом TCP-сегменте, в сообщении Timestamp Request передается временная метка, основанная на показаниях локальных часов. В подтверждении приема данных это значение возвращается обратно, что позволяет рассчитать RTT по временной метке и

времени возврата ответа. Эти сообщения обозначаются как Timestamp Request и Timestamp Reply.

Последний тип ICMP-сообщений, важных при устранении неполадок, — Redirect (Перенаправление). Он используется, чтобы сообщать отправителям IP-дейтаграмм об оптимальном маршруте от источника к адресату. Так как большинство хостов поддерживает минимальные таблицы маршрутизации, эта информация используется для повышения эффективности маршрутизации, уменьшения времени передачи и числа ошибок. Обнаружив сообщения Redirect, знайте, что ваш трафик перенаправляется к узлу назначения.

Анализ фрагментации, восстановления и детальных сведений об ошибках

Чтобы детальнее изучить фрагментацию и восстановление IP-дейтаграмм, введите **netsh interface ip show ipstats**. Результат будет выглядеть примерно так:

```
MIB-II IP Statistics
---------------------------
Forwarding is:              Enabled
Default TTL:                128
In Receives:                24219
In Header Errors:           0
In Address Errors:          250
Datagrams Forwarded:        0
In Unknown Protocol:        0
In Discarded:               0
In Delivered:               23969
Out Requests:               20738
Routing Discards:           0
Out Discards:               0
Out No Routes:              0
Reassembly Timeouts:        60
Reassembly Required         0
Reassembled Ok:             0
Reassembly Failures:        0
Fragments Ok:               0
Fragments Failed:           0
Fragments Created:          0
```

Как видите, статистика IP показывает значение TTL по умолчанию для исходящих пакетов, созданных на этом компьютере для передачи. В данном случае TTL равен 128. Это означает, что между этим компьютером и получателем может

быть до 128 переходов. Если пакету придется преодолевать большее число переходов, пакет будет отброшен и отправитель получит сообщение Time Exceeded.

Значение In Receives показывает, сколько входящих пакетов было получено. Реальное число пакетов отражается значением In Delivered, и разница между этими значениями возникает из-за следующих входящих пакетов:

- полученных с ошибками, указанными в строке In Header Errors или In Address Errors;
- пересылаемых другим IP-узлам и указываемым в строке Datagrams Forwarded;
- использующих неизвестный протокол и указываемым в строке In Unknown Protocol;
- отброшенных, например из-за превышения порогового значения TTL, и указываемым в строке In Discarded.

В данном примере разница между значениями In Receives и In Delivered составляет 250 дейтаграмм (из-за ошибок адресации в 250 входящих пакетах).

Число переданных пакетов сообщается в строке Out Requests. Любые ошибки, возвращаемые как результат передачи, записываются в соответствии с типом. Если маршрутизатор или другой узел возвращает сообщение Destination Unreachable (Адресат недоступен), оно обычно фиксируется как Routing Discard. Сообщения о других ошибках вроде Parameter Problem или Source Quench могут быть сохранены как Routing Discards или Out Discards. Если маршрут не найден, возвращается сообщение No Route, и такие сообщения могут быть записаны как Out No Routes.

Когда данные передаются извне локальной сети через маршрутизаторы, они обычно фрагментированы и восстанавливаются, как было описано ранее. Также регистрируется статистика восстановления исходных дейтаграмм и фиксируется состояние принятых фрагментов.

Анализ текущих TCP- и UDP-соединений

Брандмауэры и прокси-серверы способны повлиять на возможность подключения к системам в локальной или удаленной сетях. Обычно администратор должен открывать TCP- или UDP-порты, чтобы разрешить соединения между компьютером в локальной сети и удаленным компьютером или се-

тью. Каждый тип приложений или утилит, используемых вами, может потребовать открытия разных портов. Полный список TCP- и UDP-портов, используемых общеизвестными службами, хранится в \%*SystemRoot*%\System32\Drivers\Etc\Services.

Однако иногда нужной вам утилите не сопоставлена общеизвестная служба, и придется экспериментировать, чтобы определить, с какими TCP- или UDP-портами эта утилита работает. Один из способов выяснить это — запустить ее и использовать слушатели TCP или UDP, чтобы определить, какие порты активны в данный момент.

Работа с TCP

Доступ к TCP-портам открывается пассивно, поэтому обычно говорят, что порт доступен для приема запросов. Когда клиент хочет использовать доступный порт, он должен попытаться установить соединение. TCP-соединение — это двухстороннее соединение между двумя клиентами с использованием протоколов прикладного уровня в IP-сети. Конечные точки TCP идентифицируются парой «IP-адрес — TCP-порт». Существуют локальная конечная точка TCP и удаленная, с помощью которых можно идентифицировать замкнутые на себя соединения на локальной машине, а также стандартные соединения между локальной и удаленной машинами. TCP-соединения устанавливаются с применением трехэтапного согласования. Вот как выглядит эта схема.

1. Первый клиент, которому нужно задействовать некий порт, посылает запрос на открытие порта (SYN).
2. Второй клиент подтверждает запрос, отправляя SYN-ACK.
3. Первый клиент посылает финальное подтверждение (ACK).

Данные, передаваемые по TCP-соединению, делятся на сегменты. Сегменты отправляются как IP-дейтаграммы с TCP-заголовком и TCP-данными. При установлении соединения также устанавливается максимальный размер сегмента (maximum segment size, MSS). Обычно максимальное значение для MSS равно 65 495 байтам, т. е. из 65 535 байтов IP-дейтаграммы следует вычесть минимальные размеры IP- и TCP-заголовков (по 20 байтов). С технической точки зрения, сообщения SYN, SYN-ACK и ACK — это сегменты SYN, SYN-ACK и ACK.

Вы можете увидеть статистику текущего TCP-соединения, введя **netsh interface ip show tcpconn**. Для автоматического обновления статистики укажите параметр Rr=*ЧастотаОбновления*. Результат покажет вам, какие TCP-порты прослушиваются, по каким TCP-портам установлены соединения и какие порты находятся в состоянии ожидания:

```
MIB-II TCP Connection Entry

Local Address  Local Port Remote         Address    Remote PortState
-------------------------------------------------------------------
0.0.0.0        42         0.0.0.0        18520      Listen
0.0.0.0        53         0.0.0.0        16499      Listen
0.0.0.0        88         0.0.0.0        45165      Listen
0.0.0.0        135        0.0.0.0        2176       Listen
0.0.0.0        389        0.0.0.0        2256       Listen
0.0.0.0        1025       0.0.0.0        43054      Listen
0.0.0.0        1026       0.0.0.0        35016      Listen
0.0.0.0        1028       0.0.0.0        53398      Listen
0.0.0.0        3069       0.0.0.0        43189      Listen
0.0.0.0        3268       0.0.0.0        43230      Listen
0.0.0.0        3269       0.0.0.0        36957      Listen
127.0.0.1      389        127.0.0.1      1033       Established
127.0.0.1      389        127.0.0.1      1034       Established
127.0.0.1      389        127.0.0.1      1035       Established
127.0.0.1      389        127.0.0.1      1039       Established
127.0.0.1      1033       127.0.0.1      389        Established
127.0.0.1      1034       127.0.0.1      389        Established
127.0.0.1      1035       127.0.0.1      389        Established
127.0.0.1      1039       127.0.0.1      389        Established
127.0.0.1      3073       0.0.0.0        10251      Listen
192.168.1.50   135        192.168.1.56   1040       Listen
192.168.1.50   139        0.0.0.0        12369      Listen
192.168.1.50   389        192.168.1.50   3287       Established
192.168.1.50   3287       192.168.1.50   389        Established
192.168.1.50   3289       192.168.1.50   135        Wait
192.168.1.50   290        192.168.1.50   1025       Wait
```

Записи 0.0.0.0 представляют широковещательные TCP-пакеты, а записи 127.0.0.1 — локальные замкнутые на себя порты, используемые локальным компьютером. Вы также видите записи для физических IP-адресов, задействованных для замкнутых соединений локального компьютера. В данном случае это записи, где локальный и удаленный IP-адреса совпадают и равны 192.168.1.50. Вам наиболее интересны те записи, где

удаленный IP-адрес отличается от локального, — это соединения с другими системами и сетями.

Столбцы Local Port и Remote Port показывают сопоставление локальных TCP-портов с удаленными. В данном случае локальный порт 135 по IP-адресу 192.168.1.50 связан с удаленным портом 1040 по IP-адресу 192.168.1.56. Кроме того, у каждого TCP-соединения есть такой параметр, как состояние. Наиболее распространенные значения этого параметра приведены в табл. 15-2.

Табл. 15-2. Состояния TCP-соединения

Состояние	Описание
Closed	TCP-соединение закрыто
Listen	Протокол прикладного уровня вызвал функцию пассивного открытия, чтобы разрешить запросы на входящие соединения по указанному порту. Это не создает TCP-трафик
Syn Sent	Клиент, использующий протокол прикладного уровня, вызвал функцию активного открытия (SYN), что привело к созданию и передаче первого сегмента в процессе трехэтапного согласования TCP
Syn Rcvd	Клиент, использующий протокол прикладного уровня, получил SYN и отправил подтверждение (SYN-ACK)
Established	Получено финальное подтверждение ACK, и установлено TCP-соединение. Данные можно передавать в обоих направлениях
Wait	TCP-соединение завершено, и это подтверждено как локальным, так и удаленным клиентом (FIN-ACK)

Для просмотра дополнительной статистики TCP введите **netsh interface ip show tcpstats**. Результат должен быть примерно таким:

```
MIB-II TCP Statistics
---------------------------
Timeout Algorithm:          Van Jacobson's Algorithm
Minimum Timeout:            300
Maximum Timeout:            120000
Maximum Connections:        Dynamic
Active Opens:               381
Passive Opens:              443
Attempts Failed:            0
Established Resets:         26
```

```
Currently Established:          25
In Segments:                    27852
Out Segments:                   27518
Retransmitted Segments:         611
In Errors:                      0
Out Resets:                     4
```

Дополнительная статистика TCP показывает:

- задействованные минимальный и максимальный периоды ожидания;
- общее число активных и пассивных открытий портов с момента запуска программного обеспечения TCP/IP на данном компьютере;
- любые соединения, попытка установления которых закончилась неудачей;
- любые соединения, которые были установлены, а затем сброшены (reset);
- число соединений, установленных в данный момент;
- количество отправленных (in segments) и полученных (out segments) TCP-сегментов;
- количество повторно отправленных сегментов;
- количество сегментов, полученных с ошибками (in errors).

Работа с UDP

В отличие от TCP, ориентированного на логические соединения, UDP не использует такие соединения, а значит, доставка UDP-сообщений не гарантируется. UDP-порты отделены от TCP-портов, даже если их номера совпадают. Так как UDP-сообщения отправляются неупорядоченно и без согласования, они ненадежны в отличие от TCP, который очень надежен. Работая с UDP, вы имеете дело лишь с парами «локальный адрес — локальный порт», которые представляют прослушиваемые порты. Для просмотра таких портов введите **netsh interface ip show udpconn**. Вот пример вывода:

```
MIB-II UDP Listener Entry
Local Address       LocalPort
-------------       ----------
0.0.0.0             42
0.0.0.0             445
0.0.0.0             500
0.0.0.0             1030
```

```
0.0.0.0          1032
0.0.0.0          1701
0.0.0.0          3002
0.0.0.0          3103
0.0.0.0          3114
0.0.0.0          4500
127.0.0.1        53
127.0.0.1        123
127.0.0.1        1036
127.0.0.1        3101
127.0.0.1        3102
192.168.1.50     53
192.168.1.50     67
192.168.1.50     137
192.168.1.50     138
192.168.1.50     389
192.168.1.50     464
192.168.1.50     2535
```

Записи 0.0.0.0 представляют широковещательные UDP-пакеты, а записи 127.0.0.1 — локальные замкнутые на себя порты, используемые локальным компьютером. Записи с физическим IP-адресом сетевого адаптера сообщают порты, по которым прослушиваются пакеты.

UDP-сообщения передаются как IP-дейтаграммы и состоят из UDP-заголовка и UDP-данных. Чтобы увидеть дополнительную статистику UDP, введите **netsh interface ip show udpstats**. Результат должен быть примерно таким:

```
MIB-II UDP Statistics
---------------------
In Datagrams:              42640
In Invalid Port:           732
In Erroneous Datagrams:    20
Out Datagrams:             72217
```

Дополнительная статистика UDP показывает:

- общее число дейтаграмм, принятых через UDP-порты;
- количество дейтаграмм, отброшенных из-за их адресации к неправильным портам;
- число ошибочных дейтаграмм, принятых и отброшенных;
- количество дейтаграмм, переданных через UDP-порты.

Выявление и устранение проблем в TCP/IP-сетях

Докопаться до причины проблем в TCP/IP-сетях бывает весьма непросто. Вот почему так много инструментов для анализа того, что происходит в TCP/IP-сети. Прежде чем приступать к анализу проблем, вы должны хорошо освоить концепции и процедуры, обсуждавшиеся в разделе «Поддержка TCP/IP-сетей» ранее в этой главе. Инструменты и методики, рассмотренные там, помогут вам выявить и диагностировать некоторые из наиболее сложных проблем, возникающих в TCP/IP-сети. Этот раздел посвящен анализу проблем, связанных с поддержкой соединений и конфигурированием.

Просмотр диагностической информации

Многие проблемы с TCP/IP-сетями вызваны неверной настройкой сетевых компонентов, и вы обнаружите, что контекст Diag в Netsh действительно помогает выяснить, в чем дело. Начнем с просмотра сводной информации о конфигурации командой **netsh diag show all**. Такие сведения о конфигурации CorpSvr02 показаны на листинге 15-2.

Примечание Не упускайте из виду полезность Netsh в решении проблем с удаленными компьютерами. Netsh избавляет от необходимости сидеть за компьютером пользователя или подключаться к нему через Remote Desktop (Удаленный рабочий стол). Достаточно запустить Netsh с параметром –R, в котором вы указываете имя нужного удаленного компьютера.

Листинг 15-2. Вывод команды netsh diag show all

```
Default Outlook Express Mail (pop3.cpandl.com / mail.cpandl.com)

Default Outlook Express News (Not Configured)

Internet Explorer Web Proxy (Internet Explorer is not using
the proxy)

Loopback (127.0.0.1)

Computer System (CORPSVR02)

Operating System (Microsoft(R) Windows(R) Server 2003,
Standard Edition)
```

```
Version (5.2.3790)

Modems

Network Adapters
   1. [00000001]   Intel(R) PRO/100 VE Network Connection
   2. [00000002]   1394 Net Adapter
   3. [00000003]   RAS Async Adapter
   4. [00000004]   WAN Miniport (L2TP)
   5. [00000005]   WAN Miniport (PPTP)
   6. [00000006]   WAN Miniport (PPPOE)
   7. [00000007]   Direct Parallel
   8. [00000008]   WAN Miniport (IP)
Network Clients
   1. Microsoft Terminal Services
   2. Microsoft Windows Network
   3. Web Client Network
```

Вы также можете получить детальные сведения о конфигурации командой **netsh diag show all /v**. Но обычно это дает слишком много информации, так что лучше исследовать одну проблему за раз. Потом вам, как правило, потребуется проверить настройки сетевого адаптера на компьютере. Для просмотра сводной информации о сетевых адаптерах, настроенных на компьютере, введите **netsh diag show adapter**. В результате вы увидите, какие адаптеры доступны, например:

```
Network Adapters
   1. [00000001]   Intel(R) PRO/100 VE Network Connection
   2. [00000002]   1394 Net Adapter
   3. [00000003]   RAS Async Adapter
   4. [00000004]   WAN Miniport (L2TP)
   5. [00000005]   WAN Miniport (PPTP)
   6. [00000006]   WAN Miniport (PPPOE)
   7. [00000007]   Direct Parallel
   8. [00000008]   WAN Miniport (IP)
```

В данном случае у компьютера следующая сетевая конфигурация.

1. Сетевая плата Intel Ethernet на 100 Мбит/с.
2. Адаптер IEEE 1394 (FireWire).
3. Асинхронный адаптер RAS (это означает, что на компьютере установлен RAS).
4. IPSec-порт для L2TP (Layer Two Tunneling Protocol).

5. IPSec-порт для PPTP (Point-to-Point Tunneling Protocol).
6. IPSec-порт для PPP (Point-to-Point Protocol) через Ethernet.
7. Параллельный порт принтера.
8. IP-порт (Internet Protocol).

Затем вам, вероятно, понадобится детальная информация о конфигурации этих адаптеров, для чего добавьте параметр /V. Обычно следует ограничиться конкретным адаптером, указав в команде индекс проверяемого адаптера или его полное/частичное название. Рассмотрим несколько примеров.

Вывод детальных сведений о сетевом адаптере с индексом 1:

```
netsh diag show adapter 1 /v
```

Вывод детальных сведений о конфигурации сетевого адаптера, название которого начинается со слова «Intel»:

```
netsh diag show adapter intel* /v
```

Пример вывода детальных сведений о сетевом адаптере представлен на листинге 15-3. Как видите, здесь показываются конфигурация основных шлюзов, параметры динамического назначения IP-адреса через DHCP, а также параметры DNS и WINS.

