Contents

Naming: 1
Features for unpack/repack 1
Auto mode: 2
Unpack mode: 2
Repack mode: 2
No-mode: 2
Additional options: 3
How it works? 3
Change default settings: 4
Some example uses: 4

Note: To use SAT you must run it with root access (needed to mount files).

Naming:

SPARSE_IMG - .img file that you want to unpack
RAW_IMG – it is a unpacked .img file, it can be mount and modify on M_DIR
M_DIR – mount point directory, when RAW_IMG is mounted (default: /mnt/sat/loop)
F_SPARSE_IMG - final image file (after repack)
(you can pass full path or just name to the above values)

Features for unpack/repack

- possible to provide custom names/directories for RAW_IMG, F_SPARSE_IMG, M_DIR
- SPARSE_IMG is a base name for RAW_IMG and F_SPARSE_IMG
- finding alternative names/dirs if busy (RAW_IMG and F_SPARSE_IMG)
- can create new mount point directories if default/provided M_DIR is busy
- each mounted RAW_IMGs informations are stored, so no need to specify M_DIRs
- prompt warning before overwriting files
- script checks sense of used options eg. -a with -u
- unmount and remove all M_DIRs created by program and stored informations about it (-c option)
- colored UI
- print list of mounted RAW_IMGs with corresponding M_DIRs (-ml option)
- set up avalaible free space of mounted RAW_DIR
- delete source SPARSE_IMG after unpack it (useful for Android phones with low free memory)
- and many more . . .

Auto mode:

Unpack SPARSE_IMG, then repack (it makes sense when using some additional options)
Usage:

```
./sat.sh -a SPARSE_IMG F_SPARSE_IMG
```

or:

```
./sat.sh -a SPARSE_IMG
```

(F_SPARSE_IMG name will be generated automatically)

Unpack mode:

unpack SPARSE_IMG to RAW_IMG, then mount to not busy M_DIR
Usage:

```
./sat.sh -u SPARSE_IMG RAW_IMG
```

or:

```
./sat.sh -u SPARSE_IMG
```

(RAW_IMG name will be generated automatically)

Repack mode:

Usage:

```
./sat.sh -r RAW_IMG SPARSE_IMG
```

Repack RAW_IMG to SPARSE_IMG
or:

```
./sat.sh -r RAW_IMG
```

Repack RAW_IMG to SPARSE_IMG (automatically generated name)
or:

```
./sat.sh -r
```

will repack last created RAW_IMG to SPARSE_IMG (automatically generated name)

No-mode:

This mode is trigerred: - after each unpack - before each repack - while auto mode is used - when none of previos modes (-u,-r,-a) are used

It can be uses with some additional options. It operates on last created RAW_IMG (can be changed by -m option)

Additional options:

-m M_DIR

change mountpoint directory to M_DIR

-o

overwrite all files (if you don't want overwrite files, script will create new names/dirs)

-dm

(for unpack) disable automatic RAW_IMG mounting

-c

(for no-mode) umount and delete all M_DIR's

-vndk X

where X is one of numbers: 26, 27, 28, 29 (you can pass how many numbers do you want). It will automatically delete corresponding vndk folders: * /lib/vndk-X, * /lib/vndk-sp-X, * /lib64/vndk-X, * /lib64/vndk-sp-X

-ab2a

converts system from AB architecture to A-only.

-debug

allow to display errors (by defualt some errors and messages are not displayed)

-ml

prints list of mounted M_DIR's

-dc

disable colorful UI

-resizeoff

disable resize2fs -M RAW_IMG command before repacking

-free SIZE

(unpack option) set up free space for unpacked RAW_DIR, where SIZE is values in MB

Note: when RAW_DIR will be mounted the available free space will be a little lower than SIZE

-ds

(unpack option) delete source SPARSE_IMG after unpack it

-update

just update the script (your changes in default.conf will be kept)

How it works?

SAT basically follows with below proccess:

Unpack mode:

```
simg2img SPARSE_IMG RAW_IMG
mkdir -p M_DIR
mount RAW_DIR M_DIR
```

No-mode

Here SAT makes some changes in M_DIR (for example when -vndk, -ab2a etc. options is used).

Repack-mode

```
umount M_DIR
e2fsck -fy RAW_IMG
resize2fs -M RAW_IMG
img2simg RAW_IMG F_SPARSE_IMG
```

Auto-mode
Just perform all whole process.

Change default settings:

Some of default settings can be changed using „default.conf” text file. List of available values below:
enable_color=true/false
when set to true, the tool will turn on colorful UI
use_tool_binaries=true/false
when set to true, the tool uses tool's binaries. Otherwise it will use system packages.
do_resize=true/false
when set to true, resize2fs -M RAW_IMG command is always called before repacking
M_DIR=PATH
change default M_DIR directory to PATH
m_mount_dir=PATH
change directory, where tool creates new M_DIR's to PATH

Some example uses:

I. reduce size of Generic System Image (GSI)
You must know, which vndk folders you can delete. It depends of your device's vendor. If you don't know, then check vndk version using Treble Info app (available in Google Play). To reduce size of .img file delete unnecessary vndk folders by running:

```
./sat.sh -a SPARSE_IMG -vndk 26 27 29
```

(in that case you will remove all folders related to 26, 27, 29 vndk version)

It will unpack SPARSE_IMG to RAW_DIR, then mount it in M_DIR, deleting vndk folders, resize RAW_DIR and repack to F_SPARSE_IMG.

II. converting system from AB architecture to A-only.

```
./sat.sh -a SPARSE_IMG -ab2a
```

Note : If you want you can do 1. and 2. operation by running:

```
./sat.sh -a SPARSE_IMG -vndk 26 27 29 -ab2a
```

Tip : When you using -a option, it is nice to use it with -o (if you sure that tool won't overwrite important files)

III. Unpack SPARSE_IMG and mount, do something with files, repack it

```
./sat.sh -u SPARSE_IMG
(do something with files in M_DIR)
./sat.sh -r
```

IV. Unpack SPARSE_IMG, delete it, set up 300 MB of free space for RAW_DIR and mount it on M_DIR

```
./sat.sh -u SPARSE_IMG -ds -free 300
(now you can add some files in M_DIR)
```