

ПРАКТИЧЕСКАЯ РАБОТА

Создание образа системы. Восстановление системы

Содержание: название, формулировка темы и цели, время выполнения, необходимое обеспечение, теоретическая часть, практическая часть, технология выполнения, описание заданий, рефлексия самоконтроля, контрольные вопросы

Время выполнения: 2 ч.

Материально-техническое обеспечение: персональный компьютер (системный блок, монитор, клавиатура, мышь)

Программное обеспечение: ОС Windows, интегрированная среда разработки, офисные программы Microsoft Office.

Учебно-методическое обеспечение: конспект лекций, описание работы, встроенная справочная система ОС Windows и сред разработки.

Порядок выполнения практической работы: изучить теоретический материал, выполнить задания, составить отчёт с выводами о проделанной работе, ответить на контрольные вопросы, сдать выполненную работу и отчёт.

Требования к отчёту: отчёт должен быть оформлен в соответствии с «ГОСТ 19.XXX ЕСПД».

Цель: приобретение навыков создания образа системы, восстановления системы.

Теоретическая часть

Создание образа системы

Образ операционной системы - это полный слепок состояний этой самой системы, засунутый в файл. Слпок включает в себя не только саму ОС, но и все настройки, конфигураций и все программы, установленные на нее. После того, как файл образа ОС создан, его можно загружать в различные облачные и виртуальные машины. После распаковки и выполнения образа вы получаете в своё распоряжение полностью рабочую машину с уже установленными программами и прописанными настройками.

Зачем нужен образ установленной ОС? У этой технологии есть три основных способа применения:

1.	Автоматизация рутинной работы	Системным администраторам по долгу службы постоянно нужно ставить и настраивать кучу программ. Как правило, это сводится к набору однообразных и шаблонных операций. Например, поставить веб-сервер, запустить его, открыть порты доступа к серверу извне, настроить журналирование событий, открыть доступ к определенным файлам. Для этого нужно выполнить на рабочей машине ряд однотипных команд. Эти рутинные операции могут и даже должны быть автоматизированы. Это можно сделать, например, с помощью файлов сценариев для ОС. Запускаешь новый сервер и после этого активируешь на нём сценарий, который автоматически выполнит команды установки программ и редактирования конфигурационных файлов. Готовые образы операционной системы дают принципиально иной подход к автоматизации. Админ один раз устанавливает ОС на сервер и конфигурирует все необходимые сервисы вручную. После этого ОС со всеми потрохами сохраняется в образ. Когда надо запустить новый сервер, администратор распаковывает файл образа на новой машине. Все необходимые сервисы и программы со всеми настройками сразу же появляются на новой машине под Windows, Linux или любой другой нужной ОС.
2.	Готовые приложения в один клик от сторонних разработчиков	Многие администраторы начали создавать свои образы ОС. Это привело к тому, что некоторые из них начали выкладывать свои файлы образов в публичный доступ чтобы поделиться ими с другими админами. Этот подход взяли на вооружение поставщики облачных решений. Их админы создают набор готовых к использованию и настроенных ОС с предустановленными и сконфигурированными программами. Достаточно просто выбрать в админской панели облачного сервиса необходимый файл

		- и он будет автоматически распакован и установлен на новом сервере. Провайдеры облачных решений создали большое количество готовых виртуальных образов операционной системы для автоматизации рутинных нужд IT-команд - преднастроенные серверы баз данных, веб-серверы, заранее сконфигурированные системы сбора статистики и среды для запуска серверных приложений. У нас (MCS) базы данных подключаются здесь, а другие многие другие настроенные средства разработки - на маркете приложений. Благодаря образам клиентам облачных сервисов больше не нужно самим готовить базовый образ для автоматизации своих задач - все сервисы уже подготовлены и настроены опытнейшими и бородатейшими админами с огромным опытом работы.
3.	Резервное копирование инфраструктуры	Когда речь заходит о резервном копировании, на ум приходят, прежде всего, бэкапы баз данных и файлов пользователей. Но инструменты создания и управления образами дают возможность делать резервные копии того, что раньше нельзя было предохранить от повреждений - серверы. Зачем делать бэкапы серверов целиком? Конечно, вы всегда можете сохранить копии ключевых конфигурационных файлов и просто накатывать конфиги на новые машины (или на старые, если вы заporоли настройки и хотите вернуться к предыдущему стабильному состоянию), но это требует времени и написания скриптов автоматизации. И вот этот процесс управления конфигами и их бэкапов можно и нужно упростить, ускорить и автоматизировать с помощью образов операционной системы.

Пример. Допустим, вы проводите перенастройку вашей инфраструктуры в дата-центре. Процесс перенастройки довольно трудоёмок и включает в себя правки десятков и сотен конфигурационных файлов различных сервисов. Если в ходе работ что-то пошло не так, то придётся разбираться в изменениях конфигов и искать в них правки, вызвавшие сбои. С резервными копиями ОС подход меняется. С определенной регулярностью система управления образами создает полные слепки состояния системы, вместе со всеми файлами настроек. В случае правок конфигов и поломки сервера можно не копать в сообщениях системы, выискивая корень зла. Можно просто откатить систему к предыдущему стабильному рабочему состоянию и начать вносить изменения в работающие конфиги заново. Почему этот подход лучше? Потому что модифицировать состояние работающей системы - это всегда проще поиска поломок. Вы всегда знаете, что вы изменили в сервере и как это сказалось на работе системы. В случае ошибочной правки вы уже знаете, где искать проблему. Такой способ положительно сказывается на надежности и отказоустойчивости вашего бизнеса - в случае сбоев можно моментально выполнить восстановление, вернуть всё в рабочее состояние и подумать о дальнейших действиях ещё раз, в спокойной обстановке - всё стабильно и работает, а вам не нужно тушить пожары упавшей инфраструктуры. Этот подход с использованием образов готовой ОС продиктован одним из ключевых принципов философии DevOps (искусство разработки и поддержания информационных систем) - «Не чинить сломанное, а сразу выбрасывать и заменять на работающее».

Зачем вам изучать использование образов ОС? Все современные технологии управления и автоматизации работы с серверами нужны для того, чтобы выпускать в свет свои продукты быстрее, качественнее и с большей степенью надежности, тратя на это как можно меньше денег, времени и сил. Готовые образы ОС с преднастроенными сервисами позволяют инженерам и программистам сфокусироваться на создании конечного продукта, не отвлекаясь на задачи создания и поддержки инфраструктуры для тестов и разработки. Порог вхождения в технологии управления образами операционных систем сегодня невысок, а готовые решения от поставщиков облачных вычислительных мощностей делают этот порог ещё ниже. Беглое прочтение документации, несколько кликов в веб-интерфейсе - и вы получаете полностью рабочую инфраструктуру высочайшего качества. Отсекайте рутину, экономьте свое время и силы для действительно важной работы, которую заметят и оценят ваши пользователи и клиенты.

ISO-образ

ISO-образ - это неформальный термин для обозначения образа оптического диска, содержащего файловую систему стандарта ISO 9660. В более общем смысле, термин относится к образу любого оптического диска. Этот образ представляет собой файл с расширением .iso. Его можно использовать (в совокупности со специальными программными средствами) вместо компакт-диска. Следует иметь в виду, что ISO-образ содержит меньше информации, чем исходный компакт-диск. На компакт-диске содержится

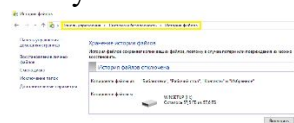
служебная информация, которая может, в частности, использоваться для защиты от копирования. Возможностью сохранять подобную информацию (в своих собственных форматах образа диска) обладают некоторые из программ для работы с компакт-дисками. Как правило, просмотреть и извлечь содержимое ISO-образа можно программой-архиватором, поддерживающим чтение таких файлов.

Работа с ISO-образами

В операционных системах Windows начиная с Windows 7 появилась возможность записать образ диска на оптический диск, а в Windows 8 его можно монтировать и работать так же, как с физическим диском. В MacOS средствами операционной системы можно монтировать и работать с этим образом точно так же, как с физическим компакт-диском. Также, существуют программы, которые на основе файла-образа создают в системе виртуальный привод CD-ROM: UltraISO, Alcohol 120%, DAEMON Tools. Для Windows есть программа ImDisk Virtual Disk Driver, которая интегрируется в контекстное меню проводника и подключает ISO - образ как виртуальный диск на первую свободную букву диска в системе. Редакторы ISO-образов: ImgBurn, ISO Master, PowerISO, UltraISO.

Создание образа системы и Диска (флеш-накопителя) восстановления Windows 10

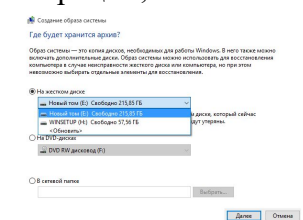
Для создания Образа системы и Диска восстановления системы в Windows 10 необходимо выполнить следующее. Щёлкаем правой кнопкой мыши (ПКМ) по значку Пуск в Панели задач, выбираем «Панель управления» => «Система и безопасность» => «История файлов». В открывшемся окне, слева внизу нажимаем «Резервная копия образа системы».



В следующем открывшемся окне в левой колонке выбираем и нажимаем «Создать диск восстановления»).



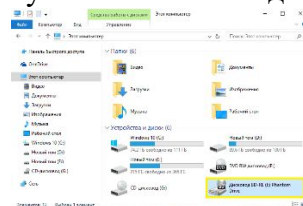
Открывается окно, в котором Windows предложит Вам выбрать локальный диск жесткого диска Вашего компьютера. Если у Вас имеется в наличии внешний накопитель, то до начала выполнения данных операций, желательно подключить его к компьютеру и выбрать его для сохранения Образа системы.



Внизу нажимаем ссылку «Далее», и в открывшемся окне внизу «Архивировать». После завершения архивации Windows предложит Вам создать Диск восстановления системы. Для этого установите в дисковод чистый готовый к записи CD/DVD RW диск и нажмите Создать.

Создание Флеш-накопителя восстановления системы

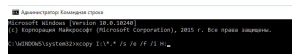
В некоторых из современных ПК дисковод CD/DVD дисков отсутствует. В этом случае существует способ создания Флешки восстановления системы с использованием виртуального дисковода. Для создания такого дисковода в настоящее время существует довольно таки много программ таких как: Ultra ISO, Alcohol 120, Phantom Creator, Deamon Tools, Power ISO и т.п., каждый может выбрать ту, которая ему более подойдет. Для примера в данном случае рассматривается способ создания Флеш-накопителя восстановления с использованием программы Phantom Creator. Запустите программу и выберете «Создать пустой CD-R». Убедитесь что в проводнике появился виртуальный дисковод.



Далее переходим на страничку со ссылкой «Создание образа системы», а ниже выбираем и нажимаем Создание диска восстановления (одинаково для всех ОС Windows).



В открывшемся окне выбираем наш виртуальный дисковод, жмем «Далее» и затем «Создать». После создания виртуального Диска восстановления системы вставляем чистую отформатированную флешку в USB дисковод или гнездо порта USB контроллера, нажимаем сочетание клавиш Win+R и в командной строке вводим команду: `хсору I:*.* /s /e /f /i H:` где I: это буква названия Вашего виртуального дисковода. Нажимаем Enter и ждем окончания операции.



Ваша Флешка восстановления успешно создана. Теперь у Вас все готово для восстановления Вашей операционной системы в любой момент и при любых возникших неприятностях и неожиданностях, а при сохранении Образа системы на внешний накопитель, даже при поломке и выхода из строя всего жесткого диска или необходимости его замены по какой-либо причине на другой.

Программы для создания образа системы, восстановления системы, загрузки официальных ISO образов ОС, создания установочных и загрузочных носителей:

- balenaEtcher (Etcher) – бесплатная утилита для Windows, MacOS и Linux, которая помогает быстро и просто создавать загрузочные USB флешки и SD карты из установочных образов операционных систем;
- Помощник по обновлению до Windows;
- Media Creation Tool от Microsoft позволяет обновить компьютер до последней версии ОС Windows 10, выполнить чистую установку системы, скачать образ Windows 10 ISO и создать установочный носитель;
- Universal USB Installer – портативная программа для создания загрузочных USB-накопителей с операционными системами Windows и Linux, а также другими загрузочными программами;
- WinToHDD – бесплатная программа для установки и клонирования операционной системы Windows, без использования загрузочных DVD или USB-дисков;
- Rufus – утилита для создания загрузочных USB-носителей из ISO-образов с выбранной операционной системой (Windows, Linux, UEFI и др.);
- UNetbootin - полезный инструмент, который позволяет устанавливать различные дистрибутивы Linux / BSD и создавать загрузочные USB-устройства флеш-памяти с минимальными усилиями.
- DAEMON Tools – популярная программа для монтирования образов дисков, подключения виртуальных приводов, а также создания, изменения и конвертирования образов дисков различных форматов;
- Alcohol 120% Free Edition – бесплатная программа для эмуляции образов дисков и виртуальных приводов, а также создания и записи образов на компакт диски;
- Fido от Rufus – скрипт PowerShell для загрузки официальных ISO-образов Windows 10 и Windows 8.1 с серверов Microsoft;
- Daemon Tools - популярный инструмент для эмуляции приводов и создания образов. Предлагает основные функции резервного копирования, дополнительные возможности по работе с образами и виртуализации приводов;
- Утилита WoeUSB предназначена для создания загрузочных USB-носителей с установочной версией Windows из под операционной системы Linux;
- WinSetupFromUSB – это портативная программа для создания мультизагрузочных USB-флешек с установочными версиями Windows, Linux, а также различными загрузочными образами и утилитами.

Программы для восстановления системы: Acronis True Image, Aomei Backupper Standard, Macrium Reflect, Windows Handy Backup, Windows Repair.

Восстановление системы

Восстановление системы (англ. System restore) - компонент операционной системы Windows (процесс `rstrui.exe`), предназначенный для восстановления работоспособности ОС путём отката (восстановления предыдущего состояния ПК) системных файлов, ключей реестра, установленных программ и т. д. Восстановление системы присутствует в системах Windows ME, Windows XP, Windows Vista, Windows 7, Windows 8 а также Windows 10. Семейство Windows Server не включает в себя функцию Восстановление системы. Восстановление системы, встроенное в Windows XP, может быть установлено на устройстве, работающем под Windows Server 2003, хотя эта операция не поддерживается Microsoft. В

Windows Vista данный компонент обладает улучшенным интерфейсом и построен на технологии Shadow Copy, в отличие от предыдущих версий, построенных на файловом фильтре, который отслеживал изменения в определённом наборе расширений файлов и затем копировал файлы, прежде чем они перезаписывались. Преимущество использования технологии Shadow Copy в том, что блок-уровневые изменения в файлах, находящихся в любой папке на диске, могут быть отслежены и резервные копии созданы независимо от местонахождения файлов.

