Scripting: Bashing IoT

scott.murphy@arrow-eye.com

2018-12-06

Bash scripting for the win...

Announcements

Linux-Ottawa
linux-ottawa.org

Welcome to OCLUG

OCLUG Mission

OCLUG promotes and supports the use of Linux in the community of Ottawa.

OCLUG History

OCLUG was founded in March, 1997. Its membership is diverse, ranging from people without computer knowledge to proverbial gurus. OCLUG was created to help promote Linux in the Ottawa area. We maintain several mailing lists, to help accomplish this goal and to provide technical support. OCLUG holds monthly general meetings.

Free computer help is provided for those who want to use a computer with Linux. This technical help is provided by volunteers and is available primarily through the Mailing list (linked on the wiki home page).

Due to some convoluted ownership issues with the domain name, we are currently operating on the linux-ottawa.org domain.

Linux-Ottawa
linux-ottawa.org

So... Welcome to Linux–Ottawa

aka OCLUG

We have a few preliminary items to go over before we start the actual talks.

Linux-Ottawa
linux-ottawa.org

First: Newcomers Welcome

- Welcome aboard to any new people.
- Feel free to ask questions, in general interruptions are welcome and will either be answered or deferred to a upcoming point.
- If you are looking for help with a project or just getting started, we have a pre-meeting for that purpose.
    - Come during the hour prior to the meeting with any queries and we'll see about helping you.
- The website is: https://linux-ottawa.org
- The wiki is: https://wiki.linux-ottawa.org (Most information is here)
- The mailing list archive is located at: https://lists.linux-ottawa.org

Linux-Ottawa
linux-ottawa.org

Venue Notes

As you know, given that you are here, we have yet another venue this month.

I'm not sure where we will be next month, as I have not found a good venue for an extended trial. Centerponte seems to be OK, but it is booked on our night for all of 2019. We'll see how this room does, but it does have the 20:30 end of night, so we may have to consider a different location. On the other hand, this does allow for an earlier trip to a pub.

I'm open to suggestions that are not very expensive.

Linux-Ottawa
linux-ottawa.org

Normal Meeting format

- 18:00 – 19:00 : New2Linux hour
- 19:00 – 21:00 : Main talk(s)
  - Typical format
    - Opening comments (like these)
    - Usually a shorter talk
    - Short break while switching speakers
    - A longer talk with questions
- 21:00 – 21:15 : Wrap up, decision on if there will be a post meeting event
- 21:30 – Done : Beer Sig (social event)

As you can see, this may need adjustment due to venue restrictions.

Linux-Ottawa
linux-ottawa.org

We now return to our regularly scheduled talk

Linux-Ottawa
linux-ottawa.org

Our target IoT device

How many of you have WeMo units?

Linux-Ottawa
linux-ottawa.org

Belkin WeMo devices

Belkin produces a set of IoT devices for home automation.

The popular ones are wall plugs and light switches. They have added a power utilization wall plug, a dimmer unit, as well as a crock pot unit. They also updated the wall unit from the one I have to a slimmer unit.

I have one of the original plugs here to pass around.

Linux-Ottawa
linux-ottawa.org

Control

The short configuration method is:

- Plug the unit in.
- Join the newly visible access point it creates via the app.
- Provide credentials for your wireless network through the app.
- It will join your network and also attempt to configure remote access. This you can disable after the fact.
- You can now controll it via the app.

Seems simple enough, right?

Linux-Ottawa
linux-ottawa.org

Not quite

The app is only available for IOS or Android. You can't control them from your desktop.

That doesn't sound too bad now does it?

What if you...

- want to use them with your own home automation?
- want to gather the state of all your switches and show them?
- just want to turn on or off something from your desktop system?
- etc.

You can!

It is remarkably easy to do so, as long as you don't mind a bit of scripting.

With a little bash and curl, you can get the devices to perform their tasks without having to get your phone or tablet out.

Of course, this can only happen after they have been configured with the app.

