

```
#!/bin/bash

# This script uses a macOS 10.13 and later installer application to repair the
# Recovery volume (APFS) or partition (HFS+).
#
# Based on the following script:
#
# https://gist.github.com/jhttps://github.com/rtrouton/create_macos_recovery
# .gitonathantneal/f20e6f3e03d5637f983f8543df70cef5

# Provide custom colors in Terminal for status and error messages

msg_status() {
    echo -e "\033[0;32m-- $1\033[0m"
}
msg_error() {
    echo -e "\033[0;31m-- $1\033[0m"
}

# Explanation of how the script works

usage() {
    cat <<EOF
Usage:
$(basename "$0") "/path/to/Install macOS [Name].app"

Description:
This script uses a macOS 10.13 and later installer application to repair the
Recovery volume (APFS) or partition (HFS+).
Requirements:

Compatible macOS installer application for the Mac's installed OS.
Account with the ability to run commands using sudo, to allow specific
functions to run with root privileges.

EOF
}

# Set the macOS installer path as a variable.

macOS_installer="$1"
mount_point="$macOS_installer/Contents/SharedSupport"

# Remove trailing slashes from input paths if needed

macOS_installer=${macOS_installer%/}

# Check to see if the path to the macOS installer application has been provided
# as part of running the script.
```

```

if [[ -z "$1" ]] || [[ ! -d "$1" ]]; then
    msg_error "The path to the macOS installer application is required as the
    first argument."
    usage
    exit 1
else
    msg_status "macOS installer is $macOS_installer"
fi

# Check for the OS information of both the macOS installer application
# and that of the Mac this script is running on.

macos_version=$(/usr/bin/sw_vers -productVersion)
macos_version_digits_only=$(echo "$macos_version" | awk -F'[^\0-9]*'
'$0=$1$2$3')
installer_version=$(/usr/libexec/PlistBuddy -c 'Print :System\ Image\
Info:version' "$macOS_installer/Contents/SharedSupport/InstallInfo.plist")
installer_version_digits_only=$(echo $installer_version | awk -F'[^\0-9]*'
'$0=$1$2$3')
installer_qualifies=$(echo $installer_version_digits_only | head -c4)
installer_mounted_volume=$(echo "$macOS_installer" | grep -o 'Install.*' | sed
's/....$//')

# If the installed OS is only four digits long, add a zero to the end.

if [[ $(echo -n "$macos_version_digits_only" | wc -m | tr -d '[:space:]') =
"4" ]]; then
    macos_version_digits_only="$macos_version_digits_only"0
fi

# If the installer's OS version is only four digits long, add a zero to the
end.

if [[ $(echo -n "$installer_version_digits_only" | wc -m | tr -d '[:space:]')
= "4" ]]; then
    installer_version_digits_only="$installer_version_digits_only"0
fi

# Define download URL and name for the RecoveryHDUUpdate installer package
# which includes the tools needed to rebuild the Recovery volume or partition.

recovery_download="http://swcdn.apple
.com/content/downloads/14/54/001-08570-A_XCP1PVIXQK/xg4hvg6zgpoxonqkgiw2e2cqo1
c0isl97q/SecUpd2020-004HighSierra.RecoveryHDUUpdate.pkg"
recovery_package="SecUpd2020-004HighSierra.RecoveryHDUUpdate.pkg"

# Detect if the macOS installer application is running macOS 10.13.x or later.
If the
# macOS installer application is for 10.12.x or earlier, stop and display an
error message.

```

```

if [[ "$installer_qualifies" -lt 1013 ]]; then
    msg_error "This script supports repairing Recovery for macOS 10.13.0 and
    later."
    msg_error "Please use an installer app which installs macOS 10.13.0 or
    later."
    exit 1
else
    msg_status "Installer application for macOS $installer_version detected.
    Proceeding...."
fi

if [[ "$installer_version_digits_only" -lt "$macos_version_digits_only" ]];
then
    msg_error "The macOS installer app provided installs $installer_version's
    Recovery, which is earlier than macOS $macos_version"
    msg_error "Please use an installer app which installs macOS $macos_version
    or later."
    usage
    exit 1
fi

# Identify the target disk

boot_drive=$(/usr/sbin/diskutil info "$(bless --info --getBoot)" | awk -F':'
'/Volume Name/ { print $2 }' | sed -e 's/^[[:space:]]*//')
msg_status "Target disk is ${boot_drive}."

# Detect the target disk's filesystem type (APFS or HFS+)

filesystem_type=$(/usr/sbin/diskutil info "$boot_drive" | awk '$1 == "Type" {
print $NF }')
msg_status "Target filesystem is ${filesystem_type}."

msg_status "Downloading $recovery_package into /private/tmp"
/usr/bin/curl "$recovery_download" --progress-bar -L -o
/private/tmp/"$recovery_package"
pkgutil --expand /private/tmp/"$recovery_package"
/private/tmp/recoveryupdate"$installer_version"

if [[ -d /private/tmp/recoveryupdate"$installer_version" ]]; then
    return_code=0
    if [[ "${filesystem_type}" == "apfs" ]]; then
        msg_status "Running ensureRecoveryBooter for APFS target volume:
        ${boot_drive}"
        /private/tmp/recoveryupdate"$installer_version"/Scripts/Tools/dm
        ensureRecoveryBooter "$boot_drive" -base "$mount_point/BaseSystem.dmg"
        "$mount_point/BaseSystem.chunklist" -diag
        "$mount_point/AppleDiagnostics.dmg"
        "$mount_point/AppleDiagnostics.chunklist" -diagmachineblacklist 0
        -installbootfromtarget 0 -slurpappleboot 0 -delappleboot 0
        -addkernelcoredump 0
    fi
fi

```

```

return_code=$(( $return_code + $? ))
  if [[ "$return_code" == 0 ]]; then
    msg_status "Successfully created $installer_version Recovery volume
    on ${boot_drive}."
  else
    msg_error "Failed to create $installer_version Recovery volume on
    ${boot_drive}."
    msg_error "Recovery tools returned the following non-zero exit code:
    $return_code"
    exit $return_code
  fi
elif [[ "${filesystem_type}" == "hfs" ]]; then
  msg_status "Running ensureRecoveryPartition for non-APFS target volume:
  ${boot_drive}"
  /private/tmp/recoveryupdate"$installer_version"/Scripts/Tools/dm
  ensureRecoveryPartition "$boot_drive" "$mount_point/BaseSystem.dmg"
  "$mount_point/BaseSystem.chunklist" "$mount_point/AppleDiagnostics.dmg"
  "$mount_point/AppleDiagnostics.chunklist" 0 0 0
  return_code=$(( $return_code + $? ))
  if [[ "$return_code" == 0 ]]; then
    msg_status "Successfully created $installer_version Recovery
    partition on ${boot_drive}."
  else
    msg_error "Failed to create $installer_version Recovery partition on
    ${boot_drive}."
    msg_error "Recovery tools returned the following non-zero exit code:
    $return_code"
    exit $return_code
  fi
else
  msg_error "Failed to create $installer_version Recovery partition on
  ${boot_drive}."
  msg_error "${boot_drive} does not have an APFS or HFS+ filesystem."
fi
else
  msg_error "Unable to locate the following directory:
  /private/tmp/recoveryupdate"$installer_version"
  exit 1
fi

# Clean up
/bin/rm -rf /private/tmp/"$recovery_package"
/bin/rm -rf /private/tmp/recoveryupdate"$installer_version"

```