   各种安装包打包发布工具（安装制作工具）评测

   作为程序员，经常遇到需要打包自己劳动成果的时候，这也算是广大码工辛苦奔波之后的杀青之作，收尾之笔。因此，安装包制作的好坏也直接关系到整个工程项目的质量。这里HAWARK STUDIO对几种常见的程序打包封装工具进行一次大串联，让大家少走点弯路。

1.小兵安装包制作工具--不得不承认，国产兵兵的营销做得很牛逼，只要打开百度搜索“安装包制作工具”，满屏的小兵软件连接啊……这优化做得，百度就成了他家似的。当然，小兵安装制作工具中文操作界面，使用简单，界面直观，很容易上手，这些都是优点。不过，国产软件毕竟是简单了些，要做一点复杂的安装程序就不给力了，而且还要收费，不交钱用测试版的话，做出来的小兵安装包会弹出广告对话框，而网上的破解版又是非常老的，目前最新的破解版是V5.1的，比小兵6.1差了一个版本也就算了，关键还破解得不到位，总是有这样那样的不如意。倒不是说破解者没责任心，只是说小兵打包软件太不出众了，才没引来破解者的关注。一句话：广告打得震天响，实际效果得掂量，简单任务很方便，复杂一点就很面。

2.Install Shield XXX-安装制作工具中的元老级软件，也是标准的恐龙级软件，动不动就上百兆的身躯，不愧是安装制作工具中的大哥大（不过太大的东西，有些人总觉得不舒服）！什么功能都有，连Linux和WindowsMobile下的安装都可以做，MSI、EXE、CAB等都可以通通搞定，属于全能型大将。网上破解版的也挺多的，down下来就能用，操作上相对专业一些，不过也不算太复杂，汉化版的也早出来了，上手不难。一句话：大型软件，专业操作，又大又强！

3.Advanced Installer-弱弱的问一句，我可以复制么？不管了，直接复制了：安装制作工具中的元老级软件，也是标准的恐龙级软件，动不动就上百兆的身躯，不愧是安装制作工具中的巨无霸！什么功能都有，连Linux和WindowsMobile下的安装都可以做，MSI、EXE、CAB等都可以通通搞定，属于全能型大将。网上破解版的也挺多的，down下来就能用，最新版的是V10.0，这个软件操作上相对专业一些，不过也不算太复杂，上手比Install Shield还简单。一句话：大型软件，专业操作，又大又强！

吹牛B的和国际知名的巨无霸软件介绍完了，下面该是小家碧玉型的亲民性软件了（大小不超过20M，一般5M以下）。

4.小颖安装程序制作专家-YingSoft出品的一枚国产软件，很飘逸的软件名字，有着汉服美女版的小清新。这个软件总体还算不错的，界面简单，操作容易，可以制作出较为专业的安装效果来。不过，软件打包思路有点陈旧，还是使用Win2000时代的安装部署方式，需要再Windows 的System目录下安插一个Uninstall程序，搞得每次做出来的安装包都有一个在系统目录植入文件的操作。而目前，各类杀毒软件对这种操作比女性私处还敏感，你这么大张旗鼓地安装个文件，杀毒软件还不都刷刷刷地出来报警，你让一般啥都不懂的电脑用户情何以堪？软件厂商总不能每次都去甩一大堆口水来抚慰那些被杀毒软件报警吓得胆战心惊的用户吧，所以这个国产软件，确实亟待来个彻底改善，不然真心没出路的。

5.WinRAR-这个软件HAWARK就不多解释了，对付小型的非专业性的程序封装，想必那是极好，简单、快捷、高效，十分具体方法，请在百度里搜索 “ WinRar制作安装程序 ” 这几个关键字，定有度娘亲自指点！

6.SFXMaker1.2.1-全英文界面，类似WinRAR的安装包压缩制作工具，十分地小巧，但是比WinRAR来得相对专业些。这个软件的压缩效率很高，就是里面的7zS.sfx容易被不明真相的杀毒软件报毒（如果被误杀，则无法完成安装包的制作）。这个软件比较老了，2007年出来的，就是个打包工具，要加什么net环境，java环境什么的，根本就是奢望，要想在安装后运行什么readme，弹出什么网页之类，也基本不可能。但人家毕竟简单小巧，完成个小程序封装还是小case的。