(Other methods are possible, but that is something we can save for a future talk).

A WeMo Script

We will examine the script I have been using. I had a simpler one, but I ran across this one.

```
#!/bin/bash

USAGE="wemo.sh [IP|network] [on|off|getstate|getsignal|getname|find]"

if [[ $# -ne 2 ]] ; then
  echo "$USAGE"
  exit 0
fi

IP="$1"
CMD="$2"


# Wemo devices typically listen on 49152 or 49153, we need to test for that

getPort () {
  PORTTEST=$(curl -s "$IP":49152 | grep "404")

  if [ "$PORTTEST" = "" ] ; then
    PORT=49153
  else
    PORT=49152
  fi
}
```

Linux-Ottawa
linux-ottawa.org

Script Page 2

```
turnON () {
getPort
curl -0 -A '' -X POST -H 'Accept: ' -H \
  'Content-type: text/xml; charset="utf-8"' -H \
  "SOAPACTION: \"urn:Belkin:service:basicevent:1#SetBinaryState\"" --data \
   '<?xml version="1.0" encoding="utf-8"?>
    <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
     s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <s:Body><u:SetBinaryState xmlns:u="urn:Belkin:service:basicevent:1">
    <BinaryState>1</BinaryState></u:SetBinaryState></s:Body>
    </s:Envelope>' \
  -s http://$IP:$PORT/upnp/control/basicevent1 | grep "<BinaryState" | cut -d">" \
  -f2 | cut -d "<" -f1 | sed 's/0/OFF/g' | sed 's/1/ON/g'
}
```

Linux-Ottawa
linux-ottawa.org

Script Page 3

```
turnOFF () {
getPort
curl -0 -A '' -X POST -H 'Accept: ' -H \
  'Content-type: text/xml; charset="utf-8"' -H \
  "SOAPACTION: \"urn:Belkin:service:basicevent:1#SetBinaryState\"" --data \
   '<?xml version="1.0" encoding="utf-8"?>
    <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
     s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <s:Body><u:SetBinaryState xmlns:u="urn:Belkin:service:basicevent:1">
    <BinaryState>0</BinaryState></u:SetBinaryState></s:Body>
    </s:Envelope>' \
  -s http://$IP:$PORT/upnp/control/basicevent1 | grep "<BinaryState"  | cut -d">" \
  -f2 | cut -d "<" -f1 | sed 's/0/OFF/g' | sed 's/1/ON/g'
}
```

Linux-Ottawa
linux-ottawa.org

Script Page 4

```
getName () {
getPort
curl -0 -A '' -X POST -H 'Accept: ' -H \
  'Content-type: text/xml; charset="utf-8"' -H \
  "SOAPACTION: \"urn:Belkin:service:basicevent:1#GetFriendlyName\"" --data \
  '<?xml version="1.0" encoding="utf-8"?>
   <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <s:Body><u:GetFriendlyName xmlns:u="urn:Belkin:service:basicevent:1">
   <FriendlyName></FriendlyName></u:GetFriendlyName></s:Body>
   </s:Envelope>' \
  -s http://$IP:$PORT/upnp/control/basicevent1 |     grep "<FriendlyName" | cut \
  -d">" -f2 | cut -d "<" -f1
}
```

Linux-Ottawa
linux-ottawa.org

Script Page 5

```
getState () {
  getPort
  curl -0 -A '' -X POST -H 'Accept: ' -H \
    'Content-type: text/xml; charset="utf-8"' -H \
    "SOAPACTION: \"urn:Belkin:service:basicevent:1#GetBinaryState\"" --data \
     '<?xml version="1.0" encoding="utf-8"?><s:Envelope
      xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
      s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
     <s:Body><u:GetBinaryState xmlns:u="urn:Belkin:service:basicevent:1">
     <BinaryState>1</BinaryState></u:GetBinaryState></s:Body></s:Envelope>' \
    -s http://$IP:$PORT/upnp/control/basicevent1 | grep "<BinaryState"  | cut \
    -d">" -f2 | cut -d "<" -f1 | sed 's/0/OFF/g' | sed 's/1/ON/g'
}
```