Листинг 15-3. Детальный вывод команды netsh diag show adapter

```
Network Adapters
   1. [00000001] Intel(R) PRO/100 VE Network Connection
      ArpAlwaysSourceRoute = (empty)
      ArpUseEtherSNAP = (empty)
      Caption = [00000001] Intel(R) PRO/100 VE Network
Connection
      DatabasePath = %SystemRoot%\System32\drivers\etc
      DeadGWDetectEnabled = (empty)
      DefaultIPGateway =  192.168.1.1 Same Subnet
         192.168.1.2 Same Subnet
      DefaultTOS = (empty)
      DefaultTTL = (empty)
      Description = Intel(R) PRO/100 VE Network Connection
      DHCPEnabled = FALSE
      DHCPLeaseExpires = (empty)
      DHCPLeaseObtained = (empty)
      DHCPServer = (empty)
      DNSDomain = (empty)
```

```
DNSDomainSuffixSearchOrder = (empty)
DNSEnabledForWINSResolution = FALSE
DNSHostName = corpsvr02
DNSServerSearchOrder = 192.168.1.50
                       192.168.1.67
DomainDNSRegistrationEnabled = FALSE
ForwardBufferMemory = (empty)
FullDNSRegistrationEnabled = TRUE
GatewayCostMetric = 1
                    2
IGMPLevel = (empty)
Index = 1
InterfaceIndex = 65539
IPAddress =  192.168.1.50
             192.168.2.12
IPConnectionMetric = 20
IPEnabled = TRUE
IPFilterSecurityEnabled = FALSE
IPPortSecurityEnabled = (empty)
IPSecPermitIPProtocols = 0
IPSecPermitTCPPorts = 0
IPSecPermitUDPPorts = 0
IPSubnet =   255.255.255.0
             255.255.255.0
IPUseZeroBroadcast = (empty)
IPXAddress = (empty)
IPXEnabled = FALSE
IPXFrameType = (empty)
IPXMediaType = (empty)
IPXNetworkNumber = (empty)
IPXVirtualNetNumber = (empty)
KeepAliveInterval = (empty)
KeepAliveTime = (empty)
MACAddress = 00:E0:B8:53:05:F1
MTU = (empty)
NumForwardPackets = (empty)
PMTUBHDetectEnabled = (empty)
PMTUDiscoveryEnabled = (empty)
ServiceName = E100B
SettingID = {A908BB00-F027-4E25-8EE8-47FD6E7DA507}
TcpipNetbiosOptions = 0
TcpMaxConnectRetransmissions = (empty)
TcpMaxDataRetransmissions = (empty)
TcpNumConnections = (empty)
TcpUseRFC1122UrgentPointer = (empty)
TcpWindowSize = (empty)
```

```
    WINSEnableLMHostsLookup = TRUE
    WINSHostLookupFile = (empty)
    WINSPrimaryServer = 192.168.1.102
    WINSScopeID = (empty)
    WINSSecondaryServer = 192.168.1.108
```

Проблемы с клиентами почты, новостей и прокси

На листинге 15-2 первая строка показывает, что почтовой программой по умолчанию является Outlook Express и что она настроена на серверы pop3.cpandl.com и mail.cpandl.com (соответственно для приема и передачи почты):

```
Default Outlook Express Mail (pop3.cpandl.com / mail.cpandl.com)
```

Вы можете получить эту информацию отдельно, набрав **netsh diag show mail**. Если вы подозреваете проблему в конфигурации электронной почты, просмотрите более детальные сведения о конфигурации командой **netsh diag show mail /v**. Вот пример:

```
Default Outlook Express Mail (pop3.cpandl.com /
mail.cpandl.com)
   InBoundMailPort = 110
   InBoundMailServer = pop3.cpandl.com
   InBoundMailType = POP3
   OutBoundMailPort = 25
   OutBoundMailServer = mail.cpandl.com
   OutBoundMailType = SMTP
```

Здесь входящая почта настроена на протокол POP3 по порту 110; при этом pop3.cpandl.com является сервером входящей почты. Исходящая почта настроена на протокол SMTP по порту 25, а mail.cpandl.com — сервер исходящей почты. Если что-то из этой информации неправильно, перенастройте почту.

На листинге 15-2 после сведений о параметрах почты по умолчанию, выводится конфигурация клиентов Usenet и Web-прокси Internet Explorer. Если эти клиенты не сконфигурированы, вы сразу же заметите это. Если ваша организация использует такие клиенты, введите **netsh diag show news /v** или **netsh diag show ieproxy /v**; это должно помочь выявить любые проблемы в конфигурации.

Проблемы с базовой конфигурацией компьютера

Контекст Diag в командной оболочке Netsh предоставляет три диагностические команды для анализа проблем с базовой конфигурацией компьютера:

- **netsh diag show computer** — показывает базовую информацию о конфигурации компьютера;
- **netsh diag show os** — показывает базовую информацию о конфигурации операционной системы;
- **netsh diag show version** — выводит номер версии операционной системы, например 5.1.2600, где 5.1 является номером версии, а 2600 — номер сборки.

В сводных сведениях эти команды сообщают лишь имя компьютера, редакцию и версию операционной системы. Детальный вывод более полезен в диагностике. Пример такой информации о компьютере, полученной с помощью команды **netsh diag show computer /v**, представлен на листинге 15-4.

Листинг 15-4. Подробная информация о конфигурации компьютера

```
Computer System (CORPSVR02)
    AdminPasswordStatus = 3
    AutomaticResetBootOption = TRUE
    AutomaticResetCapability = TRUE
    BootOptionOnLimit = (empty)
    BootOptionOnWatchDog = (empty)
    BootROMSupported = TRUE
    BootupState = Normal boot
    Caption = CORPSVR02
    ChassisBootupState = 3
    CreationClassName = Win32_ComputerSystem
    CurrentTimeZone = 480
    DaylightInEffect = FALSE
    Description = AT/AT COMPATIBLE
    DNSHostName = corpsvr02
    Domain = cpandl.com
    DomainRole = 5
    EnableDaylightSavingsTime = TRUE
    FrontPanelResetStatus = 3
    InfraredSupported = FALSE
    InitialLoadInfo = (empty)
    InstallDate = (empty)
    KeyboardPasswordStatus = 3
```

```
    LastLoadInfo = (empty)
    Manufacturer = Gateway
    Model = Gateway 800EA2
    Name = CORPSVR02
    NameFormat = (empty)
    NetworkServerModeEnabled = TRUE
    NumberOfProcessors = 1
    OEMStringArray = SMBIOS 2.3
    Customer Reference Platform
    PartOfDomain = TRUE
    PauseAfterReset = -1
    PowerManagementCapabilities = (empty)
    PowerManagementSupported = (empty)
    PowerOnPasswordStatus = 3
    PowerState = 0
    PowerSupplyState = 3
    PrimaryOwnerContact = (empty)
    PrimaryOwnerName = wrs
    ResetCapability = 1
    ResetCount = -1
    ResetLimit = -1
    Roles =   LM_Workstation
              LM_Server
              Primary_Domain_Controller
              Timesource
              Print
              DialIn
              NT
              Master_Browser
              DFS
    Status = OK
    SupportContactDescription = (empty)
    SystemStartupDelay = 30
    SystemStartupOptions = "Windows Server 2003, Standard"
/fastdetect
          "Microsoft Windows XP Home Edition" /fastdetect
    SystemStartupSetting = 0
    SystemType = X86-based PC
    ThermalState = 3
    TotalPhysicalMemory = 535805952
    UserName = CPANDL\administrator
    WakeUpType = 6
    Workgroup = (empty)
```

Что представляют собой элементы конфигурации компьютера, показано в табл. 15-3.

Табл. 15-3. Элементы конфигурации компьютера и их описание

Параметр	Описание
AdminPasswordStatus	Статус пароля администратора: 1 = отключен, 2 = включен, 3 = не введен в действие, 4 = неизвестен
AutomaticResetBootOption	Сообщает, включена ли автоматическая пере загрузка при загрузке
AutomaticResetCapability	Сообщает, разрешена ли автоматическая перезагрузка
BootOptionOnLimit	Действия, которые должна предпринять система по достижении значения ResetLimit: 1 = зарезервировано, 2 = операционная система, 3 = системные утилиты, 4 = не перезагружаться
BootOptionOnWatchDog	Действия при загрузке, которые должны быть предприняты после того, как сработает Watchdog-таймер (таймер, ведущий обратный отсчет от заданного значения при загрузке; как только он обнуляется, выполняются указанные действия, обычно перезагрузка): 1 = зарезервировано, 2 = операционная система, 3 = системные утилиты, 4 = не перезагружаться
BootROMSupported	Сообщает, поддерживается ли загрузчик из ROM
BootupState	Сообщает, как должна запускаться система: «Normal boot», «Fail-safe boot» или «Fail-safe with network boot»
Caption	Имя системы
ChassisBootupState	Состояние загрузки базовых компонентов системы: 1 = другое, 2 = неизвестное, 3 = безопасное, 4 = предупреждение, 5 = критическое, 6 = невосстановимое
CreationClassName	Имя класса, из которого создан объект
CurrentTimeZone	Разница (в минутах) между показаниями часов компьютера и всемирным координированным временем (Coordinated Universal Time, UTC)
DaylightInEffect	Сообщает, действует ли режим перехода на летнее время
Description	Описание компьютера
DNSHostName	Имя сервера в DNS

Табл. 15-3. *(продолжение)*

Параметр	Описание
Domain	Имя домена, к которому принадлежит компьютер
DomainRole	Роль компьютера в домене: 0 = автономная рабочая станция, 1 = рядовая рабочая станция в домене, 2 = автономный сервер, 3 = рядовой сервер, 4 = резервный контроллер домена, 5 = основной контроллер домена
EnableDaylightSavings-Time	Указывает, включен ли режим перехода на летнее время: TRUE (система переводит часы при переходе на зимнее/летнее время) или FALSE (система не переводит часы)
FrontPanelResetStatus	Аппаратная защита для кнопки Reset: 0 = отключена, 1 = включена, 2 = не реализована, 3 = неизвестно
InfraredSupported	Сообщает, есть ли в системе инфракрасный порт (IR)
InitialLoadInfo	Данные, нужные для поиска устройства начальной загрузки или сервиса загрузки, чтобы запросить запуск операционной системы
InstallDate	Дата и время установки программного обеспечения
KeyboardPasswordStatus	Сообщает состояние пароля клавиатуры: 0 = отключен, 1 = включен, 2 = не реализован, 3 = неизвестно
LastLoadInfo	Запись свойства InitialLoadInfo, которая содержит данные о процессе запуска загруженной в данный момент операционной системы
Manufacturer	Название компании — производителя компьютера
Model	Название модели компьютера
Name	Имя компьютера
NameFormat	Указывает, как формируется имя компьютера
NetworkServerMode-Enabled	Сообщает, включен ли режим сервера сети
NumberOfProcessors	Число включенных процессоров в компьютере
OEMStringArray	Список строк с описаниями, заданных OEM
PartOfDomain	Определяет, является ли компьютер частью домена: TRUE (компьютер входит в домен) или FALSE (компьютер входит в рабочую группу)

(см. след. стр.)

Табл. 15-3. (*продолжение*)

Параметр	Описание
PauseAfterReset	Задержка (в мс) перед началом перезагрузки, инициируемой после выключения питания или сброса
PowerManagement-Capabilities	Функциональность логического устройства в управлении электропитанием: 0 = неизвестна, 1 = не поддерживается, 2 = отключена, 3 = включена, 4 = режимы энергосбережения включаются автоматически, 5 = состояние энергосбережения можно настраивать, 6 = поддерживается циклическое включение/выключение электропитания, 7 = поддерживается включение электропитания по таймеру
PowerManagement-Supported	Указывает, можно ли управлять электропитанием устройства
PowerOnPasswordStatus	Состояние пароля при включении электропитания: 0 = отключен, 1 = включен, 2 = не реализован, 3 = неизвестно
PowerState	Сообщает текущее состояние электропитания компьютера: 0 = неизвестно, 1 = полная мощность, 2 = энергосбережение (режим малой мощности), 3 = энергосбережение (ждущий режим), 4 = энергосбережение (неизвестно), 5 = циклическое электропитание, 6 = электропитание отключено, 7 = энергосбережение (предупреждение)
PowerSupplyState	Состояние блока питания при последней загрузке: 1 = другое, 2 = неизвестно, 3 = безопасное, 4 = предупреждение, 5 = критическое, 6 = невосстановимое
PrimaryOwnerContact	Контактная информация владельца компьютера
PrimaryOwnerName	Имя владельца компьютера
ResetCapability	Указывает, можно ли перезагрузить компьютер при помощи кнопок включения и Reset (или других аппаратных средств): 1 = другое, 2 = неизвестно, 3 = отключено, 4 = включено, 5 = невосстановимо
ResetCount	Число автоматических перезагрузок с момента последней перезагрузки вручную. Значение, равное –1, сообщает, что это число неизвестно

Табл. 15-3. (*окончание*)

Параметр	Описание
ResetLimit	Число последовательных попыток перезагрузки. Значение –1 указывает, что это число неизвестно
Roles	Системные роли
Status	Текущее состояние компьютера: «OK», «Error», «Degraded», «Unknown», «Pred Fail», «Starting», «Stopping», «Service»
SupportContact-Description	Информация о службе технической поддержки компьютера
SystemStartupDelay	Задержка загрузки в секундах
SystemStartupOptions	Список вариантов загрузки системы
SystemStartupSetting	Индекс профиля загрузки по умолчанию
SystemType	Тип архитектуры системы, например «X86-based PC» или «64-bit Intel PC»
ThermalState	Температура системы при последней загрузке: 1 = другое, 2 = неизвестно, 3 = безопасная, 4 = угрожающая, 5 = критическая, 6 = невосстановимое состояние
TotalPhysicalMemory	Общий объем физической памяти в байтах
UserName	Имя зарегистрированного в данный момент пользователя
WakeUpType	Событие, вызвавшее запуск системы: 0 = зарезервировано, 1 = другое, 2 = неизвестно, 3 = APM-таймер, 4 = звонок по модему, 5 = сетевой запрос, 6 = включение электропитания, 7 = PCI PME#, 8 = восстановление электропитания
Workgroup	Сообщает имя рабочей группы, если компьютер является членом рабочей группы

Как видите, детальная информация многое говорит о конфигурации компьютера. Аналогичные сведения об операционной системе можно получить командой **netsh diag show os /v**. Пример см. на листинге 15-5.

Листинг 15-5. Детальные сведения о конфигурации операционной системы

```
Operating System (Microsoft(R) Windows(R) Server 2003,
Standard Edition)
    BootDevice = \Device\HarddiskVolume1
    BuildNumber = 3790
```

```
    BuildType = Uniprocessor Free
    Caption = Microsoft(R) Windows(R) Server 2003, Standard
Edition
    CodeSet = 1252
    CountryCode = 1
    CreationClassName = Win32_OperatingSystem
    CSCreationClassName = Win32_ComputerSystem
    CSDVersion = (empty)
    CSName = CORPSVR02
    CurrentTimeZone = -480
    Debug = FALSE
    Description = (empty)
    Distributed = FALSE
    EncryptionLevel = 168
    ForegroundApplicationBoost = 2
    FreePhysicalMemory = 357176
    FreeSpaceInPagingFiles = 1114384
    FreeVirtualMemory = 1471560
    InstallDate = 3:53:12 PM 11/21/2004
    LargeSystemCache = 1
    LastBootUpTime = 10:37:11 AM 11/19/2005
    LocalDateTime = 10:42:00 AM 11/19/2005
    Locale = 0409
    Manufacturer = Microsoft Corporation
    MaxNumberOfProcesses = -1
    MaxProcessMemorySize = 2097024
    Name = Microsoft Windows Server 2003 Standard
 Edition|C:\WINDOWS|\Device\Harddisk0\Partition1
    NumberOfLicensedUsers = 500
    NumberOfProcesses = 33
    NumberOfUsers = 2
    Organization = wrs
    OSLanguage = 1033
    OSProductSuite = 272
    OSType = 18
    OtherTypeDescription = (empty)
    PAEEnabled = FALSE
    PlusProductID = (empty)
    PlusVersionNumber = (empty)
    Primary = TRUE
    ProductType = 2
    QuantumLength = 0
    QuantumType = 0
    RegisteredUser = wrs
    SerialNumber = 38383-022-1234343-43434
    ServicePackMajorVersion = 0
```

```
ServicePackMinorVersion = 0
SizeStoredInPagingFiles = 1280320
Status = OK
SuiteMask = 272
SystemDevice = \Device\HarddiskVolume1
SystemDirectory = C:\WINDOWS\system32
SystemDrive = C:
TotalSwapSpaceSize = (empty)
TotalVirtualMemorySize = 1803568
TotalVisibleMemorySize = 523248
Version = 5.2.3790
WindowsDirectory = C:\WINDOWS
```

Что представляют собой элементы конфигурации операционной системы, показано в табл. 15-4.