7.MakeInstall - 2009年就不更新了，也是比较早期的一款安装包制作工具，其2009版的注册码NAME:piaodown Serial:lse3Xh8yuMO7Ajhe7，为什么HAWARK STUDIO敢在这里肆无忌惮的贴出注册码，因为这个软件压根就装不上楼主的电脑！想评论都没着啊，不过目测做出来的安装包顶多是Win2000风格的。

8.ksis - 这个软件是安装包脚本制作工具，忘了说了，目前安装包制作工具分两大类，一是GUI方式进行设计，还一种是脚本方式设计，而KSIS就属于脚本设计的一种，通过脚本设计出安装包程序，也算是四两拨千斤的气魄了。KSIS脚本通过编译器可以实现安装的自动控制，很有时尚感，不过这是适合高人玩的工具，反正HAWARK STUDIO没有认真细研过，毕竟谁都喜欢有GUI的东西。

9.Innosetup - 好像现在的版本是5.5，网上有绿色版的，据说功能很强大……强大……大（这是回音）。又是一款基于脚本的安装包制作工具，需要有深厚的脚本知识，喜欢的朋友可以尝试下去，有免费破解版的，达人必备。

10.CreateInstall - 最新版的是2005年的V5.19 ，年代久远的古董级软件，英文界面，没看到有汉化破解版的，有30天试用期，做出来的安装包是XP风格的，只有英文的，对中文支持不是很好。这个软件的界面不是很复杂，操作起来也不难，不过专业性不是太强，只适合搞点普通程序的打包发布，如果给外国友人做产品，还是可以满足需要的。

11.EazySetup - 又一款古董级小型软件，2004年就木见更新了网上一共就三个版本，汉化版的V2.08，破解版的V3.0，最新版的V3.14。这个软件的GUI所见即所得，虽然界面古老，但设计模式很前卫，就算现在的Install Shield也没敢这么玩。EazySetup制作的安装软件具有win98的风格，功能上讲也还凑合，什么环境变量、注册版更改、安装前运行程序、SN验证等一应俱全，相对于1.27M的体积，真是五脏俱全啊。不过这个软件做的安装包，在进行卸载的时候，会在temp目录下加入一个del.bat文件，并加入到系统自启动，又有躁动的杀毒软件蹦出来报警了。还好只是在卸载的时候有，安装的时候基本无恙的。总体感觉——软件太老，做出来的东西复古，适合怀旧玩家。

12.QSetup Installation Suite-有7.51的汉化版和10.0的破解版，最新的也就是2009年的了。软件界面很另类，有类似qq的抽屉按钮，做出来东西有winxp安装包风格，蛮像那么回事，不过Qsetup和小颖软件有同样的弊病，放硬盘上还没运行许就被杀毒软件疯狂报警，说是什么KVM病毒，坑爹啊。

13.ActualInstaller-各位亲爱的读者，很高兴您还这么有兴致看到最后，当然，这放到最后的，一定是值得推荐的好东东。这款软件目前有汉化的3.8和没有被破解的4.8，其功能总体而言比较不错，界面简单，操作方便，功能强大，做的东西也很专业，有点类似小兵和小颖的样子，不过人家做的安装包不会被报警，尤其是汉化版的3.8，你懂的。说到缺点，对话框不能定制，不能编辑，不能增加额外的，只能用自带的几个，有种天然呆的赶脚。

14.Astrum Setup Winzard-各位亲爱的读者，很高兴您还这么有兴致看到最后(有眼熟的感觉吧)。这个软件的目前版本是2.XXX，可没有OO哦，有汉化版的，非常强大，界面可以定制，安装环境、环境变量、什么都可以编辑，总体感觉相当彪悍，最重要的是软件小巧，也就3M不到的样子，跟Install Shield有的一拼。当然说的瑕疵，也不是没有，一是界面复杂，需要花大把时间研究，二是什么呢，也是最要命的，汉化版的对安装包图标的支持太逊了，仅支持16X16 四位的图标，尼玛现在随便找个ICON编辑器都搞不出这么低级的图标来。如果你想投机取巧，准备用资源编辑器编辑做好的安装包的图标，那你就错了，这个安装程序加入了CRC校验，不允许对ICON资源进行修改，一改程序就出异常，没办法。都2013年了，你做个win98图标样子的安装包给用户，难道是想证明自己穿越么？当然，如果用户也是穿越剧的FANS，你用这个打包你的程序，效果还是相当不错的。