Точки восстановления

Пользователь может создавать новую точку восстановления вручную, производить откат к уже существующей точке или изменять конфигурацию восстановления системы. Более того, восстановление само по себе может быть обратимым. Старые точки восстановления сбрасываются для того, чтобы поддерживать использование объёма диска в точно определённом размере. Пользователи, для которых важны производительность или используемый объём диска, могут также выбрать полное отключение восстановления системы. Для файлов, хранящихся на дисках, но которые служба восстановления не отследила, не будут созданы резервные копии и такие файлы никогда не будут восстановлены. Восстановление системы создаёт резервные копии системных файлов определённых расширений (.exe, .dll и т. д.) и сохраняет их для дальнейшего восстановления и использования. Также создаются резервные копии реестра и большинства драйверов. Точки восстановления создаются:

- когда программа устанавливается с помощью Windows Installer, Package Installer или других установщиков, знающих о существовании восстановления системы;
- когда Windows Update устанавливает новые обновления;
- когда пользователь устанавливает драйвер без цифровой подписи Windows Hardware Quality Labs;
- каждые 24 часа пользования компьютером (10 часов в Windows Me) или каждые 24 часа календарного времени, в зависимости от того, что истечёт раньше. Эту установку можно конфигурировать через реестр. Подобные точки восстановления известны как точки проверки системы. Для создания таких точек восстановления
- системы требуется Task Scheduler. Более того, точки проверки системы создаются лишь тогда, когда система находится в бездействии в течение некоторого времени;
- когда система запускается, будучи отключенной более 24 часов;
- по требованию пользователя.

На Windows Vista теневые копии, создаваемые во время процедуры File Backup и Complete PC Backup, могут также использоваться как точки восстановления.

Техническая реализация

Восстановление системы сохраняет точки восстановления в папку System Volume Information. По умолчанию (при установке на диск с файловой системой NTFS) Windows XP запрещает доступ пользователей к папке System Volume Information, разрешая доступ лишь системе (пользователю SYSTEM).

Диск восстановления

Компакт-диск с резервной копией системы и настроек компьютера, сделанной производителем или пользователем компьютера, обычно называется диском восстановления. Использование такого диска предполагает форматирование жесткого диска компьютера и копирование системных файлов и данных из резервной копии системы с компакт-диска на диск компьютера, то есть приведение системы компьютера в состояние на момент создания диска восстановления. Обычно диски восстановления - это загрузочные диски с собственной операционной системой, которая не зависит от установленной на компьютере системы. Такие диски могут использоваться как для полного восстановления системы, так и для диагностики компьютера и копирования отдельных файлов с поврежденной операционной системы на USB-носители или серверы. Операционная система для диска восстановления обычно выбирается по основной операционной системе компьютера. То есть используется облегченная версия основной операционной системы. Наиболее распространенная операционная система на загрузочных дисках - это Windows PE разных версий. Такое распространение этой операционной системы связано с тем, что она позволяет запускать множество стандартных программ Microsoft Windows, не требующих расширенных динамических библиотек и ресурсов, что вполне достаточно для восстановления работоспособности компьютера и копирования данных с поврежденной системы. Помимо Windows PE, диски восстановления строятся и на других операционных системах. Диск восстановления можно создать как своими силами, так и с помощью специального программного обеспечения от производителя компьютера или от стороннего разработчика.

Как работает восстановление системы? Восстановление системы Windows использует точки восстановления, чтобы восстановить предыдущее состояние системных файлов и настроек, не влияя на личные файлы. Точки восстановления создаются еженедельно автоматически, а также перед значительными системными событиями, такими как установка программы или драйвера устройства. Контрольную точку восстановления можно создать вручную.

Можно ли отменить изменения? Да. Во время каждого использования средства восстановления системы перед внесением изменений создается точка восстановления, с помощью которой можно отменить внесенные изменения, если они не исправляют проблему. Если средство восстановления системы используется во время работы в безопасном режиме или с помощью меню «Параметры восстановления системы», то операцию восстановления будет невозможно. Однако, можно запустить средство восстановления системы ещё раз и выбрать другую имеющуюся точку восстановления.

Как выбрать точку восстановления? Восстановление системы автоматически рекомендует последнюю контрольную точку восстановления, созданную перед внесением значительных изменений, например, перед установкой программы. Можно также выбрать ее из списка точек восстановления. Попробуйте воспользоваться контрольной точкой восстановления, созданной непосредственно перед возникновением проблем. Описания точек восстановления, созданных автоматически, соответствуют названиям событий, например, установка обновления Windows. Восстановление системы восстанавливает состояние компьютера, в котором он был до создания выбранной точки восстановления.

В течение какого времени сохраняются точки восстановления? Точки восстановления хранятся пока не будет заполнено дисковое пространство, выделенное для функции восстановления. При создании новых точек восстановления старые удаляются. Если на диске отключить защиту системы (функцию, которая создает точки восстановления), из него удаляются все точки восстановления. Если включить защиту системы снова, создадутся новые точки восстановления.

Что делать, если не удалось исправить ошибку? Если не удалось исправить ошибку с помощью функции восстановления системы, можно отменить операцию восстановления или выбрать другую точку восстановления. Если средство восстановления системы не отражает доступных точек восстановления, убедитесь, что защита системы включена и осталось менее 300 Мбайт свободного места на диске объемом 500 Мбайт и более или 50 Мбайт на диске объемом менее 300 Мбайт. Если не удалось устранить неполадки с помощью функции восстановления системы, можно воспользоваться дополнительным методом восстановления.

Программы для восстановления системы: Acronis True Image, Aomei Backupper Standard, Macrium Reflect, Windows Handy Backup, Windows Repair.

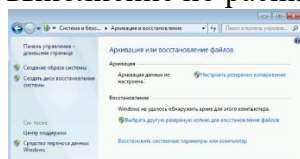
Практическая часть

Создание образа системы

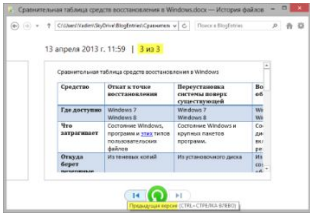
Настройка создания полного образа Windows по расписанию

Технологии резервного копирования Windows

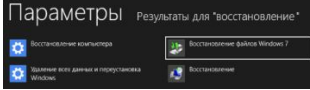
В Windows возможности резервного копирования состоят из архивации файлов и создания образа системы, в который можно включить произвольные разделы. Для обеих операций предусмотрено выполнение по расписанию, а образ системы также можно создать однократно.



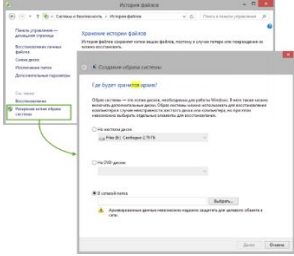
В истории файлов не используется сжатие, зато настройка резервного копирования максимально упрощена, а восстановление удобно и наглядно позволяет просматривать все сохраненные версии файлов.



WIM-образ, создаваемый утилитой `wbadmin`, заточен под работу функции Refresh Your PC. Она является средством восстановления работы Windows, нежели резервного копирования системы в традиционном смысле. Разница в этих понятиях проявляется как в содержимом создаваемого образа, так и в механизме восстановления, и незнание тонкостей может привести к неожиданным и неприятным результатам.



Создание образа системы в графическом интерфейсе Windows



Создание образа системы утилитой `wbadmin`

В состав Windows входит утилита `wbadmin`, с помощью которой можно создавать образ системы из командной строки. В клиентских ОС утилита обладает урезанным набором функций по сравнению с серверными системами, но ее достаточно для создания образа одной командой. Справку по ней вы можете посмотреть, выполнив:

```
wbadmin start backup /?
```

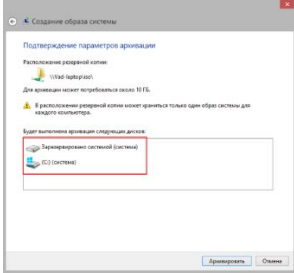
Ниже вы видите пример команды, сохраняющей на диске F образ системы, в который включен раздел C, а также все прочие разделы, необходимые для полноценного восстановления Windows на чистом диске.

```
wbadmin start backup -backupTarget:F:-include:C:-allCritical -quiet
```

Значения параметров таковы:

-backupTarget	Задаёт букву диска, на котором сохраняется резервный образ в папку WindowsImageBackup. Эта папка автоматически создается в корне диска при создании первого образа, и вручную создавать ее не надо.
-include	Задаёт буквы диска или пути на основе GUID (перечисляются через запятую) для тех разделов, которые вы хотите включить в резервный образ системы.
-allCritical	Обеспечивает автоматическое включение в образ всех разделов, необходимых для полноценного восстановления системы, в дополнение к перечисленным в параметре -include. В первую очередь речь идет о разделе, содержащем диспетчер и конфигурацию загрузки, иначе после восстановления система не загрузится.
-quiet	Обеспечивает тихое выполнение команды без лишних вопросов.

В графическом интерфейсе Windows это выглядело бы примерно так:



Запустив команду, вы увидите ход ее выполнения, в том числе и список разделов, включенных в образ.

Регулярное создание образа по расписанию
Преимущества этого подхода:

- Актуальность образа. В аудитории этого блога много опытных пользователей, но даже здесь (исходя из опроса) лишь четверть читателей обладает образом, созданным в течение последних 30 дней. Я же не вижу особого смысла в однократном создании образа системы, поскольку он имеет свойство устаревать по мере установки обновлений и программ, а также изменений в пользовательской среде.
- Эффективное распоряжение дисковым пространством. Сначала Windows создает исходный образ системы, а затем обновляет его инкрементно.

Рассмотрим два варианта создания запланированного задания.

- Вариант 1: создание обычного задания в планировщике. Этот вариант работает во всех современных ОС Windows. Вы можете воспользоваться графическим интерфейсом планировщика заданий или командной строкой. Второе для меня проще, в том числе и с точки зрения объема инструкций:

Не забудьте указать свои буквы дисков в команде для создания образа. Задание MyBackup будет выполняться в 10 часов вечера каждые 14 дней (/mo 14). Альтернативно, можно задать расписание на основе недель: /SC weekly /mo 2 /d FRI. В этом случае задание будет выполняться каждые две недели по пятницам. Проверить работу задания можно запуском в графическом интерфейсе планировщика или из командной строки:

Во время выполнения задания вы можете работать за ПК, поскольку утилита `wbadmin` сначала делает снимок теневой копии тома, а затем уже на ее основе создает образ. Конечно, во время резервного копирования производительность системы может несколько снизиться.

Задание, создающее образ по расписанию

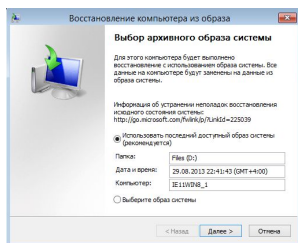
```
schtasks /create /tn MyBackup /xml "Полный путь к XML-файлу"
```

В планировщике журнал событий отключен, и включив его в правой панели, вы сможете узнать время и результат последнего запуска задания.

Но еще проще и даже важнее посмотреть время последнего обновления образа и его параметры:

Обратите внимание, что в описании образа указано, можно ли развернуть его на «голой» системе (bare metal recovery в английской терминологии). Это результат использования ключа -allCritical.

Восстановление Windows из резервного образа



Как справиться с включением в образ нежелательных разделов

Причина	Решение
Необходимые для загрузки файлы (MBR) размещены вне системного диска	Перенос файлов загрузки на системный раздел или Исключение раздела из образа (потребуется восстановления загрузки после применения образа)
В библиотеки входят папки, расположенные на другом разделе	Исключение раздела из образа
С системного раздела на другой раздел ведут жесткие или символические ссылки, либо соединения	Исключение раздела из образа
Файлы, необходимые для работы сторонней службы или драйвера, размещены на другом разделе (пример со службой обновления игры)	Поиск по букве диска (D:\) в указанном ниже разделе реестре и удаление программы HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\ или Исключение раздела из образа
Программа или игра установлена на другой раздел	Удаление программы. В Windows 10 в Параметры - Приложения есть фильтр по дискам. или Исключение раздела из образа

Общим знаменателем решений является исключение раздела из образа. Вопрос лишь в том, как это реализовать в команде wbadm. Ключ к решению – правильная формулировка задачи. Казалось бы, можно просто перечислить нужные буквы разделов в параметре -include, но не все так просто! Посмотрите на картинку ниже и представьте, что нежелательным диском является E.



Если использовать ключ -allCritical, в образ включаются все три раздела. Если его не использовать, в образ не попадает раздел System Reserved, содержащий файлы, необходимые для загрузки системы! Конечно, можно назначить ему букву диска в оснастке, но тогда раздел будет «болтаться» в файловом менеджере. Поэтому правильно задача формулируется не «как исключить разделы из образа», а «как включить в образ раздел с файлами загрузки, если у него нет буквы диска». И у нее есть элегантное решение.

Как определить раздел с диспетчером загрузки?

Команда bcdedit показывает необходимую информацию только в том случае, когда диспетчер загрузки находится на томе с назначенной буквой.

Диспетчер загрузки Windows

```
идентификатор {bootmgr}
device partition=C:
description Windows Boot Manager
```

Это не годится в ситуациях, когда файлы загрузки расположены на разделе «Зарезервировано системой»:

Диспетчер загрузки Windows

```
идентификатор {bootmgr}
device partition=\Device\HarddiskVolume1
description Windows Boot Manager
```

Путь вида \Device\HarddiskVolume1 не подходит для утилиты wbadm, потому что она понимает только буквы диска или имена на основе GUID. На помощь приходит PowerShell:

```
gwm i win32_volume | where {$_.SystemVolume -eq "true"} | select systemvolume,caption,driveletter,label | fl
```

Команда опрашивает все тома и выбирает из них только те, на которых есть диспетчер загрузки. Они помечены как системные (systemvolume), а путаницу в понятиях между системным и загрузочным томами я уже разбирал. Результат команды выглядит примерно так:

```
systemvolume : True
caption      : \\?\Volume{2f80a0d4-346d-11e1-9268-806e6f6e6963}\
driveletter :
label       : Зарезервировано системой
```

Дальше вы с равным успехом можете задействовать параметр caption или driveletter. Кстати, аналогичную задачу можно решить утилитой MultiBoot, но можно и одной командой:

```
wbadmin start backup -backupTarget:F: -include:C:,\?\Volume{2f80a0d4-346d-11e1-9268-806e6f6e6963}\ -quiet
```

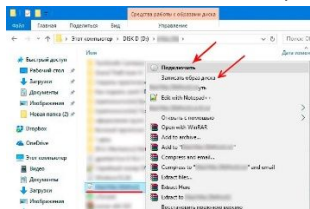
В образ попадут только разделы с системой и файлами, необходимыми для ее загрузки. Любопытно, что в этом случае команда wbadmin get versions не сообщает о возможности развертывания образа на «голой» системе, даже если включаются те же самые разделы, что и с ключом -allCritical.

