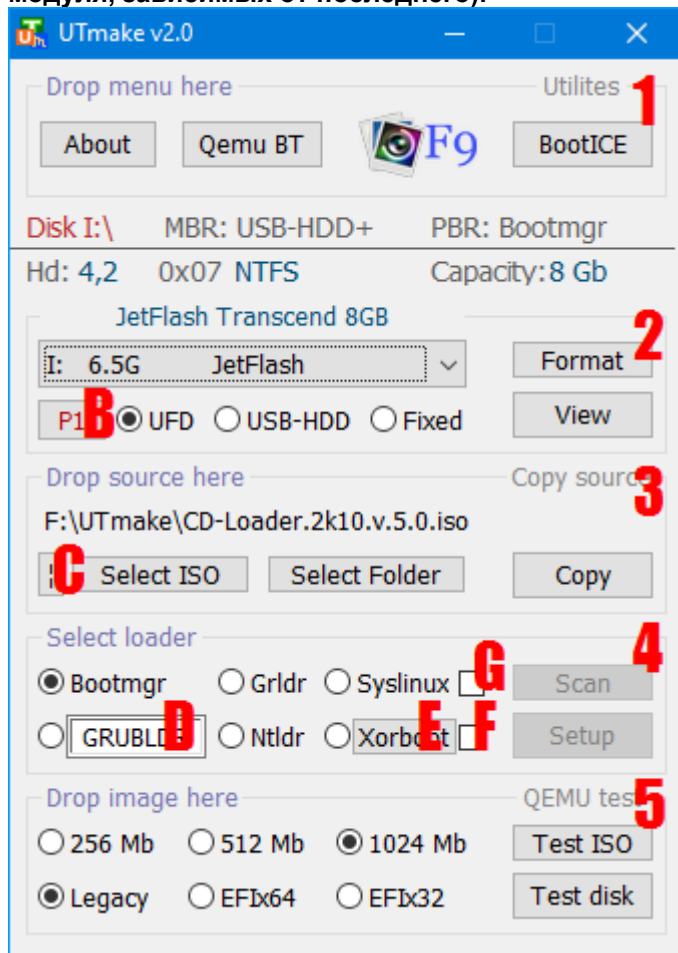


UTmake (утилита для подготовки накопителей в качестве WinPE-загрузочных)

Мое видение того, как должна выглядеть утилита для разворачивания PE-сборки на флешку (USB-HDD, HDD).
Окно утилиты разбито на 5 независимых модулей (точнее, модуль утилит, модуль выбора накопителя и 3 модуля, зависящих от последнего).



Интегрирован скрипт **Ufd2Hdd**. Можно "преобразовать" для конкретной ОС(PE) мультираздельный *removable*-накопитель (флешку или карту памяти в ридере) в **USB-HDD** (кнопка **UFD>HDD**). Ну и обратная процедура имеется, соответственно (**UFD<HDD**). В систему, при необходимости, устанавливается драйвер **diskmod**. Кнопка активна только для мультираздельных *removable*-накопителей.

1. Модуль внешних утилит. Включает QEMU BootTester, BootICE и утилиту для снятия скриншотов WinSnap (активируется по F9). Также работает как аналог EasyBootICE - на него можно "бросить" (*перетащить мышкой из окна проводника*) меню bootmgr (BCD) или grldr (*.lst/grldr) для редактирования с помощью BootICE.

2. Модуль выбора диска. Выпадающее меню выбора диска, список определяется типом накопителя (UFD - флешки, USB-HDD - USB-харды и флешки с отключенным битом "Removable", Fixed - обычные HDD). "P2"("P1") - переключение на другой раздел флешки (если она двухраздельная). Кнопка форматирования служит для вызова модуля форматирования. Для обзора содержимого накопителя служит "View", а над модулем выбора диска доступна детальная информация о разделе - его номер, ID / файловая система, активность раздела и номинальная ёмкость всего накопителя.

3. Модуль копирования. Можно выбрать образ или папку (*либо перетащить их на модуль мышкой*). При нажатии на "Copy" образ будет предварительно смонтирован как виртуальный CDRom. Для копирования используется утилита KillCopy. Настройки копирования: не перезаписывать более новые файлы и содержимое пользовательских папок с именем User.* (User.Image, User.WinPE, User.Config, ...). С учетом настроек копирования модуль можно использовать для обновления более старой версии сборки до актуальной.

4. Модуль подключения загрузчика. Кнопка "Scan" сканирует носитель на наличие загрузчиков (вначале в корне диска, потом в корневых папках, потом в подкаталогах корневых папок - глубина сканирования 2 уровня от корня). Кнопка "Setup" устанавливает выбранный загрузчик.

5. Модуль тестирования (*объект также можно перетащить на модуль мышкой*). Можно выбрать объем выделяемой памяти, среду эмуляции (Bios/EFI). Кнопка "Test ISO" позволяет сразу запустить эмуляцию загрузки образа (если он выбран в п.3), а "Test disk" - эмуляцию загрузки выбранного в п.2 диска.

В - Кнопка смены видимого ОС раздела флешки. Появится только для двухраздельных Removable-накопителей (к которым ОС относит флешки). По нажатии 1-й (видимый) и 2-й (невидимый ОС) разделы обменяются местами, и в ОС мы получим доступ к скрытому разделу. Для "возврата" к первому разделу нужно нажать ещё раз. Под Windows 10 с поддержкой многораздельных флешек (крайние билды) может менять цвет на красный, если этот раздел будет невидим в младших ОС Windows XP...8.1.

