

● Softwarebeheer in Linux met AppImage ●

Hans Luning

Inleiding

In SoftwareBus 2023-5 schreef ik over ‘Softwarebeheer in Linux met Flatpak’. Daarbij ging ik eerst in op de wijze waarop software voor Linux wordt verspreid, en de rol die pakketbeheersystemen daarin vervullen. Er worden twee soorten pakketbeheersystemen onderscheiden, te weten specifieke en universele systemen. Universele systemen zijn ontwikkeld om tegemoet te komen aan de nadelen van specifieke systemen. Een pakket dat volgens een universeel systeem is samengesteld kan in principe in elke versie van elke distributie van Linux worden gebruikt. De belangrijkste nu beschikbare universele systemen zijn Flatpak, Snap en AppImage. Flatpak kwam de vorige maal aan de orde. Nu is de beurt aan AppImage.

AppImage

AppImage (appimage.org) bestaat al sinds 2004 onder de naam Klik, werd vanaf 2011 PortableLinuxApps genoemd en vanaf 2013 AppImage. Het is ontwikkeld door een zekere Simon Peter en is geheel opensource. Het centrale magazijn is AppImages (appimage.github.io) met ruim 1400 applicaties. Je vindt ze ook op AppImageHub (www.appimagehub.com).



Een AppImage pakket van een applicatie bevat naast de applicatie zelf alle hulpsoftware (met name bibliotheken) die nodig is om de applicatie te draaien en deelt geen hulpsoftware met andere AppImages. Dat houdt in dat een AppImage helemaal op zich zelf staat. Er is geen installatieprocedure voor nodig, maar het kan naar believen ergens in het bestandssysteem worden geplaatst en uitvoerbaar maakt. Een mogelijke locatie is ergens in de thuismap van de gebruiker. Rootrechten zijn er dan niet voor nodig. AppImages zijn daarom gemakkelijk overdraagbaar (portable) van het ene Linux-systeem naar het andere.

Technisch gezien bevat een AppImage een stuurprogramma en een gecomprimeerd bestandssysteem (*squashfs*) met de applicatie en alle benodigde hulpsoftware¹. De compressie van het ingebedde bestandssysteem bespaart behoorlijk veel schijfruimte. Bij uitvoering van een AppImage koppelt (*mount*) het stuurprogramma het ingebedde bestandssysteem aan een verborgen eenmalige map onder */tmp*, waarna de applicatie vanuit die map wordt gestart. De koppeling aan een map in */tmp* wordt gedaan met *fuse* (Filesystem in Userspace), zodat hiervoor geen rootrechten nodig zijn. Voor *fuse* is de *library libfuse.so.2* nodig. Ook al staan AppImages op zichzelf, ze maken wel gebruik van enkele systeembibliotheken, waaronder *libfuse*, waarvan de juiste versie in het systeem beschikbaar moet zijn.

Om AppImages met alleen hun naam aan te kunnen roepen is het handig om ze in een map van het zoekpad \$PATH naar te

1. Het gaat hier om AppImage type 2. Er is ook nog een ouder formaat, type 1, dat tot 2016 werd gebruikt. Ik ga niet in op type 1, omdat dat type nog maar heel weinig zal voorkomen.

zetten. Een geschikte map waar geen rootrechten voor nodig zijn is een map ‘bin’ in de thuismap van de gebruiker: *\$HOME/bin*. In de meeste gevallen zal die map nog wel moeten worden aangemaakt en aan het PATH worden toegevoegd. Andere gebruikers van het systeem hebben dan echter geen toegang tot die AppImages. Wil je ze aan alle gebruikers ter beschikking stellen dan is */usr/local/bin* een geschikte map, maar die vereist voor plaatsing wel rootrechten.

De namen van AppImages zijn vaak nogal lang. Als je ze vanaf de opdrachtregel wilt starten zou het wel handig zijn om ze met een verkorte naam te kunnen aanroepen. Dat kan met een symbolische link (*symlink*), bijvoorbeeld:

```
In -s scribus-1.5.8-linux-x86_64.AppImage scribus
```

Dank zij deze symlink kan de AppImage van scribus nu worden aangeroepen met alleen ‘scribus’.

AppImages zijn weliswaar heel gemakkelijk overdraagbaar van het ene Linux-systeem naar het andere, maar ze maken nog wel gebruik van de thuismap van de gebruiker en van de configuratie daarin. De daarin opgeslagen gegevens worden bij draaien op een ander systeem dus niet meegenomen. Om hierin te voorzien hebben nieuwere AppImages een ‘portable’ modus, waarbij in dezelfde map als de AppImage een bijbehorende home- en een config-map worden gebruikt.