elf是linux下可执行文件的格式，对应windows中的PE格式，并不是一种打包方式。

linux下的打包方式有很多：

sh脚本方式、deb安装包、rpm安装包

终端下 使用ldd命令查看需要的连接库。之后自己编写shell程序，在其中增加环境变量路径，加上自己的dll文件目录。

Linux打包：（不推荐，一般都会比较大）

如果是linux下，可以使用Qt Installer Framework(可以制作在线安装)，或者使用其余打包软件。相比较Windows，linux可以直接使用ldd命令查看相关的依赖。但是因为linux存在过多有差异的发行版，linux真打包起来需要很多很多so文件，于是最终打包文件一般都会很大。

无依赖打包形式：Flatpak、Snap、AppImage(下面有例子)、sfpkg

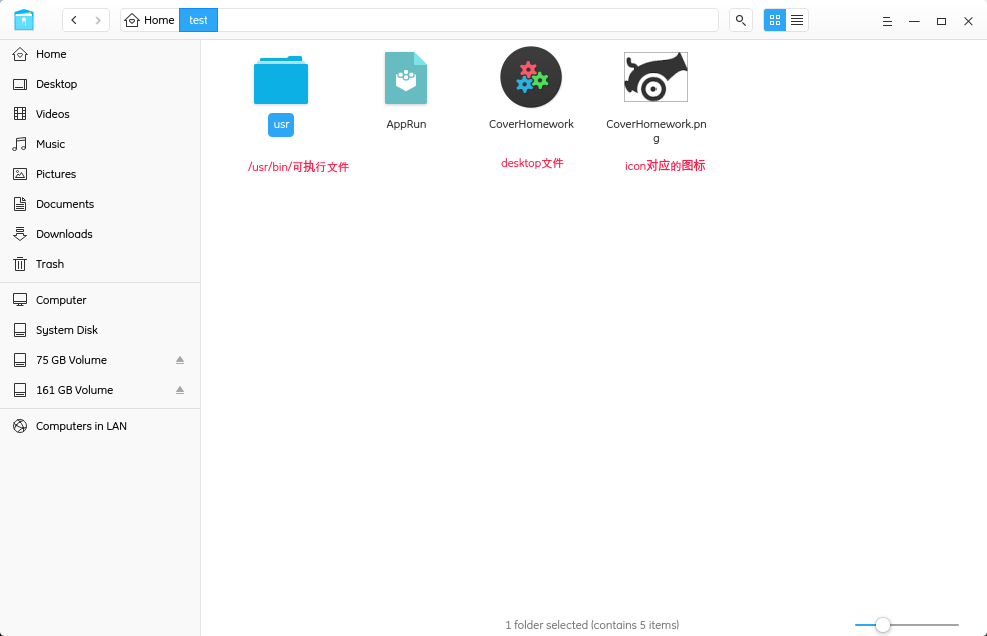
AppImage将所有依赖打包，解决的是依赖问题；Flatpak提供独立环境，各种各样的runtime（比如deepin自己就弄了一个deepin runtime，没能解决发行版过多带来的花式依赖问题，不过runtime目前才800+MB，一种缩小版系统的感觉），因为有独立环境，可以将runtime中缺少的依赖直接打包到最后的文件中，解决的是依赖冲突问题。

比较： https://github.com/AppImage/AppImageKit/wiki/Similar-projects#comparison

AppImage打包：

appimage其实就是一个压缩包,打包了所有的依赖,运行的时候就解压缩然后运行在里面。

appimagetool打包应用，appimaged注册和取消注册应用于系统daemon服务...

不过qt使用的是https://github.com/probonopd/linuxdeployqt，创建一个目录，二进制文件放到./bin下，qml放到/qml下，linuxdeployqt name -appimage就开始打包，缺少的目录与文件会自动默认创建。下面的文件是必须的：

so文件缺少要安装对应版本库：虽然qt中没有用到这个库，缺少不影响运行，但是linuxdeployqt借助ldd递归检查依赖的时候，要求一个也不能少：

ERROR: ldd outputLine: "libpq.so.5 => not found"

ERROR: for binary: "/opt/Qt5.10.1/5.10.1/gcc\_64/plugins/sqldrivers/libqsqlpsql.so"

ERROR: Please ensure that all libraries can be found by ldd. Aborting.