С - Кнопка для вычисления хешей файлов исходника и сверки их с получателем. Можно применять для генерации

текстового файла с хешами (выбор одного из алгоритмов CRC32, MD5, SHA1) исходника. Результат будет сохранен в файл %Temp%\FileHash.txt, при выходе из UTmake не удаляется.

D - Поле ввода для нестандартного имени загрузчика (например, BootXXX, Grubmgr, Grubldr). Ограничение - 7 английских букв/цифр (при вводе не проверяется).

E - Кнопка выбора файла произвольного MBR-загрузчика (Xorboot, Grub4Dos, PLOP, Grub2, Wee). Ограничение - размер загрузчика должен быть не более 63 сектора (32256 байт). Этот загрузчик будет установлен при выборе радиокнопки слева, по нажатию "Setup".

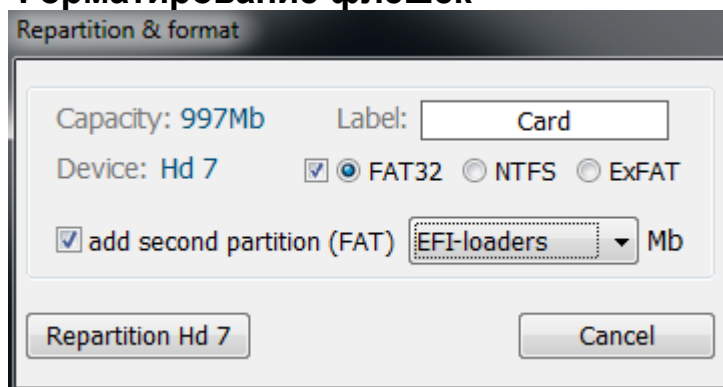
F - Чекбокс для установки Xorboot в качестве первичного загрузчика для "прозрачной" загрузки Bootmgr/Grldr/Ntldr/Grubldr (из произвольной папки).

G - Чекбокс для установки Syslinux в качестве первичного загрузчика для "прозрачной" загрузки Bootmgr/Grldr/Ntldr/Grubldr (из произвольной папки).

Если активирован "Контроль учётных записей пользователей" (UAC), утилита при старте запросит права администратора.

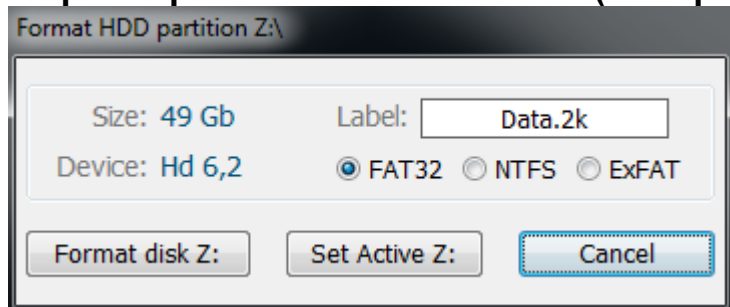
Комплект на базе **Grub2 (Flash2Ptn.exe)** предназначен для второго (невидимого) раздела флешки. Применяется для запуска **EFI-загрузчика Windows** из первого раздела. Поддерживаются **EFI x32/x64** (включая режим **Secure Boot**), файловые системы **NTFS/ExFAT**.

Форматирование флешек



При нажатии на "Format" откроется окно форматирования с возможностью выбора файловой системы и метки диска. Также есть чекбокс для создания 2-го раздела с возможностью выбора его размера. Размер 2-го раздела определяется из перечня 16/64/256/1024/2048/4096/8192 (Мб), либо указанием вручную (Manually select), либо по размеру источника (если предварительно выбран каталог или образ со сборкой, тогда размер рассчитывается на 10% больше), либо (при наличии файла Utilites\Flash2Ptn.exe) - размером файлов в этом архиве (EFI-loaders). Flash2Ptn.exe - это обычный 7z SFX-архив, для примера добавлен вариант под сборку 2k10. Файловая система первого раздела будет выбрана автоматически (для флешек меньше 2Гб - FAT, и т.д.). Если не устраивает - нажимаем чекбокс перед "FAT", он разблокирует все варианты (если они и так доступны, чекбокс недоступен). Файловая система второго раздела - автоматически FAT (если размер раздела до 2Гб) или FAT32 (если раздел от 2Гб до 512Гб). Кнопка "Repartition" переразбивает флешку (создает MBR, первый раздел, и, если отмечен 2-й - то и второй, если выбран вариант EFI-loaders - тогда сразу производится распаковка этого архива на 2-й раздел). После "Repartition" в MBR/PBR автоматически устанавливается загрузчиком bootmgr, а первый раздел становится активным (т.е., для сборок с bootmgr в качестве главного загрузчика дополнительно устанавливать загрузчик не нужно). Кнопка "Format only" предназначена только для форматирования текущего раздела флешки, можно использовать для быстрой очистки видимого раздела или смены файловой системы (если имеется второй раздел, он не затрагивается). Если раздел неактивен (например, флешка ранее не была загрузочной), кнопка "Format only" недоступна, нужно сделать Repartition.

