

Full macOS Model name to Inventory (Custom Field)

Description

Need to know the full name of "Retina, 15-inch, Late 2013)" rather than the Model Identifier "MacBookPro11,3". The provided script will look up the name and write it to inventory as a custom field

If necessary, please view the following for details on creating [Custom Fields](#).

Information

Since the introduction of Apple's new randomized serial numbers, the method of using Apple's online lookup service does not work with these devices. However, all modern devices have a store of a product name locally on the device. The below method will check for the existence of a local name, but if not found, will query Apple instead based upon the Serial Number of the device

Note, that the name stored locally may differ slightly from the one provided by Apple's lookup server. To give an indication of the difference in details see the below table (sometimes they will match):

Examples:

Name as indicated by Apple's servers and About This Mac	Name as stored in local file
MacBook Pro (13-inch, M1, 2020)	MacBook Pro (13-inch, M1, 2020)
MacBook Pro (Retina, 15-inch, Mid 2015)	15" MacBook Pro with Retina display (Mid 2015)
iMac (21.5-inch, Late 2009)	iMac Intel Core 2 Duo (widescreen, Late 2009)
Mac mini (Late 2012)	Mac mini (Late 2012)

Virtual Devices

Virtual machines do not have legitimate Apple serial numbers by default and you may use virtual machines with the FileWave client. Since the serial number is used to determine the macOS model, additionally the script has been designed to allow for machines built using VMware. Similar consideration could be taken for devices built with other virtualisation software: e.g. Parallels, VirtualBox, etc.

Script

Although this uses a local file, for older hardware information may be missing. In this instance it will attempt to pull the information directly from Apple:

```
#!/bin/zsh

mac_model=$(sysctl -n hw.model)

if [[ "$mac_model" =~ ^"VMware" ]]
then
    echo "$mac_model"
    exit 0
fi

# File that stores device information. Old models may not have this file. Additionally some models only have a
description
# Information is not exactly the same as that reported through About This Mac, but is similar enough to suffice in
most instances
apple_locale=$(defaults read /Library/Preferences/.GlobalPreferences.plist AppleLocale)
attr_file="/System/Library/PrivateFrameworks/ServerInformation.framework/Versions/A/Resources/${apple_locale}.lproj/SIMachineAttributes.plist"

if [ -e "$attr_file" ]
then

    # Use marketingModel if available
    model_name=$(/usr/libexec/PlistBuddy -c "Print :${mac_model}:_LOCALIZABLE_:marketingModel" "$attr_file"
2>/dev/null)
```

```

# If not found, use description if available
if [ -z $model_name ]
then
    model_name=$(/usr/libexec/PlistBuddy -c "Print :${mac_model}:_LOCALIZABLE_:description"
"$attr_file" 2>/dev/null)
fi
fi

# If not found in plist check ioreg. Apple seemed to have stopped populating the plist file, but do populate
ioreg data now
if [ -z "$model_name" ]
then
    model_name=$(ioreg -c IOPlatformDevice | awk -F "\"" '/product-name/ || /product-description/ {print $(NF-1);
exit}')
fi

# If still not found, try to get the details from Apple
if [ -z "$model_name" ]
then
    # Redirect standard error as macOS 10.10.x has an unfixed bug with system_profiler
    serial_number=$(ioreg -l | awk -F "\"" '/IOPlatformSerialNumber/ {print $(NF-1)}')

    # Get serial number length, note wc includes new line character
    # For serials of 11 characters, last 3 digits of serial required
    # For serials of 12 characters, last 4 digits of serial required
    serial_number_length=$(echo $serial_number | wc -c)

    # wc also adds empty spaces
    case $serial_number_length in

        *13*)
            serial_number=$(echo $serial_number | awk '{print substr( $NF, length($NF) - 3, length($NF) ) }')
            ;;
        *12*)
            serial_number=$(echo $serial_number | awk '{print substr( $NF, length($NF) - 2, length($NF) ) }')
            ;;
        *)
            echo "$mac_model"
            exit 0
            ;;
    esac

    model_name=$(curl -s https://support-sp.apple.com/sp/product?cc=$serial_number | sed 's|.*<configCode>\(.*\)|
</configCode>.*|\1|')

    # If this still fails, just return the hardware model details
    if [[ "$model_name" =~ "error" ]]
    then
        model_name="$mac_model"
    fi
fi

if [ -z $model_name ]
then
    echo "$mac_model"
else
    echo "$model_name"
fi

exit 0

```