

Windows Install Media

```
./create_win_usb.sh ~/Downloads/Win10.iso /dev/sdc
```

```
#!/bin/bash
# -----
# Create Bootable Windows 10/11 USB in Linux (Debian/RPM-based)
# Version: 1.0
# Author: LinuxConfig.org
# Date: 23-06-2023
# License: GPL-3.0
# -----
# Input parameters: Path to the ISO file and USB block device location
ISO_PATH=$1
USB_BLOCK=$2

echo "ISO Path is: $ISO_PATH"
echo "USB block device is: $USB_BLOCK"

# Check for required commands
REQUIRED_COMMANDS=("rsync" "parted" "wipefs" "mkfs.vfat" "mkfs.ntfs" "udisksctl" "sha256sum" "mount"
"umount" "mkdir" "cp" "sync" "mktemp")

echo "Checking for required commands..."
for cmd in "${REQUIRED_COMMANDS[@]}"; do
    if ! command -v $cmd >/dev/null 2>&1; then
        echo "Missing command $cmd. Please install the corresponding package and rerun this script."
        exit 1
    fi
done
echo "All required commands are available."

# 1. Checking ISO file
read -p "Do you want to calculate the ISO file checksum? This could take some time. Press 'y' to continue or any
other key to skip: " response
if [[ $response =~ ^[Yy]$ ]]
then
    echo "Calculating ISO file checksum..."
```

```
ISO_CHECKSUM=$(sha256sum $ISO_PATH)
echo "Checksum is: $ISO_CHECKSUM"
read -p "Please verify the checksum. Press 'y' to continue: " response

if [[ ! $response =~ ^[Yy]$ ]]
then
    echo "Aborting due to user response."
    exit 1
fi
fi

# Warning about formatting and data loss
echo "WARNING: All data on $USB_BLOCK will be lost!"
read -p "Are you sure you want to continue? [y/N]: " confirm
confirm=${confirm:-N}
if [[ $confirm =~ ^[Yy]$ ]]
then
    # 2. Formatting the USB drive
    echo "Formatting USB drive..."
    wipefs -a $USB_BLOCK

    parted $USB_BLOCK mklabel gpt
    parted $USB_BLOCK mkpart BOOT fat32 0% 1GiB
    parted $USB_BLOCK mkpart INSTALL ntfs 1GiB 100%
    parted $USB_BLOCK unit B print

    # Create temporary directories for mounting
    ISO_MOUNT=$(mktemp -d)
    VFAT_MOUNT=$(mktemp -d)
    NTFS_MOUNT=$(mktemp -d)

    # 3. Mounting the ISO
    echo "Mounting ISO..."
    mount $ISO_PATH $ISO_MOUNT

    # 4. Formatting and copying to the USB
    echo "Copying to USB..."
    mkfs.vfat -n BOOT ${USB_BLOCK}1
    mount ${USB_BLOCK}1 $VFAT_MOUNT

    rsync -r --progress --exclude sources --delete-before $ISO_MOUNT/ $VFAT_MOUNT/
```

```
mkdir -p $VFAT_MOUNT/sources
cp $ISO_MOUNT/sources/boot.wim $VFAT_MOUNT/sources/

mkfs.ntfs --quick -L INSTALL ${USB_BLOCK}2
mount ${USB_BLOCK}2 $NTFS_MOUNT

rsync -r --progress --delete-before $ISO_MOUNT/ $NTFS_MOUNT/

# 5. Unmounting and power off
echo "Unmounting drives and syncing data... This may take a while, do not disconnect your USB drive."
umount $NTFS_MOUNT
umount $VFAT_MOUNT
umount $ISO_MOUNT
sync

# Remove temporary mount directories
rmdir $ISO_MOUNT $VFAT_MOUNT $NTFS_MOUNT

udisksctl power-off -b $USB_BLOCK

echo "Done! You can now safely disconnect your USB drive."
else
echo "Aborted by user."
exit 1
fi
```

Revision #2

Created 15 December 2023 02:29:29 by Michael Cleary

Updated 15 December 2023 02:34:28 by Michael Cleary