Форматирование USB-HDD/HDD (или флешек с отключенным Removable Bit)



В окне форматирования отображается информация о размере раздела (Size), физической нумерации раздела (Device). Можно сменить метку и выбрать файловую систему для раздела. Кнопка "Format диск: \" - предназначена для очистки раздела или смены файловой системы путем быстрого форматирования, без изменения MBR. После форматирования раздел активируется, и обновляется метка диска. Также возможна отдельная активация диска ("Set Active") без форматирования (кнопка видима только для неактивных разделов). После выполнения любой из операций произойдет возврат в главное окно программы.

Ограничения по форматированию

Если размер флешки более 32Гб, форматирование в FAT32 также доступно (ограничение ОС Windows, которое можно обойти с помощью сторонних утилит).

Минимальный размер раздела - 1Мб; если ОС не поддерживает ExFAT (ОС ниже 7-ки без установленного обновления поддержки ExFAT), то эта файловая система будет недоступна для выбора; FAT32 поддерживается для разделов до 512Гб включительно. Если в метке диска имеются неподдерживаемые символы, метка будет укорочена до приемлемого имени (ограничение FAT/FAT32 - не поддерживаются символы "точка", "запятая", ...). Второму разделу автоматически присваивается метка Part_II. Если размер раздела не более 2Гб, при переразбивке такой раздел форматируется в FAT независимо от выбора пользователя. При обычном форматировании - согласно выбору пользователя. Поддержка форматирования в NTFS для ОС Windows XP/2003 в сильно урезанных WinPE не гарантируется. При ошибке форматирования раздел форматируется в FAT32.

Подключение загрузчика

По загрузчикам: кроме стандартных загрузчиков (Bootmgr/GRLDR/NTLDR) добавлена возможность установки Syslinux (4.07) и Xorboot (0.72). Поиск загрузчиков производится так: вначале ищутся файлы в корне, если не найдены - в корневых каталогах, если и там не найдены - то в подпапках корневых каталогов. Т.е., загрузчик может находиться в корне или в любом каталоге до второго уровня. Розыск ведется по именам: Xorboot.bin (Xorboot), syslinux.exe/syslinux.cfg/isolinux.cfg (Syslinux), Ntldr (Ntldr), Grldr (Grldr), Bootmgr (Bootmgr), и имени в окне редактирования - до 7-ми символов (Grldr). Кнопка "Scan" сканирует носитель на наличие загрузчиков и отсеивает не найденное. Но её использование не обязательно (если уверены, что нужный загрузчик имеется).

Если файловая система носителя ExFAT, автоматически блокируется установка загрузчиков Xorboot/Syslinux и Ntldr (они не работают с этой ф/с).

Для подключения Syslinux достаточно, чтобы было найдено меню суслика (файл syslinux.cfg или isolinux.cfg). При этом сам загрузчик (файл ldlinux.sys) будет создан рядом с cfg-файлом (isolinux.cfg будет переименован на syslinux.cfg).

Простейшее меню syslinux.cfg, позволяющее прозрачно загрузить GRLDR из папки 2k10/GRUB4DOS

```
PROMPT 0
TIMEOUT 1
DEFAULT BOOT
LABEL BOOT
BOOT /2K10/GRUB4DOS/GRLDR
```

Теперь сборщики могут просто включить подобное меню в свою сборку (закинуть в любую удобную папку в пределах 2 уровней от корня), и при использовании UTmake пользователь получит возможность сделать флешку "повышенной загрузаемости" с помощью костыля на базе суслика.

Загрузчик Xorboot можно установить, если (в порядке нарастания приоритета):

- а) в пределах 2 уровней от корня сборки будет найден файл загрузчика xorboot.bin
- б) файл с таким именем имеется в папке Utilites (для примера в комплект включен файл от сборки 2k10)
- в) файл был выбран по нажатию кнопки "Xorboot" - "4"/"Е".

Самым приоритетным является именно последний файл, причем MBR-загрузчик может быть не только Xorboot, но и MBR-варианты Grub4Dos, PLOP, Grub2, Wee. Размер загрузчика должен быть не более 63 секторов (32256 байт).

Установка Xorboot и Syslinux возможна как в режиме основного загрузчика (радиокнопка слева от кнопки Xorboot / Syslinux), так и в режиме "костыля" (чекбокс справа) для прозрачной загрузки основного загрузчика Bootmgr/GRLDR/NTLDR/Grubldr (указанное в "D" имя). Костыль реализован в самой UTmake, при этом основной загрузчик может находиться в пределах 2 уровней от корня (общая длина пути не более 31-го символа, без пробелов и кириллицы).

Два раздела на флешке - зачем это нужно и реализация этого варианта в UTMake

Предистория: для гарантированной загрузки в EFI-режиме носитель должен быть отформатирован в FAT/FAT32, поскольку EFI производит поиск загрузчика по пути `efi\boot\bootx64.efi` (или `efi\boot\bootia32.efi` для 32-битовой реализации EFI - она встречается на x86-планшетах) только в FAT/FAT32-разделах. И тут появляется противоречивое требование: FAT32-разделы не поддерживают файлы размером более 4Гб (4 294 967 295 байт или 2 в степени 32-1), а большинство дистрибюв (`install.wim`) превышают его. Т.е., обязательно использовать NTFS. А, с другой стороны - обязателен FAT32, иначе EFI не загрузит этот носитель.

Ради справедливости отмечу, что в последних реализациях UEFI некоторые производители (в частности, Asus) начали добавлять в UEFI драйвер NTFS, т.е., на таких ПК возможна EFI-загрузка из NTFS-носителя... но тут без гарантий.