Deze twee mappen kunnen worden aangemaakt met de AppImage opties *--appimage-portable-home* en *appimage-portable-config*. Hun naam is gelijk aan die van de AppImage, gevolgd door *.home* resp. *.config*. Die mappen hoeven niet per se met behulp van de opties te worden aangemaakt: je kunt het ook zelf doen.

De veiligheid van AppImages is niet standaard geverifieerd. Ze zouden dus kwaadaardige software, zoals virussen of malware, kunnen bevatten. Daarom is het belangrijk AppImages alleen van vertrouwde bronnen te downloaden, zoals AppImageHub. Let dus altijd goed op en verifieer de authenticiteit en integriteit van de bron voordat u een AppImage ervan downloadt en uitvoert.

Ter wille van de veiligheid kunnen AppImages met behulp van Firejail in een *sandbox* draaien, waardoor hun toegang tot netwerk- en systeembronnen zoveel mogelijk wordt beperkt. Hoe dit moet worden gedaan staat (in het Engels) op de website van *firejail* (zie onder Links). Firejail kan vanuit de eigen software-repositories van de Linux-distributie worden geïnstalleerd. In Linux Mint gaat dat het gemakkelijkst via Programmabeheer.

Aardig is het nog om te vermelden dat er een Linux-distributie is die focust op het gebruik van AppImage voor applicaties: *Nitrux*. Zie onder Links.

AppImage-opties

Opties die bij starten van een AppImage vanaf een opdrachtregel worden meegegeven, worden door het stuurprogramma doorgesluist naar de ingebedde applicatie. Het stuurprogramma heeft zelf echter ook een aantal opties. Hun naam begint met *--appimage* om ze te onderscheiden van de opties van de applicatie. Het zijn:

--appimage-help	Toon deze lijst van opties
--appimage-extract [<patroon>]	Onttrek de inhoud van het ingebedde bestandssysteem. Als een patroon is opgegeven worden alleen de passende bestanden onttrokken.
--appimage-mount	Koppel (mount) het ingebedde bestandssysteem, laat zien wat het koppelpunt is, en wacht op beëindiging met Ctrl-C.
--appimage-offset	Laat zien bij welke byte offset het ingebedde bestandssysteem begint.
--appimage-portable-home	Maak een overdraagbare thuismap voor gebruik als \$HOME
--appimage-portable-config	Maak een overdraagbare configuratie map voor gebruik als \$XDG_CONFIG_HOME
--appimage-signature	Toon de digitale handtekening die in de AppImage is opgenomen, indien beschikbaar.
--appimage-updateinfo[rmation]	Toon de update-informatie die in de AppImage is opgenomen, indien beschikbaar.
--appimage-version	Toon de versie van AppImageKit, de software waarmee de AppImage is gemaakt.

Bij onttrekking van het ingebedde bestandssysteem wordt dat geplaatst in de submap *squashfs-root* van de actuele map.

Beheersoftware voor AppImage

Een nadeel van AppImages is dat ze niet in het systeem worden geïntegreerd. Zo is er geen mechanisme om AppImages bij te werken, en worden ze ook niet automatisch in het menu opgenomen. Het beheer wordt dus volledig aan de gebruiker overgelaten. AppImage ‘an sich’ is dus eigenlijk geen echt pakket-beheersysteem. Gelukkig is door een aantal mensen software gemaakt met behulp waarvan AppImage beter in het systeem kan worden geïntegreerd, en zo meer op een echt beheersysteem gaat lijken. Ik noem er hier enkele die ik heb getest:

AppImageLauncher

AppImageLauncher is een programma dat AppImages voor de gebruiker in het systeem integreert door ze in een eigen map, standaard *\$HOME/Applications* (*\$HOME* is de thuismap van de gebruiker) te plaatsen en voor hen een desktopbestand² aan te maken in de map *\$HOME/.local/share/applications*. Dat leidt er toe dat de AppImages ook in het systeemmenu van de gebruiker verschijnen en dat hun contextmenu wordt uitgebreid met de keuzes ‘Update AppImage’ en ‘Remove AppImage from system’. De keuze ‘Update AppImage’ verschijnt overigens alleen als de AppImage daar de mogelijkheid toe biedt, en dat is lang niet altijd het geval. Bovendien zijn er versies van AppImageLauncher waarin de Update-mogelijkheid niet is opgenomen. Informatie voor het desktopbestand wordt deels ontleend aan de AppImage zelf.

