Bluetooth support in Android Apps

There are some serious limitations in access to bluetooth in the AAS. Here we document some known solutions. The original source for much of this information is: the forum

- Petefoth SailfishOS: Bluetooth in Android AppSupport
- rdomschk notes

Petefoth SailfishOS: Bluetooth in Android AppSupport

Pete documents his solutions to allowing android apps (the lxc container) bluetooth access in his repo:

https://codeberg.org/petefoths-projects/SfOS-Bluetooth-in-AAS/

or

https://codeberg.org/poetaster/SfOS-Bluetooth-in-AAS

Implementation of a method in SailfishOS (SfOS) to allow user to switch Bluetooth (BT) support between SfOS and Android AppSupport (AAS).

Current status

As stated in this forum post

> I have something similar working using `qCommand` app: switching BT to AAS is working fine; switching it back to SfOS is still ‘work-in-progress’ - sometimes it works, sometimes not.

Current status of my qCommand stuff. (For both commands I've commented out the `if systemctl is-active` condition, because it's not working as expected)

`qc_BT-to-Android` : mostly works. I can connect to and sync my watch in the Garmin Connect Android app, and connect to and control my hearing aids in the BeMore Android app.

Code

```
if [[ ! $(whoami) = "root" ]]; then
echo "must be run as root" >&2
exit 1
fi

# start by setting up the config file,even if it exists already
echo "[Config]" > /opt/appsupport/etc/appsupport.conf.d/99-passthrough.conf
echo
"Proxies=android.hardware.bluetooth@1.1::IBluetoothHci,android.hardware.bluetooth@1.0::IBluetoothHci/d
efault" >> /opt/appsupport/etc/appsupport.conf.d/99-passthrough.conf

# Stop the BT services in SfOS
systemctl stop bluetooth
systemctl stop bluebinder

# Start AAS if it isn't running already:
# returns 3 if inactive
# 0 if active
if [[ ! $(systemctl is-active --quiet appsupport@defaultuser.service) = 0 ]]; then
  echo "starting appsupport"
  systemctl reload-or-restart appsupport@defaultuser
fi

# Start the Android BT process
appsupport-attach pm enable com.android.bluetooth
```

qc-BT-to-SfOS: mostly works. Occasionally gives an error when starting `bluebinder` times out. (I haven't checked the status of BT in SfOS if the error occurs. Nor have I looked into detail in "journalctl -xe". That, and sorting out the operations conditional on `systemctl is-active` are the next jobs :slight_smile:

Code

```
if [[ ! $(whoami) = "root" ]]; then
  echo "must be run as root" >&2
  exit 1
fi
```

```
# If AAS is running, stop / disable its BT  process
#if [[ $(systemctl is-active --quiet appsupport@defaultuser.service) = 0 ]]; then
echo "DisablingBT in Android"
  appsupport-attach pm disable com.android.bluetooth
echo "Stopping AAS"
  systemctl stop appsupport@defaultuser
#fi

# Start the BT services in SfOS
echo "Starting bluebinder"
systemctl start bluebinder
echo "Starting AAS"
systemctl start appsupport@defaultuser
echo "Restarting bluetooth"
systemctl  restart bluetooth
```

Occasional error

```
Job for bluebinder.service failed because a timeout was exceeded.
See "systemctl status bluebinder.service" and "journalctl -xe" for details.`
```

Background

The means to allow BT access for Android apps running under AAS is described in [this post in the SfOS communtiy forum](). As described in that post and replies, BT can be available EITHER to SfOS OR to AAS, but not to both. The aim of this project is to proved a means for users to switch BT availability between SfOS and AAS.

Use cases

My use cases:

1. Garmin Smartwatch: download golf course data from Garmin Connect Android app running on the phone. As far as I know, this is the only way to achieve this: it is not possible to download Golf Course data from PC to watch. (Note this is a multi-sport watch and golf courses ***are not*** pre-loaded: they must be loaded from the phone. For dedicated Golf watches, with courses preloaded, course updates ***can*** be downloaded from PC or

Web)
2. BeMore Hearing Aid control app. Proprietary Android app that allows fine-grained control of hearing aids - volume, noise filters, favourite settings

These operations are both short in duration: the workflow would be

1. Enable BT in AAS
2. Perform the operation
3. Disable BT in AAS and re-enable it in SfOS

The proposed solution would also meet the needs of users who wish to have BT access available to AAS

1. all the time: never available to SfOS
2. for as long as AAS is running

Implementation

I think this solution can be implemented by

1. Creating two shell scripts:

- `bt-to-aas.sh` enables BT in AAS
- `bt-to-sfos.sh` enables BT in SfOS

2. Implementing a mechanism to call thses scritps from the SfOS UI

Enable BT in AAS (`bt-to-aas.sh`)

1. stop AAS if it is running - (is this doable from a script?). Or leave it running while we do steps 2 & 3, and ***restart*** ASS in step 4
2. stop bluetooth and bluebinder services
3. add (or uncomment) this line (using `sed`?) to `/opt/appsupport/etc/appsupport.conf.d/10-hybris.conf` `Proxies=android.hardware.bluetooth@1.1::IBluetoothHci,android.hardware.bluetooth@1.0::IBluetoothHci/default`
4. start (or restart) AAS - (is this doable from a script, rather than from SFOS settings menu)

Enable BT in SfOS (`bt-to-sfos.sh`) I am less sure about what is needed here. A 'quick and dirty solution might be to edit the `10-hybris.conf` file then reboot the phone. I would prefer a slightly more elegant and less destructive solution :slight_smile:

1. do I need to stop bluetooth and bluebinder services again, or are they still not still not running since being stopped in the `bt-to-aas.sh` script?