Как поступить? Если бы мы имели дело с HDD, вывод простой: использовать два раздела. Один - FAT32 с EFI-загрузчиком. Другой - NTFS с дистрибутивами. Но тут мы наталкиваемся на другое ограничение, уже Windows: эта ОС "не знает", что Removable-накопители (а к ним относится и флешка!) могут иметь несколько разделов. И поэтому она отображает только один раздел (первый указанный в MBR). Для решения этой проблемы есть, как минимум, два пути.

Первый из них - кардинальное решение. Заключается в том, чтобы найти (например, [тут](#) или [тут...](#)) и использовать утилиту для "прошивки" флешки. Ведь вся хитрость в том, что за идентификацию накопителя (Removable) отвечает один бит в её прошивке или чипе-контроллере. И в некоторых утилитах предусмотрено "превращение" флешки в USB-HDD. После такой операции флешка будет распознаваться Windows как HDD со всеми вытекающими. Кстати, сейчас это становится очень актуально: всё больше появляется флешек 64/128Гб, на которых FAT32 использовать почти невозможно (т.е., можно, но через ж...). Для таких флешек приходится использовать NTFS либо ExFAT (последняя - это вариация FAT32 с поддержкой для больших файлов... но, увы, эта файловая система очень слабо поддерживается загрузчиками). Ограничение первого варианта - это потеря гарантии, возможное повреждение флешки, и т.п.. Т.е., сама операция несложная, но некоторая подготовка нужна: определить контроллер флешки и флеш-память (помогут утилиты ChipEasy и/или GetFlashInfo), найти подходящую программу и разобраться с настройками...

Другой путь - чисто программное решение. Всё дело в том, что для DOS/Bios/EFI правильно отформатированная (т.е., многораздельная) флешка и USB-HDD - это одно и то же. Т.е., при EFI-загрузке из такой флешки достаточно, чтобы на ней был FAT32-раздел с EFI-загрузчиком. Т.е., можно (нужно!) разделить сборку на два раздела: первый - с NTFS/ExFAT (будет доступен под Windows), и второй, загрузочный (обязательно первичный/активный) - под FAT32. Методика подготовки такой флешки: используем менеджер дисков (например, удобно это сделать в Paragon HDM) для разбивки флешки на разделы: первый - под NTFS/ExFAT, второй - под FAT32. Размер второго определяем, исходя из размера загрузочных файлов, которые не будут нужны после загрузки ОС/WinPE. В применении к нашим реалиям это будут загрузчики, их окружение и wim-ядро, т.е. папки `2k10+efi`. Определяем с некоторым запасом этот размер, и получим размер первого раздела как разницу между полной емкостью флешки и размером второго раздела. После разбивки ОС присвоит букву первому разделу, а второй будет недоступен... но ничего страшного. Теперь используем утилиту BootICE. Выбираем в ней свой накопитель, нажимаем "Parts Manage". В управлении раздела выбираем второй раздел, кнопка "Set Accessible". Всё, теперь первый раздел станет невидимым, но мы получим доступ ко второму разделу. На этот раздел мы копируем необходимое (загрузчики и прочее). Если нужно, чтобы флешка была DOS-загрузочной, подключаем свой загрузчик. После выполнения всех операций повторяем манипуляции в BootICE - "Set Accessible" для первого раздела. Теперь первый раздел будет всегда у нас под руками в ОС/PE, а второй будет содержать загрузочные файлы, и из него будет производиться загрузка.

Ещё один вариант реализации - это помещение на второй (скрытый) раздел только файлов, необходимых для EFI-загрузки. В частности, это папка EFI и загрузчики `bootmgr.efi` и `bootm32.efi` (для x64 и x32 соответственно). Но при использовании варианта загрузки от MicroSoft придётся вручную в EFI BCD-меню (`efi\microsoft\boot\BCD`) указывать путь к ядрам/шрифтам/`boot.sdi` на первом разделе КОНКРЕТНОГО накопителя (т.е., эту операцию не автоматизируешь). Более простой вариант - использовать на втором разделе `grub2.efi`, в котором есть возможность подключения драйвера NTFS/ExFAT и последующего поиска и запуска EFI-загрузчика от MicroSoft на NTFS/ExFAT разделе (т.е., первом). Тогда значительно упрощается работа со вторым разделом (по-сути, нам нужно только один раз сконфигурировать второй раздел). Как вариант, можно использовать мою заготовку для 2k10 **Flash2Ptn.exe**. Нужно только подправить путь к файлу-маркеру (поиск производится именно по нему, чтобы исключить ложный переход к папке EFI с ОС пользователя). Путь к маркеру прописан в `efi\boot\grub.cfg`:

search --set=root -f /2k10/winpe/w8x64pe.wim

В таком варианте EFI-загрузка будет происходить следующим образом: UEFI -> Grub2.efi -> Загрузка драйвера NTFS/ExFAT -> Поиск файла `/2k10/winpe/w8x64pe.wim` -> Загрузка из EFI-папки в корне с файлом `/2k10/winpe/w8x64pe.wim` (т.е., загрузка EFI-меню MicroSoft). Что хорошо в данном варианте - так это то, что Grub2.efi полностью сертифицирован под UEFI. Более того, сам UEFI - это и есть оболочка на базе Grub2. Кроме того, Grub2.efi позволяет загружать не только загрузчик от мелко-мягких, но и многие другие. Но это уже отдельная история.