2. Een *desktopbestand* is een configuratiebestand met extensie *.desktop*, dat beschrijft hoe een applicatie wordt gestart, met welke data het kan omgaan, en hoe het in een menu verschijnt met een pictogram. Systeembreed geldige desktopbestanden voor het systeemmenu staan in de map */usr/share/applications*. De alleen voor een bepaalde gebruiker geldige desktopbestanden staan in de map *\$HOME/.local/share/applications*, waarin *\$HOME* de thuismap van de gebruiker is. Applicaties kunnen ook op het bureaublad (de *desktop*) van een gebruiker worden gezet door hun desktopbestand in de map *\$HOME/Bureaublad* (ofwel *\$HOME/Desktop*) op te nemen.

AppImageHub raadt AppImageLauncher aan voor het integreren van AppImages in het systeem.

Het AppImageLauncher-pakket bevat vier programma's: *AppImageLauncher*, *appimagelauncherd*, *ail-cli* en *AppImageLauncherSettings*.

appimagelauncherd is een zgn. daemon (vandaar de ‘d’ aan het eind van zijn naam), dat is een programma dat zich in het geheugen nestelt en op de achtergrond draait. *appimagelauncherd* houdt in de gaten welke programma's worden gestart en roept het eigenlijke beheerprogramma *AppImageLauncher* aan zo gauw er een nog niet geïntegreerde AppImage wordt gestart.

Bij de eerste opstart van *AppImageLauncher* verschijnt het venster van afbeelding 3.



Afbeelding 3: AppImageLauncher eerste opstart

Daarin staat aangevinkt dat je bij start van een nieuwe AppImage zal worden gevraagd of hij naar een centrale locatie moet worden verplaatst en zo ja, wordt die centrale locatie gesteld op *\$HOME/Applications*. Beide instellingen kun je indien gewenst veranderen.

Als je het vinkje laat staan geeft *AppImageLauncher* na klikken op OK de keus uit ‘Integreren en starten’ en ‘Eén keer starten’. Bij een volgende start komt er nog de keus ‘Annuleren’ bij (afbeelding 4).



Afbeelding 4: AppImageLauncher

Bij keuze van integreren wordt de AppImage naar de standaardmap verplaatst en wordt voor het systeemmenu een desktopbestand aangemaakt. Als een al geïntegreerde AppImage wordt gestart komt AppImageLauncher niet meer tussen beide.

Als je het vinkje weghaalt moet je er zelf voor zorgen dat de AppImage naar de standaardmap wordt verplaatst. Nadat je dat hebt gedaan zorgt AppImageLauncher ervoor dat een desktopbestand wordt aangemaakt. Bij starten van een AppImage komt AppImageLauncher in dit geval nooit tussen

beide.

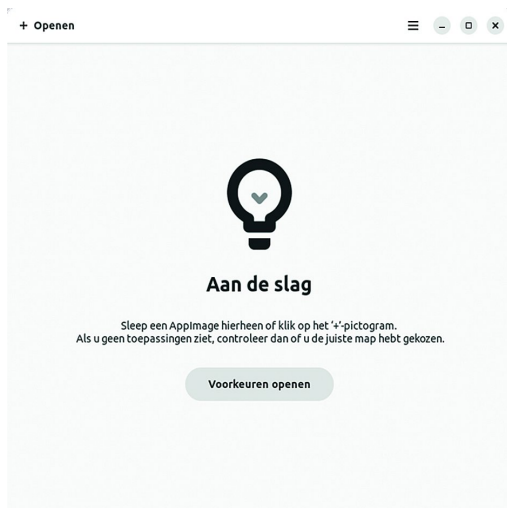
Bij verplaatsing door ApplianceLauncher van een Appliance naar de standaardmap wordt de naam van de Appliance uitgebreid met een nummer van 32 hexadecimale cijfers. Dat gebeurt niet als je de Appliance zelf verplaatst. Het desktop-bestand krijgt in beide gevallen een naam die begint met *appliancekit_*, eveneens gevolgd door een (ander) nummer van 32 hexadecimale cijfers en de naam van de applicatie. *ail-cli* is een versie van ApplianceLauncher voor gebruik op de opdrachtregel. Dat kan handig zijn voor het maken van scripts.

Met *ApplianceLauncherSettings* kunnen indien gewenst de standaard-instellingen voor ApplianceLauncher en *appliance-launcherd* worden gewijzigd.

Sommige Appliances kunnen niet in het systeem worden geïntegreerd. Dan verschijnt een foutmelding: 'Kon Appliance niet vastleggen in systeem middels libappliance'. Oorzaak lijkt te zijn dat de Appliance in kwestie informatie mist voor het te maken desktopbestand. Bij niet-integratie wordt de Appliance wel naar de standaardmap verplaatst, maar zonder uitvoeringspermissie, en wordt er geen desktopbestand aangemaakt.

