AppImageKit-Toolbox README

Kurt Pfeifle <kurt.pfeifle@gmail.com>

2018-01-01

Contents

AppImageKit-Toolbox

The Complete Suite of Tools Provided by the AppImage Project to Build AppImages – Shipping as a Single AppImage!

The AppImageKit Toolbox ships as an AppImage containing all tools provided by the AppImage project and its contributors. The individual tools are provided as AppImages too (even though you may not recognize them as such at first).

To the outside world it appears as a "multicall binary", similar to the *busybox* suite of tools.

When called without an argument (or with the `--help` parameter), it simply prints a help overview page.

At this time it is only built for the x86_64 CPU architecture.

Make Executable

For security reasons, after downloading an file, it is not by default executable. To make use of this AppImage, you have to set the 'executable' bit first, just as is common to all AppImages:

```
 $>  chmod a+x ./appimagekit-toolbox*-x86_64.AppImage
```

Renaming the AppImage

You may find the original name not so comfortable to use. Feel free to rename this AppImage (you do not even need to keep the .AppImage suffix) any way you like, and it should keep working the same way.

For example:

```
  # Copy the file to one with a shorter name, putting it into your $PATH...
  $> cp -a  ./appimagekit-toolbox*-x86_64.AppImage  $HOME/bin/aiktb

  # ...or creating a symlink...
  $> ln -s  $(pwd)/appimagekit-toolbox*-x86_64.AppImage  $HOME/aiktb

  # ...or renaming the AppImage:
  $> mv  appimagekit-toolbox*-x86_64.AppImage  aiktb
```

Usage

You can use all the tools included in this "Meta"-AppImage without unpacking it. In this case you need to invoke the functionality of the embedded tool through the respective sub-command.

Sub-Commands

To list all available sub-commands, run the AppImage with the `--listexe` flag:

```
$>  ./appimagekit-toolbox*-x86_64.AppImage --listexe

    AppImageUpdate (GUI)  # GUI utility to check for and apply available updates of AppImage
    appimaged             # Daemon to auto-integrate stored AppImages into desktop environment
    appimagetool          # Utility to generate AppImages from AppDirs and to inspect them
    appimageupdatetool    # CLI tool to check for and apply available updates of AppImage
    linuxdeployqt         # Utility to prepare an AppDir, its contained executables and
                            libraries for AppImage creation; can also create the AppImage
    zsync2                # Utility to efficiently update + syncronize changed files remotely
                            by applying binary delta patches to existing file
    zsyncmake2            # Utility to create a .zsync file for an existing file so it can be
                            later updated via `zsync2`
    AppRun                # Not meant to be executed by users -- provided for debugging only
                            (Not an AppImage)
    runtime               # Not meant to be executed by users -- provided for debugging only
                            (Not an AppImage)
```

General Help

To list all available ways how to invoke this AppImage, run:

```
$>  ./appimagekit-toolbox*-x86_64.AppImage --help
```

Other Options

By running the AppImage with *--help*, you should easily be able to discover the other available options:

```
--listexe                  # Enumerate all executables embedded in AppImage
--listfile                 # Enumerate all files embedded in AppImage

--listman                  # List all embedded manual pages
--man <name-of-manpage>    # Show the embedded man page specified
                             (keeps AppImage mounted)
--listhtml                 # List all embedded HTML pages
--html <path/to/htmlfile>  # Open (in default browser) the embedded HTML page specified
                             (keeps AppImage mounted)
--listpdf                  # List all embedded PDF documents
--pdf <path/to/pdffile>    # Open (in default browser) the embedded PDF doc specified
                             (keeps AppImage mounted)
--listepub                 # List all embedded HTML pages
--epub <path/to/epubfile>  # Open (in default app) the embedded EPUB doc specified
                             (keeps AppImage mounted)
--listreadme               # Enumerate all READMEs embedded in AppImage
--readme <path/to/readme>  # Show embedded README specified
                             (keeps AppImage mounted)
--listdir                  # List all directory paths embedded in AppImage
--dir <path/to/dir>        # Open specified directory path
                             (keeps AppImage mounted)
```