В UTMake имеется возможность реализации последнего варианта при переразбивке флешки. Если для флешки выбрана файловая система NTFS/ExFAT, то станет доступным создание второго раздела... Также, при наличии `sfx-архива` (файл `Flash2Ptn.exe` в папке `Utilites` - это заготовка для EFI-загрузки) с набором необходимых для загрузки файлов производится расчёт необходимого размера этого раздела (чуть больше размера файлов) - режим "Autosize". После запуска переразбивки на флешке создаются первый раздел (ёмкость флешки минус

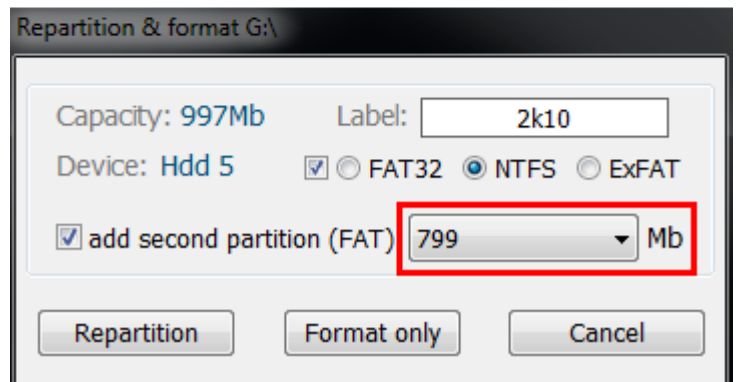
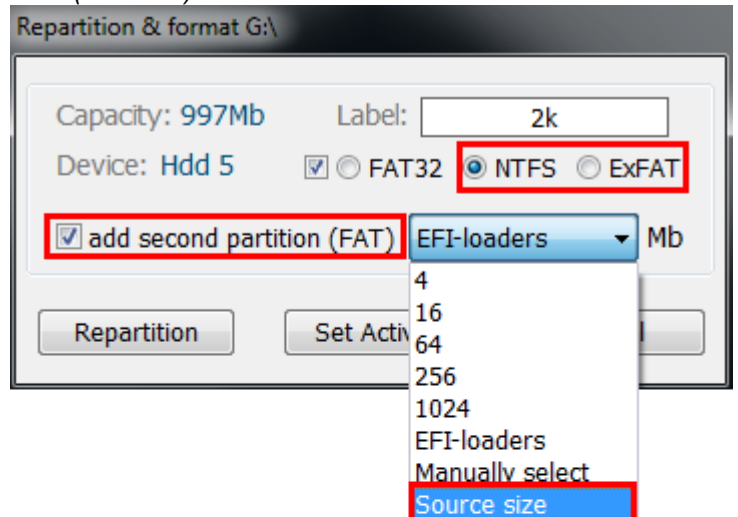
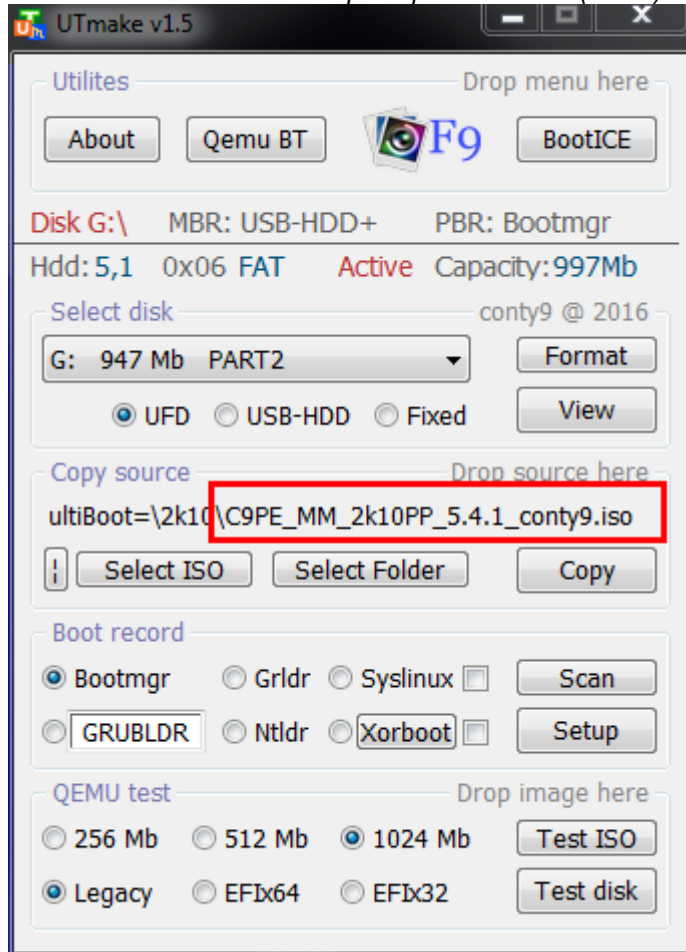
резерв для второго раздела) и второй. После этого второй раздел временно устанавливается видимым и на него копируется содержимое Flash2Ptn.exe. После чего видимым вновь назначается первый (основной) раздел. Также вручную можно реализовать второй вариант - разместить все загрузочные файлы на скрытом разделе - папки 2k10+efi. Инструментом для этого служит кнопка обмена разделов на флешке ("P2" - "B") и прямые руки пользователя. При обычном форматировании (как средствами утилиты, так и средствами ОС) второй раздел не затрагивается.