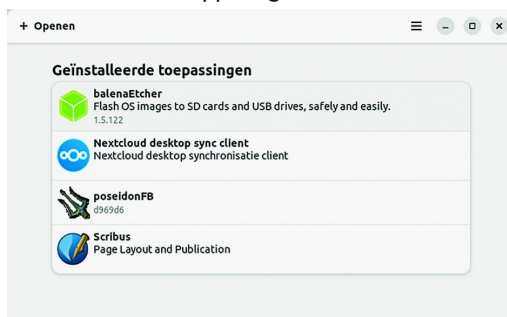
ApplianceLauncher is beschikbaar als *.deb*- en *.rpm*-pakket. Er is ook een Appliance met een 'lite' versie die zonder root-rechten draait. De laatste versie is 2.2.0 van september 2020. Enige documentatie vind je op de website ostechnix.com (zie onder Links).

Gearlever



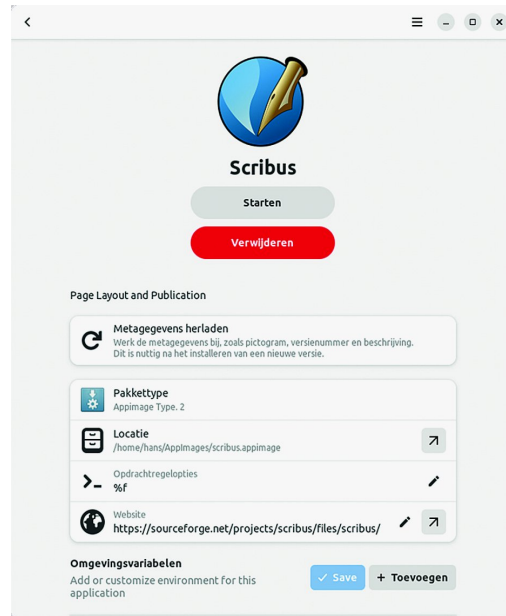
Afbeelding 5:
Gearlever -
Aan de slag

Ook Gearlever is een programma dat Appliances voor de gebruiker in het systeem integreert. Hiertoe plaatst Gearlever ze eveneens in een eigen map, standaard *\$HOME/ Appliances*, en maakt voor hen een desktopbestand aan in de map *\$HOME/.local/share/applications*. Dat leidt ertoe dat de Appliances ook in het systeemmenu van de gebruiker verschijnen. Informatie voor het desktopbestand wordt deels ontleend aan de Appliance zelf.



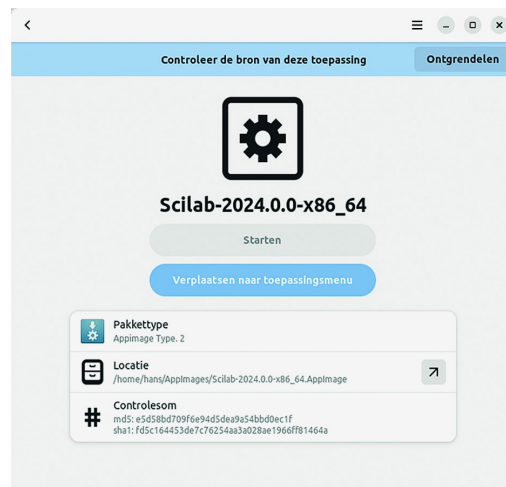
Afbeelding 6:
Gearlever toe-
passingsmenu

Gearlever toont geïntegreerde Appliances in een eigen toepassingsmenu van waaruit worden gestart en verwijderd.



Afbeelding 7:
Gearlever
venster voor
geïntegreerde
applicatie

Ook niet door Gearlever geïntegreerde Appliances kunnen naar wens in het toepassingsmenu worden opgenomen, mits ze in het systeemmenu staan (afbeelding 8). Overigens spreekt Gearlever niet van *integratie*, maar van *installatie*.



Afbeelding 8:
Gearlever ven-
ster voor niet
geïntegreerde
applicatie

Gearlever maakt geen gebruik van een daemon op de achtergrond, zoals ApplianceLauncher. Nadat Gearlever is gestart kan een Appliance worden geopend door links bovenin het Gearlever venster op '+' 'Openen' te klikken en dan een Appliance te selecteren. 'Drag and drop' op het venster van Gearlever is ook mogelijk. Omwille van de veiligheid vergrendelt Gearlever om te beginnen zo'n Appliance, maar geeft de mogelijkheid hem te ontgrendelen nadat de gebruiker de bron heeft gecheckt. Daarna kan hij worden gestart en eventueel worden geïnstalleerd.

