FileWave and Slack

Description

In some instances it is desirable to receive automated notifications from devices, perhaps based upon time or hardware conditions changing. The following is a practical example using Slack and is provided as an idea of how IT teams can retrieve information. This particular example is based upon devices reporting if the system drive is filling up, with two thresholds set.

> ⓘ **Third Party Software**
>
> Slack is a third party application. The details provided are for example only and are unsupported by FileWave.

Requirements

- Slack Account and Webhook
- Provided Fileset

The provided Fileset is for macOS, but similar could be applied to Windows.

SlackDiskUsage.fileset.zip

Information

Slack updated their Webhooks. For full compatibility the Fileset includes configuration for Legacy Webhooks, which are ignored by the newer App Webhooks.

```
\"username\": \"Helpdesk\",
\"channel\": \"#fw_messages\",
\"icon_emoji\": \":computer:\",
```

The App Webhooks are dedicated to a Channel and the name and icon are now set via Slacks's Website

Details on Legacy Webhooks:

https://api.slack.com/legacy/custom-integrations/incoming-webhooks

Directions

Slack Webhook

Slack is a useful tool for communication and the basic option is free. There is also provision for automated posting of messages to a Slack channel, which requires a generated Slack Webhook, as per their guide:

https://api.slack.com/incoming-webhooks

Once generated, the Fileset may be configured to implement this Webhook.

Fileset Configuration

The Fileset has the following components:

- The script to report back to Slack
- A pre-instal script to create a plist configuration file to provide details for the Webhook and desired thresholds for the script
- A LaunchDaemon to trigger the script
- An Activation and Pre-Uninstallation Script, to load/unload the daemon

Plist Configuration File

Once the Webhook has been generated, the Fileset should be edited to match the provided URL by replacing 'PLACE WEBHOOK HERE' with the created Webhook. This needs to be done by:

- Editing the Fileset
- Selecting 'slack_config_disk_usage.sh'
- Choose 'Get Info'
- Select 'Executable'
- Select 'Environment Variables' tab
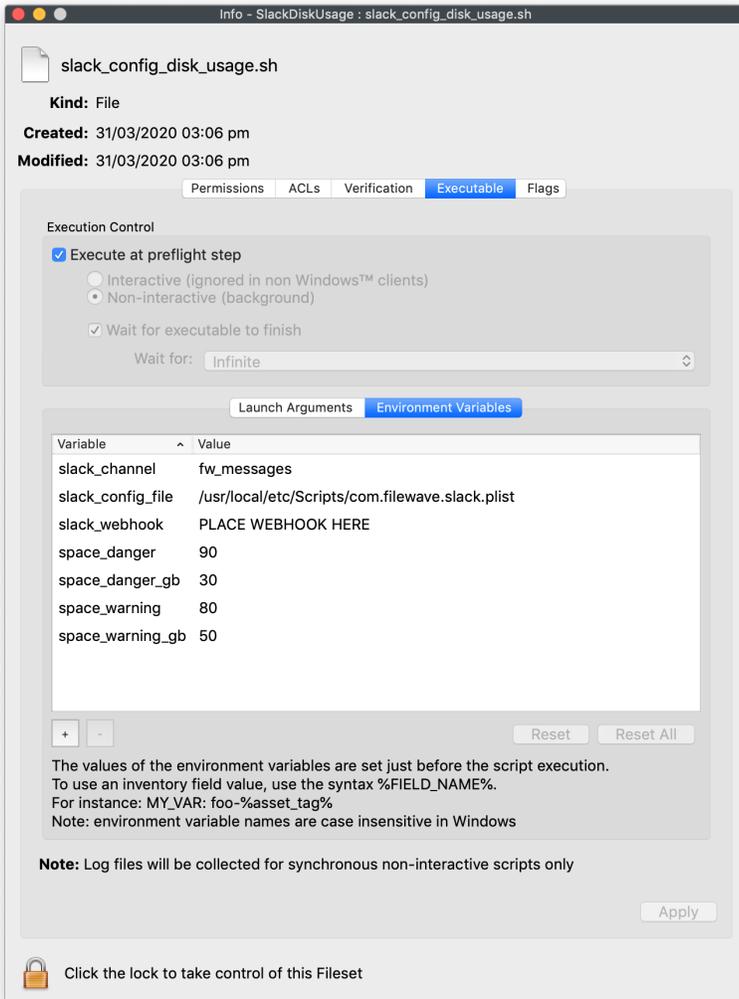
Similarly, if using Legacy Webhooks, the slack_channel variable needs to be edited to match the name of the created channel; for App Webhooks it will be ignored. In the example, this channel is called 'fw_messages'