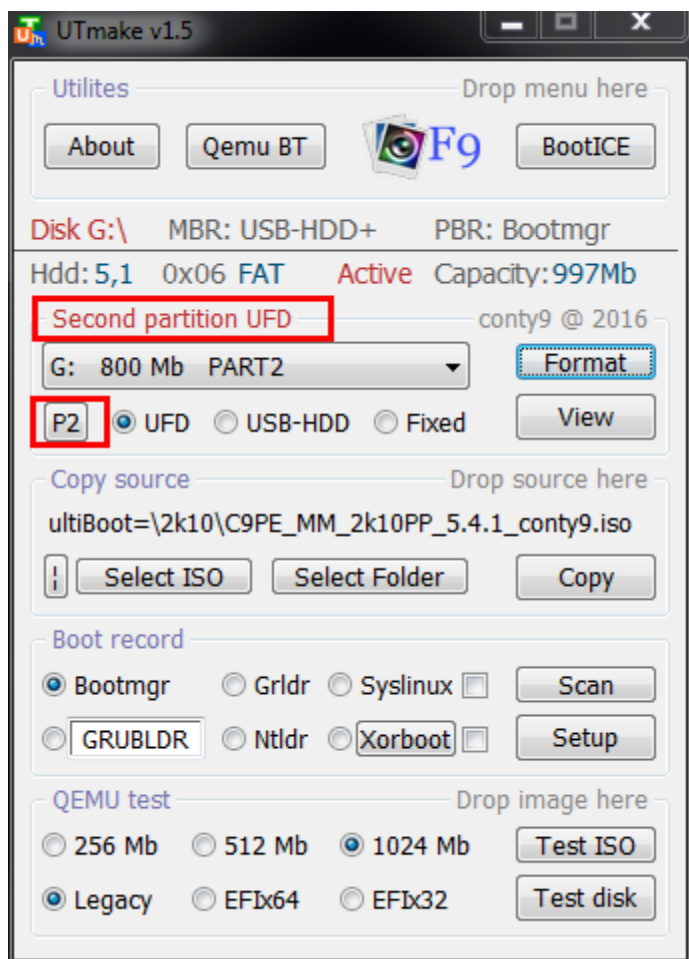
Создание флешки с вторым ("невидимым" под Windows) разделом и размещение на нем WinPE

Начиная с версии 1.5, **UTmake** позволяет разворачивать (и обновлять) загрузочные сборки на второй раздел флешки. Этот раздел недоступен под Windows (хотя без проблем доступен для Linux/DOS). **UTmake** позволяет создавать полноценную двухраздельную флешку, при этом доступны различные варианты "деления разделов":

1. Фиксированный размер второго раздела, доступны значения **4, 16, 64, 256, 1024** Мб.
2. **EFI-loaders** - минимальный второй раздел с автозаливкой на него EFI-загрузчика Grub2. Вариант доступен только при наличии в подкаталоге Utilites SFX-набора Flash2Ptn.exe. Размер второго раздела определяется размером содержимого архива Flash2Ptn.exe с небольшим запасом. В версиях до 1.5 этот пункт назывался "Autosize".
3. **Manually select** - определяемый пользователем размер 2-го раздела. При выборе откроется окно, в котором можно указать желаемый размер (в Мб). После выбора, если значение приемлемо (т.е., не больше емкости флешки или 32Гб /ограничение для FAT32/), это значение появится в списке и будет автоматически выбрано.
4. **Source size** - по размеру источника. Вариант доступен, если предварительно (до входа в меню форматирования) был выбран источник - образ или папка. При выборе этого пункта производится анализ размера (образы автоматически монтируются) источника для заливки. Значение полученного размера (с 10%-ным запасом) появится в списке и будет автоматически выбрано.

*** *Файловая система второго раздела FAT (<2Gb) / FAT32 (2...32Gb).*





Комментарии к картинкам: выбран источник - образ C9PE [1]; выбираем размер 2-го раздела по источнику (создание 2-го раздела доступно только, если для 1-го выбрана файловая система **NTSF** или **ExFAT**) [2]; после анализа размера содержимого образа (с запасом) в списке автоматически выбрано это значение [3]; состояние после форматирования - "видимым" стал второй раздел, появилась кнопка смены разделов "**P2**" [4].

После переразбивки (**Repartition**) флешки под два раздела, в интерфейсе главного окна произойдут некоторые изменения (см. последний скриншот):

1. Появится кнопка "**P2**" ("**P1**"). Число **1/2** указывает на то, какой из разделов в данный момент доступен как буква диска под Windows (т.е., является "видимым"). Нажатие на кнопку позволит обменять разделы местами ("видимый" станет "невидимым" и наоборот).
2. Если "видимым" выбран второй (по размещению физически) раздел, у нас загорится красным "**Second partition UFD**".
3. После завершения переразбивки "видимым" будет 1-й раздел, если был выбран вариант 2-х разделов **EFI-loaders**. Он же будет назначен активным. Если же при переразбивке флешки под два раздела были выбраны другие варианты, тогда после завершения форматирования "видимым" будет выбран второй раздел, он же будет назначен активным. Это сделано так, поскольку предполагается последующая заливка на этот раздел загружаемых сборок.
4. Для неактивного раздела на флешке теперь доступна кнопка "**Set Active**". Что позволяет оперативно сделать раздел активным. Т.е., в принципе, можно держать на двух разделах 2 сборки и быстро переключаться на нужную.

