How-to handle devices with interactive pagination or without multi-page support

Sometimes you have devices which do not allow you to disable multi-page support. For example on a Cisco router, the system automatically paginates the output into a screen so you can read it. On Linux most commands do not have pagination built in, you use less or more to handle that.

This page is going to describe features of the phrasebook system used by opConfig to handle all the different vendors and products we support.

- Disable Pagination
- Handling Interactive Pagination
    - Add a macro to the phrasebook
    - Creating a Command Set to invoke a macro
    - Run the command set with a macro

Disable Pagination

The best option is to disable pagination, in Cisco IOS the command would be "term length 0", in linux "unset PAGER". This would be set in the opConfig phrasebook for "enable_paging" and "disable_paging".

Handling Interactive Pagination

Add a macro to the phrasebook

First we need to add a new macro to the phrasebook, in this example I am using the bash PB (/usr/local/omk/conf/phrasebooks/unix/bash/pb) which Linux uses by default. I am adding a test for using the macro to handle paging, so I am adding the following to the existing phrasebook, which will deliberately add a pager to be handled.

```
macro pager_test
    send ps -ef | more
    follow /--More--/ with ' '
```

The two quotes ‘ are wrapped around a space so '<space>’.

The full phrasebook /usr/local/omk/conf/phrasebooks/unix/bash/pb looks like:

```
prompt user
    match /[Uu]sernam: $/

prompt pass
    match /[Pp]assword( for.*)?:/

# workaround for non-fix of rt.cpan#92376, which makes the pass 'command'
# match against prompt GENERIC even though we're trying to enter PRIVILEGED
# mode - then, after generic matches, it continues to check if privileged ALSO matches...
# line 108 in /usr/share/perl5/Net/Appliance/Session/Engine.pm
prompt generic
    match /\w+@.+(\$|#) $/

prompt privileged
    match /\[?root@.+# $/

macro begin_privileged
    send sudo -Hi bash
    match pass or privileged

macro end_privileged
    send exit
    match generic

macro disconnect
    send logout

macro pager_test
    send ps -ef | more
    follow /--More--/ with ' '

macro enable_paging
    send export PAGER=less
        match generic

macro disable_paging
        send unset PAGER
        match generic
```

Creating a Command Set to invoke a macro

Now we need a command set which uses that macro, which was a feature recently added, to invoke this the command starts with "_", so in the command set this would be:

```
            "command" : "_pager_test",
```

The full command set would look like this, file /usr/local/omk/conf/command_sets.d/pager_test.json:

```
{
   "PAGER_TEST" : {
      "scheduling_info" : {
         "run_commands_on_separate_connection" : "false"
      },
      "purging_policy" : {
         "autoprotect_first_revision" : "true",
         "keep_last" : 1000,
         "purge_older_than" : 2592000
      },
      "commands" : [
         {
            "tags" : [
               "troubleshooting",
               "operatingsystem"
            ],
            "multipage" : "true",
            "command" : "_pager_test",
            "privileged" : "true"
         }
      ],
      "os_info" : {
         "os" : "/(Linux|CentOS|Ubuntu)/"
      }
   }
}
```

Run the command set with a macro

Now to run opConfig and use that macro, you could use Virtual Operator in the GUI or the following command:

```
/usr/local/omk/bin/opconfig-cli.pl nodes=odem act=run_command_sets names=PAGER_TEST debug=3
```

Voila, opConfig executes the macro which handles the interactive paging.