There are also two pairs of thresholds. One pair for percentage of disk used:

- space_danger
- space_warning

and a pair for the amount of disk space available (value in GB):

- space_danger_gb
- space_warning_gb

From the example Fileset, a warning message will be delivered if either:

- The percentage used exceeds 80%, the value for space_warning
- The amount of disk space is lower than 50GB, the value of space_warning_gb

However, a more stringent message will be delivered if either:

- The percentage used exceeds 90%, the value of space_danger
- The amount of disk space is lower then 30GB, the value of space_danger_gb

These values should be edited to match desired requirement. Since the script is loaded as a launchd process, once loaded, changes to the script will only be adhered to if the launchd process is stopped and restarted. Providing an additional file, allows for values to be updated in the Fileset which the script will act upon when next triggered; without the need to restart the launchd process.

- LaunchDaemon

The LaunchDaemon 'com.filewave.slack_drive_space.plist' is set to run once a week, based upon a random day (Mon-Fri) and a random time (09:00- 18:00). The random nature of the daemon is configured by the Activation Script. Having a random day and time per device, prevents all devices attempting checking in at the same day and time. Default values have been configured, but will be overwritten during Fileset Activation.

Details around the configuration of launchd may be seen at the following resource, should different timings be desired. Note that this would involve editing the Activation Script as well.
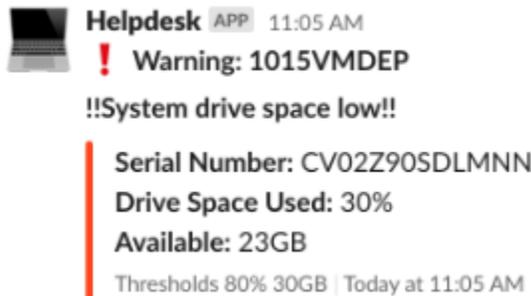
https://www.launchd.info

- Example messages

Warning - Percentage drive space outside of threshold range and available disk space between Thresholds of 20GB and 40GB:
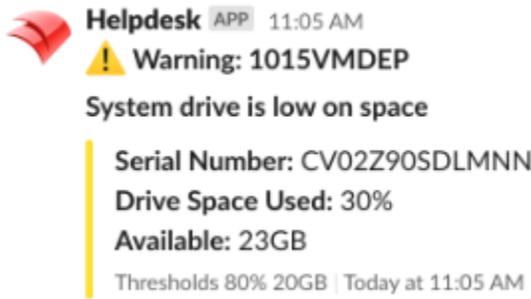
Danger - Percentage drive space outside of threshold range and available disk space lower than Thresholds of 30GB and 50GB:

For self designing messages, the above main scripts were based upon Slack's Secondary Attachments. Although Slack have only deprecated some items, indication is that all 'attachments' should not be relied upon and to consider their new Block feature.