Табл. 15-4. Элементы конфигурации операционной системы

Параметр	Описание
BootDevice	Дисковое устройство, с которого загружается операционная система Win32
BuildNumber	Номер сборки операционной системы
BuildType	Тип сборки используемой операционной системы, например «retail build» (розничная версия) или «checked build» (проверочная версия)
Caption	Имя операционной системы
CodeSet	Кодовая страница, используемая операционной системой
CountryCode	Код страны, используемый операционной системой
CreationClassName	Имя класса, из которого создан объект
CSCreationClassName	Имя класса, из которого создан объект «компьютер»
CSDVersion	Сообщает, какой Service Pack (пакет обновлений) установлен на компьютере. Значение равно NULL, если пакет обновлений не установлен
CSName	Имя компьютера, сопоставленное с объектом класса «компьютер»
CurrentTimeZone	Разница (в минутах) между временем в локальной операционной системе и средним временем по Гринвичу (Greenwich Mean Time, GMT). Это значение может быть положительным, отрицательным или нулевым

(см. след. стр.)

Табл. 15-4. (*продолжение*)

Параметр	Описание
Debug	Сообщает, является ли версия операционной системы проверочной. Если значение равно TRUE, установлена отладочная версия User.exe
Description	Описание операционной системы Windows
Distributed	Указывает, работает ли операционная система со множеством узлов. Если да, эти узлы сгруппированы в кластер
EncryptionLevel	Уровень шифрования для безопасных транзакций (40-, 128- или *n*-битный)
ForegroundApplicationBoost	Задает приоритет для программы «переднего плана». В Windows NT 4 и Windows 2000 динамическое повышение приоритета приложения реализуется за счет выделения этому приложению большего процессорного времени. Возможные значения: 0 = нет, 1 = минимальное, 2 = максимальное (по умолчанию)
FreePhysicalMemory	Объем неиспользуемой в данный момент оперативной памяти в Кб
FreeSpaceInPagingFiles	Объем свободного места в страничных файлах операционной системы в Кб
FreeVirtualMemory	Размер неиспользуемой виртуальной памяти в Кб
InstallDate	Дата и время установки операционной системы
LargeSystemCache	Указывает, оптимизируется ли использование памяти для программ или системного кэша. Возможные значения: 0 = оптимизируется для программ, 1 = оптимизируется для системного кэша
LastBootUpTime	Время последней загрузки системы
LocalDateTime	Локальные дата и время компьютера
Locale	Идентификатор языка и региональных стандартов в данной операционной системе
Manufacturer	Производитель операционной системы. Для Win32-систем это значение — «Microsoft Corporation»
MaxNumberOfProcesses	Максимальное число контекстов процессов, поддерживаемое операционной системой. Если максимум не задан, выводится 0
MaxProcessMemorySize	Максимальный объем памяти в Кб, который может быть выделен процессу. Нулевое значение сообщает, что максимум не задан
Name	Имя экземпляра операционной системы

Табл. 15-4. (*продолжение*)

Параметр	Описание
NumberOfLicensed-Users	Число пользовательских лицензий для данной операционной системы: 0 = не ограничено, −1 = неизвестно
NumberOfProcesses	Текущее число контекстов процессов в системе
NumberOfUsers	Текущее число сеансов пользователей
Organization	Название компании, задаваемое для зарегистрированного пользователя системы
OSLanguage	Языковая версия операционной системы
OSProductSuite	Тип установленного пакета системных продуктов: 1 = Small Business, 2 = Enterprise, 4 = BackOffice, 8 = Communication Server, 16 = Terminal Server, 32 = Small Business (Restricted), 64 = Embedded NT и 128 = Data Center
OSType	Тип операционной системы: 1 = другая, 18 = Windows NT или более поздняя
OtherTypeDescription	Задает дополнительное описание; используется, когда OSType = 1
PlusProductID	Серийный номер Windows Plus! (если установлен)
PlusVersionNumber	Номер версии Windows Plus! (если установлен)
Primary	Указывает, является ли данная операционная система основной
ProductType	Сообщает, для каких целей предназначена данная лицензия операционной системы: 1 = для рабочей станции, 2 = для контроллера домена, 3 = для рядового сервера
QuantumLength	Длина квантума процессорного времени в тактах системных часов: 1 = неизвестно, 2 = один такт, 3 = два такта
QuantumType	Тип длины квантума процессорного времени: 1 = неизвестно, 2 = фиксированная, 3 = переменная. При переменной длине величина кванта процессорного времени, выделяемого активному и фоновым приложениям, может быть разной, а при фиксированной длине эта величина одинакова
RegisteredUser	Имя зарегистрированного пользователя системы
SerialNumber	Серийный номер операционной системы
ServicePackMajor-Version	Старший номер версии пакета обновлений, установленного в системе. Если ни одного пакета обновлений не установлено, это значение равно 0 или NULL

(*см. след. стр.*)

Табл. 15-4. (*окончание*)

Параметр	Описание
ServicePackMinor-Version	Младший номер версии пакета обновлений, установленного в системе. Если ни одного пакета обновлений не установлено, это значение равно 0 или NULL
SizeStoredInPaging-Files	Общий объем страниц памяти в Кб, записанных в страничные файлы. Если это значение равно 0, значит, в системе нет страничных файлов
Status	Текущее состояние объекта: «OK», «Error», «Unknown», «Degraded», «Pred Fail», «Starting», «Stopping» и «Service»
SuiteMask	Битовые флаги, идентифицирующие программные пакеты в данной системе: 1 = Small Business, 2 = Enterprise, 4 = Back Office, 8 = Communications, 16 = Terminal, 32 = Small Business Restricted, 64 = Embedded NT, 128 = Data Center
SystemDevice	Дисковое устройство, на котором установлена операционная система
SystemDirectory	Системный каталог операционной системы
SystemDrive	Раздел дискового устройства, в котором установлена операционная система
TotalSwapSpaceSize	Доступное для подкачки пространство в Кб. Это значение может быть не определено (NULL), если пространство для подкачки (swap space) не отличается от страничных файлов (page files)
TotalVirtualMemory-Size	Размер виртуальной памяти в Кб
TotalVisibleMemory-Size	Общий объем физической памяти в Кб, доступный операционной системе
Version	Номер версии операционной системы
WindowsDirectory	Каталог Windows операционной системы

Проблемы с конфигурациями IP, DNS и WINS

Контекст Diag в Netsh поддерживает команды для просмотра конфигураций IP, DNS и WINS.

- **Netsh diag show ip** Показывает IP-адреса, используемые сетевыми адаптерами компьютера. Например:

```
IPAdress
    1. [00000001] Intel(R) PRO/100 VE Network Connection
       IPAddress = 192.168.1.50
```

```
2. [00000002] Intel(R) PRO/100 VE Network Connection
   IPAddress = 192.168.2.108
```

Сетевые адаптеры перечисляются по порядку. Если на компьютере два сетевых адаптера, выводится две записи. Отключенные сетевые адаптеры (или недоступные по какой-то другой причине) не отображаются.

- **Netsh diag show gateway** Показывает интернет-шлюзы, определенные для сетевых адаптеров компьютера, например:

```
Default Gateways
   1. [00000001] Intel(R) PRO/100 VE Network Connection
      DefaultIPGateway =  192.168.1.1 Same Subnet
         192.168.1.2 Same Subnet
   2. [00000002] Intel(R) PRO/100 VE Network Connection
      DefaultIPGateway =  192.168.2.1 Same Subnet
```

Шлюзы перечисляются по адаптерам и в том порядке, в каком они используются. Если на компьютере несколько сетевых адаптеров, вы увидите запись для каждого задействованного сетевого адаптера. Если шлюз находится в той же подсети, что и IP-адреса, используемые адаптером, это также отмечается в выводе. Но неправильно сконфигурированный адаптер (т. е. не относящийся к той же подсети) не помечается каким-то особым образом. В этом случае может быть просто не указан основной шлюз для адаптера. Или, если для одного адаптера задано несколько шлюзов, неверно настроенный адаптер может быть исключен из списка. Если вы подозреваете нечто подобное, сравните вывод команды **netsh diag show gateway** с результатом **netsh interface ip show config**. Хотя неверно заданный адрес шлюза не войдет в результат команды *netsh diag show gateway*, он появится в выводе команды *netsh interface ip show config*.

- **Netsh diag show dns** Показывает DNS-серверы, определенные для сетевых адаптеров компьютера, например:

```
DNS Servers
   1. [00000001] Intel(R) PRO/100 VE Network Connection
      DNSServerSearchOrder = 192.168.1.50
                             192.168.1.67
   2. [00000002] Intel(R) PRO/100 VE Network Connection
      DNSServerSearchOrder = 192.168.2.10
                             192.168.2.20
```

Настроенные DNS-серверы показываются в порядке поиска. Проверьте правильность их IP-адресов и порядка поиска.

- **Netsh diag show wins** Показывает WINS-серверы, определенные для сетевых адаптеров компьютера, например:

```
WINS Servers
    1. [00000001] Intel(R) PRO/100 VE Network Connection
       WINSPrimaryServer = 192.168.1.102
       WINSSecondaryServer = 192.168.1.108
    2. [00000002] Intel(R) PRO/100 VE Network Connection
       WINSPrimaryServer = 192.168.2.205
       WINSSecondaryServer = 192.168.2.227
```

Настроенные WINS-серверы показываются в порядке поиска. Проверьте правильность их IP-адресов и порядка поиска.

Примечание Хотя эти команды поддерживают параметр /V, он не дает никакой дополнительной информации.

Проблемы с TCP/IP-соединениями

Контекст Diag в Netsh предоставляет команды, пригодные для установления TCP/IP-соединений с целью устранения неполадок. Специфические команды позволяют устанавливать TCP/IP-соединения и потом проверять их при работе с почтой, Usenet и прокси Internet Explorer, а универсальная команда обеспечивает соединение с TCP-хостом через определенный порт.

Для проверки возможности почтовых соединений с использованием почтового клиента по умолчанию введите **netsh diag connect mail**. Если есть какие-то проблемы с соединением, это будет отражено в выводе команды. В следующем примере компьютеру не удается установить соединение с серверами входящей и исходящей почты:

```
Default Outlook Express Mail  (pop3.cpandl.com /
mail.cpandl.com)
   InBoundMailPort = 110
   InBoundMailServer = pop3.cpandl.com
   Unable to connect to pop3.cpandl.com port 110
   OutBoundMailPort = 25
   OutBoundMailServer = mail.cpandl.com
   Unable to connect to mail.cpandl.com port 25
```

Здесь компьютер, возможно, не подсоединен к сети или неправильно заданы настройки электронной почты.

Для проверки возможности соединений с использованием клиента Usenet по умолчанию введите **netsh diag connect news**. Как и в случае с почтовым клиентом, если есть какие-то проблемы с подключением к серверу новостей, это будет отражено в выводе команды. Если ни один новостной клиент не настроен, вы увидите, например:

```
Default Outlook Express News (Not Configured)
```

Для проверки прокси-соединений Internet Explorer через Web-прокси по умолчанию введите **netsh diag connect ieproxy**. В итоге вы получите состояние текущего соединения, например:

```
Internet Explorer proxy (cpandlproxy)
   IEProxyPort = 80
   IEProxy = cpandlproxy
   Server appears to be running on port(s) [80]
```

В данном случае Netsh удалось установить соединение с Web-прокси. Прокси-сервер cpandlproxy работает с использованием порта 80, который является стандартным портом для Web-серверов.

Кроме того, вы можете установить соединение с любым IP-хостом, использующим назначенный TCP-порт. Синтаксис этой команды:

```
netsh diag connect iphost ХостИмя НомерПорта
```

где *ХостИмя* задается как IP-адрес, имя компьютера или полное доменное имя (FQDN) хоста, соединение с которым вы пытаетесь установить, а *НомерПорта* определяет TCP-порт, через который вы устанавливаете соединение. Некоторые из наиболее распространенных TCP-портов и связанные с ними протоколы перечислены в табл. 15-5.

Табл. 15-5. Наиболее распространенные TCP-протоколы и порты

Протокол	Порт
FTP	21
Telnet	23
SMTP	25
Time Server	37
Nameserver	42
DNS	53
HTTP	80

(см. след. стр.)

Табл. 15-5. *(окончание)*

Протокол	Порт
Kerberos	88
POP3	110
NNTP	119
IMAP	143
HTTPS	443
Microsoft Directory Services	445
WINS	1512
PPTP	1723

Чтобы лучше понять, как подключаться к определенным хостам, рассмотрим несколько примеров.

Подключение к 192.168.1.100 через порт 37:

```
netsh diag connect iphost 192.168.1.100 37
```

Подключение к corpdc07 через порт 445:

```
netsh diag connect iphost corpdc07 445
```

Подключение к services.cpandl.com через порт 443:

```
netsh diag connect iphost services.cpandl.com 443
```

Проверка возможности соединений

Используя контекст Diag в Netsh, вы можете попытаться проверить возможность соединений с разными удаленными хостами. Базовые команды тестируют возможность соединения с сервером определенного типа. Так, команды **netsh diag ping mail**, **netsh diag ping news** и **netsh diag ping ieproxy** позволяют проверить возможность соединений с почтовым сервером, сервером новостей и прокси-сервером Internet Explorer соответственно. Рассмотрим следующий пример:

```
netsh diag ping mail
```

В данном случае вы проверяете возможность соединения через почтовый клиент по умолчанию. В выводе, приведенном ниже, обнаруживаются проблемы с соединением или конфигурацией:

```
Default Outlook Express Mail (pop3.cpandl.com /
mail.cpandl.com)
    InBoundMailServer = pop3.cpandl.com
    Ping request could not find host pop3.cpandl.com. Please
check the name and try again.
    OutBoundMailServer = mail.cpandl.com
    Ping request could not find host mail.cpandl.com. Please
check the name and try again.
```

Прочие команды, предназначенные для проверки соединений, немного сложнее.

- **Netsh diag ping iphost** Проверяет соединение с удаленным хостом по указанному IP-адресу, имени компьютера или полному доменному имени. Например, если вы хотите проверить соединение между компьютером, на котором вы сейчас работаете, и компьютером с IP-адресом 192.168.1.100, введите:

  ```
  netsh diag ping iphost 192.168.1.100
  ```

- **Netsh diag ping adapter** Проверяет конфигурацию TCP/IP для сетевых адаптеров. Обычно вас будет интересовать конфигурация конкретного адаптера. Для этого добавьте в командную строку индекс адаптера или его полное/частичное имя. Например, если вы хотите проверить конфигурацию сетевого адаптера с индексом 1, введите:

  ```
  netsh diag ping adapter 1
  ```

- **Netsh diag ping dhcp** Проверяет настройки DHCP-сервера для сетевых адаптеров. Вы можете ограничиться конкретным адаптером, выполнив эту команду с указанием индекса адаптера или его полного/частичного имени. Например, если вам нужно протестировать конфигурацию DHCP для сетевого адаптера, имя которого начитается с «3com», введите:

  ```
  netsh diag ping dhcp 3com*
  ```

- **Netsh diag ping dns** Проверяет настройки DNS-сервера для сетевых адаптеров. Вы можете ограничиться конкретным адаптером, выполнив эту команду с указанием индекса адаптера или его полного/частичного имени. Например, если вы проверяете конфигурацию DNS для сетевого адаптера, имя которого начинается с «Intel», введите:

  ```
  netsh diag ping dns Intel*
  ```

- **Netsh diag ping gateway** Проверяет параметры шлюзов для сетевых адаптеров. Вы можете ограничиться конкретным адаптером, выполнив эту команду с указанием индекса адаптера или его полного/частичного имени. Например, чтобы проверить конфигурацию основного шлюза для сетевого адаптера, имя которого начинается с «3com», введите:

```
netsh diag ping gateway 3com*
```

- **Netsh diag ping ip** Проверяет допустимость IP-адресов, назначенных сетевым адаптерам. Вы можете ограничиться конкретным адаптером, выполнив эту команду с указанием индекса адаптера или его полного/частичного имени, например:

```
netsh diag ping ip Intel*
```

- **Netsh diag ping wins** Проверяет настройки WINS-сервера для сетевых адаптеров. Вы можете ограничиться конкретным адаптером, выполнив эту команду с указанием индекса адаптера или его полного/частичного имени. Например, если вам нужно проверить конфигурацию WINS для сетевого адаптера с индексом 2, введите:

```
netsh diag ping wins 2
```

Наиболее полезна команда **netsh diag ping adapter**. Она проверяет всю TCP/IP-конфигурацию адаптеров, в том числе IP, DHCP, DNS, WINS и настройки основного шлюза. Пример вывода этой команды на листинге 15-6.