Options Common to All (Recent) AppImages

You can run this AppImage (as all other AppImages which are built according to our recommended best practices) with the *--appimage-help* parameter. This enumerates options which are common to all AppImages:

Unpack this AppImage

You can unpack this AppImage into a local sub-directory (currently named *squashfs-root*) by running this command:

```
./appimagekit-toolbox*-x86_64.AppImage --appimage-extract
```

Now you can find the binaries then in *./squashfs-root/usr/bin/*. These are mostly AppImages themselves, even if they here carry names which do not make use of the .AppImage suffix. This can be useful if you want to run one or all of the tools directly (not as a sub-command from the "mother" AppImage).

```
 $>  ls -l ./squashfs-root/usr/bin

    -rwxrwxr-x 1 kp kp   271792 Jan  1 00:00 appimaged
    -rwxrwxr-x 1 kp kp   468400 Jan  1 00:00 appimagetool
    -rwxrwxr-x 1 kp kp 10253744 Jan  1 00:00 AppImageUpdate
    -rwxrwxr-x 1 kp kp  8754608 Jan  1 00:00 appimageupdatetool
    -rwxrwxr-x 1 kp kp    10872 Jan  1 00:00 AppRun
    -rwxrwxr-x 1 kp kp 14546352 Jan  1 00:00 linuxdeployqt
    -rwxrwxr-x 1 kp kp   116144 Jan  1 00:00 runtime
    -rwxrwxr-x 1 kp kp  2381496 Jan  1 00:00 zsync2
    -rwxrwxr-x 1 kp kp  2389688 Jan  1 00:00 zsyncmake2
```

Extract Individual Tools (or other files) from AppImage

If you only want to extract individual tools or other files from this AppImage, you do not need to use *--appimage-extract*. You can run it with *--appimage-mount*. This will mount the AppImage and print the mount point, keeping it mounted. Now you can use any utility you want (either a graphical file manager or a different terminal/console instance) to go into the mounted file hierarchy, look around and copy files to any location outside of that file tree.

To umount again, simply go to the first terminal and hit *"ctrl+c"*.

Invoking Built-in Help

For a general overview simply run

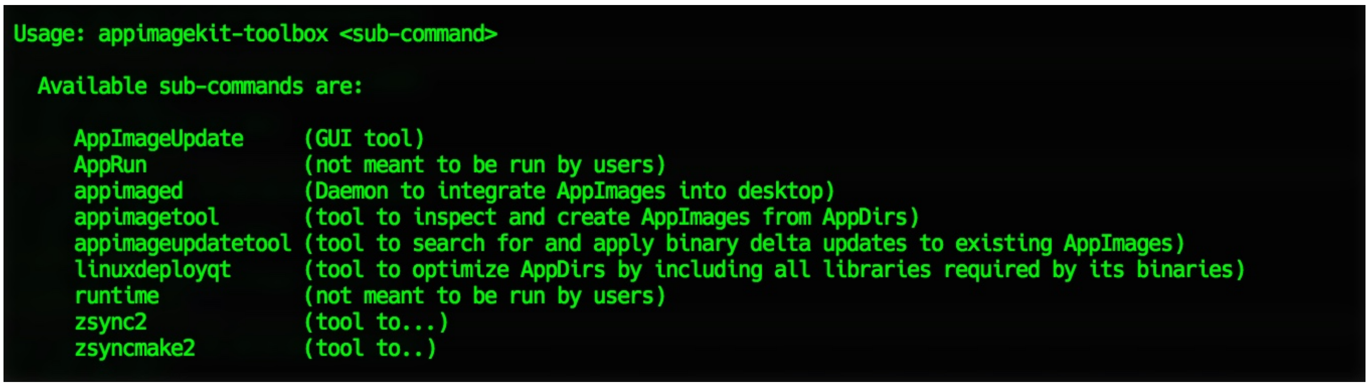
```
./appimagekit-toolbox*-x86_64.AppImage --help
```

