[1]

```
cat /boot/loader/entries/ostree-rhel-atomic-host-0.conf
initrd /ostree/rhel-atomic-
host-3fdd39fe51b1b9b84a3d6e4c27041d0bd2ac2f1b7750275b1822a2bfeb48e5b9/
initramfs-3.10.0-514.6.1.el7.x86_64.img
options no_timer_check console=tty1 console=ttyS0,115200n8
crashkernel=auto rd.lvm.lv=atomicos/root root=/dev/mapper/atomicos-
root ostree=/ostree/boot.0/rhel-atomic-host/
3fdd39fe51b1b9b84a3d6e4c27041d0bd2ac2f1b7750275b1822a2bfeb48e5b9/0
ipv6.disable=1
title Red Hat Enterprise Linux Atomic Host 7.3 7.3.2 (ostree)
linux /ostree/rhel-atomic-
host-3fdd39fe51b1b9b84a3d6e4c27041d0bd2ac2f1b7750275b1822a2bfeb48e5b9/
vmlinuz-3.10.0-514.6.1.el7.x86_64
version 1
```

[2]

```
grub2-mkconfig > /boot/loader/grub.cfg.new
Generating grub configuration file ...
/etc/grub.d/10_devmode+: line 28: _OSTREE_GRUB2_BOOTVERSION: unbound
variable
```

[3]

```
cat /etc/grub.d/10_devmode+
#!/bin/bash
#
# Copyright (C) 2015-2016 Red Hat, Inc.
#
# This file is part of atomic-devmode.
#
# atomic-devmode is free software: you can redistribute it
# and/or modify it under the terms of the GNU Lesser General
# Public License as published by the Free Software
# Foundation, either version 3 of the License, or (at your
# option) any later version.
#
# atomic-devmode is distributed in the hope that it will be
# useful, but WITHOUT ANY WARRANTY; without even the implied
# warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
# PURPOSE.  See the GNU Lesser General Public License for
# more details.
#
# You should have received a copy of the GNU Lesser General
# Public License along with atomic-devmode. If not, see
# <http://www.gnu.org/licenses/>.

set -euo pipefail
IFS=$'\n\t'

# The BLS dir we operate on is the one that *will* become the new
```

```
loader
# symlink, not necessarily the current one.
BLS_DIR="/boot/loader.${_OSTREE_GRUB2_BOOTVERSION}/entries/"

GRUB2_ENTRY_TITLE="Developer Mode"

NOCLOUD_SEEDFROM="/usr/share/atomic-devmode/cloud-init."
NOCLOUD_HOSTNAME="localhost"
NOCLOUD_INSTANCE_ID="devmode"

# NB: using local is cool, but do not call functions/programs in the
# initialization step. Otherwise it won't exit even if we have errexit
on.

main() {

    # check which hook called us
    if [[ $0 == *+ ]]; then
        main_add_entry
    elif [[ $0 == *- ]]; then
        main_remove_entry
    else
        echo "ERROR: This script should not be called directly." >&2
    fi
}

main_add_entry() {

    # read in the most updated BLS entry that has atomic-devmode
    local entry_filename
    entry_filename=$(get_bls_entry)

    local entry_text
    entry_text=$(cat $entry_filename)

    # modify it for our needs
    local new_entry_text
    new_entry_text=$(modify_bls_entry "$entry_text")

    # save it back to disk
    local new_entry_filename="${entry_filename%.conf}-devmode.conf"
    echo "$new_entry_text" > "$new_entry_filename"
}

main_remove_entry() {

    # make sure the directory exists and is writeable
    if [ ! -w "$BLS_DIR" ]; then
        echo "ERROR: Directory \"$BLS_DIR\" not writeable." >&2
        exit 1
    fi

    # remove any leftover devmode BLS entries
    find "$BLS_DIR" -name '*-devmode.conf' -delete
}

get_bls_entry() {

    # make sure the directory exists and is writeable
    if [ ! -w "$BLS_DIR" ]; then
        echo "ERROR: Directory \"$BLS_DIR\" not writeable." >&2
        exit 1
    fi

    local best_entry
    local best_version=-1

    local entry
    for entry in $(find "$BLS_DIR" -name '*.conf'); do

        local ostree=$(get_bls_entry_ostree "$entry")
        if ! ostree_has_devmode "$ostree"; then
            continue
        fi

        local version=$(get_bls_entry_version "$entry")
        if [ $version -gt $best_version ]; then
            best_entry="$entry"
            best_version=$version
        fi
    done

    if [ -z "$best_entry" ]; then
        echo "ERROR: No BLS entries found in \"$BLS_DIR\"." >&2
        exit 1
    fi

    echo $best_entry
}

get_bls_entry_version() {
    local entry=$(cat "$1")
    local line=$(echo "$entry" | grep ^version)
    echo "${line#* }"
}

get_bls_entry_ostree() {
    local entry=$(cat "$1")
    local line=$(echo "$entry" | grep ^options)
    line=${line#* ostree=}
    echo "${line%% *}"
}

ostree_has_devmode() {
    [ -f "${1}${NOCLOUD_SEEDFROM}meta-data" ] && \
    [ -f "${1}${NOCLOUD_SEEDFROM}user-data" ]
}

# $1 - original entry
modify_bls_entry() {
    local entry="$1"

    # get the title used in the original entry
    orig_title=$(grep title <<< "$entry" | cut -f 2 -d ' ')

    # use the pretty name if possible, otherwise re-use the same
    title=$(get_pretty_name "$orig_title")

    # tag it and escape it
    title=$(sed_escape "$title $GRUB2_ENTRY_TITLE")

    # update the entry
    entry=$(sed "/^title / s/ .*/ $title/" <<< "$entry")

    # add cloud-init arg
    local args="ds=nocloud"
    for arg in "h=$NOCLOUD_HOSTNAME" \
               "i=$NOCLOUD_INSTANCE_ID" \
               "s=$NOCLOUD_SEEDFROM"; do
        args=${args}$(sed_escape "\;$arg")
    done
    entry=$(sed "/^options / s/$/ $args/" <<< "$entry")

    # set version to 0 so we're always last
    entry=$(sed "/^version / s/ .*/ 0/" <<< "$entry")

    echo "$entry"
}

# $1 - default
# NB: we use a subshell to keep clean namespace
get_pretty_name() (
    local default="$1"

    if [ -f /etc/os-release ]; then
        . /etc/os-release
    elif [ -f /usr/lib/os-release ]; then
        . /usr/lib/os-release
    fi
    echo ${PRETTY_NAME:-$1}
)

# $1 - value to escape
sed_escape() {
    # http://stackoverflow.com/questions/407523
    sed 's/[\/&]/\\&/g' <<< "$1"
}

main "$@"
```