Работа с ISO-образами в Windows 10

Монтирование ISO-образа диска или его запись на диск в Windows 10 максимально упрощено. Для этого достаточно воспользоваться функциональной особенностью самой системы, автоматически исключающей необходимость загрузки и установки на компьютер программ сторонних разработчиков. Функция монтирования образов в проводнике Windows 10 стало одним из лучших, полезных и актуальных нововведений операционной системы. Конечно же, для получения более расширенных возможностей в работе с файлами-образами дисков, пользователю не обойтись без дополнительно установленной на компьютер программы.

Смонтировать, записать ISO/IMG в проводнике

Откройте проводник в вашем компьютере и перейдите в папку с вашим образом диска. Для монтирования дважды щелкните на него или задействуйте правую кнопку мыши, где выберите пункт меню «Подключить», а для записи диска выберите «Записать образ диска».



Смонтировать, записать ISO/IMG в PowerShell

Для монтирования дисков можно воспользоваться командой в PowerShell. Запустите командную оболочку Powershell и введите следующую команду:

```
Mount-DiskImage -ImagePath "C:\Users\пользователь\Downloads\Veeam.iso"
```

Следующая команда безопасно извлечет его из привода:

```
Dismount-DiskImage -ImagePath "C:\Users\пользователь\Downloads\Veeam.iso"
```

Использование монтирования дисков с помощью PowerShell подходит для случаев, когда после установки архиватора WinRAR, файлы-образы автоматически ассоциируются с этой программой и вместо «Подключения» открываются как архивы.

Записать образа диска

За запись дисков в Windows 10 отвечает встроенная утилита isoburn.exe. Располагается она в системном каталоге C:/ Windows/system32, для записи же образов используется следующая команда в PS или CMD:

```
isoburn.exe /Q D: "\\netcomputer\Distribs\ISO\7 Professional x64 SP1 Russian.iso"
```

или

```
isoburn.exe /Q E: D:\Window10.ISO
```

Аргумент [/ Q] в данном примере указывает, что запись начнется незамедлительно, [CD:] - это буква вашего физического привода, а path to iso - полный путь к образу.

Bat-скрипт получения информации из образа Windows

Рассмотрим скрипт, который используя команды системы DISM извлекает из образа Windows различную информацию и показывает её пользователю на экран, либо экспортирует в файл. Данный скрипт будет полезен системным администраторам, создающим свои «сборки», либо тестирующим чужие. Иногда может возникнуть необходимость посмотреть некоторую информацию об установленной ОС - в этом случае также пригодится данный скрипт. Все команды DISM для получения информации из образа Windows реализованы в данном скрипте. Информация из образа, которую можно получить, используя данный скрипт:

- Драйвера устройств (наличие);
- Пакеты обновлений (наличие);
- Системные компоненты (состояние);
- Языковая поддержка (состояние);
- Редакции Windows (состояние);
- Классические приложения (наличие);
- Универсальные приложения (наличие);
- Ассоциации файлов (состояние).

В скобках указан аспект информации. Если некий тип объектов может быть добавлен или удалён из образа, то информация будет показывать наличие объектов данного типа в образе. Если некоторое свойство операционной системы может быть переключено в образе (например, разрешено/запрещено), то информация будет отображать состояние данного свойства. Два последних пункта в списке доступны только на ОС начиная с Windows 8. Если в папке запуска скрипта присутствует автономный образ операционной системы Windows - файл `install.wim`, то из него запрашивается информация об имеющихся индексах и пользователю предлагается ввести номер, с которым будет вестись работа. В следующем подменю о выбранном индексе отображается расширенная информация и пользователю предлагается смонтировать данный образ для продолжения работы. После монтирования образа на экране отображается основное меню скрипта. Если в папке запуска скрипта отсутствует файл `install.wim`, то сразу происходит переход в основное меню, а запросы информации будут направлены к текущей ОС. Работать с основным меню очень просто - достаточно нажать соответствующую клавишу для вывода информации на экран. По некоторым пунктам объём данных может быть большим, поэтому добавлены пункты для экспорта информации в создаваемый текстовый файл в папке запуска скрипта. Экспорт в файл - буква в верхнем регистре. Вначале командой `set` происходит установка переменных. Можно изменить предполагаемое имя файла образа `install.wim` (например, на `install.esd`). Можно изменить имя папки монтирования (по умолчанию `mount`) или задать путь, если папка монтирования должна находиться за пределами папки запуска. Также вначале определяется версия ОС, на которой запущен скрипт. Исходя из этого, делается вывод о том, какая версия системы DISM будет принимать команды скрипта, что влияет на доступность последних двух пунктов меню.

:pre_menu	Предварительное меню. Получение основной информации о <code>wim</code> -файле с контролем ошибок. Если отсутствует файл образа, то запуск в режиме Online.
:ind_menu	Индексное меню. Получение расширенной информации о выбранном «индексе» в <code>wim</code> -файле с контролем ошибок. Предложение смонтировать «индекс».
:adv_menu	Основное (расширенное) меню скрипта. После нажатия клавиши соответствующей выбранному пункту происходит переход к соответствующей метке в теле скрипта с последующим выполнением требуемых команд и возврат в данное меню. Все пункты вывода информации на экран однотипны и включают некоторую команду DISM. Нет необходимости их разбирать. Чем отображение информации на экране больший интерес представляет экспорт в текстовый файл. Информация записывается в табличной форме, что удобно для восприятия и сравнения. Для каждого типа информации - текстовый файл с определённым именем. Повторный экспорт по некоторому пункту не заменяет файл, а создаёт новый, добавляя числовой индекс к имени. Это удобно когда происходит сбор информации с нескольких разных индексов образа.
:mount	Монтирование образа. Предварительно создаётся папка монтирования. Контроль ошибок. Изменяется переменная определяющая спецификацию образа, теперь указывает на путь к автономному образу. Монтирование осуществляется в режиме <code>/ReadOnly</code> , что означает отсутствие возможности внесения каких-либо изменений в образ, включая даты и атрибуты файла образа
:unmount	Размонтирование образа. Если выполнялось интерактивное обслуживание (<code>/Online</code>), то размонтировать не нужно. Возвращение переменных к исходным значениям. Размонтирование с ключом <code>/Discard</code> , так как монтирование осуществлялось только для чтения. Отсутствие ключа неприемлемо.

Bat-файл получения информации из образа Windows:

```
@echo off
title Getting information from Windows image

set _file=install.wim
set _img=Online
set _mnt=mount
set _tool=7
set _word=Wim

ver | find "6.1" > nul || set _tool=8
if %_tool% GTR 7 set _word=Image

:pre_menu
cls
if not exist %_file% goto :adv_menu
dism /English /LogLevel:1 /Get-%_word%Info /%_word%File:%_file%
echo -----
if %ERRORLEVEL% NEQ 0 pause & exit
set /p _ind=Input index or press [Enter] for quit: || exit
if %_ind% EQU 0 goto :adv_menu
if %_ind% GTR 0 if %_ind% LEQ 24 goto :ind_menu
goto :pre_menu

:ind_menu
cls
dism /English /LogLevel:1 /Get-%_word%Info /%_word%File:%_file% /Index:%_ind%
echo -----
if %ERRORLEVEL% NEQ 0 pause & goto :pre_menu
choice /c abcdefghijklmnopqrstuvwxyz /n /m "Mount selected image? [m] "
if %ERRORLEVEL% EQU 13 goto :mount
goto :pre_menu

:adv_menu
cls
if %_img%==Online (echo DISM: Online Servicing)^
else (dism /English /LogLevel:1 /Get-Mounted%_word%Info)
echo -----
echo ~~~Menu~~~
echo [d] drivers ([D] - export)
echo [p] packages ([P] - export)
echo [f] features ([F] - export)
echo [l] languages
echo [e] editions
echo [a] app
if %_tool% GTR 7 echo [x] appx ([X] - export)
if %_tool% GTR 7 echo [s] assoc ([S] - export)
echo [q] quit (unmount)
echo -----
choice /c dDpPfFleaxXsSq /n /cs /m "Your choice: "
if %ERRORLEVEL% EQU 1 goto :drivers
if %ERRORLEVEL% EQU 2 goto :drivers-export
if %ERRORLEVEL% EQU 3 goto :packages
if %ERRORLEVEL% EQU 4 goto :packages-export
if %ERRORLEVEL% EQU 5 goto :features
if %ERRORLEVEL% EQU 6 goto :features-export
if %ERRORLEVEL% EQU 7 goto :languages
if %ERRORLEVEL% EQU 8 goto :editions
if %ERRORLEVEL% EQU 9 goto :app
if %ERRORLEVEL% EQU 10 if %_tool% GTR 7 (goto :appx) else (goto :adv_menu)
if %ERRORLEVEL% EQU 11 if %_tool% GTR 7 (goto :appx-export) else (goto :adv_menu)
if %ERRORLEVEL% EQU 12 if %_tool% GTR 7 (goto :assoc) else (goto :adv_menu)
if %ERRORLEVEL% EQU 13 if %_tool% GTR 7 (goto :assoc-export) else (goto :adv_menu)
goto :unmount

:drivers
cls
```

```

dism /%_img% /English /LogLevel:1 /Get-Drivers /All | more
echo -----
choice /c b /n /m "Back [b] "
goto :adv_menu

:drivers-export
cls
if exist drivers%_numd%.txt set /a _numd+=1 & goto :drivers-export
echo Getting list of drivers. Please wait...
dism /%_img% /English /LogLevel:1 /Get-Drivers /All /Format:Table > drivers%_numd%.txt
goto :adv_menu

:packages
cls
dism /%_img% /English /LogLevel:1 /Get-Packages | more
echo -----
choice /c b /n /m "Back [b] "
goto :adv_menu

:packages-export
cls
if exist packages%_nump%.txt set /a _nump+=1 & goto :packages-export
echo Getting list of packages. Please wait...
dism /%_img% /English /LogLevel:1 /Get-Packages /Format:Table > packages%_nump%.txt
goto :adv_menu

:features
cls
dism /%_img% /English /LogLevel:1 /Get-Features | more
echo -----
choice /c b /n /m "Back [b] "
goto :adv_menu

:features-export
cls
if exist features%_numf%.txt set /a _numf+=1 & goto :features-export
echo Getting list of features. Please wait...
dism /%_img% /English /LogLevel:1 /Get-Features /Format:Table > features%_numf%.txt
goto :adv_menu

:languages
cls
dism /%_img% /English /LogLevel:1 /Get-Intl
echo -----
choice /c b /n /m "Back [b] "
goto :adv_menu

:editions
cls
dism /%_img% /English /LogLevel:1 /Get-CurrentEdition
echo -----
dism /%_img% /English /LogLevel:1 /Get-TargetEditions
echo -----
choice /c b /n /m "Back [b] "
goto :adv_menu

:app
cls
if %_img%==Online goto :adv_menu
dism /%_img% /English /LogLevel:1 /Get-Apps
echo -----
dism /%_img% /English /LogLevel:1 /Get-AppPatches
echo -----
choice /c b /n /m "Back [b] "
goto :adv_menu

:appx

```

```

cls
dism /%_img% /English /LogLevel:1 /Get-ProvisionedAppxPackages | more
echo -----
choice /c b /n /m "Back [b] "
goto :adv_menu

:appx-export
cls
if exist appxes%_numx%.txt set /a _numx+=1 & goto :appx-export
echo Getting list of appx. Please wait...
dism /%_img% /English /LogLevel:1 /Get-ProvisionedAppxPackages > appxes%_numx%.txt
goto :adv_menu

:assoc
cls
dism /%_img% /English /LogLevel:1 /Get-DefaultAppAssociations | more
echo -----
choice /c b /n /m "Back [b] "
goto :adv_menu

:assoc-export
cls
if exist associat%_nums%.txt set /a _nums+=1 & goto :assoc-export
echo Getting list of assoc. Please wait...
dism /%_img% /English /LogLevel:1 /Get-DefaultAppAssociations > associat%_nums%.txt
goto :adv_menu

:mount
cls
md %_mnt%
dism /English /LogLevel:1 /Mount-%_word% /%_word%File:%_file% /Index:%_ind% /MountDir:%_mnt% /ReadOnly
if %ERRORLEVEL% NEQ 0 rd %_mnt% & pause & exit
set _img=Image:%_mnt%
goto :adv_menu

:unmount
cls
if not exist %_file% exit
if not %_img%==Online (
dism /English /LogLevel:1 /Unmount-%_word% /MountDir:%_mnt% /Discard
rd %_mnt%
)
set _img=Online
goto :pre_menu

```

Bat-скрипт добавления пакетов обновления в образ Windows

Рассмотрим скрипт, оперирующий командами DISM, который автоматизирует процесс и избавляет системных администраторов от рутинных действий. Универсальность скрипта состоит в том, что с его помощью можно установить пакеты обновления на «живую» систему, либо интегрировать в образ Windows - файл install.wim. Гибкость скрипта состоит в том, что его работу можно настроить, изменив значение нескольких переменных в начале всего кода. Будем добавлять к образу Windows так называемое Convenience rollup update (KB3125574), а также необходимое Servicing stack update (KB3177467). Но после интеграции в образ KB3177467, он (образ - install.wim), становится непригодным для дальнейших интеграций. Если устанавливать KB3177467 на «живую» ОС, то проблем не возникает. У данной проблемы есть два решения: 1) интегрировать KB3177467 последним 2) заменить «проблемное» обновление его предыдущей версией - KB3020369. Выбран второй вариант, реализовав в скрипте ветвление по условию. Файлы пакетов обновлений, указанные в скрипте, взяты для образца. По месту нахождения скрипта должны быть созданы папки x64 и x86 в которые нужно разместить нужные пакеты обновлений. Сам файл скрипта можно сохранить в кодировке ANSI если не использовать кириллицу для вывода информационных сообщений. А если хочется видеть русский текст, то нужно cmd-файл сохранить в кодировке OEM 866. Как уже отмечалось, скрипт может установить обновления или интегрировать их. Если в папке запуска отсутствует файл образа - install.wim, то скрипт выполняет установку обновлений в полностью автоматическом режиме. Если в папке запуска присутствует файл образа - install.wim, то скрипт

выполняет считывание из него информацию об имеющихся «индексах» и предлагает ввести номер. После этого отображается расширенная информация о выбранном «индексе» выдаётся запрос на монтирование. Нажатие любой клавиши приводит к возврату, а нажатие клавиши [m] запускает следующую цепочку действий: монтирование образа, интеграция обновлений, размонтирование образа, возврат в меню выбора «индекса». После чего можно выбрать другой «индекс» для интеграции обновлений. Выбор «индекса» под номером 0 запускает установку обновлений на «живую» ОС. Вначале командой set происходит установка переменных. Можно изменить предполагаемое имя файла образа install.wim (например, на install.esd). Можно изменить имя папки монтирования или задать путь, если папка монтирования должна находиться за пределами папки запуска. Можно изменить «уровень» логирования, для исключения информационных сообщений из файла лога системы.