Листинг 15-6. Вывод команды netsh diag ping adapter

```
Network Adapters
   1. [00000001] Intel(R) PRO/100 VE Network Connection
      DefaultIPGateway = 192.168.1.1 Same Subnet
         Pinging 192.168.1.1 with 32 bytes of data:
         Reply from 192.168.1.1: bytes=32 time<1ms TTL=0
         Reply from 192.168.1.1: bytes=32 time<1ms TTL=0
         Reply from 192.168.1.1: bytes=32 time<1ms TTL=0
         Reply from 192.168.1.1: bytes=32 time<1ms TTL=0
         Ping statistics for 192.168.1.1:
            Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
         Approximate round trip times in milliseconds:
            Minimum = 0ms, Maximum = 0ms, Average = 0ms
         Pinging 192.168.1.2 with 32 bytes of data:
         Request timed out.
```

```
    Request timed out.
    Request timed out.
    Request timed out.
    Ping statistics for 192.168.1.2:
        Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
DNSServerSearchOrder = 192.168.1.50
    Pinging 192.168.1.50 with 32 bytes of data:
    Reply from 192.168.1.50: bytes=32 time<1ms TTL=0
    Reply from 192.168.1.50: bytes=32 time<1ms TTL=0
    Reply from 192.168.1.50: bytes=32 time<1ms TTL=0
    Reply from 192.168.1.50: bytes=32 time<1ms TTL=0
    Ping statistics for 192.168.1.50:
        Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
    Approximate round trip times in milliseconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
        192.168.1.67
    Pinging 192.168.1.67 with 32 bytes of data:
    Request timed out.
    Request timed out.
    Request timed out.
    Request timed out.
    Ping statistics for 192.168.1.67:
        Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
IPAddress = 192.168.1.12
    Pinging 192.168.1.12 with 32 bytes of data:
    Reply from 192.168.1.12: bytes=32 time<1ms TTL=0
    Reply from 192.168.1.12: bytes=32 time<1ms TTL=0
    Reply from 192.168.1.12: bytes=32 time<1ms TTL=0
    Reply from 192.168.1.12: bytes=32 time<1ms TTL=0
    Ping statistics for 192.168.1.12:
        Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
        192.168.2.12
    Pinging 192.168.2.12 with 32 bytes of data:
    Reply from 192.168.2.12: bytes=32 time<1ms TTL=0
    Reply from 192.168.2.12: bytes=32 time<1ms TTL=0
    Reply from 192.168.2.12: bytes=32 time<1ms TTL=0
    Reply from 192.168.2.12: bytes=32 time<1ms TTL=0
    Ping statistics for 192.168.2.12:
        Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
    Approximate round trip times in milliseconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
WINSPrimaryServer = 192.168.1.102
    Pinging 192.168.1.102 with 32 bytes of data:
    Request timed out.
```

```
    Request timed out.
    Request timed out.
    Request timed out.
    Ping statistics for 192.168.1.102:
        Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
WINSSecondaryServer = 192.168.1.108
    Pinging 192.168.1.108 with 32 bytes of data:
    Request timed out.
    Request timed out.
    Request timed out.
    Request timed out.
    Ping statistics for 192.168.1.108:
        Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

Просмотрев этот листинг, вы обнаружите просто мириады возможных проблем с соединениями или конфигурацией. Сразу возникают вопросы, требующие проверки.

- **Подключен ли компьютер к сети?** Видя так много сообщений об ошибках, первое, что я бы сделал, — проверил, действительно ли компьютер подключен к сети. Если компьютер может установить соединение с одним из своих основных шлюзов (и этот шлюз не настроен на тот же IP-адрес, что и компьютер), значит, он подключен к сети. Если же компьютер не в состоянии связаться ни с одним из своих шлюзов, не исключено, что сетевой кабель отсоединен или сетевой адаптер неисправен.
- **Исправен ли сетевой адаптер?** Если компьютеру не удается связаться ни с одним из указанных для него шлюзов, возможно, его сетевой адаптер неисправен. Чтобы выяснить это, проверьте записи IP-адресов. Эти записи показывают результаты соединения компьютера со своими сетевыми адаптерами. Если тест на соединение с одним из этих IP-адресов постоянно или периодически не проходит, то скорее всего один из сетевых адаптеров неисправен.

Неудачи с подключением к DNS и WINS тоже предмет для беспокойства. Проблемы связи с заданным DHCP-сервером обычно дают сходную картину. Если компьютер соединяется с основным шлюзом, но не способен связаться с DNS-, DHCP- или WINS-сервером, то, возможно, сервер выключен, в конфигурации неправильно указан IP-адрес или имеется проблема с каналом связи между вашим компьютером и этим сервером.

Еще одна полезная команда Netsh Diag – SHOW TEST. Введите **netsh diag show test /v** и вы запустите полную диагностику соединений для:

- почтового клиента по умолчанию;
- клиента новостей по умолчанию;
- прокси по умолчанию (Internet Explorer);
- локальной петли по адресу 127.0.0.1;
- всех модемов;
- всех сетевых адаптеров.

Приложение

Справочник по основным утилитам командной строки

В этой книге я рассмотрел многие инструменты и сценарии командной строки. Данное приложение — краткий справочник по синтаксису и использованию этих средств, а также других команд и утилит, которые, возможно, вам пригодятся. Все средства перечисляются в алфавитном порядке. Если не оговорено иное, в Microsoft Windows Server 2003 и Windows XP Professional эти инструменты работают одинаково. Кроме того, если какая-то утилита не входит в стандартный дистрибутив Windows, я указываю, откуда она берется. Например, «Windows Server 2003 Resource Kit» означает, что данное средство доступно только в Microsoft Windows Server 2003 Resource Kit.

arp

Показывает или изменяет используемые протоколом ARP (address resolution protocol) таблицы преобразования IP-адресов в физические.

```
arp -a [IP-адрес] [-n АдресИнтерфейса]
arp -d IP-адрес [АдресИнтерфейса]
arp -s IP-адрес Ethernet-адрес [АдресИнтерфейса]
```

assoc

Показывает или изменяет сопоставления расширений файлов.

```
assoc [.расширение[=[ТипФайла]]]
```

attrib

Показывает или изменяет атрибуты файла или группы файлов.

```
attrib [+r|-r] [+a|-a] [+s|-s] [+h|-h]
[[Диск:] [Путь] ИмяФайла] [/s] [/d]
```

cacls

Выводит или изменяет ACL (access control list) файла.

```
cacls ИмяФайла [/t] [/e] [/c] [/g Пользователь:Доступ]
[/r Пользователь [...]] [/p Пользователь:Доступ [...]]
[/d Пользователь [...]]
```

call

Вызывает сценарий или выполняет текущий сценарий как процедуру, начиная с заданной метки.

```
call [Диск:][Путь]ИмяФайла [Параметры]
call :Метка [Аргументы]
```

cd

Показывает текущий каталог или выполняет его смену.

```
chdir [/d] [Диск:][Путь]
chdir [..]
cd [/d] [Диск:][Путь]
cd [..]
```

chdir

См. CD.

chkdsk

Проверяет логический диск на наличие ошибок и показывает отчет.

```
chkdsk [Диск:][[Путь]ИмяФайла]
[/f][/v][/r][/x][/i][/c][/l[:Размер]]
```

chkntfs

Сообщает состояние томов. Включает или отключает автоматическую проверку томов при загрузке операционной системы.

```
chkntfs [/x | /c] Том: [...]
chkntfs /t[:Время]
chkntfs /d
```

choice

Создает список, из которого пользователь может выбирать значения при выполнении пакетных сценариев.

```
choice [/c Клавиши] [/n] [/cs] [/t ВремяОжидания /d Клавиша]
[/m "Текст"]
```

cipher

Показывает или изменяет параметры шифрования папок и файлов NTFS-томов.

В Windows Server 2003:

```
cipher [/e| /d] [/s:Каталог] [/a] [/i] [/f] [/q]
       [/h] [[Путь]ИмяФайла [...]]
cipher [/k | /r:ИмяФайла | w:Каталог]
cipher /u [/n]
cipher /x[:EFS-файл] [ИмяФайла]
```

В Windows XP Professional:

```
cipher [/e| /d] [/s:Каталог] [/a] [/i] [/f] [/q]
       [/h] [[Путь]ИмяФайла [...]]
cipher [/k | /r:ИмяФайла | w:Каталог]
cipher /u [/n]
```

clip

При конвейеризации переадресует вывод командной строки в буфер обмена Windows.

```
[Команда |] clip
```

Примечание В данном случае «|» — это символ конвейеризации.

cls

Очищает окно консоли.

```
cls
```

cmd

Запускает новый экземпляр командной оболочки Windows.

```
cmd [/a | /u] [/q] [/d] [/e:on | /e:off]
[/f:on | /f:off] [/v:on | /v:off]
[[/s] [/c | /k] Команда]
```

cmdkey

Создает и изменяет сохраненные имена и пароли пользователей.

```
cmdkey [{/add | /generic}:ЦелевоеИмя
{/smartcard | /user:Пользователь@Домен
{/pass{:Пароль}}} | /delete{:ЦелевоеИмя
| /ras} | /list{:ЦелевоеИмя}]
```

Команда доступна только в Windows Server 2003.

color

Задает цвета фона (Ф) и текста (Т) в окне командной оболочки.

```
color [ФТ]
```

comp

Сравнивает содержимое двух файлов или наборов файлов.

```
comp [Данные1] [Данные2] [/d] [/a] [/l]
[/n=Количество] [/c] [/offline]
```

compact

Показывает или изменяет параметры сжатия файлов в NTFS-разделах.

```
compact [/c | /u] [/s[:Каталог]] [/a] [/i] [/f]
[/q] [ИмяФайла [...]]
```

convert

Преобразовывает файловую систему тома из FAT в NTFS.

```
convert Диск: /fs:NTFS [/v] [/x]
[/cvtarea:ИмяФайла] [/nosecurity]
```

copy

Копирует или объединяет файлы.

```
copy [/d][/v][/n][/y|/-y][/z][/a|/b] Источник [/a | /b]
[+ Источник [/a | /b] [+ ...]][Результат [/a|/b]]
```

date

Показывает или изменяет системную дату.

```
date [/T | мм-чч-гг]
```

dcgpofix

Приводит в исходное состояние используемые по умолчанию объекты политики группы.

```
dcgpofix [/ignoreschema]
[/target: {domain | dc | both}]
```

Команда доступна только в Windows Server 2003.

defrag

Дефрагментирует том жесткого диска.

```
defrag Том [/a] [/v]
defrag Том [/f]
```

del

Удаляет один или несколько файлов.

```
del [/p] [/f] [/s] [/q] [/a[[:]Атрибуты]]
[Диск:][Путь]ИмяФайла[...]
```

dir

Показывает список файлов и подкаталогов заданного каталога.

```
dir [Диск:][Путь][ИмяФайла] [/p] [/w] [/d]
[/a[[:]Атрибуты]] [/o[[:]ПорядокСортировки]]
[/t[[:]Время]] [/s] [/b] [/l] [/n]
[/x] [/c] [/q] [/4]
```

diskcomp

Сравнивает содержимое двух дискет.

```
diskcomp [Диск1: [Диск2:]]
```

diskcopy

Копирует содержимое одной дискеты на другую.

```
diskcopy [Диск1: [Диск2:]] [/v]
```

diskpart

Вызывает командную оболочку DISKPART, в которой вводятся и выполняются внутренние команды, управляющие дисками, разделами и томами.

```
diskpart
```


Примечание О работе с DISKPART см. главы 8, 9 и 10.

doskey

Позволяет редактировать командные строки, повторно вызывать команды Windows и создавать макросы DOSKey.

```
doskey [/reinstall] [/listsize=Размер]
[/macros[:all | :ИсполняемыйФайл]]
[/history] [/insert | /overstrike]
[/exename=ИсполняемыйФайл]
[/macrofile=Файл] [Макрос=[Текст]]
```

driverquery

Показывает список всех установленных драйверов устройств и свойства этих драйверов.

```
driverquery [/s Компьютер [/u [Домен\]Пользователь
[/p [Пароль]]]]
[/fo {table|list|csv}] [/nh] [/v] [/si]
```

dsadd computer

Создает учетную запись компьютера в службе каталогов Active Directory.

```
dsadd computer DNКомпьютера [-samid SAM-имя] [-desc Описание]
[-loc Местонахождение] [-memberof DNГруппы ...] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [{-uc |
-uco | -uci}]
```

Для использования команды в Windows XP Professional нужно установить Windows Server 2003 Administration Tools Pack.

dsadd group

Создает учетную запись группы в Active Directory.

```
dsadd group DNГруппы [-secgrp {yes | no}] [-scope {l | g | u}]
[-samid SAM-имя] [-desc Описание] [-memberof Группа ...]
[-members Член ...] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-q] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional нужно установить Windows Server 2003 Administration Tools Pack.

dsadd user

Создает учетную запись пользователя в Active Directory.

```
dsadd user DNПользователя [-samid SAM-имя] [-upn UPN] [-fn
Имя] [-mi Отчество] [-ln Фамилия] [-display ОтображаемоеИмя]
[-empid ИдентификаторСотрудника] [-pwd {Пароль | *}] [-desc
Описание] [-memberof Группа ...] [-office Офис] [-tel Телефон]
[-email E-mail] [-hometel ДомашнийТелефон] [-pager Пейджер]
[-mobile МобильныйТелефон] [-fax Факс] [-iptel IP-телефон]
[-webpg Web-страница] [-title Должность] [-dept Отдел]
[-company Компания] [-mgr Менеджер] [-hmdir ОсновнойКаталог]
[-hmdrv БуквaДиска:] [-profile Путь] [-loscr Путь] [-mustchpwd
{yes | no}] [-canchpwd {yes | no}] [-reversiblepwd {yes | no}]
[-pwdneverexpires {yes | no}] [-acctexpires ЧислоДней]
[-disabled {yes | no}] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-q] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional нужно установить Windows Server 2003 Administration Tools Pack.

dsget computer

Показывает свойства учетных записей компьютеров. Синтаксис для вывода свойств нескольких компьютеров выглядит так:

```
dsget computer DNКомпьютера ... [-dn] [-samid] [-sid] [-desc]
[-loc] [-disabled] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [-l] [{-uc | -uco
| -uci}] [-part DNРаздела [-qlimit] [-qused]]
```

Синтаксис вывода информации о членстве одного компьютера:

```
dsget computer DNКомпьютера [-memberof [-expand]] [{-s Сервер
| -d Домен }] [-u ИмяПользователя] [-p {Пароль | *}] [-c] [-q]
[-l] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional нужно установить Windows Server 2003 Administration Tools Pack.

dsget group

Показывает свойства учетных записей групп. Синтаксис для вывода свойств нескольких групп выглядит так:

```
dsget group DNГруппы ... [-dn] [-samid] [-sid] [-desc]
[-secgrp] [-scope] [{-s Сервер | -d Домен}] [-u ИмяПользова-
теля] [-p {Пароль | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
[-part DNРаздела [-qlimit] [-qused]]
```

А вот синтаксис для вывода информации о членстве группы:

```
dsget group DNГруппы [{-memberof | -members} [-expand]] [{-s
Server | -d Домен}] [-u ИмяПользователя] [-p {Пароль | *}]
[-c] [-q] [-l] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional нужно установить Windows Server 2003 Administration Tools Pack.

dsget server

Показывает свойства контроллеров домена. Синтаксис для вывода основных свойств заданного контроллера домена выглядит так:

```
dsget server DNСервера ... [-dn] [-desc] [-dnsname] [-site]
[-isgc] [{-s Сервер | -d Домен}] [-u ИмяПользователя] [-p
{Пароль | *}] [-c] [-q] [-l] [{-uc | -uco | -uci}]
```

Синтаксис для вывода списка участников системы безопасности, которым принадлежит наибольшее число объектов каталога на заданном контроллере домена, имеет вид:

```
dsget server DNСервера ... [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [-l] [{-uc | -uco
| -uci}] [-topobjowner ЧислоВыводимыхОбъектов]
```

Наконец, синтаксис для вывода DN (составных имен) разделов каталога на заданном сервере:

```
dsget server DNСервера ... [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [-l] [{-uc | -uco
| -uci}] [-part]
```

Для использования команды в Windows XP Professional нужно установить Windows Server 2003 Administration Tools Pack.

dsget user

Показывает свойства учетных записей пользователей. Синтаксис для вывода свойств нескольких пользователей:

```
dsget user DNПользователя ... [-dn] [-samid] [-sid] [-upn]
[-fn] [-mi] [-ln] [-display] [-empid] [-desc] [-office] [-tel]
[-email] [-hometel] [-pager] [-mobile] [-fax] [-iptel]
[-webpg] [-title] [-dept] [-company] [-mgr] [-hmdir] [-hmdrv]
[-profile] [-loscr] [-mustchpwd] [-canchpwd]
[-pwdneverexpires] [-disabled] [-acctexpires] [-reversiblepwd]
[{-uc | -uco | -uci}] [-part DNРаздела [-qlimit] [-qused]]
```

```
[{-s Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль |
*}] [-c] [-q] [-l]
```

Синтаксис для вывода информации о членстве пользователя в группах:

```
dsget user DNПользователя [-memberof [-expand]] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-c] [-q]
[-l] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional нужно установить Windows Server 2003 Administration Tools Pack.

dsmod computer

Изменяет атрибуты одной или нескольких учетных записей компьютеров в каталоге.

```
dsmod computer DNКомпьютера ... [-desc Описание] [-loc
Местонахождение] [-disabled {yes | no}] [-reset] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-c] [-q]
[{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsmod group

Изменяет атрибуты одной или нескольких учетных записей групп в каталоге.

```
dsmod group DNГруппы ... [-samid SAM-имя] [-desc Описание]
[-secgrp {yes | no}] [-scope {l | g | u}] [{-addmbr | -rmmbr |
-chmbr} DNУчастника ...] [{-s Сервер | -d Домен}]
[-u ИмяПользователя] [-p {Пароль | *}] [-c] [-q] [{-uc | -uco
| -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsmod server

Изменяет свойства контроллера домена.

```
dsmod server DNСервера ... [-desc Описание] [-isgc {yes | no}]
[{-s Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль |
*}] [-c] [-q] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsmod user

Изменяет атрибуты одной или нескольких учетных записей пользователей в каталоге.

