App Installer

A simple-to-use, robust tool for installing Android apps.

Installing

- Go to the [releases](#) page
- Download the latest Setup.exe release (or prerelease, if you really want to)
- Run the file, the program will automatically open when the installer is done
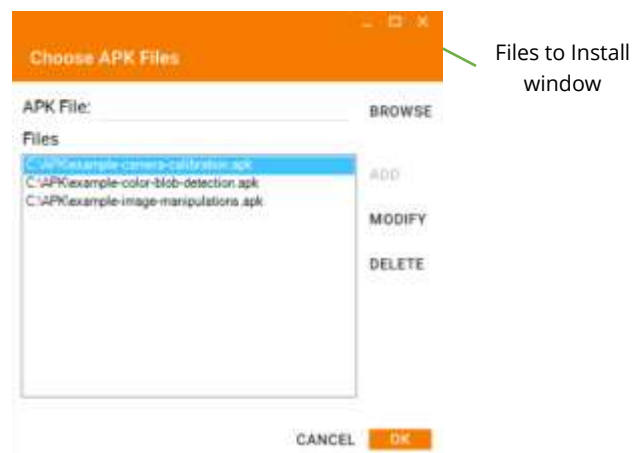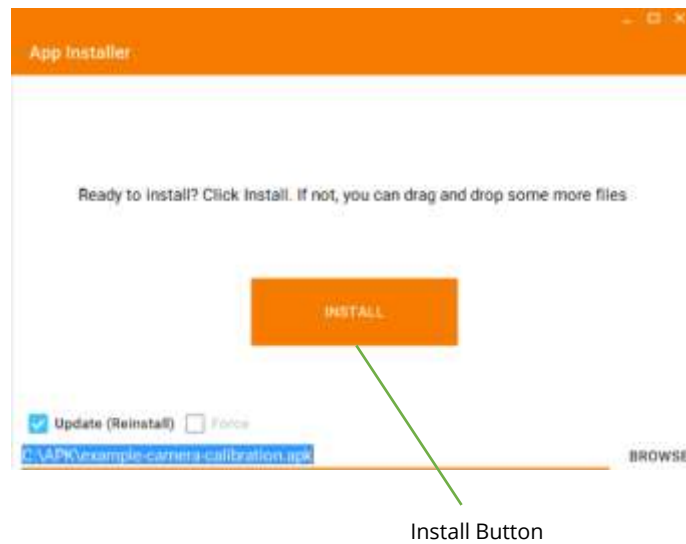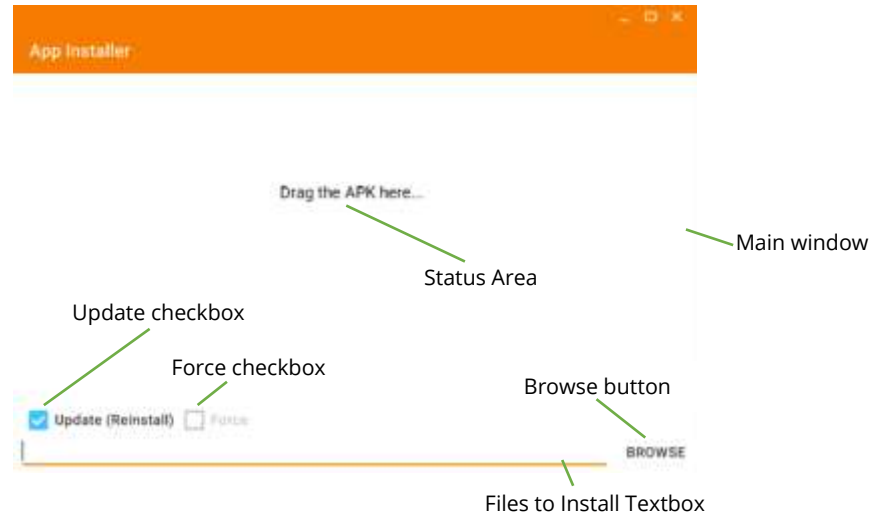
Usage

Simple

Obtain an APK file (or a few APK files) that you want to install. Drag-and-drop the file(s) on to the program's window. Make sure an Android device (in development mode with USB debugging enabled) is connected to the computer, if you are unsure about the phone consult the [guide](#) to set it up. Click the Install button on the App Installer window when you are ready.