:pre_menu	Предварительное меню. Получение основной информации о wim-файле с контролем ошибок. Если ошибка 2 - отсутствует wim-файл, то запуск в режиме Online. Максимальное количество «индексов» в одном образе равно 24.
:ind_menu	Индексное меню. Получение расширенной информации о выбранном «индексе» в wim-файле с контролем ошибок. Предложение смонтировать «индекс».
:bit	Определение разрядности ОС. Для интерактивного обслуживания считываются системные переменные среды, а для автономного обслуживания из образа запрашивается необходимая информация и в ней ищется ключевая строка.
:add	Добавление пакетов. Чтобы не делать лишней работы вначале происходит запрос списка уже добавленных (установленных или интегрированных) пакетов, а потом перед каждой командой запуска осуществляется вызов псевдофункции для проверки.
:exist	Псевдофункция проверки. Работа описана в статье Проверка присутствия пакета обновления в системе перед установкой. Код слегка подправлен.
:mount	Монтирование образа. Предварительно создаётся папка монтирования. Контроль ошибок. Изменяется переменная определяющая спецификацию образа, теперь указывает на путь к автономному образу.
:unmount	Размонтирование образа. Если выполнялось интерактивное обслуживание (/Online), то размонтировать не нужно. Возвращение переменных к исходным значениям.

Bat-скрипт добавления пакетов обновления в образ Windows:

<pre>@echo off Title Convenience Rollup - most important updates after SP1 set _arch=x64 set _file=install.wim set _img=Online set _mnt=mount set _lev=2 :pre_menu cls dism /Get-WimInfo /LogLevel:%_lev% /WimFile:%_file% echo ----- if %ERRORLEVEL% NEQ 0 if %ERRORLEVEL% EQU 2 (goto :bit) else (pause & exit) set /p _ind=Input index or press [Enter] for quit: exit if %_ind% EQU 0 goto :bit if %_ind% GTR 0 if %_ind% LEQ 24 goto :ind_menu goto :pre_menu :ind_menu cls dism /Get-WimInfo /LogLevel:%_lev% /WimFile:%_file% /Index:%_ind% echo ----- if %ERRORLEVEL% NEQ 0 pause & goto :pre_menu choice /c abcdefghijklmnopqrstuvwxyz /n /m "Mount selected image? [m] " if %ERRORLEVEL% EQU 13 goto :mount goto :pre_menu :bit if %_img%==Online (if "%PROCESSOR_ARCHITECTURE%" == "x86" if not defined PROCESSOR_ARCHITEW6432 set _arch=x86)^ else (dism /Get-WimInfo /LogLevel:%_lev% /WimFile:%_file% /Index:%_ind% /English find "Architecture : x86" > nul && set _arch=x86) goto :add</pre>
--

```

:add
cls
echo Getting packages list. Please wait...
dism /%_img% /Get-Packages /LogLevel:%_lev% /English > %TEMP%\packages.txt
echo -----
if not %_img%==Online (
echo Add: Servicing stack update KB3020369
call :exist .\%_arch%\Windows6.1-KB3020369-%_arch%.cab ||^
dism /%_img% /Add-Package /LogLevel:%_lev% /PackagePath:.\%_arch%\Windows6.1-KB3020369-%_arch%.cab /NoRestart
echo -----
) else (
echo Add: Servicing stack update KB3177467
call :exist .\%_arch%\Windows6.1-KB3177467-%_arch%.cab ||^
dism /%_img% /Add-Package /LogLevel:%_lev% /PackagePath:.\%_arch%\Windows6.1-KB3177467-%_arch%.cab /NoRestart
echo -----
)
echo Add: Convenience rollup update KB3125574
call :exist .\%_arch%\Windows6.1-KB3125574-v4-%_arch%.cab ||^
dism /%_img% /Add-Package /LogLevel:%_lev% /PackagePath:.\%_arch%\Windows6.1-KB3125574-v4-%_arch%.cab /NoRestart
echo -----
del %TEMP%\packages.txt
if not exist %_file% exit
pause
goto :unmount

:exist
dism /%_img% /Get-PackageInfo /LogLevel:%_lev% /PackagePath:%1 /English | find "Package Identity" | findstr /g:/ %TEMP%\packages.txt > nul
exit /b

:mount
cls
md %_mnt%
dism /Mount-Wim /LogLevel:%_lev% /WimFile:%_file% /Index:%_ind% /MountDir:%_mnt%
if %ERRORLEVEL% NEQ 0 rd %_mnt% & pause & exit
set _img=Image:%_mnt%
goto :bit

:unmount
cls
if not %_img%==Online (
dism /Unmount-Wim /LogLevel:%_lev% /MountDir:%_mnt% /Commit
rd %_mnt%
)
set _arch=x64
set _img=Online
goto :pre_menu

```

Ват-скрипт отключения компонентов Windows

Рассмотрим скрипт, отключающий «ненужные» компоненты. Команды скрипта обращаются к средствам DISM и могут быть применимы как к установленной системе, так и к автономному образу. Скрипт определяет версию утилиты dism.exe и подставляет нужные команды. Реализовано ветвление по версии целевой ОС, таким образом, что в одном скрипте можно указать отключаемые компоненты для разных версий Windows. Отключение компонентов данным скриптом обратимо. Сделано это главным образом для того, чтобы убрать из меню Пуск ссылки на ненужные мне программы. Здесь для примера указано по два компонента для каждой версии Windows, которые будут отключены. Тем, кто будет использовать данный скрипт в своей работе, необходимо дополнить его. Определите компоненты, которые Вам не понадобятся и добавьте их отключение в скрипт. Путём нехитрых изменений, можно наоборот - включать компоненты. Для своих нужд оставим только Internet Explorer, Media Player, Windows Search и .NET Framework 3.5.1 (последний отключён производителем в новых версиях Windows, с удалением

файлов). Данный скрипт может отключать компоненты как в онлайн системе, так и в автономном образе. Если в папке запуска отсутствует файл образа - install.wim, то скрипт выполняет отключение компонентов в полностью автоматическом режиме. Если в папке запуска присутствует файл образа - install.wim, то скрипт выполняет считывание из него информацию об имеющихся «индексах» и предлагает ввести номер. После этого отображается расширенная информация о выбранном «индексе» выдаётся запрос на монтирование. Нажатие любой клавиши приводит к возврату, а нажатие клавиши [m] запускает следующую цепочку действий: монтирование образа, отключение компонентов, размонтирование образа, возврат в меню выбора «индекса». После чего можно выбрать другой «индекс» для отключения компонентов. Выбор «индекса» под номером 0 запускает отключение компонентов на «живой» ОС. Вначале командой set происходит установка переменных. Можно изменить предполагаемое имя файла образа install.wim (например, на install.esd). Можно изменить имя папки монтирования или задать путь, если папка монтирования должна находиться за пределами папки запуска. От установки «уровня» логирования, как в предыдущем скрипте по пакетам обновления, решил отказаться и напрямую прописал во всех командах информирование только об ошибках - /LogLevel:1. Также во всех командах добавлен ключ /English чтобы все сообщения отображались на английском языке. Скрипт может быть запущен на разных версиях ОС, и таким образом обращаться к разным версиям системы DISM, которые отличаются набором команд. Так, в версиях после 6.1 во всех командах, слово Wim заменено на Image, хотя и оставлены старые «наименования» команд для обратной совместимости. В самом начале работы скрипта определяется версия утилиты dism.exe и в дальнейшем во все команды подставляется нужное слово. При обслуживании образа вышестоящей версии на системе нижестоящей версии возможны зависания и ошибки. Т. е., не нужно пытаться на Windows 8.1 обрабатывать файл образа Windows 10. Также для успешного выполнения всех команд требуется отключить Контроль учётных записей - параметр EnableLUA установить в значение 0.

:pre_menu	Предварительное меню. Получение основной информации о wim-файле с контролем ошибок. Если отсутствует файл образа, то запуск в режиме Online. Максимальное количество «индексов» в одном образе равно 24.
:ind_menu	Индексное меню. Получение расширенной информации о выбранном «индексе» в wim-файле с контролем ошибок. Предложение смонтировать «индекс».
:version	Определение версии целевой системы. Если находится необходимая строка, то осуществляется переход к указанной метке. Если строка не обнаруживается, то происходит размонтирование.
:disable-7 :disable-9 :disable-A	Метки целевых систем обслуживания. Я называю Windows 8.1 - девяткой, а Windows 10 - Windows A (кто в теме тот поймёт), поэтому такие имена у меток. Вначале данного блока происходит запрос состояния всех компонентов с сохранением в файл features.txt. В дальнейшем, чтобы ускорить процесс отключения и не пытаться отключить то, что и так отключено, происходит предварительная проверка состояния каждого компонента. В конце блока - удаление временных файлов и размонтирование.
:state	Проверка состояния компонента. Псевдо-функция, возвращающая результат по коду ошибки в глобальную переменную %ERRORLEVEL%. Строка с именем компонента ищется в файле features.txt и если находится и также обнаруживается слово «Enable», то сигнал к отключению.
:mount	Монтирование образа. Предварительно создаётся папка монтирования. Контроль ошибок. Изменяется переменная определяющая спецификацию образа, теперь указывает на путь к автономному образу.
:unmount	Размонтирование образа. Если выполнялось интерактивное обслуживание (/Online), то размонтировать не нужно. Возвращение переменных к исходным значениям.

Bat-скрипт отключения компонентов Windows:

<pre>@echo off title Disabling features in Windows image set _file=install.wim set _img=Online set _mnt=mount set _tool=7 set _word=Wim dism /English /LogLevel:1 /Get-Help find "Version: 6.1" > nul set _tool=8 if %_tool% GTR 7 set _word=Image :pre_menu cls</pre>

```

if not exist %_file% goto :version
dism /English /LogLevel:1 /Get-%_word%Info /%_word%File:%_file%
echo -----
if %ERRORLEVEL% NEQ 0 pause & exit
set /p _ind=Input index or press [Enter] for quit: || exit
if %_ind% EQU 0 goto :version
if %_ind% GTR 0 if %_ind% LEQ 24 goto :ind_menu
goto :pre_menu

:ind_menu
cls
dism /English /LogLevel:1 /Get-%_word%Info /%_word%File:%_file% /Index:%_ind%
echo -----
if %ERRORLEVEL% NEQ 0 pause & goto :pre_menu
choice /c abcdefghijklmnopqrstuvwxyz /n /m "Mount selected image? [m] "
if %ERRORLEVEL% EQU 13 goto :mount
goto :pre_menu

:version
dism /%_img% /English /LogLevel:1 /Get-Help | find "Image Version: 6.1" > nul && goto :disable-7
dism /%_img% /English /LogLevel:1 /Get-Help | find "Image Version: 6.3" > nul && goto :disable-9
dism /%_img% /English /LogLevel:1 /Get-Help | find "Image Version: 10" > nul && goto :disable-A
goto :unmount

:disable-7
cls
echo Getting list of features. Please wait...
dism /%_img% /English /LogLevel:1 /Get-Features /Format:Table > %TEMP%\features.txt
echo -----
set /a _num+=1
echo %_num% Disable: WindowsGadgetPlatform
call :state WindowsGadgetPlatform &&^
dism /%_img% /English /LogLevel:1 /Disable-Feature /FeatureName:WindowsGadgetPlatform /NoRestart
echo -----
set /a _num+=1
echo %_num% Disable: MediaCenter
call :state MediaCenter &&^
dism /%_img% /English /LogLevel:1 /Disable-Feature /FeatureName:MediaCenter /NoRestart
echo -----
del %TEMP%\features.txt
if not exist %_file% exit
goto :unmount

:disable-9
cls
echo Getting list of features. Please wait...
dism /%_img% /English /LogLevel:1 /Get-Features /Format:Table > %TEMP%\features.txt
echo -----
set /a _num+=1
echo %_num% Disable: Printing-Foundation-Features
call :state Printing-Foundation-Features &&^
dism /%_img% /English /LogLevel:1 /Disable-Feature /FeatureName:Printing-Foundation-Features /NoRestart
echo -----
set /a _num+=1
echo %_num% Disable: Windows-Defender-Default-Definitions
call :state Windows-Defender-Default-Definitions &&^
dism /%_img% /English /LogLevel:1 /Disable-Feature /FeatureName:Windows-Defender-Default-Definitions /NoRestart
echo -----
del %TEMP%\features.txt
if not exist %_file% exit
goto :unmount

:disable-A
cls
echo Getting list of features. Please wait...
dism /%_img% /English /LogLevel:1 /Get-Features /Format:Table > %TEMP%\features.txt
echo -----

```

```

set /a _num+=1
echo %_num% Disable: MicrosoftWindowsPowerShellV2Root
call :state MicrosoftWindowsPowerShellV2Root &&^
dism /%_img% /English /LogLevel:1 /Disable-Feature /FeatureName:MicrosoftWindowsPowerShellV2Root /NoRestart
echo -----
set /a _num+=1
echo %_num% Disable: NetFx4-AdvSrvs
call :state NetFx4-AdvSrvs &&^
dism /%_img% /English /LogLevel:1 /Disable-Feature /FeatureName:NetFx4-AdvSrvs /NoRestart
echo -----
del %TEMP%\features.txt
if not exist %_file% exit
goto :unmount

:state
findstr %1 %TEMP%\features.txt | find "Enable" > nul
exit /b

:mount
cls
md %_mnt%
dism /English /LogLevel:1 /Mount-%_word% /%_word%File:%_file% /Index:%_ind% /MountDir:%_mnt%
if %ERRORLEVEL% NEQ 0 rd %_mnt% & pause & exit
set _img=Image:%_mnt%
goto :version

:unmount
cls
if not %_img%==Online (
dism /English /LogLevel:1 /Unmount-%_word% /MountDir:%_mnt% /Commit
rd %_mnt%
)
set _img=Online
goto :pre_menu

```

Bat-скрипт удаления универсальных приложений «Магазина» Windows (MicrosoftStore)

Представленное здесь решение предназначено для обслуживания образов Windows 8, 8.1, 10. Может применяться как к автономным файлам образов в формате *.wim, так и к развёрнутой (установленной) системе. Как и в предыдущих моих скриптах использую средства системы DISM. Однако в ходе разработки выяснилось, что с помощью команды `dism` нет возможности удалить универсальные приложения из установленной системы (обслуживание онлайн) - только средствами PowerShell. Пришлось делать выбор: либо писать скрипт полностью на PS, либо использую возможности командного интерпретатора (CMD), осуществить вызов отдельных операций из PS. Я выбрал второй вариант, в связи с чем, представленное решение состоит из двух файлов: основного скрипта (сохранить с любым именем *.cmd) и дополнительного (сохранить с именем RemoveAppxes.ps1). Вначале командой `set` происходит установка переменных. Можно изменить предполагаемое имя файла образа `install.wim` (например, на `install.esd`). Можно изменить имя папки монтирования (по-умолчанию `mount`) или задать путь, если папка монтирования должна находится за пределами папки запуска.

