## Toolchain

#!/bin/bash

set -e

OPENWRT\_VERSION="23.05.3"

TARGET="ath79"

SUBTARGET="generic"

SDK\_FILENAME="openwrt-sdk-${OPENWRT\_VERSION}-${TARGET}-${SUBTARGET}\_gcc-12.3.0\_musl.Linux-x86\_64.tar.xz"

SDK\_URL="https://downloads.openwrt.org/releases/${OPENWRT\_VERSION}/targets/${TARGET}/${SUBTARGET}/${SDK\_FILENAME}"

SDK\_DIR\_LONG="openwrt-sdk-${OPENWRT\_VERSION}-${TARGET}-${SUBTARGET}\_gcc-12.3.0\_musl.Linux-x86\_64"

SDK\_DIR\_SHORT="${TARGET}-sdk"

cd /root

if [ -d "${SDK\_DIR\_SHORT}" ]; then

rm -rf "${SDK\_DIR\_SHORT}"

fi

if [ -d "${SDK\_DIR\_LONG}" ]; then

rm -rf "${SDK\_DIR\_LONG}"

fi

if [ -f "${SDK\_FILENAME}" ]; then

rm -f "${SDK\_FILENAME}"

fi

wget "${SDK\_URL}"

tar xf "${SDK\_FILENAME}"

mv "${SDK\_DIR\_LONG}" "${SDK\_DIR\_SHORT}"

cd "${SDK\_DIR\_SHORT}"

./scripts/feeds update -a

./scripts/feeds install -a

echo "--- SETUP COMPLETE ---"

echo "SDK is ready in $(pwd)"

Phonebook on file server:

Location: http://hb9bla-vm-tunnelserver.local.mesh/filerepo/Phonebook/AREDN\_PhonebookV2.csv

Content:

first\_name,name,callsign,ip-address,telephone,email,club,mobile,street,City,International Number,Privat

Francois,Burri,HB9IBI,122630@122630.local.mesh,122630,,,+41,,,1411226,

Lars,Nef,HB9VBE,122730@122730.local.mesh,122730,,,+41,,,1411227,

Yves,Bornick,HB9HHH-1,141830@141830.local.mesh,141830,yves.bornick@bornick.ch,,+41 79 426 61 11 ,Champ du Clos 6,Vuarrens,1411418,

Yves,Bornick,HB9HHH-2,141831@141831.local.mesh,141831,yves.bornick@bornick.ch,,+41 79 426 61 11 ,Champ du Clos 6,Vuarrens,,

Vincent,Jouby,HB9VCJ,152830@152830.local.mesh,152830,hb9vcj@hb9vcj.ch,,+41763484117,Ch. Champ Cavin 14,Surpierre,1411528,

Vincent,Jouby,HB9VCJ,152831@152831.local.mesh,152831,hb9vcj@hb9vcj.ch,,+41763484117,Ch. Champ Cavin 15,Surpierre,,

Beat ,Monnard,HB9IIV,169930@169930.local.mesh,169930,hb9iiv@monnard.net,HB9CF,+41 79 239 74 51,,Porsel FR,1411699,

Store on HAP: /usr/bin/phonebook\_fetcherV1

# Make it start and stop automatically

## 1) files/postinst

Place this file in package/phonebook-fetcher/files/postinst (make it executable):

#!/bin/sh

# package post-install script

[ "$IPKG\_INSTROOT" != "/" ] && exit 0

# Enable & start on install (or upgrade)

[ -x /etc/init.d/phonebook\_fetcher ] && {

/etc/init.d/phonebook\_fetcher enable

/etc/init.d/phonebook\_fetcher restart

}

exit 0

## 2) files/prerm

Place this in package/phonebook-fetcher/files/prerm (also executable):

#!/bin/sh

# package pre-removal script

[ "$IPKG\_INSTROOT" != "/" ] && exit 0

# Stop & disable on remove

[ -x /etc/init.d/phonebook\_fetcher ] && {

/etc/init.d/phonebook\_fetcher stop

/etc/init.d/phonebook\_fetcher disable

}

exit 0

## 3) Update your Makefile

Ensure your Makefile copies these into the opkg info directory. Add the following under your package definition:

define Package/phonebook-fetcher/install

$(INSTALL\_DIR) $(1)/usr/bin

$(INSTALL\_BIN) $(PKG\_BUILD\_DIR)/phonebook\_fetcher $(1)/usr/bin/

$(INSTALL\_DIR) $(1)/etc/init.d

$(INSTALL\_BIN) $(PKG\_BUILD\_DIR)/files/phonebook\_fetcher.init \

$(1)/etc/init.d/phonebook\_fetcher

# install opkg control scripts

$(INSTALL\_DIR) $(1)/usr/lib/opkg/info

$(INSTALL\_BIN) $(PKG\_BUILD\_DIR)/files/postinst \

$(1)/usr/lib/opkg/info/phonebook-fetcher.postinst

$(INSTALL\_BIN) $(PKG\_BUILD\_DIR)/files/prerm \

$(1)/usr/lib/opkg/info/phonebook-fetcher.prerm

endef

## Result

With that in place, when you run:

opkg install phonebook-fetcher\_1.0-1\_mips.ipk

it will:

1. Drop the binary and init script into place.
2. Run postinst, which **enables** and **starts** the service for you.

When you later run:

opkg remove phonebook-fetcher

it will:

1. Run prerm, which **stops** and **disables** the service before uninstalling.

Now there’s **no manual /etc/init.d/... step** required—the package handles it all.

Start service:  
root@HB9BLA-HAP-2:~# /etc/init.d/sip-proxy stop

root@HB9BLA-HAP-2:~# ps | grep sip-proxy | grep -v grep

root@HB9BLA-HAP-2:~# /etc/init.d/sip-proxy start

root@HB9BLA-HAP-2:~# ps | grep sip-proxy | grep -v grep

4546 root 1676 S< /usr/bin/sip-proxy