Batch

Obtain your APK files Drag-and-mode your APK files on to the window (you can drag more than one file on to the window at once). If you ever make a mistake, double-click on the "files to install" textbox to open a "Files to Install" window to modify what files you are going to install. When you are finished modifying what files to be installed, click OK on the "Files to Install" window. Make sure an Android device (in development mode with USB debugging enabled) is connected to the computer. If you are unsure about the phone, consult the [guide](#) to set it up. Click the Install button on the App Installer window when you are ready.

Interface

Drag the APK here...

Main window

Status Area

Update checkbox

Force checkbox

Browse button

Update (Reinstall)   Force

BROWSE

Files to Install Textbox

App Installer

Ready to install? Click Install. If not, you can drag and drop some more files

INSTALL

Update (Reinstall)   Force

C:\APK\example-camera-calibration.apk

BROWSE

Install Button

Choose APK Files

Files to Install window

APK File: BROWSE

Files

C:\APK\example-camera-calibration.apk
C:\APK\example-color-blob-detection.apk
C:\APK\example-image-manipulations.apk

ADD

MODIFY

DELETE

CANCEL   OK

Options

Force
This removes any existing package with the same package name and all of the associated data before installing the given APK on the connected device.
The id is version-independent of both the App Installer, and the APK version. The id is also independent of the APK file name. The id, also referred to as the *package name*, is given to the APK file by its internal manifest, which was compiled in.

Update (Reinstall)
Reinstalls (or updates the app) with the given APK.

The APK used determines if the process would be considered an upgrade or a re-install. If the APK file being used is different than the APK file that was previously used to install the package, the process would be *upgrading.* Otherwise, as long as a previously installed version of the same package is on the phone, the process would be *reinstalling.* If the application is not currently installed on the phone, then this option is ignored.

Command Line Options

`--force`
`-f`
Removes any existing application and wipes any associated application data for that application, before installing.

`--no-prompt`
`-np`
Does not prompt user for option *[Implicit with options: -r, -f]*

`--update`
`--reinstall`
`-r`
Reinstalls the app represented by the given APK. Does nothing if the app is not already installed.

`-!np`
`--prompt`
Prompts the user about options *[Overrides: -np]*

`file`
The path of an APK file. The path should exist, be a valid APK file, and end with the file extension `.apk`

Examples:
`APKInstaller test.apk`
`APKInstaller C:\somedir\and\some\more\directories\foo.apk`

Troubleshooting

The program can't find your device?

The device needs to be in development mode, with USB debugging enabled and connected to your computer. If you are using a ADB TCP/IP bridge, try connecting the phone with a USB cable.

The program is having trouble with ADB TCP/IP bridges

The ADB versions in use are most likely too different to use the TCP/IP bridge. However, if you add your platform-tools to the system path, or if you correctly configure the `ANDROID_HOME` environment variable to point to your Android SDK. The program will use the Android SDK's copy of ADB.

Important: The program will not use any copy of ADB located under the Windows directory (`%windir%`), as that version is most likely out-of-date.

Bundled ADB version

```
$ adb version
Android Debug Bridge version 1.0.36
Revision af05c7354fe1-android
```

Configuring an Android device

1. Enable USB debugging on your device by going to Settings > Developer options.
   o Note: On Android 4.2 and newer, Developer options is hidden by default. To make it available, go to Settings > About phone and tap Build number seven times. Return to the previous screen to find Developer options.
2. Set up your system to detect your device.
   o If you're on Windows, you need to install a USB driver for adb.
      ▪ [Moto G USB Driver](#)
      ▪ [ZTE Speed Driver](#)
        Note The ZTE Speed's driver is inside a RAR file, a tool like [7-Zip](#) may have to be used to extract it.

Building

Requirements

- Visual Studio 2015+
- Git
- An internet connection (to download NuGet dependencies)

Get a copy

Clone the repo

```
git clone https://github.com/dmssargent/AppInstaller
```

Download a [zip](zip)

Open the solution in Visual Studio

Open `APKInstaller.sln` in Visual Studio (`Ctrl+O` or **File | Open > File**)

Build

Press `Ctrl+Shift+B` or **Build | Build Solution**

Run

Press `F5` or **Debug | Start Debugging**