:pre_menu	Предварительное меню. Получение основной информации о wim-файле с контролем ошибок. Если отсутствует файл образа, то запуск в режиме Online.
:ind_menu	Индексное меню. Получение расширенной информации о выбранном «индексе» в wim-файле с контролем ошибок. Предложение смонтировать «индекс».
:version	Проверка версии целевой системы. Если оказывается, что скрипт пытаются применить к образу Windows 7, то происходит размонтирование. Иначе, выполняется переход к следующей метке.
:remove (:filter, :action)	Основной блок выполнения скрипта. Вначале запрашивается список интегрированных универсальных приложений с сохранением во временный файл appxes.txt. Затем в цикле <code>for</code> из полученного списка последовательно выбираются строки и передаются в псевдофункцию <code>:filter</code> . Строки фильтруются поиском по регулярному выражению «Microsoft.*» - если найдена, то передаются в следующую псевдофункцию <code>:action</code> . Здесь по регулярному выражению «Microsoft.*_» отфильтровываются названия универсальных приложений, передаваемые в дальнейшем утилите <code>dism</code> для удаления. Чтобы лучше понять логику работы нужно взглянуть на содержимое файла <code>appxes.txt</code> . Так как файл удаляется по

	завершению работы цикла, то чтобы с ним ознакомиться нужно предварительно вручную его получить и сохранить. Одной командой происходит удаление всех ассоциаций открытия отдельных файлов универсальными приложениями. Для распространённых типов файлов остаются ассоциации со стандартными классическими приложениями Windows. Удаление (деинсталляция) универсальных приложений на установленной системе происходит вызовом составной команды на языке скриптов PowerShell. Файл дополнительного скрипта RemoveAppxes.ps1 вначале выполняет запрос списка всех универсальных приложений для всех пользователей, а затем «по конвейеру» передаёт их команде на удаление. Вроде бы ничего сложного, однако, чтобы выполнить эту операцию скрипт PS должен быть запущен с повышенными правами, причём работать от имени Администратора не достаточно. Для успешного запуска нужно вставить тот сложный код в команде powershell чтобы получить повышение прав. Защита которая легко обходится! Вот за этот «идиотизм» мне и не приглянулся PowerShell.
:mount	Монтирование образа. Предварительно создаётся папка монтирования. Контроль ошибок. Изменяется переменная определяющая спецификацию образа, теперь указывает на путь к автономному образу.
:unmount	Размонтирование образа. Если выполнялось интерактивное обслуживание (/Online), то размонтировать не нужно. Возвращение переменных к исходным значениям.

Основной bat-скрипт:

```
@echo off
title Deleting Appxes in Windows image

set _file=install.wim
set _img=Online
set _mnt=mount

dism /English /LogLevel:1 /Get-Help | find "Version: 6.1" > nul && exit

:pre_menu
cls
if not exist %_file% goto :version
dism /English /LogLevel:1 /Get-ImageInfo /ImageFile:%_file%
echo -----
if %ERRORLEVEL% NEQ 0 pause & exit
set /p _ind=Input index or press [Enter] for quit: || exit
if %_ind% EQU 0 goto :version
if %_ind% GTR 0 if %_ind% LEQ 24 goto :ind_menu
goto :pre_menu

:ind_menu
cls
dism /English /LogLevel:1 /Get-ImageInfo /ImageFile:%_file% /Index:%_ind%
echo -----
if %ERRORLEVEL% NEQ 0 pause & goto :pre_menu
choice /c abcdefghijklmnopqrstuvwxyz /n /m "Mount selected image? [m] "
if %ERRORLEVEL% EQU 13 goto :mount
goto :pre_menu

:version
dism /%_img% /English /LogLevel:1 /Get-Help | find "Image Version: 6.1" > nul && goto :unmount
goto :remove

:remove
cls
echo Getting list of Appxes. Please wait...
dism /%_img% /English /LogLevel:1 /Get-ProvisionedAppxPackages > %TEMP%\appxes.txt
echo -----
set _num=1
for /f "skip=8 tokens=3" %%a in (%TEMP%\appxes.txt) do call :filter %%a
del %TEMP%\appxes.txt
echo Removes default application associations
dism /%_img% /English /LogLevel:1 /Remove-DefaultAppAssociations
echo -----
if %_img%==Online (
echo Remove current Appxes for AllUsers
```



```

powershell -Command Start-Process powershell -ArgumentList '-NoProfile -WindowStyle Hidden -ExecutionPolicy Bypass -File RemoveAppxes.ps1' -Verb RunAs
)
if not exist %_file% exit
goto :unmount

:filter
echo %1 | findstr /ric:"Microsoft.*" > nul
if %ERRORLEVEL% EQU 0 call :action %1
exit /b

:action
echo %1 | findstr /ric:"Microsoft.*_" > nul
if %ERRORLEVEL% NEQ 0 (
set /a _num+=1
echo %_num% Remove: %1
) else (
dism /%_img% /English /LogLevel:1 /Remove-ProvisionedAppxPackage /PackageName:%1
echo -----
)
exit /b

:mount
cls
md %_mnt%
dism /English /LogLevel:1 /Mount-Image /ImageFile:%_file% /Index:%_ind% /MountDir:%_mnt%
if %ERRORLEVEL% NEQ 0 rd %_mnt% & pause & exit
set _img=Image:%_mnt%
goto :version

:unmount
cls
if not %_img%==Online (
dism /English /LogLevel:1 /Unmount-Image /MountDir:%_mnt% /Commit
rd %_mnt%
)
set _img=Online
goto :pre_menu

```

Дополнительный PowerShell-скрипт:

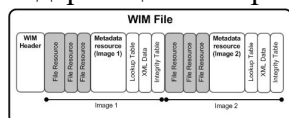
```
Get-AppxPackage -AllUsers | Remove-AppxPackage
```

Два файла должны размещаться в одной папке. Если в папке запуска отсутствует файл образа - install.wim, то скрипты выполняют работу в полностью автоматическом режиме. Если в папке запуска присутствует файл образа - install.wim, то скрипт выполняет считывание из него информации об имеющихся «индексах» и предлагает ввести номер. После этого отображается расширенная информация о выбранном «индексе» выдаётся запрос на монтирование. Нажатие любой клавиши приводит к возврату, а нажатие клавиши [m] запускает следующую цепочку действий: монтирование образа, проверка версии ОС, удаление дистрибутивов приложений (по циклу), удаление всех ассоциаций, деинсталляция установленных приложений (при запуске онлайн), размонтирование образа, возврат в меню выбора «индекса». После чего можно выбрать другой «индекс» образа для удаления приложений. Выбор «индекса» под номером 0 запускает обслуживание онлайн.

Управление компонентами образов Windows с помощью еонсольной утилиты DISM

Команда DISM используется для просмотра, установки, удаления, и настройки компонентов в образах Windows формата WIM (Windows IMaging Format). Начиная с Windows 7, утилита dism.exe (Deployment Image Servicing and Management) входит в стандартный комплект программного обеспечения установленной ОС Windows. Представляет собой средство командной строки, которое может использоваться для обслуживания образов Windows и заменяет диспетчер пакетов (Pkgmgr.exe), PEimg и Intlcfg, которые включались в Windows Vista. Функциональные возможности этих средств теперь собраны в одном инструменте dism.exe. Кроме того, были добавлены новые функции, улучшающие возможности автономного обслуживания образов. В отличие от сектор-ориентированных форматов (.iso, .tib, .bin), образ WIM является файл-ориентированным форматом, то есть его наименьшим логическим элементом является

файл. Этот формат был разработан компанией Microsoft в качестве основного средства развертывания операционных систем Windows Vista и более поздних. Его достоинствами является аппаратная независимость и возможность хранения только одной копии файла, имеющего множество ссылок в дереве файловой системы (single instance storage), что обеспечивает компактность образов и высокую скорость их развертывания. WIM файл может включать в себя несколько образов, которые определяются по числовому индексу или уникальному имени. Технология единственной копии множества одинаковых файлов single instance storage при добавлении нового образа к существующим позволяет уменьшить размер WIM-базы, если будут присутствовать совпадающие элементы. WIM-образ может быть также разбит на несколько файлов, которым присваивается расширение «.swm». WIM-образ может быть смонтирован как логический диск в среде Windows с присвоением ему буквы диска для облегчения редактирования его содержимого (благодаря этому образ WIM можно достаточно просто сконвертировать в ISO-образ). Помимо штатных утилит от Microsoft, извлечь файлы из WIM-образа можно также при помощи архиватора 7-Zip. WIM-образы могут быть загрузочными. В частности, установка всех современных операционных систем семейства Windows выполняется с использованием файла образа с именем BOOT.WIM содержащего загрузочную версию среды предустановки Windows PE (Windows Preinstall Environment), из которой выполняется процесс установки самой системы с использованием содержимого файла образа с именем INSTALL.WIM. После стандартной установки Windows, создается среда восстановления Windows RE (Recovery Environment), загрузка которой выполняется из образа WinRE.WIM для восстановления работоспособности при проблемах с загрузкой основной системы. Обычно образ winre.wim располагается в скрытом разделе, которому не присваивается буква логического диска. Структура файла WIM, содержащего 2 образа:



Как видно, wim-файл содержит 6 типов содержимого:

- WIM Header - заголовок wim-файла, описывающий содержимое, атрибуты, версию, размер, тип сжатия и т. п.
- File Resources - последовательность пакетов, содержащих упакованные данные файлов в составе образа.
- Metadata Resource - содержит информацию о файлах в составе образа, включая структуру папок и атрибуты.
- Lookup Table - таблица расположений и размеров ресурсных файлов в файле .wim
- XML Data - данные в формате XML с дополнительными сведениями об образе. Представляют собой несжатый текст в кодировке Unicode.
- Integrity Table - таблица целостности, используемая для контроля работоспособного состояния образа.

Процесс создания образа WIM:

1. Создается заголовок WIM Header файла образа, например C:\data.wim
2. В памяти создаются метаданные при сканировании и индексировании каталога исходных данных, сворачиваемого в образ, например - C:\Sources\
3. На основе метаданных содержимое файлов из каталога исходных данных записывается с сжатием в файл образа в виде порций пакетов File Resources , и в это же время в памяти строится таблица размещения порций в образе Lookup Table
4. По завершению создания образа, таблица Lookup Table записывается из памяти в файл .wim, затем дописываются данные XML и контроля целостности.
5. Обновляется начальный заголовок файла WIM.

Процесс извлечения файлов из образа:

1. Считывается заголовок WIM Header и определяется место расположения метаданных в образе WIM.
2. Метаданные переписываются в память компьютера и строится таблица расположения файлов внутри образа.
3. Создается структура каталогов на основе содержимого образа.
4. Файлы из образа обрабатываются с использованием построенных таблиц и переписываются в созданную структуру каталогов.

Выполнение DISM.EXE без параметров выводит на экран краткую справку по ее использованию. Журнал утилиты сохраняется в файле C:\Windows\Logs\DISM\dism.log

Формат командной строки

DISM.exe	{/Image:путь_к_автономному_образу		/Online}	[параметры_dism]
{служебная команда}	[аргументы служебной команды]			

Формат команды DISM отличается в зависимости от того, работает ли утилита с автономным образом или с данными ОС Windows (режим /Online), в среде которой выполняется DISM.EXE. Для работы утилиты требуется запуск командной строки от имени администратора.

Служебные команды для работы с образами дисков:

- /Split-Image - Делит существующий файл .wim или файл .ffu на несколько файлов WIM/FFU, доступных только для чтения.
- /Apply-Image - Применение новых параметров образа.
- /Get-MountedImageInfo - Отображает сведения о подключенных WIM- и VHD-образах.
- /Get-ImageInfo - Отображает сведения об образах в WIM- или VHD- файле.
- /Commit-Image - Сохраняет изменения подключенного WIM- или VHD-образа.
- /Unmount-Image - Отключает подключенный WIM- или VHD-образ.
- /Mount-Image - Подключает образ из WIM- или VHD-файла.
- /Remount-Image - Восстанавливает каталог подключения образа.
- /Cleanup-Mountpoints - Удаляет ресурсы, связанные с поврежденными подключенными образами.

