

Custom Fields and Multi-Lined Outputs

Description

Custom Fields are an excellent way to provide additional reporting information on devices that are not included as standard inventory. However, there are alternative methods to present this information where the output includes large amounts of text.

Information

Scripted Custom Fields values are based upon the output of the included script. Although the information desired may be a list of items, sometimes it may be better to return an alternate response to that information and instead store the returned list of items differently. As such, the decision then becomes:

- What is the best information to return as a value, which can concisely provide meaningful information
- Where is the best place to store the required values for reference

As an example, consider 32-bit Apps on macOS 10.15 that are no longer supported. The list of responses from such a script could be vast, would be practically unreadable and make searching responses awkward. Instead, consider perhaps counting the amount of Apps stored on a device that are 32-bit and return this as an integer. Devices may then be targeted based upon the amount of 32-bit Apps that are installed.

The required list of 32-bit Apps, though, still needs to be stored, preferably located in an easy to obtain location for analysis.

Details

Continuing with the example, as of macOS 10.15 Apple dropped support for 32-bit Apps. Consideration was then required for listing 32-bit Apps on devices prior to upgrading to 10.15, such that software upgrades or alternate software could be implemented to replace those utilised. For this example the chosen options are:

- Return the quantity of 32-bit Apps as an Integer
- Output the list of 32-bit Apps into the FileWave Client log.

As an integer, Inventory Queries may be built to target the devices with the highest quantity of installed 32-bit Apps first and over time, as software is replaced, the count should come down, until all apps have been accounted for.

The FileWave Client log is easily accessible from the Admin console. Additionally, by tailoring the output with desired key words, observing this information then becomes easy

The script could look something like the following, which consists of:

- A python section to provide a dated format to match the current log
- A function to define the output if either an Application or Framework has been located (consideration has been made to identify Apps with Apps or within Frameworks, etc)
- A default output for any other identified binary, library, etc
- A command to identify the apps, pushed into a loop to trigger the above
- A total count to report back as the overall integer value

```
#!/bin/zsh

testline=""
linecount=0
logfile=/private/var/log/fwclld.log

function time_formatted {
python - <<END
import datetime

now = datetime.datetime.now()
year = '{:04d}'.format(now.year)
month = '{:02d}'.format(now.month)
day = '{:02d}'.format(now.day)
hour = format(now.hour)
minute = '{:02d}'.format(now.minute)
second = '{:02d}'.format(now.second)
mlsec = format(now.microsecond)
the_time = '{}-{}-{} {}:{}:{}.{}'.format(year, month, day, hour, minute, second, mlsec[:3])

print(the_time)
```

```

END
}

function printline {

    if [[ "$@" != "$testline" ]]
    then
        linecount=$(( linecount + 1 ))
        echo -e "\t$linecount: $@" >> "$logfile"
    fi

    testline="$@"
}

echo $(time_formatted)"|main|INFO|CLIENT|Checking for 32bit apps" >> "$logfile"

while read line
do
    case "$line" in

        *.framework*)
            printline $(echo "$line" | awk -F "\\\.framework" '{print "32bit
app|Application|" $1 ".framework"}')
            ;;
        *.app*)
            printline $(echo "$line" | awk -F "\\\.app" '{print "32bit app|Framework|" $1 ".app"}')
            ;;
        *)
            linecount=$(( linecount + 1 ))
            echo -e "\t$linecount: 32bit app|Other|$line" >> "$logfile"
            ;;
    esac
done< <(mdfind "kMDItemExecutableArchitectures == 'i386*' && kMDItemExecutableArchitectures != 'x86*'"

echo $linecount
echo -e "\tTotal 32bit apps found: $linecount" >> "$logfile"

```

Viewing the list output to the log file may be achieved via Client Monitor > Client Log and then by searching for '32bit app', which may look something like:

Pulling a log via client monitor requires a connection from your admin directly to the client. Client firewall, NAT, and Network ACL might prevent this from happening. [Default TCP and UDP Port Usage](#)

```

2019-07-20 15:46:31.373|main|INFO|CLIENT|Checking for 32bit apps
1: 32bit app|Framework|/Users/rstehpens/Documents/Papercut MF Clent/mac/legacy/PCClient.app
2: 32bit app|Framework|/Users/rstehpens/Documents/Papercut MF Clent/mac/legacy/client-local-install.app
3: 32bit app|Framework|/Applications/BBEdit.app
4: 32bit app|Framework|/Applications/Adobe Connect/Adobe Connect.app
5: 32bit app|Framework|/Applications/RingCentral Meetings.app
6: 32bit app|Application|/System/Library/Frameworks/CoreServices.framework
7: 32bit app|Application|/System/Library/Frameworks/Carbon.framework
8: 32bit app|Other|/usr/bin/vmmap32
9: 32bit app|Other|/usr/bin/stringdups32
10: 32bit app|Other|/usr/bin/malloc_history32
11: 32bit app|Other|/usr/bin/leaks32
12: 32bit app|Other|/usr/bin/heap32
13: 32bit app|Other|/Users/rstehpens/Desktop/UserData/BigHonkingText
14: 32bit app|Application|/System/Library/Frameworks/QuickLook.framework
15: 32bit app|Framework|/System/Library/Input Methods/InkServer.app
16: 32bit app|Application|/System/Library/Frameworks/DrawSprocket.framework
17: 32bit app|Application|/System/Library/Frameworks/Scripting.framework
18: 32bit app|Application|/System/Library/Frameworks/DVComponentGlue.framework
19: 32bit app|Application|/System/Library/Frameworks/QuickTime.framework
20: 32bit app|Application|/System/Library/Frameworks/AppKitScripting.framework
21: 32bit app|Application|/System/Library/Frameworks/DiscRecording.framework
22: 32bit app|Application|/System/Library/PrivateFrameworks/CoreMediaAuthoring.framework
23: 32bit app|Application|/System/Library/PrivateFrameworks/vmutils.framework
24: 32bit app|Application|/System/Library/PrivateFrameworks/CoreMediaPrivate.framework
25: 32bit app|Application|/System/Library/PrivateFrameworks/CoreMediaIOServicesPrivate.framework

```

```

26: 32bit app|Application|/System/Library/PrivateFrameworks/GraphicsAppSupport.framework
27: 32bit app|Application|/System/Library/PrivateFrameworks/FWAVCPrivate.framework
28: 32bit app|Other|/System/Library/Printers/Libraries/libConverter.dylib
29: 32bit app|Other|/sbin/autodiskmount
30: 32bit app|Other|/usr/sbin/pictd
31: 32bit app|Other|/usr/lib/libnetsnmp.5.2.1.dylib
32: 32bit app|Other|/usr/bin/qtdefaults
33: 32bit app|Other|/usr/bin/qc2movie
34: 32bit app|Framework|/usr/local/bin/iHook.app
35: 32bit app|Other|/Users/rstehpens/Library/Android/sdk/build-tools/28.0.3/mipsel-linux-android-ld
36: 32bit app|Other|/Users/rstehpens/Library/Android/sdk/build-tools/28.0.3/i686-linux-android-ld
37: 32bit app|Other|/Users/rstehpens/Library/Android/sdk/build-tools/28.0.3/arm-linux-androideabi-ld
38: 32bit app|Other|/Users/rstehpens/Library/Android/sdk/tools/mksdcard
39: 32bit app|Framework|/Library/Application Support/Microsoft/Silverlight/OutOfBrowser/SLLauncher.app
40: 32bit app|Other|/Users/rstehpens/Scripts/bin/BigHonkingText
41: 32bit app|Other|/Users/rstehpens/Downloads/Safari/BigHonkingText106
42: 32bit app|Framework|/Users/rstehpens/Library/Application Support/WebEx Folder/T31_UMC/Meeting
Center.app
43: 32bit app|Other|/Users/rstehpens/Library/Application Support/WebEx Folder/T31_UMC/libopenh264-
0.6.6.0.dylib
44: 32bit app|Other|/Users/rstehpens/Library/Application Support/WebEx Folder/T31_UMC/libcrypto-
1.6.2.9.dylib
45: 32bit app|Other|/Users/rstehpens/Library/Application Support/WebEx Folder/T31_UMC/libssl-1.6.2.9.dylib
46: 32bit app|Framework|/Users/rstehpens/Library/Application Support/WebEx Folder/T31_UMC/atmsupload.app
47: 32bit app|Framework|/Users/rstehpens/Library/Application Support/WebEx Folder/T31_UMC/convertpdf.app
48: 32bit app|Other|/Users/rstehpens/Library/Application Support/WebEx Folder/T31_UMC/xml-31.0.0.1.dylib
49: 32bit app|Other|/Users/rstehpens/Library/Application Support/WebEx Folder/T31_UMC/cmccrypto-
31.11.0.1.dylib
50: 32bit app|Framework|/Users/rstehpens/Library/Application Support/WebEx
Folder/T31_UMC/asannotation2.app
Total 32bit apps found: 50

```

As can be seen, this list would be problematic as a returned value in a Custom Field, but the combination of a single response and the easily obtainable list provides a neater solution.

There are other ways the information could be stored or viewed. Scripts ran through Filesets, e.g. Activation, Preflight, etc. store the output and may be viewed via Fileset Status. These scripts have the difference in that they are usually ran just once, which may be better or worse depending upon the use case. Additionally, scripts can include API calls to alter values in Administration Custom Fields.

Each required report may have different styles of output, but with some consideration, reporting may be made easier with smarter application.

🔄Revision #2

★Created 2 July 2023 14:33:46 by Josh Levitsky

🔧Updated 16 July 2023 23:32:49 by Josh Levitsky