```
dsmod user DNПользователя ... [-upn UPN] [-fn Имя] [-mi
Отчество] [-ln Фамилия] [-display ОтображаемоеИмя] [-empid
ИдентификаторСотрудника] [-pwd {Пароль | *}] [-desc Описание]
[-office Офис] [-tel Телефон] [-email E-mail] [-hometel
ДомашнийТелефон] [-pager Пейджер] [-mobile МобильныйТелефон]
[-fax Факс] [-iptel IP-телефон] [-webpg Web-страница] [-title
Должность] [-dept Отдел] [-company Компания] [-mgr Менеджер]
[-hmdir ОсновнойКаталог] [-hmdrv БуквaДиска:] [-profile Путь]
[-loscr Путь] [-mustchpwd {yes | no}] [-canchpwd {yes | no}]
[-reversiblepwd {yes | no}] [-pwdneverexpires {yes | no}]
[-acctexpires ЧислоДней] [-disabled {yes | no}] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-c] [-q]
[{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsmove

Перемещает или переименовывает объекты Active Directory.

```
dsmove DNОбъекта [-newname НовоеИмя] [-newparent DNРодителя]
[{-s Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль |
*}] [-q] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery computer

Ищет учетные записи компьютеров, удовлетворяющие заданному критерию.

```
dsquery computer [{НачальныйУзел | forestroot | domainroot}]
[-o {dn | rdn | samid}] [-scope {subtree | onelevel | base}]
[-name Имя] [-desc Описание] [-samid SAM-имя] [-inactive
ЧислоНедель] [-stalepwd ЧислоДней] [-disabled] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p { Пароль | *}] [-q] [-r]
[-gc] [-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery contact

Ищет информацию о контактах, соответствующих критерию.

```
dsquery contact [{НачальныйУзел | forestroot | domainroot}]
[-o {dn | rdn}] [-scope {subtree | onelevel | base}] [-name
Имя] [-desc Описание] [{-s Сервер | -d Домен}] [-u
ИмяПользователя] [-p { Пароль | *}] [-q] [-r] [-gc] [-limit
ЧислоОбъектов] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery group

Ищет учетные записи групп, соответствующие критерию.

```
dsquery group [{НачальныйУзел | forestroot | domainroot}] [-o
{dn | rdn | samid}] [-scope {subtree | onelevel | base}]
[-name Имя] [-desc Описание] [-samid SAM-имя] [{-s Сервер | -d
Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r] [-gc]
[-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery partition

Ищет разделы Active Directory, соответствующие критерию.

```
dsquery partition [-o {dn | rdn}] [-part Фильтр] [{-s Сервер |
-d Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r]
[-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery quota

Ищет квоты дисков, соответствующие критерию.

```
dsquery quota {domainroot | DNОбъекта } [-o {dn | rdn}] [-acct
Имя] [-qlimit Фильтр] [-desc Описание] [{-s Сервер | -d
Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r]
[-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery server

Ищет контроллеры доменов, соответствующие критерию.

```
dsquery server [-o {dn | rdn}] [-forest] [-domain ИмяДомена]
[-site ИмяСайта] [-name Имя] [-desc Описание] [-hasfsmo
{schema | name | infr | pdc | rid}] [-isgc] [{-s Сервер | -d
Домен}] [-u ИмяПользователя] [-p {Пароль | *}] [-q] [-r] [-gc]
[-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery site

Ищет сайты Active Directory, соответствующие критерию.

```
dsquery site [-o {dn | rdn}] [-name Имя] [-desc Описание] [{-s
Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль | *}]
[-q] [-r] [-gc] [-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery user

Ищет учетные записи пользователей, соответствующие критерию.

```
dsquery user [{НачальныйУзел | forestroot | domainroot}]
[-o {dn | rdn | upn | samid}] [-scope {subtree | onelevel |
base}] [-name Имя] [-desc Описание] [-upn UPN] [-samid SAM-
имя] [-inactive ЧислоНедель] [-stalepwd ЧислоДней] [-disabled]
[{-s Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль |
*}] [-q] [-r] [-gc] [-limit ЧислоОбъектов] [{-uc | -uco |
-uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsquery *

Ищет объекты Active Directory любого типа, соответствующие критерию.

```
dsquery * [{НачальныйУзел | forestroot | domainroot}] [-scope
{subtree | onelevel | base}] [-filter LDAP-фильтр] [-attr
{СписокАтрибутов | *}] [-attrsonly] [-l] [{-s Сервер | -d
Домен}] [-u ИмяПользователя] [-p {Пароль| *}] [-q] [-r] [-gc]
[-limit ЧислоОбъектов] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

dsrm

Удаляет объекты Active Directory.

```
dsrm DNОбъекта ... [-subtree [-exclude]] [-noprompt] [{-s
Сервер | -d Домен}] [-u ИмяПользователя] [-p {Пароль | *}]
[-c] [-q] [{-uc | -uco | -uci}]
```

Для использования команды в Windows XP Professional установите Windows Server 2003 Administration Tools Pack.

echo

Выводит сообщения или включает/выключает отображение команд на экране.

```
echo [on | off]
echo [Сообщение]
```

endlocal

Отмечает конец локализации (локальной области видимости) переменных окружения в пакетном файле.

```
endlocal
```

erase

См. DEL.

eventcreate

Создает пользовательские события в журналах событий.

В Windows Server 2003:

```
eventcreate [/s Компьютер [/u Домен\Пользователь [/p Пароль]]
{[/l {application | system}] | [/so ИмяИсточника]} /t {success
| error | warning | information} /id ИдентификаторСобытия /d
Описание
```

В Windows XP Professional:

```
eventcreate [/s Компьютер [/u Домен\Пользователь [/p Пароль]]
{[/l {application | system}] | [/so ИмяИсточника]} /t {error |
warning | information} /id ИдентификаторСобытия /d Описание
```

eventquery

Просматривает журналы событий и выбирает записи о событиях.

```
eventquery [/s Компьютер [/u [Домен\]Пользователь [/p
Пароль]]] [/fi ИмяФильтра] [/fo {table | list | csv}]
[/r ДиапазонСобытий] [/nh] [/v] [/l [application] [system]
[security] ["dns server"] [ПользовательскийЖурнал]
[КаталогЖурнала] [*]]
```

eventtriggers /create

Создает триггер события (event trigger) и задает выполняемую операцию.

```
eventtriggers /create [/s Компьютер [/u [Домен\]Пользователь
[/p [Пароль]]]] /tr Имя [/l Журнал] [/d Описание] /tk Задание
{[/eid ИдентификаторСобытия] | [/t Тип] | [[/so Источник]}
[/ru ИмяПользователя [/rp Пароль}}
```

eventtriggers /delete

Удаляет триггер события.

```
eventtriggers /delete /tid {ИдентификаторСобытия [...] | *}
[/s Компьютер [/u [Домен\] Пользователь [/p Пароль]]]
```

eventtriggers /query

Показывает триггеры событий, установленные в данный момент в указанной системе.

В Windows Server 2003:

```
eventtriggers /query [/s Компьютер [/u [Домен\]Пользователь
[/p Пароль]]] [fo{table | list | csv}] [/nh] [/v] [/id]
```

В Windows XP Professional:

```
eventtriggers /query [/s Компьютер [/u [Домен\]Пользователь
[/p Пароль]]] [fo{table | list | csv}] [/nh] [/v]
```

exit

Завершает работу интерпретатора команд.

```
exit [/b] [КодЗавершения]
```

expand

Распаковывает файлы.

```
expand [-r] Источник Результат
expand -r Источник [Результат]
expand -d Источник.cab [-f: Файлы]
expand Источник.cab -f:Файлы Результат
```

fc

Сравнивает файлы и показывает различия.

```
fc [/a] [/c] [/l] [/lbn] [/n] [/t] [/u] [/w]
   [/Число][/offline][Диск1:][Путь1]ИмяФайла1
   [Диск2:][Путь2]ИмяФайла2
fc /b [Диск1:][Путь1]ИмяФайла1
   [Диск2:][Путь2]ИмяФайла2
```

find

Ищет текстовую строку в одном или нескольких файлах.

```
find [/v] [/c] [/n] [/i] [/offline] "Строка"
[[Диск:][Путь]ИмяФайла[ ...]]
```

findstr

Ищет в файлах строки, соответствующие регулярному выражению.

```
findstr [/b] [/e] [/l] [/r] [/s] [/i] [/x] [/v] [/n] [/m] [/o]
[/p] [/f:Файл] [/a:Атрибуты] [/c:Строка] [/d:Каталог]
[/g:Файл] [/offline] [Строки] [[Диск:][Путь]ИмяФайла[ ...]]
```

for

Выполняет заданную команду для каждого файла в наборе.

Синтаксис команды FOR при вызове из командной строки:

```
for %Переменная in (Набор) do Команда [Параметры]
for /d %Переменная in (Набор) do Команда [Параметры]
for /r [[Диск:]Путь] %Переменная in (Набор) do Команда [Параметры]
for /l %Переменная in (Начало,Шаг,Конец) do Команда [Параметры]
for /f ["Ключи"] %Переменная in (Набор) do Команда [Параметры]
```

Синтаксис команды FOR при вызове из сценария:

```
for %%Переменная in (Набор) do Команда [Параметры]
for /d %%Переменная in (Набор) do Команда [Параметры]
for /r [[Диск:]Путь] %%Переменная in (Набор) do Команда
[Параметры]
for /l %%Переменная in (Начало,Шаг,Конец) do Команда [Параметры]
for /f ["Ключи"] %%Переменная in (Набор) do Команда [Параметры]
```

forcedos

Запускает программу в командной оболочке MS-DOS (command.com), а не в командной оболочке Windows (cmd.exe):

```
forcedos [/d Каталог] ИмяФайла [Параметры]
```

forfiles

Выбирает один или несколько файлов и выполняет команду для каждого из них.

```
forfiles [/p Путь] [/m ШаблонПоиска] [/s]  [/c Команда]
[/d [+ | -] {ММ/ДД/ГГГГ | ДД}]
```

Команда доступна только в Windows Server 2003.

format

Форматирует дискету или жесткий диск.

```
format Диск: [/fs:ФайловаяСистема] [/v:Метка] [/q] [/a:Размер]
[/c] [/x]
format Диск: [/v:Метка] [/q] [/f:Размер]
format Диск: [/v:Метка] [/q] [/t:Треки /n:Секторы]
format Диск: [/v:Метка] [/q]
```

freedisk

Определяет, имеется ли на заданном диске локальной или удаленной системы нужный объем свободного дискового пространства.

```
freedisk [/s Компьютер [/u [Домен\]Пользователь
[/p [Пароль]]]] [/d Диск] [Значение]
```

Команда доступна только в Windows Server 2003.

ftp

Передает файлы.

```
ftp [-v] [-d] [-i] [-n] [-g] [-s:ИмяФайла] [-a]
[-w:РазмерБуфера] [Хост]
```

ftype

Показывает или изменяет типы файлов, используемые в сопоставлениях расширений файлов.

```
ftype [ТипФайла[=[Команда]]]
```

getmac

Показывает информацию о сетевом адаптере.

```
getmac [/s Компьютер [/u [Домен]\Пользователь [/p [Пароль]]]]
[/fo {table|list|csv}] [/nh] [/v]
```

gettype

Получает от операционной системы базовую информацию о конфигурации, в том числе имя хоста, название и версию операционной системы, а также сведения о ролях и компонентах.

```
gettype [/s Компьютер [/u [Домен]\Пользователь [/p [Пароль]]]]
[/role | /sp | /ver | /minv | /majv | /type | /build]
```

Команда доступна только в Windows Server 2003.

goto

Выполняет переход на строку сценария, обозначенную меткой.

```
goto :Метка
goto :EOF
```

gpupdate

Выполняет фоновое обновление политики группы.

```
gpupdate [/target:{computer | user}] [/force]
[/wait:<Значение>] [/logoff] [/boot] [/sync]
```

hostname

Показывает имя компьютера.

```
hostname
```

if

Выполняет команду по условию в пакетном файле.

```
if [not] errorlevel Значение Команда
if [not] [/i] Строка1== Строка2 Команда
if [not] exist ИмяФайла Команда
if [/i] Строка1 ОператорСравнения Строка2 Команда
if cmdextversion Значение Команда
if defined Переменная Команда
```

inuse

Указывает, что при перезагрузке файлы операционной системы будут заменены.

```
inuse НовыйПуть/Файл ТекущийПуть/Файл [/y]
```

Команда доступна только в Windows Server 2003.

ipconfig

Показывает конфигурацию TCP/IP.

```
ipconfig [/all] | [/release [Адаптер] | /renew [Адаптер]]
ipconfig /flushdns | /displaydns | /registerdns
ipconfig /showclassid Адаптер
ipconfig /setclassid Адаптер
[УстанавливаемыйИдентификаторКласса]]
```

label

Создает, изменяет или удаляет метку тома диска.

```
label [Диск:][Метка]
label [/mp] [Том] [Метка]
```

md

Создает каталог или подкаталог.

```
mkdir [Диск:]Путь
md [Диск:]Путь
```

memmonitor

Показывает сведения об использовании памяти отдельными процессами.

```
memmonitor {/p ИдентификаторПроцесса | /pn Имя | /ps Служба}
[/wait] [/nodbg] [/assumedbg] [/int Секунды] [/ws Значение]
[/ppool Значение] [/nppool Значение] [/vm Значение]
```

Windows Server 2003 Resource Kit.

memtriage

Помогает выявить источник утечек памяти.

```
memtriage {/m | /p | /mp | /h ИдентификаторПроцесса}
[/t ЧислоПовторов /w Длина] ФайлЖурнала.log
memtriage /s [/r ФайлПравил.ini /pid ИдентификаторПроцесса]
ФайлЖурнала.log
memtriage {/a [/pid ИдентификаторПроцесса] | /av}
ФайлЖурнала.log
```

Windows Server 2003 Resource Kit.

mkdir

См. MD.

more

Показывает вывод блоками, умещающимися в одном окне.

```
more [/e [/c] [/p] [/s] [/tn] [+n]] < [Диск:][Путь]ИмяФайла
more /e [/c] [/p] [/s] [/tn] [+n] [Файлы] ИмяКоманды | more
[/e [/c] [/p] [/s] [/tn] [+n]]
```

mountvol

Управляет точками монтирования (mount points) томов.

```
mountvol [Диск:]Путь ИмяТома
mountvol [Диск:]Путь /d
mountvol [Диск:]Путь /l
```

move

Перемещает файлы из одного каталога в другой в пределах одного диска.

```
move [/y] [/-y] [Источник] [Приемник]
```

nbtstat

Показывает состояние NetBIOS.

```
nbtstat [-a Узел] [-A IP-адрес] [-c] [-n] [-r] [-R] [-RR] [-s]
[-S] [Интервал] ]
```

Примечание Ключи команды чувствительны к регистру букв.

net accounts

Управляет политиками учетных записей и паролей.

```
net accounts [/forcelogoff:{Минуты | no}]
 [/minpwlen:Длина]
 [/maxpwage:{Дни | unlimited}]
 [/minpwage:Дни]
 [/uniquepw:Число] [/domain]
```

net computer

Добавляет компьютер в домен или удаляет компьютер из него.

```
net computer \\ИмяКомпьютера {/add | /del}
```

net config server

Показывает или изменяет конфигурацию службы сервера.

```
net config server [/autodisconnect:Время]
   [/srvcomment:"Текст"] [/hidden:{yes | no}]
```

net config workstation

Показывает или изменяет конфигурацию службы рабочей станции.

```
net config workstation [/charcount:Байты]
[/chartime:Миллисекунды]
[/charwait:Секунды]
```

net continue

Возобновляет работу приостановленной службы.

```
net continue Служба
```

net file

Показывает сведения об открытых на сервере файлах. При запуске с ключом /close закрывает файл и снимает его блокировки.

```
net file [Идентификатор [/close]]
```

net group

Показывает или изменяет глобальные группы.

```
net group [ИмяГруппы [/comment:"Текст"]]
  [/domain]
net group ИмяГруппы {/add [/comment:"Текст"]
  | /delete} [/domain]
net group ИмяГруппы ИмяПользователя [...]
  {/add | /delete} [/domain]
```

net localgroup

Показывает учетные записи локальной группы.

```
net localgroup [ИмяГруппы [/comment:"Текст"]] [/domain]
```

Создает учетную запись локальной группы.

```
net localgroup ИмяГруппы {/add [/comment:"Текст"]} [/domain]
```

Изменяет учетные записи локальной группы.

```
net localgroup [ИмяГруппы Имя [ ...] /add [/domain]
```