上面的错误意为打包目标依赖的libqsqlpsql.so缺少依赖的libpq.so.5

libmysqlclient.so.18=>libmysqlclient18

libpq.so.5=>libpq5

安装了对应库以后，so文件就会出现在/usr/lib/x86\_64-linux-gnu下

一开始我没有打包正确，得到的AppImage有150MB，文件也是不可执行的。后来重新打包，Appimage才25MB。

flatpak(xdg-app)打包： <http://docs.flatpak.org/en/latest/getting-started.html>

（中文介绍）<https://blog.csdn.net/beyond_zhangna/article/details/79774771>

<https://blog.csdn.net/qq_32768743/article/details/80024622> （About Qt按钮还提供了查看当前环境中qt版本信息）

sudo apt install flatpak flatpak-builder

sudo apt install deepin-fprt-com.deepin.platform deepin-fprt-com.deepin.sdk

关键概念：运行时、基准运行时（规定运行flatpak的发行版必须有哪些软件和实现了哪些标准）、SDK、绑定库（Bundle Libraries）、沙盒（SandBox）、Portals、存储库(remote)、缓存（结果文件夹删除以后，同样的命令会迅速创建，在当前目录下的.flatpak-builder/cache/refs/heads）(~/.local/share/flatpak/repo是一个同样文件夹结构的文件夹)（～/.cache里面虽然有flatpak的位置但是为空）。

flatpak应用要求每一个应用程序指定runtime，使用它的依赖。有freedesktop、gnome、kde、deepin等几种选择。flatpak remote-add --from gnome <https://sdk.gnome.org/gnome.flatpakrepo>像GTK的话，可以选择gnome，QT的话可以选择KDE（基于freedesktop，增加了qt、kde结构），因为他们里面自带了gui库，当然直接选择freedesktop，然后手动加入gui库也是可以的。

deepin全家桶是基于bundle的deb文件。

安装环境：环境有runtime（以Platform结尾）、打包环境（SDK）

deepin：

sudo apt install flatpak flatpak-builder deepin-fprt-com.deepin.platform-15.5 deepin-fprt-com.deepin.sdk-15.5

kde： <https://community.kde.org/Guidelines_and_HOWTOs/Flatpak>

应用命名方式与Android一样，作为容器中的ID，这个ID可能还会被用于D-BUS，所以名字7-zip应该命名org.\_7\_zip.xxxx。

应用要放入仓库（remote），而对仓库的任何提交（修改、删除、创建仓库）都需要-gpg-sign=<key>或者--no-gpg-verify。

finish-args 描述应用沙箱权限

--share=network - access the network

--socket=x11 - show windows using X11

--share=ipc - share IPC namespace with the host (necessary for X11)

--socket=wayland - show windows with Wayland

--device=dri - OpenGL rendering

--socket=pulseaudio - play sound with PulseAudio

Access to the entire bus with --socket=system-bus or --socket=session-bus should be avoided, unless the application is a development tool.

--filesystem=host - access normal files on the host, not including host os or system internals described below



--filesystem=home - access the user’s home directory

--filesystem=/path/path - access specific paths

--filesystem=xdg-download - access a specific XDG folder

--persist=path - map paths from the user’s home directory into the sandbox filesystem

--device=all can be used to access devices like controllers or webcams.

cleanup 相关属性，用于指定清除在build过程中产生的中间产物

modules 应用程序模块，相关库、依赖模块，比如qt.json

{

"name": "qt",

"disabled": true,

"buildsystem": "simple",

"build-commands": [

"./configure -prefix /app -confirm-license -opensource -nomake examples -no-angle -no-opengl -no-sql-sqlite -no-sql-sqlite -ssl -openssl-linked",

"make -j2 install"

],

"sources": [

{

"type": "archive",

"url": "https://download.qt.io/archive/qt/5.8/5.8.0/submodules/qtbase-opensource-src-5.8.0.tar.xz",