Заливка сборок на второй раздел практически ничем не отличается от обычного разворачивания сборки: копируем данные из источника (**Copy**), подключаем желаемый загрузчик (**Setup**), проверяем загрузку (**Test Disk**). Добавляется лишь один пункт - после подготовки нажимаем "**P2**", чтобы сделать видимым первый раздел и скрыть загрузочный. Также можно повторить проверку загрузки (**Test Disk**)... теперь уже - со второго раздела.

Для чего всё это нужно... Есть несколько позитивных моментов в загрузке PE-сборок со второго раздела флешки. Во-первых, это позволяет решить проблему EFI-загрузки из больших флешек; можно без проблем использовать NTFS/ExFAT для "видимого" раздела (файлопомойки); можно скрыть файлы сборки как от вирусов, так и от обычных ламеров. И при этом остается возможность изменения содержимого флешки (например, с помощью **UTmake** меняем "видимый" <> "невидимый").

Похожее решение предлагается в **MBROSTool**, но там оно жестче: второй раздел модифицируется так, чтобы быть полностью недоступным для менеджеров разделов (отображается как неразмеченное пространство), для любого изменения нужно полностью переформатировать флешку.

Какие PE-сборки можно заливать на такую флешку? До недавнего времени - только сборки, которые после загрузки в память не используют "свой" накопитель (т.е., не имеют "внешних" пакетов драйверов/программ/...).

Например, ранние сборки Стрельца, BootPass от **Joker-2013**, большинство "фирменных" WinPE-сборок (кАкронисы/Парагоны/Ой-вэй/...). Ну и, безусловно, все сборки DOS/Linux.

Но, как оказалось, путем несложной модификации ядер можно заставить WinPE "видеть" загрузочные разделы - и, соответственно, практически полноценно работать с содержимым загрузочного диска (т.е., подключать "внешний" программный пакет, драйверпаки, прочее). Единственное ограничение - невозможно будет загружать ядра в наитив-режиме.

Полноценная поддержка второго раздела реализована в версии 6.0 сборки 2k10.

Работоспособность проверена на всей линейке PE (XP/2003/7/8/10/x86/x64, правда, для x64 вроде обязательно поддержка x86-подсистемы).

Примеры применения утилиты:

Создание загрузочной флешки 'с нуля'

Выбираем флешку (2), нажимаем "Format". При желании меняем метку диска, выбираем желаемую файловую систему из доступных (чтобы снять ограничения автовыбора - отмечаем чекбокс). Если выбрана файловая система NTFS/ExFAT, будет доступна возможность создания второго раздела (можно выбрать его размер или оставить автовыбор). Жмем "Repartition" (переразбивка флешки как HDD, с созданием MBR). Если флешка ранее была загрузочной (MBR имеется), можно использовать "Format" - это обычное форматирование видимого раздела флешки. После успешного форматирования нужно выбрать образ (можно сразу же протестировать его загрузку - "5"/"Test ISO") или папку с распакованной сборкой WinPE (всё содержимое этой папки будет скопировано в корень флешки). Для начала копирования жмем "Copy". Копирование можно приостановить, изменить параметры копирования, продолжить, прервать (и продолжить позже). После успешного завершения копирования выбираем желаемый загрузчик (можно воспользоваться кнопкой "Scan" для поиска имеющихся на флешке загрузчиков). Если имя загрузчика нестандартное, нужно вписать его в поле ввода ("D"). Если отметить чекбокс "F" или "G", тогда для загрузчиков Bootmgr/Gldr/Ntldr можно установить "костыль" на базе Xorboot или Syslinux. Тогда Xorboot (или Syslinux) будет прописан в MBR с единственным пунктом меню - загрузка основного выбранного загрузчика. Такая "двухэтапная" схема загрузки улучшает загрузаемость сборок на старых ПК, рекомендуется схема Syslinux+GRLDR или XorBoot+Bootmgr.

После подключения загрузчика флешка готова к работе. При желании, можно сразу же протестировать её работу "5"/"Test disk".

Внимание!!! При использовании активного раздела **ExFAT** не работают загрузчики **Xorboot/Syslinux**, а также загрузка **C9PE** и некоторых DOS/Linux утилит будет невозможна.

Внимание!!! Если сборка размещена на невидимом разделе используем только 2 загрузчика **Bootmgr/Gldr** (без "костыля" на базе Xorboot).

Создание загрузочного USB-HDD (или обычного HDD) 'с нуля'

В любом удобном менеджере дисков (рекомендую Paragon HDM) создаем первичный раздел под сборку, желательно размером до 32Гб (на разделах крупнее могут возникнуть проблемы с запуском загрузчика). Форматируем в желаемой файловой системе (лучше всего - FAT32, если нужна работа под DOS). Делаем раздел активным. Запускаем UTMake, выбираем созданный диск (2), выбираем образ или папку с распакованной сборкой WinPE, жмем "3"/"Copy". После успешного завершения копирования устанавливаем желаемый загрузчик. После подключения загрузчика хард готов к работе (можно сразу же протестировать его работу "5"/"Test disk").