You can also access the embedded AppImageKit manual pages. For example, to view the man page for AppImage(7) invoke

```
./appimagekit-toolbox*-x86_64.AppImage --man 7 AppImage
```

TODO: complete this README with a view more topics

Screenshots

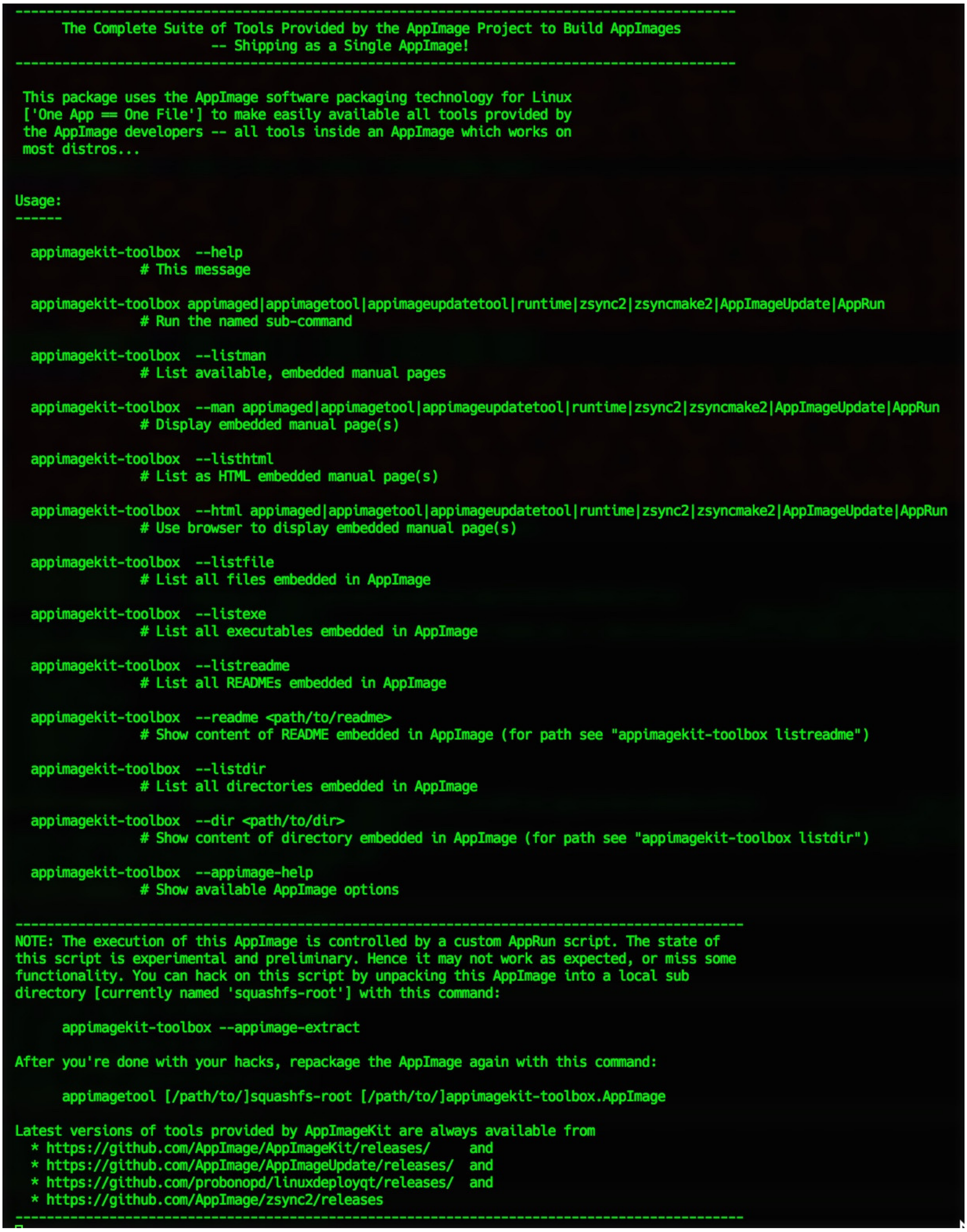


```
Usage: appimagekit-toolbox <sub-command>

  Available sub-commands are:

     AppImageUpdate     (GUI tool)
     AppRun             (not meant to be run by users)
     appimaged          (Daemon to integrate AppImages into desktop)
     appimagetool       (tool to inspect and create AppImages from AppDirs)
     appimageupdatetool (tool to search for and apply binary delta updates to existing AppImages)
     linuxdeployqt      (tool to optimize AppDirs by including all libraries required by its binaries)
     runtime            (not meant to be run by users)
     zsync2             (tool to...)
     zsyncmake2         (tool to..)
```

