

# OTA Package Tools

The [ota\\_from\\_target\\_files](https://android.googlesource.com/platform/build/+/master/tools/releasetools/ota_from_target_files) ([https://android.googlesource.com/platform/build/+/master/tools/releasetools/ota\\_from\\_target\\_files](https://android.googlesource.com/platform/build/+/master/tools/releasetools/ota_from_target_files)) tool provided in `build/tools/releasetools` can build two types of package: *full* and *incremental*. The tool takes the *target-files* .zip file produced by the Android build system as input.

## Full updates

---

A *full* update is one where the entire final state of the device (system, boot, and recovery partitions) is contained in the package. As long as the device is capable of receiving the package and booting the recovery system, the package can install the desired build regardless of the current state of the device.

Example: Using the release tools to build a full update for the hypothetical **tardis** device:

```
# first, build the target-files .zip
$ . build/envsetup.sh && lunch tardis-eng
$ mkdir dist_output
$ make dist DIST_DIR=dist_output
```

The `target-files` .zip contains everything needed to construct OTA packages.

```
$ ./build/tools/releasetools/ota_from_target_files dist_output/tardis-target_files.zip ota_update.zip
```

The `ota_update.zip` is now ready to be sent to test devices (everything is signed with the test key). For user devices, generate and use your own private keys as detailed in [Signing builds for release](https://source.android.com/devices/tech/ota/sign_builds.html) ([https://source.android.com/devices/tech/ota/sign\\_builds.html](https://source.android.com/devices/tech/ota/sign_builds.html)).

## Incremental updates

---

An *incremental* update contains a set of binary patches to be applied to the data already on the device. This can result in considerably smaller update packages:

- Files that have not changed don't need to be included.
- Files that have changed are often very similar to their previous versions, so the package need only contain an encoding of the differences between the two files.

You can install the incremental update package only on a device that has the old or source build used when constructing the package. To build an incremental update, you need the `target_files` .zip from the previous build (the one you want to update *from*) as well as the `target_files` .zip from the new build.

```
$ ./build/tools/releasetools/ota_from_target_files -i PREVIOUS-tardis-target_files.zip dist_output/tardis-target_file
```

This build is very similar to the previous build, and the incremental update package is much smaller than the corresponding full update (about 1 MB instead of 60 MB).

**Note:** To generate a [block-based OTA](https://source.android.com/devices/tech/ota/block.html) (<https://source.android.com/devices/tech/ota/block.html>) for subsequent updates, pass the `--block` option to `ota_from_target_files`.

Distribute an incremental package only to devices running exactly the same previous build used as the incremental package's starting point. Attempting to install the incremental package on a device with some other build results in the recovery error icon. Rebooting the device at this point returns the user to the old system; the package verifies the previous state of all the files it updates before touching them, so the device should not be left in a half upgraded state if this occurs.

## Update packages

---

An update package (`ota_update.zip`, `incremental_ota_update.zip`) is a .zip file that contains the executable binary `META-INF/com/google/android/update-binary`. After verifying the signature on the package, recovery extracts this binary to `/tmp` and runs it, passing the following arguments:

- **Update binary API version number.** If the arguments passed to the update binary change, this number is incremented.
- **File descriptor of the *command pipe*.** The update program can use this pipe to send commands back to the recovery binary (mostly for UI changes such as indicating progress to the user).
- **Filename of the update package .zip file.**

A recovery package can use any statically-linked binary as the update binary. The OTA package construction tools use the updater program (source in `bootable/recovery/updater`), which provides a simple scripting language that can do many installation tasks. You can substitute any other binary running on the device.

For details on the updater binary, edify syntax, and builtin functions, see [Inside OTA Packages](https://source.android.com/devices/tech/ota/inside_packages) ([https://source.android.com/devices/tech/ota/inside\\_packages.html](https://source.android.com/devices/tech/ota/inside_packages.html)).

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated June 17, 2017.*