Удаляет учетную запись локальной группы.

```
net localgroup ИмяГруппы /delete  [/domain]
```

net name

Показывает или изменяет имена получателей, используемые службой сообщений.

```
net name [Имя [/add | /delete]]
```

net pause

Приостанавливает службу.

```
net pause Служба
```

net print

Показывает задания на печать и общие очереди и управляет ими.

```
net print \\ИмяКомпьютера\ИмяПринтера
net print [\\ИмяКомпьютера] Задание# [/hold | /release |
/delete]
```

net send

Отправляет сообщение с помощью службы сообщений.

```
net send {Имя | * | /domain[:Имя] | /users} сообщение
```

net session

Выводит список сеансов или завершает сеансы.

```
net session [\\ИмяКомпьютера] [/delete]
```

net share

Показывает список сетевых (общих) принтеров и каталогов и управляет ими.

```
net share [ИмяОбщегоРесурса]
net share ИмяОбщегоРесурса[=Путь:Диск]
  [/users:Число | /unlimited]
  [/remark:"Текст"]
  [/cache:Флаг]
net share {ИмяОбщегоРесурса | ИмяУстройства |
  Диск:Путь} /delete
```

net start

Запускает сетевую службу или выводит список работающих сетевых служб.

```
net start [Служба]
```

net statistics

Показывает статистику службы рабочей станции или сервера.

```
net statistics [workstation | server]
```

net stop

Останавливает службу.

```
net stop Служба
```

net time

Показывает или синхронизирует сетевое время.

```
net time [\\ИмяКомпьютера |
  /domain[:ИмяДомена] |
  /rtsdomain[:ИмяДомена]] [/set]
net time [\\ИмяКомпьютера] /querysntp
net time [\\ИмяКомпьютера]
  /setsntp[:СписокСерверов]
```

net use

Показывает удаленные соединения и управляет ими.

```
net use [ИмяУстройства | *]
[\\ИмяКомпьютера\ИмяОбщегоРесурса[\Том] [Пароль | *]]
[/user:[Домен\]ИмяПользователя] [/user:[ИмяПользователя@Домен]
[[/delete] | [/persistent:{yes | no}]] [/smartcard]
[/savecred]
net use [ИмяУстройства | *] [Пароль | *]] [/home]
net use [/persistent:{yes | no}]
```

net user

Создает локальные учетные записи пользователей.

```
net user ИмяПользователя [Пароль | *] /add [/active:{no |
yes}] [/comment:"Текст"] [/countrycode:Код] [/expires:{{ММ/ДД
/ГГГГ | ДД/ММ/ГГГГ | ммм,дд,ГГГГ} | never}] [/fullname:"Имя"]
[/homedir:Путь] [/passwordchg:{yes | no}] [/passwordreq:{yes |
no}] [/profilepath:[Путь]] [/scriptpath:Путь] [/times:{День
[-День][,День[-День]] ,Время[-Время][,Время[-Время]] [;...] |
all}] [/usercomment:"Текст"]
[/workstations:{ИмяКомпьютера[,...] | *}] [/domain]
```

Изменяет локальные учетные записи пользователей.

```
net user [ИмяПользователя [Пароль | *] [/active:{no | yes}]
[/comment:"Текст"] [/countrycode:Код] [/expires:{{ММ/ДД/ГГГГ |
ДД/ММ/ГГГГ | ммм,дд,ГГГГ} | never}] [/fullname:"Имя"]
[/homedir:Путь] [/passwordchg:{yes | no}] [/passwordreq:{yes |
no}] [/profilepath:[Путь]] [/scriptpath:Путь] [/times:{День
[-День][,День[-День]] ,Время[-Время][,Время[-Время]] [;...] |
all}] [/usercomment:"Текст"]
[/workstations:{ИмяКомпьютера[,...] | *}]] [/domain]
```

Удаляет локальные учетные записи пользователей.

```
net user ИмяПользователя [/delete] [/domain]
```

net view

Показывает сетевые ресурсы или компьютеры.

```
net view [\\ИмяКомпьютера [/cache] |
  /domain[:ИмяДомена]]
net view /network:nw [\\ИмяКомпьютера]
```

Netsh

Запускает специальную командную оболочку, позволяющую управлять конфигурациями различных сетевых служб на локальных и удаленных компьютерах.

```
netsh
```

 **Примечание** О работе с Netsh см. главу 15.

netstat

Показывает состояние сетевых соединений.

```
netstat [-a] [-e] [-n] [/o] [-s] [-p Протокол] [-r] [Интервал]
```

nslookup

Сообщает информацию DNS.

```
nslookup [-Параметр] [Компьютер | Сервер]
```

ntbackup

Создает резервные копии файлов.

```
ntbackup backup [systemstate] "@ИмяФайлаВыборки"
 /j {"ИмяЗадания"}
 [/p {"ИмяПула"}]
 [/t {"ИмяЛенточногоУстройства"}]
```

```
[/n {"ИмяНосителя"}]
[/f {"ИмяФайла"}]
[/d {"Описание"}]
[/ds {"ИмяСервера"}]
[/is {"ИмяСервера"}]
[/g {"GUID-имя"}]
[/a] [/v:{yes|no}] [/r:{yes|no}]
[/l:{f|s|n}] [/m {ТипРезервирования}]
[/rs:{yes|no}] [/hc:{on|off}]
[/snap{on|off}]
```

path

Показывает или устанавливает путь поиска исполняемых файлов для текущего окна командной оболочки.

```
path [[Диск:]Путь[;...][;%PATH%]
path ;
```

pathping

Трассирует маршруты и выводит информацию о потере пакетов.

В Windows Server 2003:

```
pathping [-n] [-h МаксЧислоПереходов] [-g СписокХостов]
 [-i Адрес] [-p Интервал]
 [-q ЧислоЗапросов [-w Таймаут]
 ИмяАдресата [-4] [-6]
```

В Windows XP Professional:

```
pathping [-n] [-h МаксЧислоПереходов] [-g СписокХостов]
 [-i Адрес] [-p Период] [-T] [-R] [-P]
 [-q ЧислоЗапросов [-w Таймаут]
 ИмяАдресата [-4] [-6]
```

Примечание Ключи команды чувствительны к регистру букв.

pause

Приостанавливает выполнение сценария и ожидает ввода с клавиатуры.

```
pause
```

pfmon

Выводит сведения о программных и аппаратных ошибках страниц.

```
pfmon [/n | /l] [/c | /h] [/k | /K] [/p ИдентификаторПроцесса]
[/d] КомандняяСтрокаПриложения
```

Windows Server 2003 Resource Kit.

Примечание Ключи команды чувствительны к регистру букв.

ping

Определяет, можно ли установить сетевое соединение.

```
ping [-t] [-a] [-n Число] [-l Размер] [-f]
 [-i TTL] [-v ТипСлужбы]
 [-r ЧислоПереходов] [-s ЧислоПереходов]
 [[-j СписокХостов] | [-k СписокХостов]]
 [-w Таймаут] СписокАдресатов
```

pmon

Показывает снимок задействованных ресурсов и выполняемых процессов.

```
pmon
```

Windows Server 2003 Resource Kit.

popd

Делает текущим каталог, сохраненный PUSHD.

```
popd
```

print

Печатает текстовый файл.

```
print [/d:Устройство] [[Диск:][Путь]ИмяФайла[...]]
```

printdriverinfo

Показывает информацию обо всех драйверах принтеров, сконфигурированных на локальной или удаленной системе.

```
printdriverinfo [/s: ИмяСервера] [/p: ИмяПринтера]
[/d: ИмяДрайвера] [/f: ИмяФайла]
```

Windows Server 2003 Resource Kit.

prompt

Изменяет приглашение командной строки Windows.

```
prompt [Текст]
```

pushd

Сохраняет текущий каталог, затем делает текущим заданный каталог.

```
pushd [Путь | ..]
```

rd

Удаляет каталог.

```
rmdir [/s] [/q] [Диск:]Путь
rd [/s] [/q] [Диск:]Путь
```

recover

Восстанавливает сохранившуюся информацию с поврежденного или дефектного диска.

```
recover [Диск:][Путь]ИмяФайла
```

reg add

Добавляет раздел или параметр в реестр.

```
reg add ИмяРаздела [/v ИмяПараметра | /ve] [/t ТипДанных]
[/d Данные] [/f] [/s Разделитель]
```

reg compare

Сравнивает разделы или параметры реестра.

```
reg compare ИмяРаздела1 ИмяРаздела2 [/v ИмяПараметра | /ve]
[/s] [/КлючВывода]
```

reg copy

Копирует записи реестра в заданный раздел реестра локальной или удаленной системы.

```
reg copy ИмяРаздела1 ИмяРаздела2 [/s] [/f]
```

reg delete

Удаляет раздел или параметр реестра.

```
reg delete ИмяРаздела [/v ИмяПараметра | /ve | /va] [/f]
```

reg query

Перечисляет параметры и подразделы в заданном разделе реестра.

В Windows Server 2003:

```
reg query ИмяРаздела [/v ИмяПараметра | /ve] [/s] [/d Данные]
[/k] [/d] [/c] [/e] [/t Тип] [/z] [/se Разделитель]
```

В Windows XP Professional:

```
reg query ИмяРаздела [/v ИмяПараметра | /ve] [/s]
```

reg restore

Записывает в реестр параметры и разделы, ранее сохраненные в файле.

```
reg restore ИмяРаздела "ИмяФайла"
```

reg save

Сохраняет в файл копию заданных разделов, параметров и значений.

```
reg save ИмяРаздела "ИмяФайла"
```

regsvr32

Регистрирует DLL или отменяет ее регистрацию.

```
regsvr32 [/u] [/s] [/n] [/i[:КомандняяСтрока]] ИмяDLL
```

rem

Добавляет комментарий в сценарий.

```
rem [Комментарий]
```

ren

Переименовывает файл.

```
rename [Диск:][Путь]ИмяФайла1 ИмяФайла2
ren [Диск:][Путь]ИмяФайла1 ИмяФайла2
```

rmdir

См. RD.

route

Управляет таблицами маршрутизации.

```
route [-f] [-p] [Команда [Адресат] [mask СетеваяМаска] [Шлюз]
[metric Метрика]] [if Интерфейс]
```

runas

Запускает программу с разрешениями указанного пользователя.

В Windows Server 2003:

```
runas [/noprofile | /profile] [/env] [/netonly]
 [/savecred] /user: УчетнаяЗапись Программа
runas [/noprofile | /profile] [/env] [/netonly]
 [/savecred] /smartcard [/user:УчетнаяЗапись] Программа
```

В Windows XP Professional:

```
runas [/noprofile | /profile] [/env] [/netonly]
 /user: УчетнаяЗапись Программа
runas [/noprofile | /profile] [/env] [/netonly]
 /smartcard [/user: УчетнаяЗапись] Программа
```

sc config

Настраивает учетные записи регистрации и запуска служб.

```
sc [\\ИмяСервера] config ИмяСлужбы [type= {own|share|{interact
type = {own |  share}}| kernel | filesys | rec | adapt}]
  [start= {boot|system|auto|demand|disabled}]
  [error= {normal|severe|critical|ignore}]
  [binPath= ИсполняемыйФайлСлужбы]
  [group= ГруппаПорядкаЗагрузки]
  [tag= {yes|no}]
  [depend= Зависимости]
  [obj= {ИмяУчетнойЗаписи|ИмяОбъекта}]
  [DisplayName= ОтображаемоеИмя]
  [password= Пароль]
```

sc continue

Возобновляет работу приостановленной службы.

```
sc [\\ИмяСервера] continue ИмяСлужбы
```

sc failure

Задает операции, которые выполняются при сбое службы.

```
sc [\\ИмяСервера] failure ИмяСлужбы [reset=
ИнтервалСбросаСчетчика] [reboot= ШироковещательноеСообщение]
[command= КомандныйСтрока] [actions= Действие]
```

sc pause

Приостанавливает службу.

```
sc [\\ИмяСервера] pause ИмяСлужбы
```

sc qc

Показывает информацию о конфигурации заданной службы.

```
sc [\\ИмяСервера] qc ИмяСлужбы [РазмерБуфера]
```

sc qfailure

Сообщает об операции, выполняемой при сбое службы.

```
sc [\\ИмяСервера] qfailure ИмяСлужбы [РазмерБуфера]
```

sc query

Перечисляет службы, настроенные в данной системе.

```
sc [\\ИмяСервера] query ИмяСлужбы
 [type= {driver | service | all}]
 [type= {own|share|interact|kernel|filesys|rec|adapt}]
 [state= {active | inactive | all}] [bufsize= РазмерБуфера]
 [ri= ИндексВозобновления]
 [group= ИмяГруппы]
```

sc start

Запускает службу.

```
sc [\\ИмяСервера] start ИмяСлужбы [Аргументы]
```

sc stop

Останавливает службу.

```
sc [\\ИмяСервера] stop ИмяСлужбы
```

schtasks /change

Изменяет свойства запланированного задания.

В Windows Server 2003:

```
schtasks /change /tn ИмяЗадания [/s Система [/u [Домен\]
Пользователь [/p [Пароль]]]] ИзменяемыеПараметры
```

```
{[/ru Пользователь[ ]/rp Пароль] [/tr ИсполняемоеЗадание]}
[/st ВремяЗапуска] [/ri Интервал] [{/et ВремяОкончания|
/du Длительность} [/k]] [/sd ДатаЗапуска] [/ed ДатаОкончания]
[enable | disable] [/it] [/z]
```

В Windows XP Professional:

```
schtasks /change /tn ИмяЗадания [/s Система
[/u [Домен\]Пользователь [/p [Пароль]]]] ИзменяемыеПараметры
{[/ru Пользователь[ ]/rp Пароль] [/tr ЗапускаемаяПрограмма]}
```

schtasks /create

Создает задания, выполняемые по расписанию.

В Windows Server 2003:

```
schtasks /create [/s Система [/u [Домен\]Пользователь
[/p [Пароль]]]] [/ru [Домен\]ИмяПользователя [rp Пароль]]/tn
ИмяЗадания /tr ИсполняемоеЗадание /sc Расписание [/mo
Модификатор] [/d День] [/i ВремяПростоя] [/st ВремяЗапуска]
[/m Месяц [, Месяц [...]]] [/sd ДатаЗапуска] [/ed
ДатаОкончания] [/ri Интервал] [{/et ВремяОкончания | /du
Длительность} [/k]] [/it] [/z] [/f]
```

В Windows XP Professional:

```
schtasks /create [/s Система [/u [Домен\]Пользователь
[/p [Пароль]]]] [/ru [Домен\]ИмяПользователя [rp Пароль]]/tn
ИмяЗадания /tr ИсполняемоеЗадание /sc Расписание [/mo
Модификатор] [/d День] [/i ВремяПростоя] [/st Время_Запуска]
[/m Месяц [, Месяц [...]]] [/sd ДатаЗапуска] [/ed
ДатаОкончания]
```

schtasks /delete

Удаляет запланированное задание.

```
schtasks /delete /tn {ИмяЗадания | *} [/f] [/s Компьютер
[/u [Домен\]Пользователь [/p [Пароль]]]]
```

schtasks /end

Останавливает выполняемое задание.

```
schtasks /end /tn ИмяЗадания [/s Компьютер
[/u [Домен\]Пользователь [/p [Пароль]]]]
```

schtasks /query

Перечисляет задания, запланированные на локальном или на заданном компьютере.

```
schtasks /query [/s Компьютер [/u [Домен\]Пользователь
[/p [Пароль]]]] [/fo {table | list | csv}] [/nh/ [/v]
```

schtasks /run

Запускает запланированное задание.

```
schtasks /run /tn ИмяЗадания [/s Компьютер
[/u [Домен\]Пользователь [/p [Пароль]]]]
```

set

Показывает или изменяет переменные окружения Windows. Кроме того, используется для вычисления числовых выражений в командной строке.

```
set [Переменая=[Строка]]
set /a Выражение
set /p Переменная=[СтрокаПриглашения]
```

setlocal

Отмечает начало локализации (локальной области видимости) переменных окружения в пакетных сценариях.

```
setlocal
setlocal {enableext | disableext}
```

sfc

Сканирует и проверяет защищенные системные файлы.

```
sfc [/scannow] [/scanonce] [/scanboot] [/revert] [/purgecache]
[/cachesize=Размер]
```

shift

Сдвигает позиции замещаемых параметров в сценариях.

```
shift [/n]
```

shutdown

Выключает или перезапускает компьютер.