Обновление имеющейся на флешке/HDD сборки

Выбираем флешку (2), выбираем образ с новой сборкой (или папку с распакованной сборкой). Кроме выбора кнопками, можно просто перетащить образ или нужную папку из окна проводника на модуль копирования ("3"). Для начала копирования жмем "Copy". При копировании будут заменены только более "старые" версии файлов (индикация красным цветом в программе копирования KillDisk **S:23** - указывает, сколько файлов пропущено / не обновлялось). Также полностью сохраняется содержимое пользовательских папок (пользовательскими считаются все папки по маске User.* - User.Config, User.WinPE, User.Image, ...) и папок, отсутствующих в новом образе. После успешного завершения копирования можно подключить загрузчик (необязательно, если раньше он был подключен), либо сменить его на другой. Всё, можно тестировать...

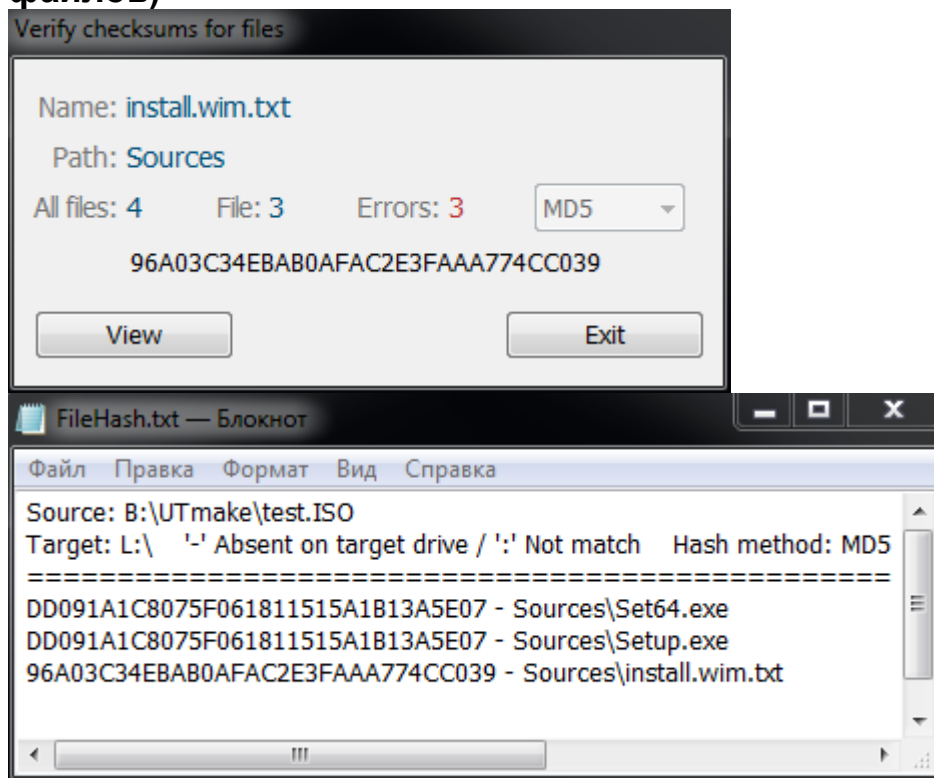
Редактирование BCD-меню или меню Grub4Dos ('Easy BootICE')

Перетаскиваем из окна проводника на модуль внешних утилит ("1") файл BCD-меню, файл меню Grub4Dos (*.lst) или сам загрузчик GRLDR (если меню встроено). После чего будет запущен BootICE в режиме редактирования меню.

Тестирование загрузочных образов ('Easy QEMU')

Перетаскиваем из окна проводника на модуль тестирования ("5") ISO-образ (файл с расширением iso/cdr/dmg), образ флоппика/HDD (расширение ima, img, также поддерживаются упакованные в Gzip/LZMA образы gz/imz) либо целый диск (C:\, D:\, E:\, ...). После этого запустится эмулятор QEMU.

Сверка соответствия файлов получателя и источника (проверка хешей файлов)



Иногда (например, флешка была использована на зараженном ПК или есть подозрение, что носитель поврежден) может возникнуть желание сверить файлы источника и получателя (проще говоря - оригинального образа и сделанной из него флешки). Выбираем свой носитель ("2") и образ, из которого он был создан ("3"). По нажатию на "С" мы попадем в окно сверки хешей. Там нужно выбрать желаемый алгоритм расчета хешей (хеши CRC32 - самые короткие) и запустить сверку. Для ускорения процесса можно отключить антивирус. Отображаемые в процессе проверки значения: имя файла (Name), относительный путь к нему (Path), общее количество файлов (All files), номер проверяемого файла (File), число ошибок (Errors, файл не совпадает или отсутствует на получателе), значение контрольной суммы. Процесс можно прервать. Если в процессе проверки были обнаружены ошибки, то будет создан файл отчета (%Temp%\FileHash.txt), в котором будут указаны отсутствующие ("-") и несовпадающие (":") файлы и их хеши (для оригинала).

Если в качестве получателя выбрать диск, на котором ЗАВЕДОМО нет файлов из источника, то отчет будет содержать хеши всех файлов оригинала. Т.е, при желании, можно сделать список хешей для любой папки или образа.

Расчет хешей производится только для непустых файлов, а в "All files" подсчитывается количество ВСЕХ файлов (включая пустые). Поэтому в результате проверки последнее значение "File" может оказаться меньше "All files" на количество "пустышек".

Утилиту можно распространять без ограничения.