Служебные команды:

- /Apply-CustomDataImage - Консервирует файлы, содержащиеся в настраиваемом образе данных.
- /Capture-CustomImage - Записывает настройки в разностный WIM-файл в системе WIMBoot. Записанные каталоги включают все вложенные папки и данные.
- /Get-WIMBootEntry - Выводит записи конфигурации WIMBoot для указанного тома диска.
- /Update-WIMBootEntry - Обновляет запись конфигурации WIMBoot для указанного тома диска.
- /List-Image - Отображает список файлов и папок в указанном образе.
- /Delete-Image - Удаляет указанный образ тома из WIM-файла, содержащего несколько образов томов.
- /Export-Image - Экспортирует копию указанного образа в другой wim-файл.
- /Append-Image - Добавление другого образа в WIM-файл.
- /Capture-Image - Записывает образ диска в новый WIM-файл. Записанные каталоги включают все подкаталоги и данные.
- /Get-MountedWimInfo - Отображает сведения о подключенных из WIM образах.
- /Get-WimInfo - Отображает сведения об образах в WIM-файле.
- /Commit-Wim - Сохраняет изменения подключенного из WIM образа.
- /Unmount-Wim - Отключает подключенный из WIM образ.
- /Mount-Wim - Подключает образ из WIM-файла.
- /Remount-Wim - Восстанавливает потерянный каталог подключения WIM.
- /Cleanup-Wim - Удаляет ресурсы, связанные с поврежденными подключенными WIM-образами.

Спецификации образа:

- /Online - Задаёт работающую операционную систему как назначение.
- /Image - Задаёт путь к корневому каталогу автономного образа Windows.

Параметры DISM:

- /English - Отображает данные, выводимые командной строкой, на английском языке.
- /Format - Задаёт формат отчёта. Допустимые значения - /Format:Table или /Format:List, определяют формат в виде таблицы или списка. По умолчанию используется список.
- /WinDir - Задаёт путь к каталогу Windows.
- /SysDriveDir - Задаёт путь к файлу системного загрузчика с именем BootMgr.
- /LogPath - Задаёт путь к файлу журнала.
- /LogLevel - Задаёт уровень ведения журнала событий (1-4).
- /NoRestart - Запрещает автоматическую перезагрузку и вывод пользователю предложений выполнить перезагрузку.
- /Quiet - Скрывает все сообщения за исключением сообщений об ошибках.
- /ScratchDir - Задаёт путь к каталогу временных файлов.

Примеры:

dism.exe /Mount-Wim /? – подсказка по команде /Mount-Wim

dism.exe /Format /? - отобразить справку о форматах вывода DISM

dism.exe /Online /Get-Drivers /? - подсказка для режима вывода сведения о драйверах в образе или работающей Windows (/Online). По умолчанию выполняется извлечение списка сторонних драйверов. Чтобы вывести весь список драйверов, используйте параметр /all

dism.exe /Online /Get-Drivers /format=table - отобразить список сторонних драйверов текущей Windows в виде таблицы

dism.exe /List-Image /Imagefile:C:\test\images\myimage.wim /index:1 - отобразить полное содержимое первого образа из файла, заданного параметром /Imagefile

dism /unmount-wim /MountDir:C:\test\offline /commit - отключить автономный образ, подключенный к каталогу C:\test\offline с сохранением сделанных изменений (сохранение с фиксацией, задаваемое параметром /commit)

dism /unmount-wim /MountDir:C:\test\offline /Discard - то же, что и в предыдущем примере, но без сохранения изменений в содержимом образа (сохранение без фиксации)

dism.exe /Image:C:\test\offline /Add-Driver /Driver:D:\Drivers /recurse - добавить в образ все драйверы из каталога D:\Drivers и его подкаталогов

dism.exe /Image:C:\test\offline /Add-Driver /Driver:D:\Drivers\Usb\Usb3.inf - добавить в автономный образ драйвер на основе информации из файла D:\Drivers\Usb\Usb3.inf

dism.exe /Image:C:\test\offline /Add-Driver /DriverName:"C:\Drivers\1.inf" /DriverName:"C:\Drivers\2.inf" /DriverName:"C:\Drivers\3.inf" - добавление нескольких драйверов в одной команде. При добавлении нескольких драйверов, их установка выполняется в том же порядке, в каком они перечисляются в командной строке DISM

dism.exe /Commit-Image /MountDir:C:\test\offline - сохранить изменения в образе, примонтированном к каталогу, заданному параметром /MountDir

dism.exe /Online /Cleanup-Image /CheckHealth - проверить наличие признака повреждения образа или хранилища компонент

dism.exe /Online /Cleanup-Image /ScanHealth - сканирование хранилища системных компонент. В Windows 7 кроме сканирования, выполняется еще и восстановление целостности. Данный вариант команды не выполняется и сопровождается сообщением Ошибка 87. Параметр ScanHealth не распознан в этом контексте, если в Windows 7 не установлено обновление KB2966583

dism.exe /Online /Cleanup-Image /RestoreHealth - проверка хранилища компонентов и автоматическое восстановление

dism.exe /Online /Cleanup-Image /StartComponentCleanup - очистить замененные компоненты и уменьшить размер хранилища (папки \Windows\WinSxS)

dism.exe /Online /Cleanup-Image /StartComponentCleanup /ResetBase - то же, что и в предыдущем примере, плюс сброс базы замененных компонентов. После сброса базы, установленные обновления Windows удалить невозможно

dism.exe /Online /Cleanup-Image /AnalyzeComponentStore - создание отчета о состоянии локального хранилища компонентов WinSxS

dism /Online /Get-Features - отобразить список компонентов текущей системы

dism.exe /Online /Enable-Feature:SNMP - включить компонент "SNMP-протокол"

dism.exe /Online /Get-Featureinfo:TelnetClient - отобразить сведения о компоненте "Клиент Telnet"

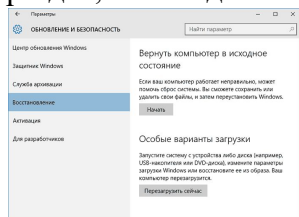
dism.exe /Online /Disable-Feature:SNMP - выключить компонент "SNMP-протокол"

Восстановление Windows 10

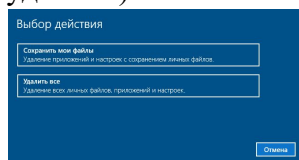
Windows 10 предлагает множество функций восстановления системы, среди которых - возврат компьютера в исходное состояние и точки восстановления, создание полного образа системы на внешнем жестком диске или DVD, а также запись USB диска восстановления (который стал лучше, чем в предыдущих системах).

- Возврат компьютера или ноутбука в исходное состояние

Первая функция восстановления, на которую следует обратить внимание - возврат Windows 10 в исходное состояние, доступ к которой можно получить, нажав по значку уведомлений, выбрав пункт «Все параметры» - «Обновление и безопасность» - «Восстановление» (есть еще один способ попасть в этот раздел, без входа в Windows 10, описан ниже).



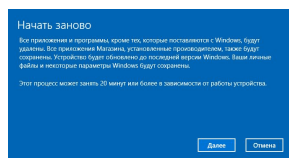
На случай, если Windows 10 не запускается, вы можете запустить откат системы с диска восстановления или дистрибутива ОС, о чем - ниже. Если в пункте «Вернуть в исходное состояние» нажать «Начать», вам будет предложено либо полностью очистить компьютер и переустановить Windows 10 (при этом, загрузочная флешка или диск не требуются, будут задействованы файлы, имеющиеся на компьютере), либо сохранить ваши личные файлы (установленные программы и настройки, тем не менее, будут удалены).



Еще один простой способ получить доступ к этой возможности, даже без входа в систему - на экране входа в систему (там, где вводится пароль), нажать по кнопке питания и, удерживая клавишу Shift, нажать «Перезагрузка». На открывшемся экране выберите «Диагностика», а затем - «Вернуть в исходное состояние». Плюсы данного способа восстановления - вам не требуется иметь дистрибутив системы, переустановка Windows 10 происходит автоматически и тем самым минимизирует вероятность некоторых ошибок, совершаемых начинающими пользователями. Более того, в последних версиях системы через эти же параметры возможно Восстановление Windows 10 из облака, то есть сброс на заводские настройки происходит с использованием оригинального образа на серверах Microsoft. Главный минус - в случае выхода из строя жесткого диска или серьезного повреждения файлов ОС восстановить систему таким образом не получится, но тут могут пригодиться следующие два варианта - диск восстановления или создание полной резервной копии Windows 10 встроенными средствами системы на отдельном жестком диске (в том числе внешнем) или DVD дисках. Подробнее о способе и его нюансах: Как сбросить Windows 10 или автоматическая переустановка системы.

- Автоматическая чистая установка Windows 10

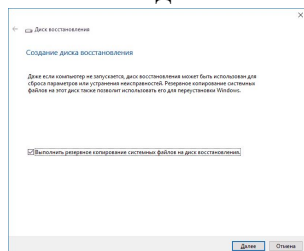
Функция «Начать заново» (или «Start Fresh») выполняет автоматическую чистую установку системы.



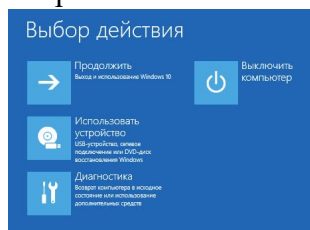
- Диск восстановления Windows 10

Под диском понимается USB-накопитель, например, обычная флешка, а название сохранилось с тех пор, когда была возможность записывать CD и DVD диски восстановления. В предыдущих версиях ОС диск восстановления содержал лишь утилиты для попыток автоматического и ручного восстановления установленной системы (весьма полезные), в свою очередь диск восстановления Windows 10, помимо них, может содержать и образ ОС для восстановления, то есть с него можно запустить возврат в исходное состояние, как это было описано в предыдущем разделе, автоматически переустановив систему на

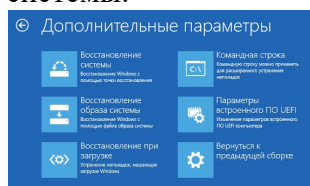
компьютере. Для записи такой флешки, зайдите в панель управления и выберите «Восстановление». Уже там вы найдете необходимый пункт - «Создание диска восстановления».



Если при создании диска вы отметите пункт «Выполнить резервное копирование системных файлов на диск восстановления», то итоговый накопитель может быть использован не только для действий по исправлению возникших проблем вручную, но и для быстрой переустановки Windows 10 на компьютере.



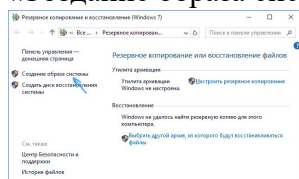
После загрузки с диска восстановления (нужно будет поставить загрузку с флешки или использовать boot menu), вы увидите меню выбора действия, где в разделе «Диагностика» (и в «Дополнительные параметры» внутри этого пункта) вы можете: вернуть компьютер в исходное состояние, используя файлы на флешке; зайти в БИОС (Параметры встроенного ПО UEFI); попробовать восстановить систему с помощью точки восстановления; запустить автоматическое восстановление при загрузке; использовать командную строку для восстановления загрузчика Windows 10 и других действий; восстановить систему из полного образа системы.



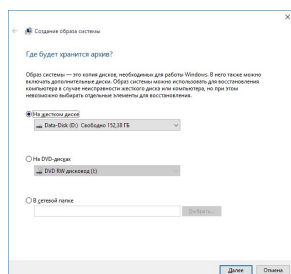
Иметь такой накопитель в чем-то может быть даже удобнее, чем просто загрузочную флешку Windows 10 (хотя и с нее можно запустить восстановление, нажав соответствующую ссылку внизу слева в окне с кнопкой «Установить» после выбора языка).

- Создание полного образа системы для восстановления Windows 10

В Windows 10 сохранилась возможность создавать полный образ восстановления системы на отдельном жестком диске (в том числе внешнем) или нескольких DVD-дисках. Ниже описан лишь один способ создания образа системы, если вас интересуют другие варианты, изложенные более подробно, см. инструкцию Резервная копия Windows 10. Отличие от предыдущего варианта заключается в том, что таким образом создается своего рода «слепок» системы, со всеми программами, файлами, драйверами и настройками, которые имеются на момент создания образа (а в предыдущем варианте мы получаем чистую систему с сохранением разве что личных данных и файлов). Оптимальное время создание такого образа - сразу после чистой установки ОС и всех драйверов на компьютере, т. е. после того, как Windows 10 была доведена до полностью работоспособного состояния, но еще не захламлена. Чтобы создать такой образ, зайдите в Панель управления - История файлов, а затем внизу слева выберите «Резервная копия образа системы» - «Создание образа системы». Еще один путь - зайти в «Все параметры» - «Обновление и безопасность» - «Служба архивации» - «Перейти в раздел «Архивация и восстановление (Windows 7)» - «Создание образа системы».



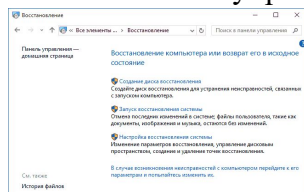
На следующих этапах вы можете выбрать, куда будет сохранен образ системы, а также какие именно разделы на дисках нужно добавить в резервную копию (как правило, это раздел, зарезервированный системой и системный раздел диска).



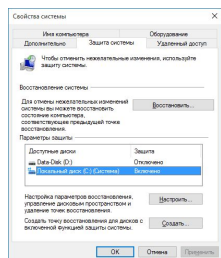
В дальнейшем, вы можете использовать созданный образ для быстрого возврата системы в то состояние, которое вам требуется. Запустить восстановление из образа можно с диска восстановления или выбрав «Восстановление» в программе установки Windows 10 (Диагностика - Дополнительные параметры - Восстановление образа системы).

- Точки восстановления

Точки восстановления в Windows 10 работают так же, как и в двух предыдущих версиях операционной системы и часто могут помочь откатить последние изменения на компьютере, вызывавшие проблемы. Подробная инструкция по всем возможностям инструмента: Точки восстановления Windows 10. Для того, чтобы проверить, включено ли автоматическое создание точек восстановления, вы можете зайти в «Панель управления» - «Восстановление» и нажать «Настройка восстановления системы».



По умолчанию, защита для системного диска включена, также вы можете настроить создание точек восстановления для диска, выбрав его и нажав кнопку «Настроить».



Точки восстановления системы создаются автоматически при изменении каких-либо системных параметров и настроек, установке программ и служб, также имеется возможность их создания вручную перед каким-либо потенциально опасным действием (кнопка «Создать» в окне настроек защиты системы). Когда потребуется применить точку восстановления, вы можете зайти в соответствующий раздел панели управления и выбрать пункт «Запуск восстановления системы» или же, если Windows не запускается, загрузиться с диска восстановления (или установочного накопителя) и найти запуск восстановления в Диагностика - Дополнительные параметры.

- История файлов

Ещё одна функция восстановления Windows 10 - история файлов, позволяющая хранить резервные копии важных файлов и документов, а также их предыдущие версии и возвращаться к ним при необходимости.

