

# Steffen Probst

## Blog

### reMarkable mit wine

#### Installation

- Die Installationsdatei bei reMarkable downloaden.
- Die Installation dann einfach mit wine64<sup>1)</sup> starten. Vorzugsweise wine vom winehq.org verwenden. Eine Anleitung ist direkt für die jeweilige Distribution dort zu finden.

```
WINEPREFIX="$HOME/Software/remarkable" \
wine64 ~/Downloads/reMarkable-3.16.1.901-win64.exe
```

#### Programm starten

- Das Programm wie folgt starten und einen Login-Code erzeugen. → <https://my.remarkable.com/device/remarkable?showOtp=true> ← Du musst dich vorher anmelden am Portal.

```
WINEPREFIX="$HOME/Software/remarkable" \
wine64 "C:\reMarkable\reMarkable.exe"
```

2025/01/23 21:35 · SProbst

[wine remarkable linux](#)

### openwrt - kleines Buildimage Script v0.2

Ich habe das Script für openwrt Buildimage etwas optimiert.

[owrt\\_buildimage.sh](#)

```
#!/usr/bin/bash

# Funktion zum Setzen der Konfigurationsvariablen
set_config() {
    KERNEL_PARTSIZE=256
    ROOTFS_PARTSIZE=2048
    OWRTVER="23.05.4"
    ARCH="bcm27xx/bcm2711"
```

```

    ARCHSUB=${ARCH////-}
    PKGS="luci-app-snmpd snmpd usbutils tio tmux vim tcpdump luci-app-
lxc luci-il8n-lxc-de lxc lxc-attach lxc-auto lxc-autostart lxc-cgroup
lxc-checkconfig lxc-common lxc-config lxc-configs lxc-console lxc-copy
lxc-create lxc-destroy lxc-device lxc-execute lxc-freeze lxc-hooks lxc-
info lxc-init lxc-ls lxc-monitor lxc-monitor-d lxc-snapshot lxc-start
lxc-stop lxc-templates lxc-top lxc-unfreeze lxc-unprivileged lxc-
unshare lxc-user-nic lxc-usernsexec docker docker-compose podman luci-
app-dockerman luci-il8n-dockerman-de luci-app-adblock luci-app-
attendedsysupgrade luci-app-firewall luci-app-opkg luci-app-snmpd luci-
app-statistics luci-il8n-adblock-de luci-il8n-attendedsysupgrade-de
luci-il8n-base-de luci-il8n-lxc-de"
    PROFIL="NONE"
    DOWNPATH="$HOME/Downloads"
    FOLDER="openwrt-imagebuilder-$OWRTVER-$ARCHSUB.Linux-x86_64"
}

# Funktion zum Herunterladen und Entpacken des Image Builders
download_and_extract() {
    [[ -d $DOWNPATH ]] || mkdir -p $DOWNPATH
    cd $DOWNPATH/
    if [[ ! -d $FOLDER ]]; then
        wget -q -O -
https://downloads.openwrt.org/releases/$OWRTVER/targets/$ARCH/openwrt-
imagebuilder-$OWRTVER-$ARCHSUB.Linux-x86_64.tar.xz | tar xfJ -
    fi
    cd $FOLDER
}

# Funktion zum Anpassen der Partitionsgrößen in der Konfiguration
adjust_partition_sizes() {
    sed -i
"s/CONFIG_TARGET_KERNEL_PARTSIZE=.*CONFIG_TARGET_KERNEL_PARTSIZE=$KERN
EL_PARTSIZE/;
s/CONFIG_TARGET_ROOTFS_PARTSIZE=.*CONFIG_TARGET_ROOTFS_PARTSIZE=$ROOTF
S_PARTSIZE/" .config
}

# Funktion zum Erstellen des Images
build_image() {
    if [[ $PROFIL == "NONE" ]]; then
        make image PACKAGES="{PKGS}"
    else
        make image PROFILE=$PROFIL PACKAGES="{PKGS}"
    fi
}

# Funktion zum Anzeigen des Ausgabepfads
show_output_path() {
    echo "Das Image ist unter $DOWNPATH/$FOLDER/bin/targets/$ARCH zu
finden."
}

