

Create custom Win USB install

```
sudo add-apt-repository ppa:nilarimogard/webupd8
sudo apt update
sudo apt install wousb
sudo add-apt-repository --remove ppa:nilarimogard/webupd8
sudo apt update
```

or build manually from git:

```
git clone https://github.com/slacka/WoeUSB.git
cd WoeUSB/
./setup-development-environment.bash
sudo apt-get install devscripts equivs gdebi-core
mk-build-deps
sudo gdebi wousb-build-deps_*.deb
dpkg-buildpackage -uc -b
sudo gdebi ../wousb*.deb
```

identify USB stick's device

```
lsblk and/or
sudo blkid
```

Create adjustable USB stick from Windows ISO file

```
sudo umount /dev/sdb1

sudo wousb --target-filesystem NTFS --device win_10.iso /dev/sdb
```

or use

```
woeusbgui
```

<https://www.cyberciti.biz/faq/create-a-bootable-windows-10-usb-in-linux/>

Create bootable win7 USB drive in linux

<https://askubuntu.com/questions/289559/how-can-i-create-a-windows-bootable-usb-stick-using-ubuntu>

Plug in USB Drive check the device name using `dmesg | tail` unmount the drive if it was mounted automatically while `/dev/sdX` is the device name shown in `dmesg`

```
sudo umount /dev/sdX1
```

Partition using fdisk:

```
sudo fdisk /dev/sdX
```

1. Type p and Enter to print the current partition table.
2. Delete all the current partitions by typing d then Enter for each partition.
3. Type n and Enter. Type p and Enter. Type 1 and then type Enter three times to create one new primary partition that uses all available space.
4. Type t and Enter (Partition 1 is automatically selected because it's the only partition). Type 7 and Enter to change the type to HPFS/NTFS/exFAT.
5. Type a and Enter. Type 1 and Enter to turn on the Boot flag.
6. To verify everything worked, type p and Enter and make sure the Boot column has an asterisk (*) set and the Id column is set to 7.
7. Finally, type w and Enter to write the changes.

```
sfdisk /dev/sdX <<EOF  
label: dos  
unit: sectors  
type=7, bootable  
EOF
```

format the first partition with ntfs:

```
sudo apt-get install ntfs-3g  
sudo mkfs.ntfs -f /dev/sdX1
```

or: use gparted using GUI

Add Grub bootloader:

```
#MBR instructions (note: EFI does not support NTFS!)  
sudo mkdir -p /mnt/usb  
sudo mount /dev/sdX1 /mnt/usb  
sudo grub-install --target=i386-pc --boot-directory="/mnt/usb/boot" /dev/sdX  
  
cat <<EOF >/mnt/usb/boot/grub/grub.cfg  
default=1  
timeout=15  
color_normal=light-cyan/dark-gray  
menu_color_normal=black/light-cyan  
menu_color_highlight=white/black  
  
menuentry "Start Windows Installation" {  
    insmod ntfs  
    insmod search_label  
    search --no-floppy --set=root --label dos --hint hd0,msdos1  
    ntldr /bootmgr  
    boot  
}
```

```
menuentry "Boot from the first hard drive" {
    insmod ntfs
    insmod chain
    insmod part_msdos
    insmod part_gpt
    set root=(hd1)
    chainloader +1
    boot
}
EOF
```

mount usb drive

```
sudo mkdir -p /mnt/usb
sudo mount /dev/sdX1 /mnt/usb
```

mount win7 iso (iso needs to be available locally):

```
sudo mkdir -p /mnt/iso
sudo mount -o loop /tmp/en_windows_7_professional_x64_dvd_X15-65805.iso
/mnt/iso
```

copy win7 iso content to usb:

```
sudo cp -av /mnt/iso/* /mnt/usb/
```

copy any additional software, drivers etc to usb drive, then unmount it.

Add Drivers to Boot Image

If specific drivers are required for the installation like USB3 Host adapter drivers, follow these steps to add them to the boot/install image:

<https://wimlib.net/>

Download wimtools, libntfs-3g88 and libwim15 from: <https://packages.debian.org/sid/libwim15> and install them using

```
sudo dpkg -i x.deb
```

mount the windows usb stick and go to the sources directory in there you'll find a boot.wim and install.wim file. make a backup copy of both just in case.

use wiminfo to list the images contained in the wim file.

```
sudo apt-get install wimtools dos2unix
```

To add a folder to an image, use this command:

create a commands.txt file and list all directories that need to be added:

```
add /home/drivers /DELL_DRIVERS
```

```
wimupdate boot.wim imageno. < commands.txt
```

repeat with install.wim and the required image number for the windows version.