Linux-Ottawa
linux-ottawa.org

Script Page 6

```
getSigStrength () {
  getPort
  curl -0 -A '' -X POST -H 'Accept: ' -H \
    'Content-type: text/xml; charset="utf-8"' -H \
    "SOAPACTION: \"urn:Belkin:service:basicevent:1#GetSignalStrength\""  --data \
    '<?xml version="1.0" encoding="utf-8"?>
     <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
      s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
     <s:Body><u:GetSignalStrength xmlns:u="urn:Belkin:service:basicevent:1">
     <GetSignalStrength>0</GetSignalStrength></u:GetSignalStrength></s:Body></s:Envelope>' \
    -s http://$IP:$PORT/upnp/control/basicevent1 | grep "<SignalStrength"  | cut \
    -d">" -f2 | cut -d "<" -f1
}
```

Linux-Ottawa
linux-ottawa.org

Script Page 7

```bash
findDevices () {
  which nmap > /dev/null || \
    ( echo "nmap is not installed and it's needed for this option" ; exit 1 )

  echo "$IP" | grep -q '\*'
  if [[ $? -ne 0 ]] ; then
    echo "You need to provide a network for this option. It should be similar to 1.1.1.*"
    exit 1
  fi

  echo "Finding. This may take a while..."
  DEVICES=$(nmap -p 49153 --open "$IP" | grep 'scan report for' | awk '{print $5}')
  if [ "$DEVICES" ] ; then
    echo "IP           NAME"
    echo "---------------------"
    for DEVICE in $DEVICES ; do
      IP="$DEVICE"
      echo "$DEVICE $(getName )"
    done
  else
    echo "Did not find any devices"
  fi
  echo ""
}
```

Linux-Ottawa
linux-ottawa.org

Script page 8

```
case "$CMD" in
  on)         turnON ;;
  off)        turnOFF ;;
  getstate)   getState ;;
  getsignal)  getSigStrength ;;
  getname)    getName ;;
  find)       findDevices ;;
  *)          echo "Unknown option" && exit 1 ;;
esac

exit 0
```

Linux-Ottawa
linux-ottawa.org

Let's examine a script run

As you can see below, if you happen to know how many are out there, you can run the find command a few times to get a proper list. I think they need to be strobed a few times to make them responsive.

```
 $ wemo 192.168.1.* find
Finding. This may take a while...
IP           NAME
---------------------
192.168.1.81 BackDeck

$ wemo 192.168.1.* find
Finding. This may take a while...
IP           NAME
---------------------
192.168.1.81 BackDeck
192.168.1.84 Front Lamp

$ wemo 192.168.1.* find
Finding. This may take a while...
IP           NAME
---------------------
192.168.1.80 Front Light
192.168.1.81 BackDeck
192.168.1.82 Guest Room
192.168.1.84 Front Lamp
```

Linux-Ottawa
linux-ottawa.org

Script run (continued)

```
$ wemo 192.168.1.84 getstate
ON
$ wemo 192.168.1.81 getstate
OFF
$ wemo 192.168.1.81 on
ON
$ wemo 192.168.1.81 getstate
ON
$ wemo 192.168.1.81 off
OFF
$ wemo 192.168.1.81 getstate
OFF
$ wemo 192.168.1.81 getsignal
89
$ for i in 80 81 82 84; do
> wemo 192.168.1.${i} getsignal
> done
76
89
91
99
$
```

Linux-Ottawa
linux-ottawa.org

Discussion

Linux-Ottawa
linux-ottawa.org

Resources & References

- The Garfield strip is from May 1982
- Belkin WeMo Units
- Raspberry Pi Home Automation
- Source for the code presented here

Linux-Ottawa
linux-ottawa.org