```

```
    echo "Einfach cd $DOWNPATH/$FOLDER/bin/targets/$ARCH in der Konsole
eingeben oder im Filer/Dateimanager suchen."
}

# Hauptfunktion
main() {
    set_config
    download_and_extract
    adjust_partition_sizes
    build_image
    show_output_path
}

# Ausführung der Hauptfunktion
main
```

2024/10/17 07:28 · [SProbst](#)  
[openwrt](#), [build](#), [script](#), [image](#)

## Firefox die 2.: /etc/firefox/policies/policies.json

Ich habe jetzt eine Standardkonfiguration für den Firefox zu bauen. Vorteil ist, dass man diverse Funktionen sich so einstellen kann, das nach einer Neueinrichtung des Systems, gleich wieder alle Funktionen zur Verfügung stehen und, zum Beispiel, die Telemetriedaten deaktiviert sind. Unabhängig davon muss trotzdem, wenn es erwünscht ist, das alte Profile von \$HOME/.mozilla umgezogen werden, wenn nicht die Synchronisierung mit Firefox<sup>2)</sup> aktiviert wurde. Das muss jeder selbst entscheiden, inwieweit er der Mozilla Foundation seine Daten anvertraut und soll hier nicht Thema sein. Solltet ihr die Synchronisierungsfunktion verwenden, müsst ihr euch ggf. die ersten zwei Zeilen policies auskommentieren oder löschen.

Ich habe mir die policies.json unter nixOS bauen lassen, da es hier gut Vorlagen gibt. Dafür habe ich mir ein flake erstellt. Die entsprechende Konfiguration findest du hier → [auf Codeberg](#).

Ein Beispiel wie die policies.json aussehen kann, siehst du hier.

### policies.json

```
{
  "policies": {
    "DisableAccounts": true,
    "DisableFirefoxAccounts": true,
    "DisableFirefoxScreenshots": true,
    "DisableFirefoxStudies": true,
    "DisablePocket": true,
    "DisableTelemetry": true,
    "DisplayBookmarksToolbar": "never",
    "DisplayMenuBar": "default-off",
    "DontCheckDefaultBrowser": true,
    "EnableTrackingProtection": {
```



```
    "installation_mode": "force_installed"
  },
  "uBlock0@raymondhill.net": {
    "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/ublock-origin/latest.xpi",
    "installation_mode": "force_installed"
  },
  "{e4a12b8a-ab12-449a-b70e-4f54ccaf235e}": {
    "install_url":
"https://addons.mozilla.org/firefox/downloads/latest/proxy_switcher_and_manager/latest.xpi",
    "installation_mode": "force_installed"
  }
},
"OverrideFirstRunPage": "",
"OverridePostUpdatePage": "",
"Preferences": {
  "browser.contentblocking.category": {
    "Status": "locked",
    "Value": "strict"
  },
  "browser.formfill.enable": {
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.feeds.section.topstories": {
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.feeds.snippets": {
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.section.highlights.includeBookmarks": {
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.section.highlights.includeDownloads": {
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.section.highlights.includePocket": {
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.section.highlights.includeVisited": {
```

```
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.showSponsored": {
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.showSponsoredTopSites": {
    "Status": "locked",
    "Value": false
  },
  "browser.newtabpage.activity-stream.system.showSponsored": {
    "Status": "locked",
    "Value": false
  },
  "browser.search.suggest.enabled": {
    "Status": "locked",
    "Value": false
  },
  "browser.search.suggest.enabled.private": {
    "Status": "locked",
    "Value": false
  },
  "browser.topsites.contile.enabled": {
    "Status": "locked",
    "Value": false
  },
  "browser.urlbar.showSearchSuggestionsFirst": {
    "Status": "locked",
    "Value": false
  },
  "browser.urlbar.suggest.searches": {
    "Status": "locked",
    "Value": false
  },
  "extensions.pocket.enabled": {
    "Status": "locked",
    "Value": false
  },
  "extensions.screenshots.disabled": {
    "Status": "locked",
    "Value": true
  }
},
"SearchBar": "unified"
}
```

Die ganzen Optionsschalter sind gut dokumentiert und lassen sich über die einschlägigen Seiten herausfinden. Optional sei auch hier wieder auf die `about:config` verwiesen.