2. stop AAS
    1. is this necessary or can we just restart it once we have changed the `10-hybris.conf` file?
    2. is this: do-able from a script ?
3. OR disable AAS Android Bluetooth package, but leave AAS running
4. remove (or comment) this line(using `sed`?) to `/opt/appsupport/etc/appsupport.conf.d/10-hybris.conf`
   `Proxies=android.hardware.bluetooth@1.1::IBluetoothHci,android.hardware.bluetooth@1.0::IBluetoothHci/default`
5. restart bluetooth bluebinder and services
6. restart AAS (optional: we could leave this to be done manually)

Problems to solve

☐ How to stop

☐ AAS if running: normally do this from from SFOS settings menu. Can we do it via a script?

☑ Stop bluetooth and bluebinder services: Solved in forum post

> “ `systemctl stop bluetooth`
>
> `systemctl stop bluebinder`

☐ How to change the contents of `10-hybris.conf`?

- answer probably involves `sed`. I need to experiment to find out how to
    1. add the needed line in `10-hybris.conf`
    2. remove the added line in `10-hybris.conf`

☐ Better approach: don't touch `10-hybris.conf`, use a different file.

- Delete it to enable BT in SfOS
- Create it to enable BT in AAS

> “ `echo "Proxies=android.hardware.bluetooth@1.1::IBluetoothHci,android.hardware.bluetooth@1.0::IBluetoothHci/default" > /opt/appsupport/etc/appsupport.conf.d/99-passthrough.conf`

☐

Where in the device filesystem will the scripts live? And how will they get there?

☑

How to invoke the scripts so that they have root / `devel-su` access to perform privileged operations (e.g. starting / stopping `systemd` services)?

- answered in SfOS forum

> “ You can run scripts as root by using qCommand app or Sailcron (at selected time). Scripts which don’t need root are possible to run from launcher by creating *.desktop file. Scripts which don’t need root are possible to run from launcher by creating *.desktop file.

Both our scripts will need root I think

rdomschk notes

Directly from the forum:

https://forum.sailfishos.org/t/potential-bluetooth-support-in-the-android-app-support-aas/9686/206

I’m a fan of ShellEx and Sudo. So I create in ShellEx two command blocks into one command. Of course, I modified the AAS file /opt/appsupport/etc/appsupport.conf.d/10-hybris.conf beforehand. I found out, this change can even remain permanent, because we can disable/enable the BT package from AAS.

Now my way of working:

1. Stop the AppSupport and open with root@tIDE editor follow file and add one line (in my case line 50 in SFOS 5.0.0.67)

```
/opt/appsupport/etc/appsupport.conf.d/10-hybris.conf
Proxies=android.hardware.camera.provider@2.4::ICameraProvider/legacy/0
#Add this line below line 49
Proxies=android.hardware.bluetooth@1.1::IBluetoothHci,android.hardware.bluetooth@1.0::IBluetoothHci/default
```

2. Start the AppSupport
3. Than create follow commands in ShellEx or in Situations!

Name 1: “BT AppSupport On”

```
sudo systemctl stop bluetooth
sudo systemctl stop bluebinder
sudo systemctl start appsupport@defaultuser
sudo appsupport-attach pm enable com.android.bluetooth
```

Name 2: “BT AppSupport Off”

```
sudo appsupport-attach pm disable com.android.bluetooth
sudo systemctl stop appsupport@defaultuser
sudo systemctl start bluebinder
sudo systemctl start appsupport@defaultuser
```

```
sudo systemctl restart bluetooth
```

4. Open Android settings and go to the second menu “Bluetooth” and see the BT devices
5. On X10III it needs time and I had to give the BT a name first time (e.g. Xperia10III). If the phone not found any BT devices under AAS its help to connect the phone form another phone via BT one time. On X10 it was not a problem but the X10III need this procedure.
6. With the Command “BT AppSupport Off” you can bring back your Phone to normal BT under SFOS.
   Don’t be surprised. The first command block runs quite quickly, provided AAS is already running. The second block takes a while because AAS has to be stopped and started. So, the gyroscope in ShellEx is normal, and only at the end does a restart of the BT complete everything.

Only as additional information - to sudo settings:

1. Install Sudo: devel-su + Password ; pkcon install sudo
2. Create a file “my-personal-settings” under `/etc/sudoers.d/`
3. Fill the file with follow lines:

```
defaultuser ALL = (ALL) NOPASSWD: ALL
defaultuser ALL = NOPASSWD: /usr/bin/dbus-send
defaultuser ALL = NOPASSWD: /usr/bin/systemctl
defaultuser ALL = NOPASSWD: /usr/bin/killall
defaultuser ALL = NOPASSWD: /usr/bin/kill
```

In Situations you can also bring the command blocks and it’s possible to create only one Situation because you can run the commands at start and stop from this Situation. So now enough text. slight_smile:

Image not found or type unknown

Postscript:

Hello - I had also trouble with the 2. part back to normal bluebinder service. The solution was to start AAS after start bluebinder and restart Bluetooth as last command. So you win time because the start from AAS and the bluebinder service is on if you start Bluetooth. You can also add the resart command on the end from your procedure.
With situations work this also good. I tested it for some minutes…