

User guides

Begrippenlijst

Lineage OS = Android ROM. Lineage wordt gebruikt als tussen laag tussen Sailfish en de hardware. Deze laag is HAL (hardware abstraction layer).

Sailfish OS = Linux besturingssysteem voor telefoons. Dit is het daadwerkelijke OS dat je ziet.

Build server user guide

Om het compilatie proces efficiënter te maken is gekozen een build server op te zetten. Door gebruik te maken van deze server kan het compilatie proces versneld worden. De server kan op de achtergrond draaien. Iedereen met toegang kan meekijken met de server en hierop compileren, meedenken en veranderingen aanbrengen.

De build server draait dezelfde ontwikkel omgeving als de ontwikkelaars. Door middel van de SSH kan worden ingelogd in de build server. Met de applicatie screen kan vervolgens in dezelfde terminal instantie worden gewerkt. Wanneer verbonden werkt de server hetzelfde als de lokale ontwikkelomgeving.

Build server verbinding

1. [Maak een SSH key aan als je deze nog niet hebt.](#)
2. SSH Key lokaal opvragen.
`cat ~/.ssh/<ssh_key>.pub`
3. Deze key moet worden toegevoegd aan de build server (in ~/.ssh/authorized_keys).
4. De server kan daarna worden gejoined door gebruik te maken van het onderstaande commando. Het wachtwoord is 'sfos'.
`ssh -p 2222 sfos@204.168.189.183`
5. Werken wordt gedaan in een [GNU screen](#).
`screen -x build`
6. In deze omgeving werkt alles hetzelfde als de lokale build environment.
7. Gebruik het onderstaande commando om de screen te verlaten.
`Ctrl + A ; D.`
8. Om de ssh verbinding te verlaten.
`exit`

Nieuwe build environment handmatig opzetten

Ten eerste is het belangrijk om even de versie van SailfishOS die je wil bouwen vast te stellen. Deze guide is geschreven voor versie 4.5.0.24.

1. Zorg dat je verbonden bent met de build server.
2. Verwijder alle potentiële overgebleven bestanden als deze nog aanwezig zijn.
 - a. Verwijder de map /srv

```
sudo chattr -i /srv/sailfishos/sdks/sfossdk/var/log/lastlog
```

```
sudo rm -rf /srv
```
 - b. Verwijder overgebleven bestanden in je home directory(~).

```
rm -rf hadk .hadk.env bin .mersdkubu.profile .scratchbox2 .mb2
```
3. Volg de [getting started guide](#).
4. Bij errors, bekijk het hoofdstuk “Troubleshooting”.

Rebuilden na veranderingen

Als je de build server opgezet hebt, maar je hebt veranderingen gemaakt aan Sailfish OS of de kernel, kun je op de volgende manier de images rebuilden.

1. Volg de installation guide vanaf het hoofdstuk “Building Boot and Recovery Image”.
2. Skip hierbij de volgende delen
 - a. Bij het hoofdstuk “Install SDK-targets and sdk-tooling” hoef je de sdk-assistant commando’s niet uit te voeren. Dit is de eerste keer al gedaan. Dit hoeft alleen bij het updaten van de sdk-assistant.
 - b. Bij het hoofdstuk “Cloning the standard configurations” hoef je de git clones allemaal niet uit te voeren. Deze bestaan allemaal al.
 - c. Bij building the middleware packages, kun je de git clones skippen. De build-packages commando’s kunnen wel handig zijn, als hier veranderingen in zijn gedaan.
3. Na het uitvoeren van Package SailfishOS, zijn er nieuwe images beschikbaar. De terminal laat de paths naar de output files zien.

Troubleshooting

Bij het bouwen van de image en het opzetten van de build environment kunnen een aantal dingen misgaan. Dit kan gebeuren doordat er een snelheidsfout wordt gemaakt, of door andere dingen die (nog) niet goed ingesteld zijn. Hieronder worden elke error die wij zijn tegen gekomen beschreven, met de context en bijhorende oplossing.

Problemen met sfossdk

Met de sfossdk (Sailfish SDK) omgeving kunnen er een aantal problemen optreden.

1. Dubbele 'Sailfish SDK' tekst in de sfossdk omgeving.

- a. Dit probleem kan verdere problemen veroorzaken. Dit kun je oplossen door in je home directory naar de .bashrc te kijken. Daar staat aan het eind dubbele code. Dit gebeurt als je meerdere keren het begin stuk in de installation guide uitvoert. Haal simpelweg de dubbele functies/code weg in je .bashrc.

Problemen met ubu-chroot

Er kunnen zich een aantal problemen voordoen wanneer je probeert te chrooten naar de HADK omgeving.

1. Missing ubu-chroot.

- a. Als de terminal aangeeft dat je ubu-chroot niet bestaat, moet je zorgen dat je de stap 'Install Tools' eerst hebt uitgevoerd. Dat is het runnen van het zypper installatie commando: `<zypper install <packages>>`

2. (...) is likely not an android environment.

- a. Dit komt, omdat je het ubu-chroot environment niet uitvoert in de Sailfish SDK omgeving. Gebruik eerst het commando `'sfosdk'`. Hierna kun je het ubu-chroot commando gebruiken.

Niet-gevonden bestanden / executables

Ook kan het voorkomen dat er bestanden of executables niet gevonden worden.

Dit zijn vaak kleine fixes. Controleer de overige hoofdstukken als de oplossing er hier niet bij staat.

1. \$HOME/.profile does not exist

- a. Bij het uitvoeren van het install-tools kopje, kan deze error voorkomen. Als dit bestand niet bestaat, is het niet nodig deze te sourcen. Deze error is dan ook te negeren. Alles wat nodig is staat al in de .bashrc.

2. Repo: command not found

- a. Bij het uitvoeren van het repo commando in het 'syncing the android base' kopje, kan het zijn dat deze niet gevonden wordt. Dit komt omdat je dit programma download naar je ~/bin folder. Deze staat in je home directory en staat niet standaard in je path. Voeg dus deze folder toe aan je path:
`'export PATH=~/.bin:$PATH'`

Problemen bij het packagen van de fingerprint daemon

Bij het packagen van de fingerprint daemon, kan het zijn dat het build-packages commando failt. Dit is simpel op te lossen door eerst verder te gaan met het bouwen van SailfishOS. Deze build zal ook failen, maar hierna kan de build van de fingerprint daemon wel succesvol uitgevoerd worden. Dit komt, omdat er een dependency is voor de fingerprint daemon die nog niet gebouwd is. Door verder te gaan met het bouwen van de SailfishOS packages, worden deze dependencies gebouwd.

Als de fingerprint daemon succesvol gebouwd is, kan je verder gaan met het packagen van SailfishOS, wat nu wel goed zou moeten gaan.

File conflicts

In de laatste stap ‘Packaging SailfishOS’ kunnen zich file conflicts voor doen. Dit komt omdat specifieke Fairphone configuraties clashen met generieke configuraties. Deze generieke configs kan je vervolgens gewoon verwijderen.

1. **Voor ofono-configs-binder**

```
$ SFOSSDK
```

```
sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -R -msdk-install zypper remove ofono-configs-binder
```

2. **Voor bluez5-configs-mer**

```
$ SFOSSDK
```

```
sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -R -msdk-install zypper remove bluez5-configs-mer
```