2024/04/07 18:02 · SProbst

## Firefox: default Einstellungen global (Win)

Ab und zu kommt man in Verlegenheit, ein Windows für einen anderen Nutzer zu konfigurieren. Da möchte ich, dass der Firefox entsprechende Standardeinstellungen<sup>3)</sup> hat, die nicht mit dem Firefox ausgeliefert werden. Hierfür lässt sich eine entsprechende Einstellung unter dem Firefox vornehmen.

Im Hauptverzeichnis der Firefox Installation unter Windows legt man `local-settings.js` in `defaults/pref` an. Die Javascriptdatei verweist dann auf die `mozilla.cfg`. Diese ist auch eine Javascriptdatei mit den Einstellungen. In der folgenden Zeile, in der `local-settings.js` gibt man dann noch mit, dass die Datei in PlainText zu lesen ist.

### local-settings.js

```
/* This Source Code Form is subject to the terms of the Mozilla Public
 * License, v. 2.0. If a copy of the MPL was not distributed with this
 * file, You can obtain one at http://mozilla.org/MPL/2.0/. */
//
// This pref is in its own file for complex reasons. See the comment in
// browser/app/Makefile.in, bug 756325, and bug 1431342 for details. Do
// not add
// other prefs to this file.

pref("general.config.filename", "mozilla.cfg");
pref("general.config.obscure_value", 0);
```

Wichtig ist, dass in der `mozilla.cfg` die erste Zeile nicht interpretiert wird und diese mit einem Kommentar aufgefüllt wird. Die `mozilla.cfg` muss im Hauptverzeichnis der Windows-Installation liegen.

**Unter Linux muss entsprechend in der jeweiligen Distribution geschaut werden, wie die Konfigurationsdateien abgelegt werden.**<sup>4)</sup> Alternativ kann ein `/etc/firefox/policies.json`<sup>5)</sup> angelegt werden, wo die entsprechenden globalen Einstellungen für Firefox unter Linux abgelegt werden. Das sollte unter allen Linux Distributionen funktionieren<sup>6)</sup>.

Meine Konfiguration hier setzt die Startseite auf die Suchmaschine Qwant. Zusätzlich wird das überprüfen, ob der Firefox der Standardbrowser ist, deaktiviert und das Caching vom Firefox deaktiviert. Dabei habe ich mich an den Einstellungsoptionen aus den Fussnoten bedient und in der `about:config` vom Firefox nachgeschaut. In jeden Fall sollte die Datei mit einem Linux Editor erzeugt werden, dass der Firefox entsprechend LF (LineFeed) Umbrüche seit Version 60 erwartet. Hier kann es beim Erstellen mit dem **Editor(notepad)** von Windows probleme geben, wenn jemand auf die kommt die Dateien unter Windows zu erstellen. Aber wer kommt schon auf solche Ideen. ;)

### mozilla.cfg

```
// IMPORTANT: Start your code on the 2nd line
pref("browser.startup.homepage", "https://www.qwant.com");
pref("browser.shell.checkDefaultBrowser", false);
```

```
pref("browser.cache.disk.enable", false);  
pref("browser.cache.disk.smart_size.enabled", false);  
pref("browser.cache.disk.capacity", 0);
```

2024/04/02 19:48 · [SProbst](#)  
[firefox](#), [einstellung](#), [standard](#)

## Debian - apt pinning

In diesem kleinen Beitrag arbeite ich das LUG Thema vom 06.03.2024 auf. Wir hatten dort das Thema Pinning von deb Paketen unter Debian. Eine weiterführende Dokumentation ist auch direkt auf der Debian Seite <sup>7)</sup> zu finden.

### Was ist apt

apt oder auch apt-get ist die Paketverwaltung unter Debian basierte Linux Distribution <sup>8)</sup>

### Was ist das Ziel?

Mit apt pinning möchte man in der Paketverwaltung bestimmte Programme bei der Installation priorisieren.

Beispiel:

Du möchtest als Standarddistribution Debian 12 bookworm verwenden, aber bestimmst Pakete aus einer anderen Version von Debian (oder einer anderen Distribution) dazu installieren.