В Windows Server 2003:

```
shutdown [{-i |-l|-s|-r|-a}] [-f] [-m [\\ИмяКомпьютера]]
[-t nn] [-c "Сообщение"] [-d [p]:n1:n2] [-h] [-e] [-p]
```

В Windows XP Professional:

```
shutdown [{-i |-l|-s|-r|-a}] [-f] [-m [\\ИмяКомпьютера]]
[-t nn] [-c "Сообщение"] [-d[u][p]:n1:n2]
```

sort

Сортирует ввод.

```
[Команда |] sort [/r] [/+n] [/m Кб] [/l
Идентификатор_локализации] [/rec СимволовЗаписи]
[Диск1:][Путь1]ИмяФайла1] [/t [Диск2:][Путь2]]
[/o [Диск3:][Путь3]ИмяФайла3]
```


Примечание В данном случае «|» — символ конвейеризации.

splinfo

Показывает информацию о спулинге печати.

```
splinfo [/z] [/v] [/d] [\\ИмяКомпьютера]
```

Windows Server 2003 Resource Kit.

start

Запускает заданную программу или команду в новом окне командной оболочки.

```
start ["Заголовок"] [/d Путь] [/i] [/min] [/max] [/separate|
/shared] [/wait] [/b] [/low | /belownormal | /normal |
/abovenormal | /high | /realtime] [Команда/Программа]
[Параметры]
```

subst

Сопоставляет заданный путь с буквой диска.

```
subst [Диск1: [Диск2:]Путь]
subst Диск1: /d
```

systeminfo

Показывает подробные сведения о конфигурации.

```
systeminfo [/s Компьютер [/u [Домен\]Пользователь
[/p [Пароль]]]] [/fo {table|list|csv}] [/nh]
```

takeown

Позволяет администратору стать владельцем одного или нескольких файлов.

```
takeown [/s Компьютер [/u [Домен\]Пользователь [/p [Пароль]]]]
/f ИмяФайла [/a] [/r [/d Запрос]]
```

Команда доступна только в Windows Server 2003.

taskkill

Завершает выполняемые процессы по имени или идентификатору.

```
taskkill [/s Компьютер] [/u [Домен\]Пользователь [/p Пароль]]
{[/fi "Фильтр" [/fi Фильтр2 [ ... ]]]
[/pid ИдентификаторПроцесса|/im ИмяОбраза]} [/f][/t]
```

tasklist

Перечисляет все выполняемые процессы.

```
tasklist [/s Компьютер [/u [Домен\]Пользователь
[/p [Пароль]]]] [{/m Модуль | /svc | /v}] [/fo {table | list |
csv}] [/nh] [/fi ИмяФильтра1 [/fi ИмяФильтра2 [ ... ]]]
```

time

Показывает или устанавливает системное время.

```
time [Время | /T]
```

timeout

Задает время ожидания или ждет нажатия клавиши в пакетном сценарии.

```
TIMEOUT /t ВремяОжидания [/nobreak]
```

title

Задает заголовок окна командной оболочки.

```
title [Строка]
```

tracert

Показывает маршрут между компьютерами.

```
tracert [-d] [-h МаксЧислоПереходов] [-j СписокХостов]
[-w ПериодОжидания] ИмяАдресата
```

type

Показывает содержимое текстового файла.

```
type [Диск:][Путь]ИмяФайла
```

ver

Показывает версию Windows.

```
ver
```

verify

Включает или отключает проверку правильности записи файлов на диск.

```
verify [on | off]
```

vol

Показывает метку и серийный номер тома диска.

```
vol [Диск:]
```

waitfor

Указывает, что перед продолжением работы система должна дождаться определенного сигнала, или отправляет сигнал.

Синтаксис отправки сигнала:

```
waitfor [/s Компьютер [/u [Домен\]Пользователь [/p [Пароль]]]]
/si Сигнал
```

Синтаксис ожидания сигнала:

```
waitfor [/t ПериодОжидания] Сигнал
```

Команда доступна только в Windows Server 2003.

where

Показывает список файлов, соответствующих шаблону поиска.

```
where [/r Каталог] [/q] [/f] [/t] Шаблон
where [/q] [/f] [/t] $Переменная:Шаблон
where [/q] [/f] [/t] Путь:Шаблон
```

Команда доступна только в Windows Server 2003.

whoami

Показывает регистрационную информацию и сведения о защите для текущего пользователя.

В Windows Server 2003:

```
whoami [/upn | /fqdn | /logonid]
whoami /all [/fo {table|list|csv}] [/nh]
```

В Windows XP Professional:

```
whoami [/all][/user [/sid] [/logonid [/sid] [/groups [/sid]
[/priv [/sid] [/noverbose]]
```

В Windows XP Professional эта команда доступна через Support Tools.

Предметный указатель

Специальные символы

$ 257
@ 33, 37, 59

A

ACL 415
Active Directory 242
— контейнеры 244
— — по умолчанию 246
— логические и физические структуры 244–245
— объекты 244, 248
— — переименование и перемещение 260
— — удаление 261–262
— поиск контроллеров домена 276–277
— публикация принтеров 348
— средства командной строки 247
Auto Check (настройка параметров) 200

B

BDC 280

C

Check Disk (Chkdsk) 196
CScript 130
— хост сценариев по умолчанию 335

D

DiskPart 172–192, 418
— команды 175–177
— сценарий
— — коды ошибок 178
— — пример 179
DN 245–246
— относительное 246
DNS (Domain Name System) 243
— сервер
— — задание 369
— — настройка 367–368

E

EFI (Extensible Firmware Interface) 207
EMF 328
Event Log (Журнал событий) 125
Eventquery 129

F

Fast User Switching 71
File and Printer Sharing for Microsoft Networks (Служба доступа к файлам и принтерам сетей Microsoft) 69
FSUtil 192, 201, 217
— команды 193–194
— применение 194–196

G

GPT 176, 185
— разделы 186

I

ICMP 378–382
IP-адрес
— динамический 366
— добавление 367
— проверка 375–377
— статический 365–366

L

LocalSystem 116

M

MBR 185
— разделы 185–186
Memory Monitor; *см.* команда, memmonitor
MTU 377

N

Netsh 359, 436
— контексты 359–361
NTDSUtil 265, 284–286

O

OU 244

P
Page Fault Monitor; *см.* команда, pfmon
PDC 280
Poolmon 167
Process Resource Monitor; *см.* команда, pmon

R
RAID 220
— создание 227–235
RAID-0 228–231
— восстановление 236–237
RAID-1 228, 231–233
RAID-5 228, 233–235
— восстановление 234, 238–239
RAW 328
Remote Desktop 73
Resource Leak Triage Tool; *см.* команда, memtriage
RTO 381
RTT 381

S
Scheduled Task Wizard (Мастер планирования заданий) 72
Scheduled Tasks (Назначенные задания) 72–73
Shutdown 120
SID 307
SNMP 344
System Configuration Utility 120
System Idle Process 150

T
Task Scheduler (Планировщик заданий) 68–69
TCP
— окно 381
— порты (наиболее распространенные) 407–408
— протоколы (наиболее распространенные) 407–408
TCP/IP
— удаление параметров 372
— получение и сохранение конфигурации 373–375
— порты
— — создание и изменение для принтеров 343–344
— — удаление (используемых принтерами) 346
— просмотр диагностической информации 389–392
TTL 371, 380

W
W3svc 116, 152
Windows Server 2003 Resource Kit Tools 15–16
Windows Support Tools for Windows XP Professional 7, 12–14
WINS-сервер (задание) 369–371

В
вложение 19

Г
группа 311
— Domain Computers (Компьютеры домена) 268
— безопасности 308
— глобальная 310
— добавление членов 319
— замена всех членов 320–321
— локальная 308, 314
— — домена 310
— определение членства 316–317
— распространения 308
— удаление членов 319–320
— универсальная 311

Д
дейтаграмма (доставка) 380
диск
— базовый 187–189
— — преобразование в динамический 190–191
— — назначение 208
— — смена 209
— — удаление 209
— дефрагментация 200–202
— динамический 187, 191
— — подключение 226
— — преобразование в базовый 191–192
— — уровни RAID 227–228
— зеркальный 228
— изменение типа 190
— проверка состояния и конфигурации 182–185
— разделы
— — EFI 203

— — GPT 207
— — MBR 204–206
— — MSR (Microsoft Reserved Partition) 207
— — получение информации 203–204
— — преобразование типов 186–187
— — расширение 218
— — создание активных 189–190
— — удаление 219
— — форматирование 210–213
— три особые области 189
— цилиндр 204
— чередование 228
— — с записью четности 228
домен 243
— контроллер 242
— — установка и удаление 275
дорожка 204

Ж

журнал событий
— запись собственных 135–137
— — просмотр и фильтрация 129–134

З

задание
— включение и выключение 83
— изменение свойств 81–82
— копирование и перемещение 82–83
— мониторинг назначенного 74–75
— на печать (приоритеты) 349
— назначение
— — через Scheduled Task Manager 76–81
— — через Schtasks 84–96
— немедленный запуск 83
— событийно-управляемое 70–71
— удаление 83
замена, горячая 182
значение (перебор наборов) 56–57

И

имя
— простое (CN) 246, 292–293
— составное; *см.* DN
интерфейс
— Local Area Connection 376
— замкнутый на себя 375
— проверка конфигурации 375–377
итератор 55

К

кавычки
— обратные 62
— одинарные 62
каталог
— глобальный 277–278
— — добавление или удаление 279
— — проверка параметров кэширования и настроек 280–281
— перебор 58–59
команда
— add 175
— add address 367
— add disk 233
— add dns 369
— add wins 370
— arp 414
— assign 175, 209
— assign letter 208
— assign mount 208
— assoc 5, 23, 414
— attrib 414
— automount 175
— behavior 193
— break 5, 175
— break disk 232, 236
— cacls 415
— call 5, 66, 415
— cd 5, 415
— chdir; *см.* cd
— chkdsk 196–198, 415
— chkntfs 199–200, 415
— choice 415
— cipher 416
— clean 176
— clean all 187
— clip 416
— cls 5, 33, 416
— cmd 416
— cmd /q 18
— cmdkey 416
— color 5, 33, 417

– comp 417
– compact 417
– connect ieproxy 407
– connect mail 406
– connect news 407
– connect to server 285
– convert 176, 186, 213–216, 417
– convert dynamic 230
– copy 5, 417
– create 176
– create partition efi 207
– create partition extended 206
– create partition logical 206
– create partition msr 207
– create partition primary 205, 207
– create volume raid 234
– create volume simple 224
– create volume stripe 231
– date 5, 417
– dcgpofix 417
– dcpromo 275–276
– defrag 75, 201–202, 418
– del 6, 418, 426
– delete 176
– delete arpcache 372
– delete partition 219, 227
– delete volume 227
– detail 176
– dir 5–6, 418
– dirty 193
– diskcomp 418
– diskcopy 418
– diskpart 172–192, 418
– doskey 419
– driverquery 99, 100, 419
– dsadd computer 264, 266–268, 419
– dsadd group 309, 312, 419
– dsadd user 288, 291–293, 419–420
– dsget computer 264, 269–271, 420
– dsget group 309, 316, 420–421
– dsget server 265, 277, 421
– dsget site 280–281
– dsget user 288, 298, 421–422
– dsget user –memberof 299
– dsmod computer 264, 272–273, 422
– dsmod group 309, 313, 422
– dsmod server 265, 279, 422
– dsmod user 288–301, 423
– dsmove 260–261, 274, 305, 321, 423
– dsmove –newname 306, 322
– dsquery 248
– – формат вывода имен 256–257
– dsquery * 250, 425
– dsquery computer 249, 252, 254–256, 423
– – документирование текущей конфигурации учетных записей 271
– dsquery contact 249, 424
– dsquery group 249, 315, 424
– dsquery ou 249
– dsquery partition 249, 424
– dsquery quota 250, 424
– dsquery server 250, 276, 424–425
– dsquery server 277
– dsquery site 250, 281, 425
– dsquery subnet 250
– dsquery user 250–255, 425
– dsrm 261–262, 275, 308, 323, 426
– echo 6, 33, 35–36, 426
– endlocal 6, 46, 426
– erase; *см.* del
– eventcreate 128, 135–137, 426
– eventquery 128, 130–131, 133–134, 426–427
– eventtriggers 128, 140–141
– eventtriggers /create 140–142, 427
– eventtriggers /delete 143–144, 427
– eventtriggers /query 142–143, 427
– eventvwr 127–128
– exit 6, 9, 176, 427
– expand 427
– extend 176, 218
– fc 428
– file 193
– find 25, 428
– findstr 428
– forcedos 428

— forfiles 429
— format 211–213, 429
— freedisk 429
— fsinfo 193
— ftp 429
— ftype 6, 23, 429
— getmac 429
— gettype 430
— goto 6, 45, 430
— — применение 65
— gpupdate 430
— hardlink 193
— help 176
— hostname 430
— import 176
— inactive 176
— interface portproxy 360
— inuse 430
— ipconfig 373, 431
— label 217, 431
— list 176
— list disk 173–174, 182–183
— list partition 173, 203–204
— list volume 173, 221
— md 6, 431
— memmonitor 163, 165–166, 431
— memsnap 167
— memtriage 163, 167–169, 431
— mkdir; *см.* md
— more 25, 432
— mountvol 432
— move 6, 432
— msconfig 120
— nbtstat 432
— net accounts 432
— net computer 432
— net config server 432
— net config workstation 433
— net continue 433
— net file 433
— net group 433
— net localgroup 309–310, 314, 433
— net name 434
— net pause 434
— net print 434
— net send 434
— net session 434
— net share 434
— net start 434
— net statistics 435
— net stop 435
— net time 435
— net use 180, 435
— net user 289–290, 296, 435–436
— net view 436
— netsh 359, 436
— netsh -c 364
— netsh dhcp 363–364
— netsh -r 362
— netstat 436
— nlsinfo 99
— now 98
— nslookup 436
— ntbackup 436–437
— objectid 193
— online 176, 226
— path 6, 20, 437
— pathping 437
— pause 6, 437
— pfmon 163–164, 438
— ping 438
— ping adapter 409–410
— ping dhcp 409
— ping dns 409
— ping gateway 410
— ping ieproxy 408
— ping ip 410
— ping iphost 409
— ping mail 408
— ping news 408
— ping wins 410
— pmon 145, 155–156, 160, 438
— popd 6, 438
— print 438
— printdriverinfo 326–329, 438
— prncnfg 341–352
— — ключи 350–352
— prncnfg -t 341, 346, 348–349
— prnjobs 354–355
— prnjobs -m 357
— prnjobs -x 358
— prnjobs -z 357
— prnmngr 336–339
— prnmngr -d 342
— prnmngr -g 341
— prnmngr -l 340
— prnport 343–344
— prnport -d 346
— prnport -l 345

— prnqctl 355–356
— prompt 6, 439
— pushd 6, 439
— quota 193
— rd 6, 439–440
— recover 439
— reg 103
— reg add 107–108, 439
— reg compare 104–106, 439
— reg copy 108, 439
— reg delete 109, 439
— reg query 104, 440
— reg restore 106–107, 440
— reg save 106, 440
— regsvr32 440
— rem 6, 33, 177, 440
— remove 177
— remove letter 210
— remove mount 210
— ren 6, 440
— rename; *см.* ren
— repair 177
— repair disk 234
— reparsepoint 193
— rescan 177
— retain 177
— rmdir; *см.* rd
— route 441
— runas 441
— sc 110
— sc config 114–116, 441
— sc continue 113, 441
— sc failure 117–119, 441–442
— sc pause 113, 442
— sc qc 112–113, 442
— sc qfailure 117, 442
— sc query 111–112, 442
— sc start 113–114, 442
— sc stop 113, 442
— schtasks 68
— schtasks /change 90–93, 442–443
— schtasks /create 84–90, 443
— — типы расписаний 85–86
— schtasks /delete 95–96, 443
— schtasks /end 94–95, 443
— schtasks /query 93, 443–444
— schtasks /run 94, 444
— select 177
— set 6, 40, 444
— set address 365–366
— set dns 368
— set wins 370
— setlocal 6, 46, 444
— setx 7, 21
— sfc 444
— shift 6, 444
— show adapter 390–392
— show all 389
— show computer 394–395
— show config 374
— show dns 405
— show gateway 404–405
— show icmp 378–379
— show ieproxy 393
— show interface 376–377
— show ip 404
— show ipaddress 376
— show ipstats 382–383
— show mail 393
— show news 393
— show os 399–401
— show tcpconn 385
— show tcpstats 386–387
— show test 413
— show udpconn 387–388
— show udpstats 388
— show wins 405
— shutdown 121–123, 444–445
— skip 61–62
— sort 445
— sparse 193
— spcheck 15
— splinfo 330–332, 445
— start 7, 445
— subst 445
— systeminfo 99–100, 445
— takeown 445–446
— taskkill 146, 161–163, 446
— tasklist 145, 148–150, 446
— tasklist /fi 153–155
— tasklist /m 152–153
— tasklist /svc 151
— time 7, 446
— timeout 446
— title 7, 33, 37, 446
— tracert 446
— type 7, 446
— usn 194
— ver 447

— verify 7, 447
— vol 7, 447
— volume 194
— waitfor 447
— where 98–99, 447
— whoami 98, 447–448
— xcopy 7
— внешняя 5, 7
— — MS-DOS 8–9
— внутренняя 5–7
— — MS-DOS 8
— группирование 27–29
— путь 20–21
— хронология 11–12
— цепочки 27–28
— эхо-отображение 35
конвейеризация 24–25
контекст
— aaaa 359
— bridge 360
— dhcp 360
— diag 360
— interface ip 360
— interface ipv6 360
— ipsec 360
— ras 361
— routing 361
— rpc 361
— wins 361
координатор
— именования доменов 282
— инфраструктуры 278
— операций
— — настройка ролей из командной строки 284–286
— — поиск 283–284
— относительных идентификаторов 282
— схемы 282
— эмулятора PDC 282
кэш
— ARP 371
— — удаление 371–372
— файловой системы 160

M

метка 64
многопоточность 159
монитор печати 328

H

набор
— зеркальный 235–237
— рабочий 166

O

область
— действия, локальная 46
— поиска 255
— чередующаяся 228
оболочка, командная
— MS-DOS 8
— инициализация 4
— коды цветов для окна 38
— математические операции 47–48
— настройка
— — заголовка и цветов 37–38
— — свойств 9–10
— определение 3
— последовательность событий при вводе команды 17–18
— управление
— — запуском 18–19
— — отображением текста и команд 35
— — эхо-отображением с помощью @ 36–37
окно
— Command Prompt Properties 10–11
— командной оболочки (очистка) 33
оператор 32–33
— for 6, 54–55, 428
— for /d 58
— for /f 60
— for /l 56
— for /r 59
— if 6, 45, 50–51, 53–54, 430
— if defined 52
— if...else 51
— if not defined 52
— if not 50, 52
— арифметический 48–49
— выбора в командной строке 50–54
— имитация возведения в степень 50
— приоритет 49–50