https://api.slack.com/reference/messaging/attachments

Block Kit

Block Kit is really no different to 'attachments' and where some features are already incompatible with App Webhooks, others have been included through Block Kit. A similar look may be achieved:
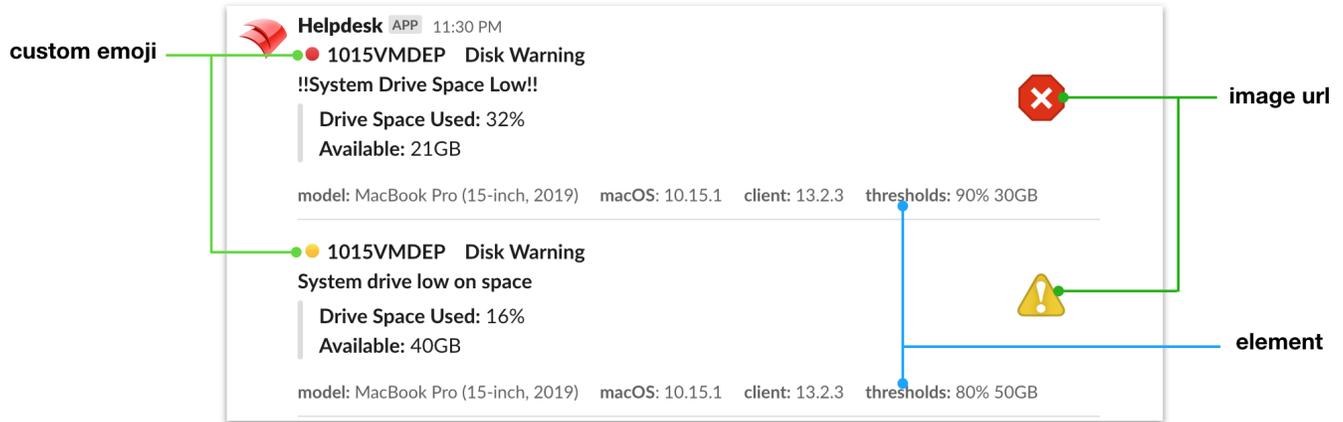
For example,

- The deprecated attachment 'icon_emoji' may no longer be updated by a message and the icon is set through the settings for the App Webhooks. However, it is possible to add an image via a URL. Images may be added to the thread and then referenced by their URL; as the above example. Alternatively, numerous App Webhooks could be created with differing icons and the relevant Webhook could be referenced in each script.
- The attachment 'footer' can be replaced with an 'elements' block
- The 'mrkdwn' for Block Quotes does not allow for colouring the vertical line
- A 'divider' line may now be included to more clearly separate messages

Further details on designing text within Blocks:

https://api.slack.com/reference/surfaces/formatting

To reference images uploaded to Slack, they must be made publicly available. A public link should be of the format:

Slack permalink_public Format

```
https://slack-files.com/{team_id}-{file_id}-{pub_secret}
```

The direct link to be referenced in a script has the format:

Direct Public Link Format

```
https://files.slack.com/files-pri/{team_id}-{file_id}/{filename}?pub_secret={pub_secret}
```

An example link may look like:

```
https://files.slack.com/files-pri/ABU1BH39Z-F0117AF5RRB/exclamation.jpg?pub_secret=5c56df27a5
```

Example json from the above. Variables have been replaced with values to allow for immediate testing:

Example Json using Block Kit

```
        json="{
             \"blocks\": [
                    {
                            \"type\": \"section\",
                            \"text\": {
                                    \"type\": \"mrkdwn\",
                                    \"text\": \":red_circle:*1015VMDEP\tDisk Warning*\n*!!System Drive
Space Low!!*\n>*Drive Space Used:* 32%\n>*Available:* 21GB\"
                            },
                            \"accessory\": {
                                    \"type\": \"image\",
                                    \"image_url\": \"https://api.slack.com/img/blocks/bkb_template_images
/beagle.png\",
                                    \"alt_text\": \"alt text for image\"
                            }
                    },
                    {
                            \"type\": \"context\",
                            \"elements\": [
                                    {
                                            \"type\": \"mrkdwn\",
                                            \"text\": \"*model:* MacBook Pro (15-inch, 2019)\"
                                    },
                                    {
                                                                        \"type\": \"mrkdwn\",
                                                                        \"text\": \"*macOS*:
10.15.1\"
                                    },
                                    {
                                            \"type\": \"mrkdwn\",
                                            \"text\": \"*client:* 13.2.3\"
                                    },
                                    {
                                            \"type\": \"mrkdwn\",
                                            \"text\": \"*thresholds:* 90% 30GB\"
                                    }
                            ]
                    },
                    {
                            \"type\": \"divider\"
                    }
             ]
        }"
```

Conclusion

The above shows how a Fileset may be delivered to devices in tandem with 3rd party tools, such that they will automatically report disk space usage at timed intervals. However, this is just an example framework and the scope for use is endless.