Unattended Windows install

An autoattend.xml file can be created using this website:

http://www.windowsafg.com/win7x86_x64.html

Mount the USB stick, create/copy/adjust an autoattend.xml file and either copy it in the root of the usb stick in MSDOS format or embed it into the Windows boot image. Windows setup will use the xml file in the root automatically when the setup recognises the device as a removable media. USB HDDs, redirected USB drives for VirtualBox and even some USB sticks etc are considered as fixed drives and the autoattend.xml file on those is being ignored. In this case, the only option is to embed it into the boot image:

```
sudo apt-get install wimtools dos2unix
sudo mkdir -p /mnt/usb /mnt/loopback
sudo mount /dev/sdX1 /mnt/usb
sudo wimmountrw /mnt/usb/sources/boot.wim 2 /mnt/loopback
sudo cp autounattend.xml /mnt/loopback
sudo unix2dos /mnt/loopback/autounattend.xml
sudo wimunmount /mnt/loopback --commit
sudo umount /mnt/usb
```

Example script:

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup"
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">
    <ImageInstall>
      <OSImage>
        <WillShowUI>Never</WillShowUI>
        <InstallTo>
          <DiskID>0</DiskID>
          <PartitionID>1</PartitionID>
        </InstallTo>
      </OSImage>
    </ImageInstall>
    <UserData>
```

```

    <ProductKey>
      <WillShowUI>Never</WillShowUI>
      <Key>XXXX-XXXX-XXXX-XXXX-XXXX</Key>
    </ProductKey>
    <FullName>Information Technology Services</FullName>
    <Organization>Amati Global Investors</Organization>
    <AcceptEula>true</AcceptEula>
  </UserData>
  <DiskConfiguration>
    <WillShowUI>Never</WillShowUI>
    <Disk>
      <DiskID>0</DiskID>
      <WillWipeDisk>true</WillWipeDisk>
      <CreatePartitions>
        <CreatePartition>
          <Order>1</Order>
          <Type>Primary</Type>
          <Extend>true</Extend>
        </CreatePartition>
      </CreatePartitions>
    </Disk>
  </DiskConfiguration>
</component>
</settings>
<settings pass="specialize">
  <component name="Microsoft-Windows-Deployment"
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">
    <RunSynchronous>
      <RunSynchronousCommand wcm:action="add">
        <Path>net user administrator /active:yes</Path>
        <Order>1</Order>
      </RunSynchronousCommand>
    </RunSynchronous>
  </component>
  <component name="Microsoft-Windows-Shell-Setup"
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">
    <ShowWindowsLive>>false</ShowWindowsLive>
  </component>
  <component name="Microsoft-Windows-IE-InternetExplorer"
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">
    <FilterLevel>High</FilterLevel>
    <ShowInformationBar>>false</ShowInformationBar>

```

```
<DisableAccelerators>true</DisableAccelerators>
<DisableDevTools>true</DisableDevTools>
<DisableFirstRunWizard>true</DisableFirstRunWizard>
<DisableOOBAccelerators>true</DisableOOBAccelerators>
<Home_Page>http://www.google.com</Home_Page>
<SearchScopes>
  <Scope wcm:action="add">
    <ScopeDefault>true</ScopeDefault>
    <ScopeDisplayName>Google</ScopeDisplayName>
    <ScopeKey>SearchProvider1</ScopeKey>
<ScopeUrl>http://www.google.com/search?q={searchTerms}</ScopeUrl>
  </Scope>
</SearchScopes>
</component>
<component name="Microsoft-Windows-ErrorReportingCore"
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">
  <DisableWER>1</DisableWER>
</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/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <InputLocale>2057:00000809</InputLocale>
    <UILanguage>en-GB</UILanguage>
    <UserLocale>en-GB</UserLocale>
  </component>
  <component name="Microsoft-Windows-Shell-Setup"
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">
    <OOBE>
      <ProtectYourPC>3</ProtectYourPC>
      <NetworkLocation>Work</NetworkLocation>
      <SkipMachineOOBE>true</SkipMachineOOBE>
      <SkipUserOOBE>>false</SkipUserOOBE>
    </OOBE>
    <UserAccounts>
      <LocalAccounts>
        <LocalAccount wcm:action="add">
          <Password>
            <Value>password</Value>
            <PlainText>true</PlainText>
          </Password>
          <Name>UserName</Name>
```

```
        <Group>Administrators</Group>
        </LocalAccount>
    </LocalAccounts>
    <AdministratorPassword>
        <Value>secretadminpassword</Value>
        <PlainText>>true</PlainText>
    </AdministratorPassword>
</UserAccounts>
<TimeZone>GMT Standard Time</TimeZone>
<ComputerName>PC-WULF</ComputerName>
</component>
</settings>
</unattend>
```

From:

<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:

<http://wuff.dyndns.org/doku.php?id=linux:win-boot-usb&rev=1597872944>

Last update: **2023/05/29 11:53**