Восстановление загрузчика Windows 10 через командную строку

Способ 1: Автоматическое восстановление

Первый метод никак не связан с «Командной строкой», поскольку восстановление загрузчика происходит в автоматическом режиме, но запускается оно через тот же установочный образ Windows 10, о создании которого мы поговорим немного позже. Поэтому мы решили все же рассказать о данном варианте более детально, чтобы упростить вам процесс восстановления, если вы не знали о наличии такой функции. В том случае, когда она уже была опробована и не принесла никаких результатов, можете сразу переходить ко второму способу. Для реализации этого и всех последующих способов придется записать установщик Windows 10 на флешку, чтобы после загрузить оснастку восстановления системы.



Создание загрузочного диска (флешки) с Windows 10

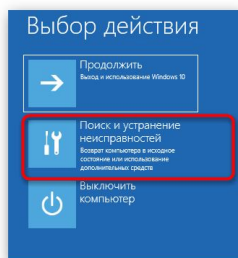
Вставьте полученную загрузочный диск в дисковод либо флешку в свободный USB-разъем и запустите компьютер. При появлении формы выбора языка осуществите это действие, а затем нажмите на кнопку «Далее».



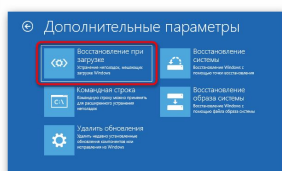
Перейдите в раздел «Восстановление системы», щелкнув по соответствующей кнопке слева внизу.



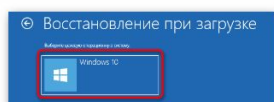
Здесь вас интересует пункт «Поиск и устранение неисправностей».



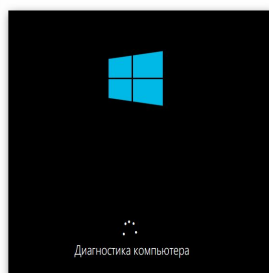
Щелкните левой кнопкой мыши по панели «Восстановление при загрузке».



Укажите целевую операционную систему, чтобы запустить этот процесс.



Дождитесь окончания диагностики компьютера. Если какие-то ошибки будут найдены и исправлены, вы получите соответствующее уведомление.

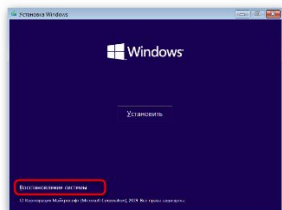


Этот метод далеко не всегда оказывается действенным, однако он самый простой в реализации, поэтому мы и решили ознакомить вас с ним перед тем, как перейти к использованию «Командной строки» для

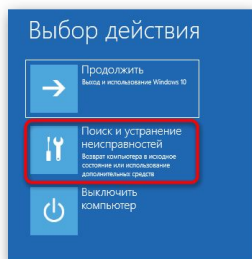
восстановления Виндовс 10. Если произведенные действия не принесли никакого результата, стоит ознакомиться с одним из следующих вариантов.

Способ 2: Ручное восстановление загрузчика

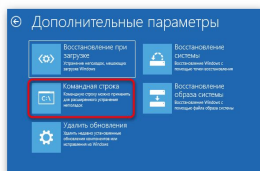
Принцип ручного восстановления заключается в перезаписи основных файлов, отвечающих за работоспособность загрузчика. Для этого придется указать диск с операционной системой при запуске специализированной утилиты. При запуске образа с ОС снова перейдите в раздел «Восстановление системы».



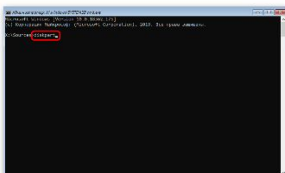
Здесь требуется нажать на ту же плитку «Поиск и устранение неисправностей».



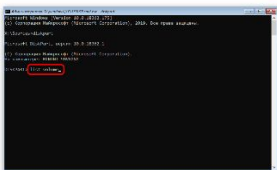
Запустите отсюда «Командную строку», кликнув по соответствующей надписи.



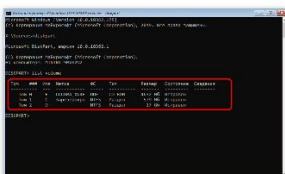
Сейчас потребуется запустить оснастку diskpart, чтобы узнать информацию о дисках. Введите название этой утилиты в строке и нажмите на Enter.



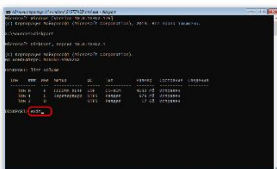
Отобразим список подключенных разделов через list volume.



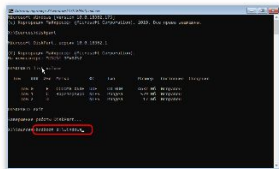
Теперь изучите отобразившиеся сведения. Здесь понадобится запомнить имя того логического тома, который выступает основным для хранения файлов Windows 10.



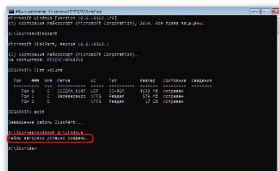
На этом взаимодействие с diskpart закончено, следует покинуть ее, введя exit.



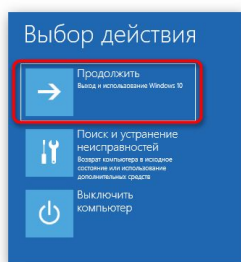
Используйте команду `bcdboot c:\windows`, заменив букву диска на определенную ранее, чтобы восстановить загрузчик.



Вы будете уведомлены о создании специальных файлов, отвечающих за дальнейшую загрузку ОС. Это значит, что все шаги были выполнены правильно.



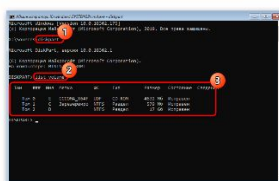
Закройте консоль и продолжите стандартную загрузку ОС.



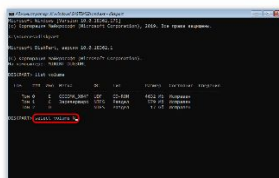
Этот метод не такой сложный в исполнении, но является довольно эффективным: в большинстве ситуаций после его выполнения загрузчик ОС успешно восстанавливается и со входом в Виндовс не возникает никаких проблем. Если решить проблему так и не удалось, советуем обратить внимание на альтернативный вариант.

Способ 3: Ручное альтернативное восстановление

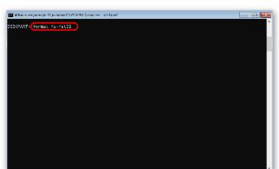
Второй вариант заключается в использовании скрытого тома, который резервируется системой и задействуется в качестве загрузчика. Сейчас следует обнаружить его и произвести определенные манипуляции. Эта операция так же осуществляется в режиме восстановления, поэтому предварительно загрузитесь с созданного ранее носителя. После этого снова запустите оснастку `diskpart`, выведите список разделов через `list volume` и отыщите скрытый том, размер которого варьируется от 99 до 600 мегабайт, в зависимости от используемого загрузчика. В случае с MBR (BIOS) он будет иметь файловую систему NTFS, а при GPT (UEFI) - FAT32.



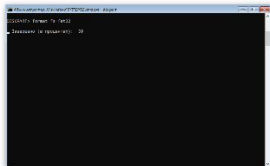
Не покидайте текущую оснастку и здесь же введите `select volume N`, где вместо N укажите номер определенного ранее раздела.



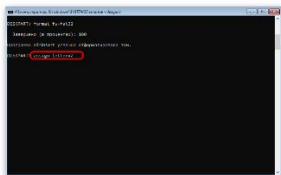
Форматируйте его в FAT32 или в NTFS, отталкиваясь от того, какая файловая система установлена на текущий момент.



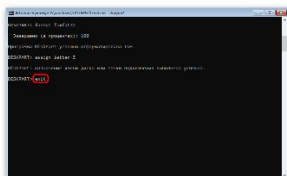
Ожидайте окончания этой процедуры - она может занять несколько минут.



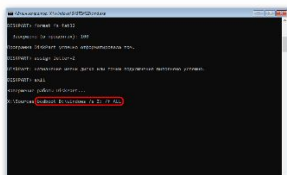
Присвойте ему букву Z, написав assign letter=Z.



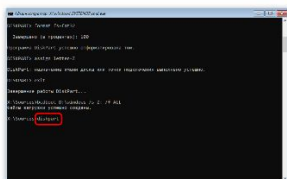
Закройте текущую оснастку через exit.



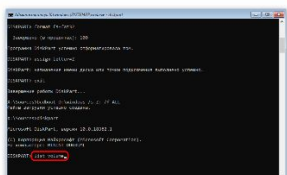
Для восстановления файлов загрузчика используйте команду bcdboot D:\Windows /s Z: /f ALL, заменив D на букву раздела с ОС.



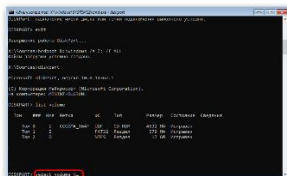
Снова введите diskpart.



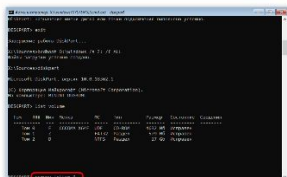
Откройте список дисков через list volume.



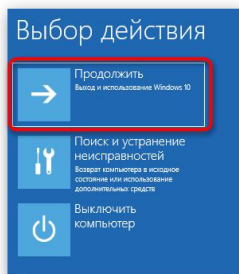
Выберите диск select volume N (тот, которому была присвоена буква Z).



Удалите букву через remove letter=Z, чтобы в будущем она не мешала взаимодействию с системой.



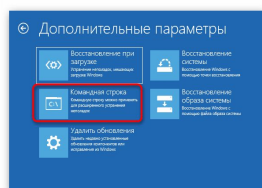
Перезагрузите ПК, чтобы проверить результативность.



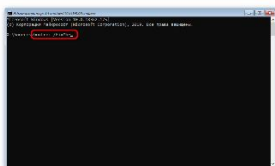
Вы можете быть уверены в том, что один из приведенных выше методов точно поможет восстановить загрузчик Windows 10, если, конечно, проблема с загрузкой не была вызвана другими проблемами.

Способ 4: Восстановление после удаления Linux

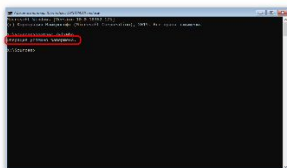
Как известно, в таких ситуациях в роли основного загрузчика выступает GRUB, но потом он удаляется, что приводит к поломкам компонентов Windows. Решается такая задача не так, как это было описано в способах выше, а намного проще. В режиме восстановления, переход в который был осуществлен через установочный образ операционной системы, запустите «Командную строку».



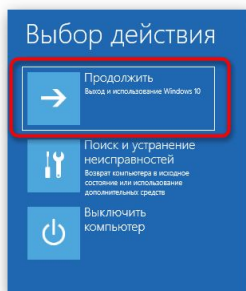
Здесь введите `bootrec /FixMbr` и нажмите на Enter.



Вы будете уведомлены о том, что операция выполнена успешно.



Закройте консоль и перезагрузите компьютер, чтобы начать корректное взаимодействие с системой.



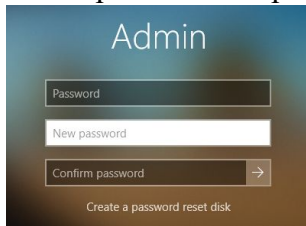
Это были все способы восстановления загрузчика Windows 10 через «Командную строку». Как видите, каждый пользователь сможет найти оптимальный для себя вариант или воспользоваться альтернативой, если выбранный метод не принес никакого результата.

Примеры решения задач

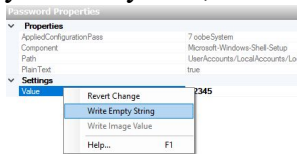
Простой файл ответов для быстрой автоустановки Windows 10

- Файл предназначен для 64-разрядной Windows 10. Сохраните его с именем `AutoUnattend.xml` в корень установочной флэшки.
- Язык. Для русской версии замените `en-US` на `ru-RU` и наоборот везде кроме `SystemLocale` (если ничего не менять, надо будет вручную указать язык интерфейса, форматов и метода ввода).

- Издание, ключ продукта и активация. Издание выбирается на основе ключа продукта в файле ответов. Если ключ не указан, выводится список всех изданий в WIM-образе за исключением случая, когда ноутбук шел с Windows 8 и новее. В этом варианте эквивалентное издание Windows 10 автоматически выбирается на основе данных UEFI BIOS, а после установки система активируется (верно на момент публикации статьи). В файле ответов ключ пустой, поэтому окно его ввода пропускается. Вы можете прописать ключ KMS для установки (но не активации) или указать свой. В любом случае ключ можно сменить в параметрах после установки.
- Учетная запись. Экран создания учетной записи Microsoft (MSA) пропускается. Это позволяет избежать генерации папки профиля по первым 5-6 буквам MSA. Локальной учетной записи не задаются три контрольных вопроса. Имя учетной записи Admin можно поменять на любое. Если вы не хотите задавать имя заранее, закомментируйте раздел UserAccounts полностью. Заметьте, что в файле также задано HideOnlineAccountScreens = true, поэтому экран создания MSA все равно не появится.
- Пароль и автоматический вход в аккаунт. Пароль закомментирован, поэтому программа установки предложит сменить его – задайте любой или продолжайте с пустым, нажав стрелку рядом с подтверждением пароля.



- Можно пропустить этот экран, настроив автоматический вход (в примере – однократный). Раскомментируйте разделы Password и Autologon, задайте пароль в открытом виде и не забудьте поменять имя учетной записи для автологона. В Windows System Image Manager можно замаскировать пароль или указать пустой, как показано на картинке.



- Управление дисками
 - UEFI. Перед установкой задайте в UEFI BIOS нативный режим загрузки UEFI (если ноутбук шел с Windows, можно просто сбросить все на дефолт).
 - GPT. В окне управления дисками нажмите Shift + F10, в командной строке очистите диск и конвертируйте в GPT, потом обновите информацию в окне:

```
diskpart
sel dis 0
clean
convert gpt
exit
```

Разобраться в интерфейсе Windows SIM и структуре данных в образе Windows не так-то просто. Но есть решение: можно использовать готовый файл autounattend.xml, созданный для русской 64-разрядной версии Windows 10 (для английской версии нужно в параметре <UILanguage> указать значение en_US).