Figure 1: appimagekit-toolbox: available sub-commands



```
----------------------------------------------------------------------------------------------
      The Complete Suite of Tools Provided by the AppImage Project to Build AppImages
                        -- Shipping as a Single AppImage!
----------------------------------------------------------------------------------------------

 This package uses the AppImage software packaging technology for Linux
 ['One App == One File'] to make easily available all tools provided by
 the AppImage developers -- all tools inside an AppImage which works on
 most distros...


Usage:
------

  appimagekit-toolbox  --help
                # This message

  appimagekit-toolbox appimaged|appimagetool|appimageupdatetool|runtime|zsync2|zsyncmake2|AppImageUpdate|AppRun
                # Run the named sub-command

  appimagekit-toolbox  --listman
                # List available, embedded manual pages

  appimagekit-toolbox  --man appimaged|appimagetool|appimageupdatetool|runtime|zsync2|zsyncmake2|AppImageUpdate|AppRun
                # Display embedded manual page(s)

  appimagekit-toolbox  --listhtml
                # List as HTML embedded manual page(s)

  appimagekit-toolbox  --html appimaged|appimagetool|appimageupdatetool|runtime|zsync2|zsyncmake2|AppImageUpdate|AppRun
                # Use browser to display embedded manual page(s)

  appimagekit-toolbox  --listfile
                # List all files embedded in AppImage

  appimagekit-toolbox  --listexe
                # List all executables embedded in AppImage

  appimagekit-toolbox  --listreadme
                # List all READMEs embedded in AppImage

  appimagekit-toolbox  --readme <path/to/readme>
                # Show content of README embedded in AppImage (for path see "appimagekit-toolbox listreadme")

  appimagekit-toolbox  --listdir
                # List all directories embedded in AppImage

  appimagekit-toolbox  --dir <path/to/dir>
                # Show content of directory embedded in AppImage (for path see "appimagekit-toolbox listdir")

  appimagekit-toolbox  --appimage-help
                # Show available AppImage options

----------------------------------------------------------------------------------------------
NOTE: The execution of this AppImage is controlled by a custom AppRun script. The state of
this script is experimental and preliminary. Hence it may not work as expected, or miss some
functionality. You can hack on this script by unpacking this AppImage into a local sub
directory [currently named 'squashfs-root'] with this command:

      appimagekit-toolbox --appimage-extract

After you're done with your hacks, repackage the AppImage again with this command:

      appimagetool [/path/to/]squashfs-root [/path/to/]appimagekit-toolbox.AppImage

Latest versions of tools provided by AppImageKit are always available from
  * https://github.com/AppImage/AppImageKit/releases/     and
  * https://github.com/AppImage/AppImageUpdate/releases/  and
  * https://github.com/probonopd/linuxdeployqt/releases/  and
  * https://github.com/AppImage/zsync2/releases
----------------------------------------------------------------------------------------------
```

Figure 2: appimagekit-toolbox: help screen