Wichtig: Wenn du Pakete aus anderen Distributionen oder Versionen mischst, kann es zu ungewollten Seiteneffekte kommen. Daher gebe ich auf die Anleitung keine Funktionsgarantie und jeder führt diese auf eigenes Risiko durch.

### Wie gehe ich vor?

- Erstelle unter `/etc/apt/source.list.d` eine `.list` mit dem entsprechenden Repo, was du verwenden willst. → Als Vorlage kann die `/etc/apt/source.list` dienen. → Kopiere diese in den Ordner `/etc/apt/source.list.d/` und benenne diese vorzugsweise um. Beispiel: `cp /etc/apt/source.list /etc/apt/source.list.d/foo.list` → Ändere die Einträge in der Datei mit einem Editor deiner Wahl auf das passende Repo ab.
- Lege unter `/etc/apt/preferences.d/` eine Datei an, in der du das Pinning konfigurierst. → Ich gebe der Datei immer voran gestellt eine Nummer und dann, was diese enthalten soll. → Beispiel: `99-debian`
- Öffne die Datei mit dem Editor deiner Wahl und ändere die Einträge wie folgt.

## 99-debian

```
Package: *  
Pin: release a=unstable  
Pin-Priority: 500
```

```
Package: *  
Pin: release a=stable  
Pin-Priority: 900  
  
Package: firefox*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: keepass*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: flameshot*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: filius*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: remmina*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: libreoffi*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: podman*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: docker*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: distrobox*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: remmina*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: qflipper*  
Pin: release a=unstable  
Pin-Priority: 1000  
  
Package: mfcuk  
Pin: release a=unstable
```

```
Pin-Priority: 1000

Package: mfoc
Pin: release a=unstable
Pin-Priority: 1000

Package: fritzing*
Pin: release a=unstable
Pin-Priority: 1000

Package: arduino*
Pin: release a=unstable
Pin-Priority: 1000
```

- Die ersten dreier Block setzen die Priorität für sid/unstable auf den Wert 500. Das bedeutet, dass die Pakete aus diesem Repo installiert werden, wenn Sie benötigt werden.
- Der zweite dreier Block setzte stable auf die Priorität 900. Dadurch liegt die Priorität für stable höher als 500 von unstable und dem Paket aus dem stabilen Zweck vorrangig installiert, wenn die dort Pakete mit einer niedrigeren Versionsnummer vorhanden sind.
- Die nachfolgenden Blöcke priorisieren die Pakete aus unstable mit einer 1000. Die Priorität 1000 sagt, dass die Pakete explizit aus unstable installiert werden sollen und die nötigen Abhängigkeiten zwingend nach gezogen werden sollen. Dabei prüft apt ob er Abhängigkeiten aus dem stable und unstable Zweig der Repositorys auflösen kann.

### Schlusswort

Wenn jemand einen Fehler findet, bitte sendet mir diesen zu. Ich korrigiere dann den Eintrag. Falls ich noch auf etwas eingehen soll, dann bitte auch eine Info. Ansonsten findet man sehr gute Anleitungen im Netz dazu, Ich hoffe ich konnte etwas Licht ins Dunkel bringen.

2024/03/07 07:33 · [SProbst](#)  
[debian](#), [linux](#), [apt](#), [pinning](#)

[Ältere Einträge >>](#)

## Index

### A

- [About me ...](#)

### B

- [Blog](#)

### N

- [Notizen](#)

L

- [Linksammlung](#)

1)

siehe Codeblock

2)

Firefox Cloud/Konto → <https://www.mozilla.org/de/account/>

3)

<https://support.mozilla.org/en-US/kb/customizing-firefox-using-autoconfig>

4)

<https://linuxconfig.org/how-to-customize-firefox-using-the-policies-json-file>

5)

<https://mozilla.github.io/policy-templates/>

6)

von mir noch nicht getestet → wird aber bei Gelegenheit nachgeholt

7)

<https://wiki.debian.org/AptConfiguration>

8)

Wie Linux Mint der Ubuntu zum Beispiel.

From:

<https://wiki.lug-wr.de/wiki/> - **Wiki der Linux User Group Wernigerode**

Permanent link:

<https://wiki.lug-wr.de/wiki/doku.php?id=user:sprobst:start&rev=1574451719>

Last update: **2019/11/22 20:41**