Базовый файл ответов AutoUnattend.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-International-Core-WinPE" processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIconfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
      instance">
      <InputLocale>en-US; ru-RU</InputLocale>
      <SystemLocale>ru-RU</SystemLocale>
      <UILanguage> ru-RU</UILanguage>
```

```

    <UserLocale> ru-RU</UserLocale>
  </component>
  <component name="Microsoft-Windows-Setup" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMICConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <UserData>
      <ProductKey>
        <Key></Key>
      </ProductKey>
      <AcceptEula>true</AcceptEula>
    </UserData>
  </component>
</settings>
<settings pass="oobeSystem">
  <component name="Microsoft-Windows-International-Core" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMICConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <InputLocale>en-US; ru-RU</InputLocale>
    <SystemLocale>ru-RU</SystemLocale>
    <UILanguage>ru-RU</UILanguage>
    <UserLocale>en-US</UserLocale>
  </component>
  <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMICConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <OOBE>
      <HideOnlineAccountScreens>true</HideOnlineAccountScreens>
      <ProtectYourPC>3</ProtectYourPC>
    </OOBE>
    <UserAccounts>
      <LocalAccounts>
        <LocalAccount wcm:action="add">
          <Group>Administrators</Group>
          <Name>Administrator</Name>
          <!-- <Password>
            <Value>password</Value>
            <PlainText>true</PlainText>
          </Password> -->
        </LocalAccount>
      </LocalAccounts>
    </UserAccounts>
    <!-- <AutoLogon>
      <Password>
        <Value>password</Value>
        <PlainText>true</PlainText>
      </Password>
      <Username> Administrator</Username>
      <LogonCount>1</LogonCount>
      <Enabled>true</Enabled>
    </AutoLogon> -->
  </component>
</settings>
</unattend>

```

В этом файле поле для ввода ключа продукта оставлено пустым: если используемый вами образ Windows включает несколько редакций, программа установки предложит выбрать нужную. На компьютерах с UEFI ключ продукта установщик автоматически подхватит из энергонезависимой памяти (при его наличии). Также можно вбить ключ прямо в поле <Key></Key>, тогда система активируется автоматически с этим ключом, как только машина соединится с интернетом. При автоматической установке Windows создается локальная учетная запись с именем Administrator. Строки, обеспечивающие создание пароля для этой учетной записи, закомментированы, поэтому инсталлятор попросит его ввести (это поле можно будет оставить пустым). Вы можете убрать теги комментария и задать пароль прямо в файле, а заодно можно

раскомментировать секцию, отвечающую за автоматический вход в систему с локальной учетной записью. Этап создания учетной записи Microsoft пропускается - за это отвечает тег:

```
<HideOnlineAccountScreens>true</HideOnlineAccountScreens>
```

С использованием этого файла ответов вам придется указать дисковый раздел, в который планируется установить систему, и ответить на парочку дополнительных вопросов программы установки. Весь остальной процесс автоматической установки Windows пройдет без вашего участия.

Форматирование существующего 1-го раздела на диске в AutoUnattend.xml

Использование этих фрагментов приведет к потере данных на существующих разделах физического диска. Первый раздел первого физического диска будет отформатирован в файловую систему NTFS, ему будет назначена буква C, а тому присвоена метка OS1. Остальные разделы затронуты не будут:

```
<DiskConfiguration>
  <WillShowUI>OnError</WillShowUI>
  <Disk wcm:action="add">
    <ModifyPartitions>
      <ModifyPartition wcm:action="add">
        <Extend>>false</Extend>
        <Format>NTFS</Format>
        <Label>OS1</Label>
        <Letter>C</Letter>
        <Order>1</Order>
        <PartitionID>1</PartitionID>
        <Active>true</Active>
      </ModifyPartition>
    </ModifyPartitions>
    <DiskID>0</DiskID>
    <WillWipeDisk>>false</WillWipeDisk>
  </Disk>
</DiskConfiguration>
```

Удаление всех разделов и создание нового основного раздела в AutoUnattend.xml

С первого физического диска будут удалены все разделы. На их месте будет создан один основной раздел, который займет все пространство на диске. Раздел будет отформатирован в файловую систему NTFS, ему будет назначена буква C, а тому присвоена метка OS1:

```
<DiskConfiguration>
  <Disk wcm:action="add">
    <CreatePartitions>
      <CreatePartition wcm:action="add">
        <Extend>true</Extend>
        <Order>1</Order>
        <Type>Primary</Type>
      </CreatePartition>
    </CreatePartitions>
    <ModifyPartitions>
      <ModifyPartition wcm:action="add">
        <Active>true</Active>
        <Extend>>false</Extend>
        <Format>NTFS</Format>
        <Label>OS1</Label>
        <Letter>C</Letter>
        <Order>1</Order>
        <PartitionID>1</PartitionID>
      </ModifyPartition>
    </ModifyPartitions>
    <DiskID>0</DiskID>
    <WillWipeDisk>true</WillWipeDisk>
  </Disk>
</DiskConfiguration>
```

Удаление всех разделов и создание двух новых разделов в AutoUnattend.xml

С первого физического диска будут удалены все разделы. На их месте будет создан один основной раздел размером 50 Гб (Vista) и один дополнительный раздел, который займет все оставшееся пространство на диске. На дополнительном разделе будет создан логический диск (Data):

```
<DiskConfiguration>
  <Disk wcm:action="add">
    <CreatePartitions>
      <CreatePartition wcm:action="add">
        <Order>1</Order>
        <Extend>>false</Extend>
        <Type>Primary</Type>
        <Size>50000</Size>
      </CreatePartition>
      <CreatePartition wcm:action="add">
        <Extend>>true</Extend>
        <Order>3</Order>
        <Type>Logical</Type>
      </CreatePartition>
      <CreatePartition wcm:action="add">
        <Extend>>true</Extend>
        <Order>2</Order>
        <Type>Extended</Type>
      </CreatePartition>
    </CreatePartitions>
    <ModifyPartitions>
      <ModifyPartition wcm:action="add">
        <Active>>true</Active>
        <Extend>>false</Extend>
        <Format>NTFS</Format>
        <Label>Vista</Label>
        <Letter>C</Letter>
        <Order>1</Order>
        <PartitionID>1</PartitionID>
      </ModifyPartition>
      <ModifyPartition wcm:action="add">
        <Active>>false</Active>
        <Extend>>false</Extend>
        <Format>NTFS</Format>
        <Label>DATA</Label>
        <Letter>D</Letter>
        <Order>2</Order>
        <PartitionID>2</PartitionID>
      </ModifyPartition>
    </ModifyPartitions>
    <DiskID>0</DiskID>
    <WillWipeDisk>true</WillWipeDisk>
  </Disk>
</DiskConfiguration>
```

Удаление всех разделов и создание основного раздела и дополнительного раздела с двумя логическими дисками в AutoUnattend.xml

С первого физического диска будут удалены все разделы. На их месте будет создан один основной раздел размером 50 Гб (OS1) и один дополнительный раздел с двумя логическими дисками. Первый логический диск (Data) будет иметь размер 100 Гб, а второй (Files) - займет все оставшееся пространство:

```
<DiskConfiguration>
  <Disk wcm:action="add">
    <CreatePartitions>
      <CreatePartition wcm:action="add">
        <Order>1</Order>
        <Size>50000</Size>
        <Type>Primary</Type>
      </CreatePartition>
      <CreatePartition wcm:action="add">
        <Extend>>true</Extend>
        <Order>2</Order>
```

```

    <Type>Extended</Type>
  </CreatePartition>
  <CreatePartition wcm:action="add">
    <Extend>false</Extend>
    <Order>3</Order>
    <Type>Logical</Type>
    <Size>100000</Size>
  </CreatePartition>
  <CreatePartition wcm:action="add">
    <Size>1000</Size>
    <Order>4</Order>
    <Type>Logical</Type>
  </CreatePartition>
</CreatePartitions>
<ModifyPartitions>
  <ModifyPartition wcm:action="add">
    <Active>true</Active>
    <Extend>false</Extend>
    <Format>NTFS</Format>
    <Label>OS1</Label>
    <Letter>C</Letter>
    <Order>1</Order>
    <PartitionID>1</PartitionID>
  </ModifyPartition>
  <ModifyPartition wcm:action="add">
    <Active>false</Active>
    <Extend>false</Extend>
    <Format>NTFS</Format>
    <Label>Data</Label>
    <Letter>D</Letter>
    <Order>2</Order>
    <PartitionID>2</PartitionID>
  </ModifyPartition>
  <ModifyPartition wcm:action="add">
    <PartitionID>3</PartitionID>
    <Order>3</Order>
    <Label>Files</Label>
    <Format>NTFS</Format>
    <Active>false</Active>
    <Extend>true</Extend>
    <Letter>E</Letter>
  </ModifyPartition>
</ModifyPartitions>
<DiskID>0</DiskID>
<WillWipeDisk>true</WillWipeDisk>
</Disk>
</DiskConfiguration>

```

В примере для создания второго логического диска на дополнительном разделе используется обходной путь. Вместо того, чтобы расширять его на все оставшееся свободное пространство при создании (CreatePartition), ему на этом этапе задается фиксированный размер (1 ГБ). А расширение раздела производится уже на этапе изменения раздела (ModifyPartition). Это связано с тем, что, похоже, при создании логических дисков на дополнительном разделе параметр Extend срабатывает только для первого диска.

Создание учетных записей с помощью файла ответов и настройка автоматического входа в систему в AutoUnattend.xml

```

<settings pass="oobeSystem">
  <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="x86"
    publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
    xmlns:wcm="http://schemas.microsoft.com/WMICConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <AutoLogon>
      <Enabled>true</Enabled>
      <LogonCount>5</LogonCount>
    </AutoLogon>
  </component>
</settings>

```

```

    <Username>Admin</Username>
  </AutoLogon>
  <TimeZone></TimeZone>
  <RegisteredOrganization></RegisteredOrganization>
  <RegisteredOwner></RegisteredOwner>
  <UserAccounts>
    <LocalAccounts>
      <LocalAccount wcm:action="add">
        <Description>Админ компьютера</Description>
        <DisplayName></DisplayName>
        <Group>Administrators</Group>
        <Name>Admin</Name>
      </LocalAccount>
      <LocalAccount wcm:action="add">
        <Description>Обычный пользователь</Description>
        <Group>Users</Group>
        <Name>User</Name>
      </LocalAccount>
    </LocalAccounts>
  </UserAccounts>
</component>
</settings>

```

Установка драйверов с USB- или DVD-диска в AutoUnattend.xml

```

<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="offlineServicing">
    <component name="Microsoft-Windows-PnpCustomizationsNonWinPE" processorArchitecture="x86"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMICConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
      <DriverPaths>
        <PathAndCredentials wcm:action="add" wcm:keyValue="1">
          <Path>C:\VistaWork\Drivers</Path>
        </PathAndCredentials>
      </DriverPaths>
    </component>
  </settings>
  <cpu:offlineImage cpu:source="catalog:c:\vistawork\dvd\sources\install_windows_vista_ultimate.clg"
xmlns:cpu="urn:schemas-microsoft-com:cpu" />
</unattend>
FirstLogonCommands
Статья FirstLogonCommands
<FirstLogonCommands>
  <SynchronousCommand wcm:action="add">
    <Order>15</Order>
    <CommandLine>cmd /c %DiskRoot%\Install\firefox_setup.exe -ms</CommandLine>
    <Description></Description>
  </SynchronousCommand>
  <SynchronousCommand wcm:action="add">
    <Order>25</Order>
    <CommandLine>cmd /c %DiskRoot%\Install\install.cmd</CommandLine>
  </SynchronousCommand>
</FirstLogonCommands>
Unattend.xml (для Windows PE)
Запуск синхронной команды в Windows PE. Статья Создание базового загрузочного диска Windows PE.
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup" processorArchitecture="x86"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMICConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
      <RunSynchronous>
        <RunSynchronousCommand wcm:action="add">
          <Order>1</Order>

```

```

    <Path>%SystemDrive%\TotalCmdExt\TOTALCMD.EXE "I=%SystemDrive%\
TotalCmdExt\Profiles\Prof_V\main.ini"</Path>
    </RunSynchronousCommand>
    </RunSynchronous>
  </component>
</settings>
<cpu:offlineImage cpu:source="catalog:c:/vistawork/dvd/sources/install_windows_vista_ultimate.clg"
xmlns:cpu="urn:schemas-microsoft-com:cpu" />
</unattend>

```

Создание профиля настроенного образа ОС в AutoUnattend.xml

```

<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="specialize">
    <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="x86"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMIconfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
      <CopyProfile>true</CopyProfile>
    </component>
  </settings>
  <cpu:offlineImage cpu:source="catalog:c:/vistawork/dvd/sources/install_windows_vista_ultimate.clg"
xmlns:cpu="urn:schemas-microsoft-com:cpu" />
</unattend>

```

Задания для самостоятельной работы

Для текущего автоматизированного рабочего места:

1. Написать bat-скрипт получения информации из образа Windows
2. Написать bat-скрипт добавления пакетов обновления в образ Windows.
3. Написать bat-скрипт отключения компонентов Windows за исключением Internet Explorer, Media Player, Windows Search и .NET Framework 4.6.1.
4. Написать bat-скрипт удаления универсальных приложений «Магазина» Windows (MicrosoftStore).
5. Сконфигурировать базовый файл ответов (AutoUnattend.xml) для быстрой автоустановки Windows: локализация - английский язык; ключа продукта – произвольный (но с учётом количества символов); имя учетной записи – имя студента; пароль – произвольный (не менее 6 символов); управление дисками - UEFI/GPT (в зависимости от текущего автоматизированного рабочего места); форматирование существующего 1-го раздела на диске; установка драйверов с USB.

Контрольные вопросы

1. Что такое «Образ операционной системы»?
2. Способы применения образа операционной системы.
3. Зачем изучать использование образов ОС?
4. Что такое «ISO-образ»?
5. Работа с ISO-образами.
6. Перечислите программы, которые на основе файла-образа создают в системе виртуальный привод.
7. Перечислите редакторы ISO-образов.
8. Создание образа системы и Диска (флеш-накопителя) восстановления Windows.
9. Создание Флеш-накопителя восстановления системы.
10. Перечислите программы для создания образа системы, восстановления системы, загрузки официальных ISO образов ОС, создания установочных и загрузочных носителей.
11. Точки восстановления.
12. Техническая реализация точек восстановления.
13. Диск восстановления.
14. Как работает восстановление системы?

15. Можно ли отменить изменения?
16. Как выбрать точку восстановления?
17. В течение какого времени сохраняются точки восстановления?
18. Перечислите программы для восстановления системы.