"sha256": "c17111ae02a44dc7be1ec2cf979a47ee9e58edf4904041a525c21f4fa53fc005"

}

]

}

flatpak-builder --repo=<repo> <build-dir> <manifest> 希望导入仓库

flatpak-builder --install <build-dir> <manifest> 不考虑仓库

modules类型：

archive - .tar or .zip archive files

git - Git repositories

bzr - Bazaar repositories

file - local file (these are copied into the source directory)

script - an array of shell commands (these are put in a shellscript file)

shell -an array of shell commands that are run during source extraction

patch - a patch (are applied to the source directory)

extra-data - data that can be downloaded at install time; this can include archive or package files

大多数flatpak命令默认在系统范围内执行。要仅为当前用户执行命令,请使用--user选项。

flatpak-builder选项介绍：

--force-clean build目标文件夹如果已经存在，则会被删除；

--disable-cache关闭缓存;

--keep-build-dirs 保留过程中文件，在当前文件夹下的.flatpak-builder，具体目录在构建的一开始有讲;

flatpak list -d --app --runtime 显示已经安装的App、Runtime

删除org.gnome.Platform/x86\_64/3.20：flatpak --user uninstall org.gnome.Platform 3.20（不加上版本号应该会移除所有版本）

flatpak remotes

flatpak remote-add --if-not-exists flathub https://dl.flathub.org/repo/flathub.flatpakrepo

flatpak remote-delete flathub

QT flatpak：

flatpak --user remote-add --no-gpg-verify --if-not-exists explosivebattery explosivebattery 建立仓库

pro文件中需要：

isEmpty(PREFIX){

PREFIX = /usr

}

target.path = $${PREFIX}/bin/

INSTALLS += target

或者直接：

target.path = /app/bin/

INSTALLS += target

Mainfest文件：

{

"app-id":"cn.hellovega.qt.CoverHomework",

"runtime":"com.deepin.Platform",

"runtime-version":"15.5.2",

"sdk":"com.deepin.Sdk",

"command":"CoverHomework",

"finish-args":[

"--socket=x11",

"--share=network"

],

"modules":[

{

"name":"CoverHomework",

"no-autogen":true,

"sources":[

{

"type":"git",

"url":"file:///home/vega/test/CoverHomework",

"branch":"master"

},

{

"type":"script",

"commands":[

"sed -i 's|/usr|/app|g' CoverHomework.pro",

"qmake PREFIX=/app"

],

"dest-filename":"configure"

}

]

}

]

}

flatpak-builder --repo=explosivebattery --force-clean 结果文件夹 Mainfest.jso 效果是在当前目录下产生一个explosivebattery文件夹，算是ref，install的时候用到

flatpak --user install explosivebattery cn.hellovega.qt.CoverHomework 安装

flatpak run cn.hellovega.qt.CoverHomework 运行

flatpak build-bundle explosivebattery cn.hellovega.qt.CoverHomework 打包

最后打包出来的因为经过压缩，仅仅400+KB（QT直接产生Release可执行文件有900+KB）。

给用户安装App的时候可以编写一个flatpakref文件，从某一个的remote添加指定名字的应用程序：

[Flatpak Ref]

Title=qBittorrent

Name=org.qBittorrent.qbittorrent

Branch=stable

Url=https://chocobo1.github.io/qBittorrent-flatpak/ Url参数似乎没有什么用

IsRuntime=False

RuntimeRepo=https://flathub.org/repo/flathub.flatpakrepo

GPGKey=xxx

Install:

flatpak install --from https://raw.githubusercontent.com/Chocobo1/qBittorrent-flatpak/master/qBittorrent.flatpakref --assumeyes

Run:

flatpak run org.qBittorrent.qbittorrent

Update:

flatpak update --app org.qBittorrent.qbittorrent

Remove:

flatpak uninstall org.qBittorrent.qbittorrent

减小打包体积：

1. 可执行文件与动态共享库文件DSO中都有符号表，可以移除来瘦身，同时我们还希望万一出事了，比如coredump了，我们能获取更多的信息，这时候我们又希望有符号表。https://blog.csdn.net/bingqingsuimeng/article/details/50522242
2. ldd -u xxx 列出不使用的so文件