— присваивания 48–49
— сравнения 54
— цикла в командной строке 54

П

параметр
— /C 213
— /CvtArea 215–216
— /NoSecurity 215
— /X 212
— +Published 348
— active 175
— Addmbr 319
— AdminPasswordStatus 396
— All 210
— AutomaticResetBootOption 396
— AutomaticResetCapability 396
— BootDevice 401
— BootOptionOnLimit 396
— BootOptionOnWatchDog 396
— BootROMSupported 396
— BootupState 396
— BuildNumber 401
— BuildType 401
— Caption 396, 401
— ChassisBootupState 396
— Chmbr 320
— CodeSet 401
— CountryCode 401
— CreationClassName 396, 401
— CSCreationClassName 401
— CSDVersion 401
— CSName 401
— CurrentTimeZone 396, 401
— DaylightInEffect 396
— Debug 402
— delims 61
— Desc 252
— Description 396, 402
— Disabled 259
— Dismount 210
— Distributed 402
— DNSHostName 396
— Domain 278–279, 397
— DomainRole 397
— EnableDaylightSavings-Time 397
— EncryptionLevel 402
— eol 61–62
— eq 132, 154
— equ 54
— ForegroundApplication-Boost 402
— FreePhysicalMemory 402
— FreeSpaceInPagingFiles 402
— FreeVirtualMemory 402
— FrontPanelResetStatus 397
— ge 132, 154
— geq 54
— gt 132, 154
— gtr 54
— Hasfsmo 283
— InfraredSupported 397
— InitialLoadInfo 397
— InstallDate 397, 402
— KeyboardPasswordStatus 397
— LargeSystemCache 402
— LastBootUpTime 402
— LastLoadInfo 397
— le 132, 154
— leq 54
— Limit 256
— LocalDateTime 402
— Locale 402
— lss 54
— lt 132, 154
— Manufacturer 397, 402
— MaxNumberOfProcesses 402
— MaxProcessMemorySize 402
— Memberof 316
— Members 316
— Model 397
— Name 397, 402
— NameFormat 397
— ne 132, 154
— neq 54
— NetworkServerMode-Enabled 397
— Newparent 305
— Noerr 175
— NumberOfLicensedUsers 403
— NumberOfProcesses 403
— NumberOfProcessors 397
— NumberOfUsers 403
— OEMStringArray 397
— Organization 403
— OSLanguage 403
— OSProductSuite 403
— OSType 403
— OtherTypeDescription 403
— PartOfDomain 397

— PauseAfterReset 398
— PlusProductID 403
— PlusVersionNumber 403
— PowerManagementCapabilities 398
— PowerManagementSupported 398
— PowerOnPasswordStatus 398
— PowerState 398
— PowerSupplyState 398
— Primary 403
— PrimaryOwnerContact 398
— PrimaryOwnerName 398
— ProductType 403
— Published 348
— QuantumLength 403
— QuantumType 403
— RegisteredUser 403
— Reset 273
— ResetCapability 398
— ResetCount 398
— ResetLimit 399
— Rmmbr 319
— Roles 399
— Samid 252
— SerialNumber 403
— ServicePackMajorVersion 403
— ServicePackMinorVersion 404
— SizeStoredInPagingFiles 404
— Stalepwd 274
— Status 399, 404
— Subtree 262
— SuiteMask 404
— SupportContactDescription 399
— SystemDevice 404
— SystemDirectory 404
— SystemDrive 404
— SystemStartupDelay 399
— SystemStartupOptions 399
— SystemStartupSetting 399
— SystemType 399
— ThermalState 399
— tokens 61
— TotalPhysicalMemory 399
— TotalSwapSpaceSize 404
— TotalVirtualMemorySize 404
— TotalVisibleMemorySize 404
— usebackq 61–62
— UserName 399
— Version 404
— WakeUpType 399
— WindowsDirectory 404
— Workgroup 399
переменная
— errorlevel 40–41
— именование 42
— локализация области действия 46–47
— окружения 40–41
— — $username$ 302
— — %ErrorLevel% 135
— — %PathExt% 22
— — %SystemRoot% 3
— — %SystemRoot%\Tasks 75
— — %UserProfile% 5
— подстановка значений 44–45
— присвоение значений 43–44
перенаправление
— ввода-вывода 24
— — в файлы 25–26
— стандартного вывода другим командам 24–25
— стандартных ошибок 26
подпрограмма 63–66
подсеть 245
принтер
— настройка свойств 346–347, 350–352
— переименование 341
— перечисление 340
— удаление 342
— установка (сетевого) 339
проблемы
— в TCP/IP-сетях 389–413
— с TCP/IP-соединениями 406–407
— с базовой конфигурацией компьютера 394–404
— с запуском заданий 81
— с клиентами почты, новостей и прокси 393
— с конфигурациями IP, DNS и WINS 404–405
— с процессами 147
— с томами 222–223
— со спулингом 352–353
процедура 63, 66–67
процесс 146
— анализ 148–149
— идентификатор 146

— мониторинг 155–160
— останов 160–163
— приоритеты 158–159
— системный и пользовательский 146–147
пулы памяти 157

Р

рабочий стол, удаленный (соединение) 73
реестр 101–109
— параметры 101
— — просмотр 104
— — типы данных 102–103
— раздел 101–102
— — добавление 107
— — копирование 108
— — корневой 102
— — сохранение и восстановление 106–107
— — сравнение 104–105
— — удаление 109

С

сайт 245
сегмент 384
сектор 204
сервер печати 336
символ
— переадресации 25
— специальный 20
служба
— запуск, останов и приостановка 113
— контроллер 110
— настройка
— — восстановления 117–118
— — запуска 114–115
— — регистрации 115–116
— просмотр списка 110–111
— управление (системными) 110–119
событие 124–125
— типы 126
состояние
— Audio CD 183
— Auto Check 199
— Closed 386
— Data Incomplete 222
— Data Not Redundant 222
— Established 386
— Failed 222
— Failed Redundancy 222
— Foreign 183
— Formatting 222
— Healthy 222
— Initializing 183
— Listen 386
— Missing 183, 234
— No Media 184
— Not Initialized 184
— Offline 184, 236
— Online 184
— Online (Errors) 184, 237
— Regenerating 223
— Resynching 223
— Stale Data 223
— Syn Rcvd 386
— Syn Sent 386
— TCP-соединения 386
— Unknown 223
— Unreadable 185
— Unrecognized 185
— Wait 386
спулер 327
— детальная информация 332–334
— сводная информация 330–331
спулинг 350
страница
— аппаратная ошибка 158
— программная ошибка 158, 164
— разделитель 348–349
строка, командная 2–16
— настройка свойств 11
сценарий 31
— добавление комментариев 33
— настройки IP 375
— передача аргументов 38–39
— создание 31
— файлы 363–364

Т

том 220
— метка 217
— набор (ограничения при расширении) 226
— простой 223
— — расширение 224–226
— — создание 223–224
— удаление 227

точка монтирования 208
— назначение 208
— смена 209
— удаление 209
триггер события 138–142

У

условие 52
— вложенное 53
учетная запись
— группы
— — добавление 310
— — переименование 322
— — перемещение 321
— — просмотр и поиск 315–316
— — удаление 322–323
— имя в SAM 257
— компьютера
— — восстановление заблокированной 273
— — настройка атрибутов и членства в группах 267–268
— — отключение и включение 272–273
— — перемещение 274
— — просмотр и поиск 269–271
— — создание 266–267
— — удаление 275
— неактивная 260
— пользователя
— — включение и отключение 302
— — восстановление просроченной 303
— — вывод 290
— — доменная 287
— — локальная 287
— — настройка атрибутов 300–301
— — настройка атрибутов (доменных) 293–294
— — переименование 306
— — перемещение 305
— — просмотр и поиск 297–299
— — создание (доменных) 291–292
— — создание (локальных) 290, 295–297
— — удаление 307–308
— — удаление (локальных) 290

Ф

файл
— appevent.evt 125
— autoexec.nt 4
— config.nt 4, 8
— dnsevent.evt 125
— localtag.txt 168
— ntbootdd.sys 189
— ntds.evt 125
— ntfrs.evt 125
— pooltag.txt 168
— secevent.evt 126
— перебор 57–58
— журнала планировщика заданий 74
— подстановочный 216
— расширения 22
— создание сопоставления 23
файловая система (преобразование в NTFS) 213–216
флаг dirty 201
флажок
— Allow Users To Connect Remotely To This Computer (Разрешить удаленный доступ к этому компьютеру) 73
— Insert Mode (Быстрая вставка) 9
— Let System Position Window (Автоматический выбор) 10
— Modify Shortcut That Started This Window (Изменить ярлык для запуска этого окна) 11
— Open Advanced Properties For This Task When I Click Finish (Установить дополнительные параметры после нажатия кнопки) 80
— QuickEdit Mode (Выделение мышью) 9
фокус 174
формат вывода 94, 100

Ц

цикл 54–55
— формы 55

Ш

шлюз 405

Об авторе

За плечами Уильяма Р. Станека (William R. Stanek) 20-летний опыт программирования и разработки приложений, и сейчас он считается одним из ведущих экспертов в области компьютерных технологий. Его советы помогли миллионам программистов, разработчиков и сетевых инженеров всего мира. На счету Уильяма немало наград за книги — он написал более двух десятков книг о компьютерах. В том числе «Microsoft Windows XP Professional. Справочник администратора», «Microsoft Windows 2000. Справочник администратора», «Microsoft Windows Server 2003. Справочник администратора» и «Microsoft IIS 6.0. Справочник администратора».

С 1991 г. Станек участвовал в разработке коммерческих Интернет-проектов. Его бизнес-качества и профессиональный опыт сформировались за 11 лет армейской службы. Уильям обладает большим опытом разработки в области серверных технологий, шифрования и Интернет-решений. Он написал массу технических статей и курсов лекций по широкому спектру проблем и весьма известен как компьютерный эксперт, обладающий опытом практической работы.

Станек получил степень магистра информационных систем с отличием и степень бакалавра компьютерных наук. Он гордится участием в боевых действиях в Персидском заливе в составе экипажа самолета радиоэлектронной борьбы. В его послужном списке несколько боевых вылетов в Ирак и девять медалей за воинскую службу, включая одну из высших авиационных наград США — Крест Военно-Воздушных Сил с отличием. В настоящее время Станек с женой и детьми живет на северо-западе Тихоокеанского побережья США.

Уильям Р. Станек

Командная строка Microsoft® Windows®. Справочник администратора

Перевод с английского по общей редакцией ***Ю. Е. Купцевича***
Компьютерная верстка и дизайн ***Е. В. Козлова***
Технический редактор ***Н. Г. Тимченко***
Корректор ***Л. А. Панчук***

Оригинал-макет выполнен с использованием
издательской системы Adobe PageMaker 6.0

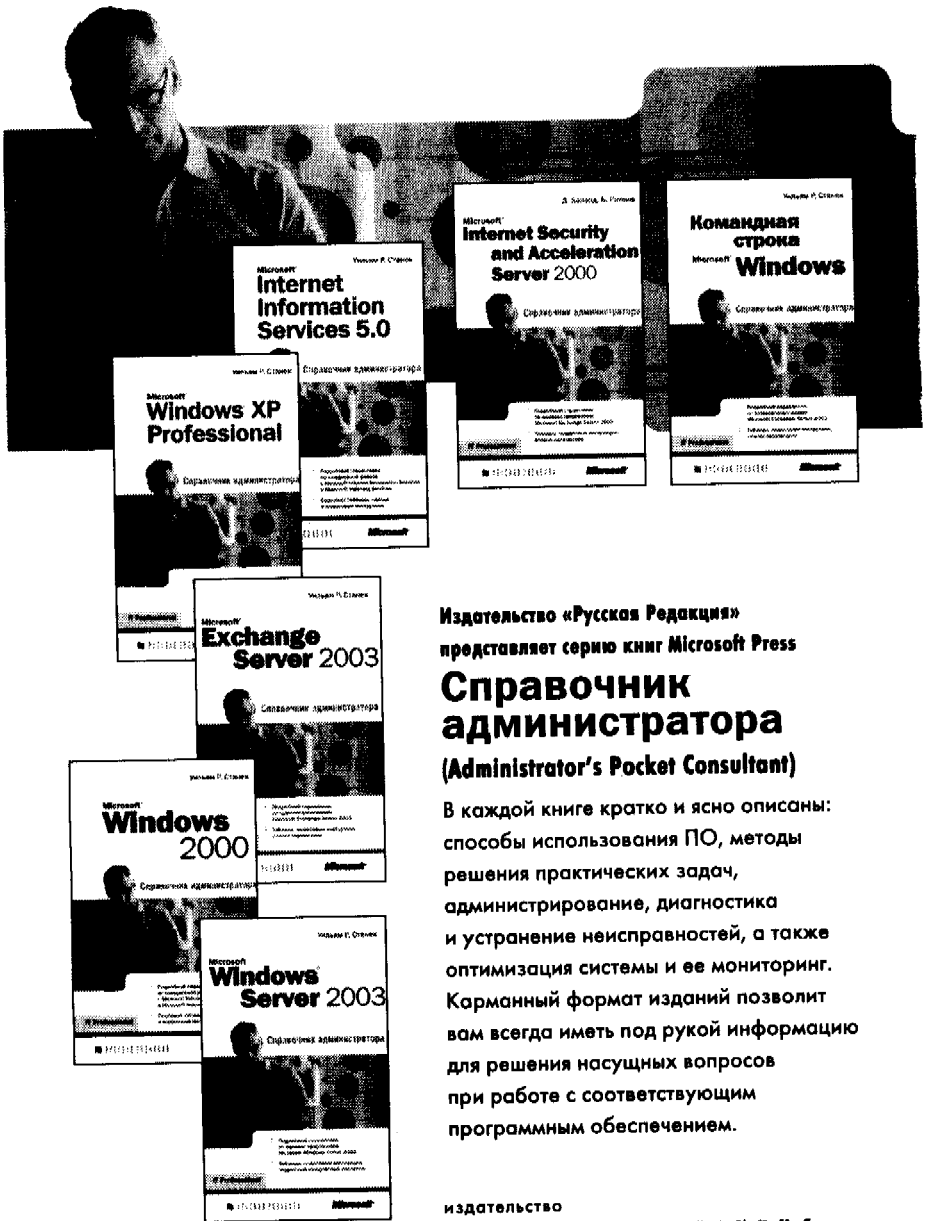
Главный редактор **А. И. Козлов**

Подготовлено к печати
Издательско-торговым домом «Русская Редакция»
121087, Москва, ул. Антонова-Овсеенко, д. 13
тел.: (095) 256-5120, тел./факс: (095) 256-4541
e-mail: info@rusedit.ru, http://www.rusedit.ru

РУССКАЯ РЕДАКЦИЯ

Подписано в печать 15.07.2004 г. Тираж 3 000 экз.
Формат 84x108/32. Физ. п. л. 15

Отпечатано в ОАО «Типография «Новости»
105005, Москва, ул. Фр. Энгельса, 46



Издательство «Русская Редакция»
представляет серию книг Microsoft Press

Справочник администратора

(Administrator's Pocket Consultant)

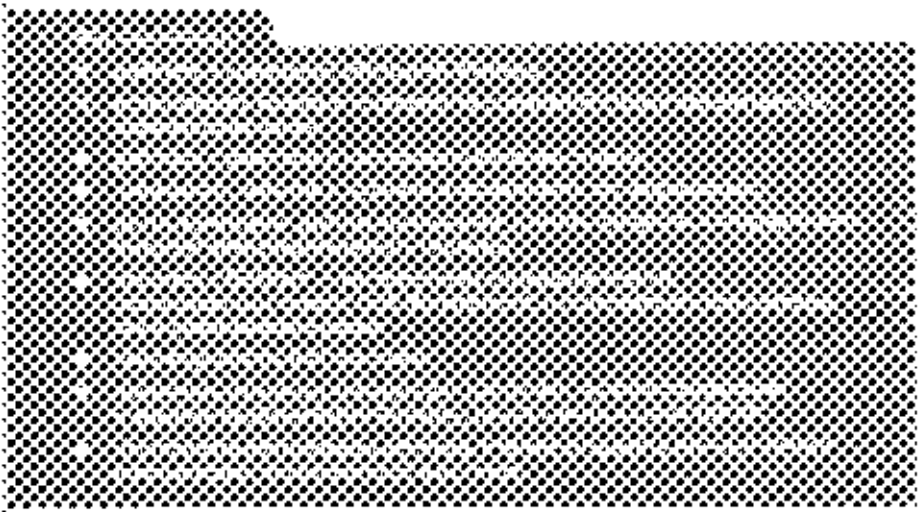
В каждой книге кратко и ясно описаны: способы использования ПО, методы решения практических задач, администрирование, диагностика и устранение неисправностей, а также оптимизация системы и ее мониторинг. Карманный формат изданий позволит вам всегда иметь под рукой информацию для решения насущных вопросов при работе с соответствующим программным обеспечением.

издательство
РУССКАЯ РЕДАКЦИЯ

ПРОДАЖА КНИГ
тел.: (095) 256-5120, тел./факс: (095) 256-4541,
e-mail: sale@rusedit.ru

Командная строка Windows

Справочник администратора



Windows Server System