De namen van de geïnstalleerde Appliances worden gewijzigd. Afhankelijk van de configuratie is dat ofwel een verkorte naam gevormd als *<naam applicatie>.appliance*, of een langere naam bestaand uit *gearlever_<naam applicatie>_<identificatie van zes hexadecimale cijfers>.appliance*. Ook het desktopbestand krijgt die langere naam, maar gevolgd door *.desktop* in plaats van *.appliance*. In beide gevallen staat de naam helemaal in kleine letters. De langere naam met identificatie is handig voor het naast elkaar bestaan van verschillende versies van de toepassing. Jammer is dat in beide varianten van de toegekende naam het versienummer van de applicatie ontbreekt terwijl dat meestal wel voorkomt in de oorspronkelijke naam.

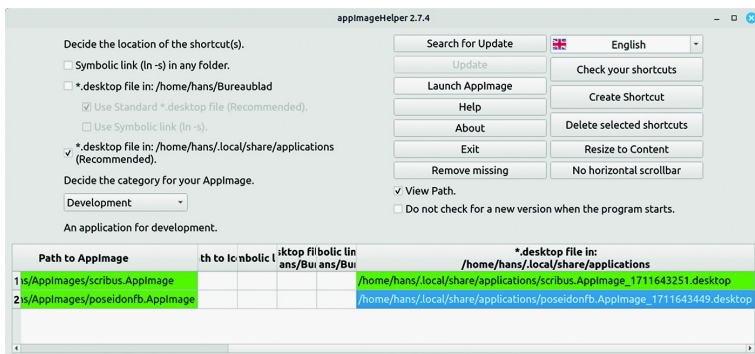
Niet alle Applimages blijken met Gearlever te kunnen worden geïnstalleerd. Ze worden wel in de installatiemap geplaatst, maar komen niet in de lijst van geïnstalleerde Applimages en er wordt ook geen desktopbestand aangemaakt. In de log van Gearlever verschijnt dan een foutmelding³.

Gearlever ondersteunt de 'portable' modus van Applimages (nog) niet.

Gearlever is beschikbaar als flatpak. In het Nederlands heet het programma 'Versnellingspook'. Met die naam komt het in het systeemmenu als Nederlands de standaardtaal is. De laatste versie is 1.5.2 van maart 2024. Enige documentatie vind je op de website ostechnix.com (zie onder Links).

applmageHelper

Met applmageHelper kan vrij eenvoudig een desktopbestand voor een Applimage worden aangemaakt door de daarvoor benodigde informatie in te voeren. Zo'n desktopbestand kan worden aangemaakt op het Bureaublad ($\$HOME/Bureaublad$) en/of in het systeemmenu ($\$HOME/.local/share/applications$).



Afbeelding 9: applmageHelper

In plaats van een desktopbestand kan op het bureaublad ook een symbolische link worden aangemaakt. Bovendien kan een symbolische link worden aangemaakt in een map naar keuze. Om de Applmage in het systeemmenu in de juiste categorie te plaatsen moet die categorie uit een lijst worden gekozen. Helaas is het niet mogelijk om meerdere categorieën op te geven. Een applicatie kan immers in meer dan één categorie passen.

Door vervolgens op de knop 'Create Shortcut' te klikken kan een nieuwe snelkoppeling (.desktop en/of symlink) worden aangemaakt. ApplmageHelper vraagt dan eerst om een pictogram (*icon*) te selecteren, en vervolgens de Applmage waarom het gaat. Een pictogram is niet nodig, zodat bij de selectie kan worden gekozen van *Cancel*. Alleen Applimages met extensie *.Applmage* worden als zodanig herkend. De extensies *.applimage* of *.applmage* werken dus niet voor ApplmageHelper.

De naam van de applicatie zoals hij in het desktopbestand wordt opgenomen en in het menu verschijnt, wordt door applmageHelper gesteld op de volledige naam van het Applmage-bestand gevolgd door *.desktop*. Je krijgt dan iets als *applimagepool-5.1.0-x86_64.Applmage.desktop* in het menu. Ik vind dat nogal onhandig. Het zou alleen de naam van de applicatie moeten zijn, in het voorbeeld dus *applimagepool*.

Er is geen specifieke locatie voor Applimages; dat wordt aan de gebruiker zelf overgelaten. Als een Applmage naderhand wordt verplaatst of verwijderd klopt het desktopbestand natuurlijk niet meer.

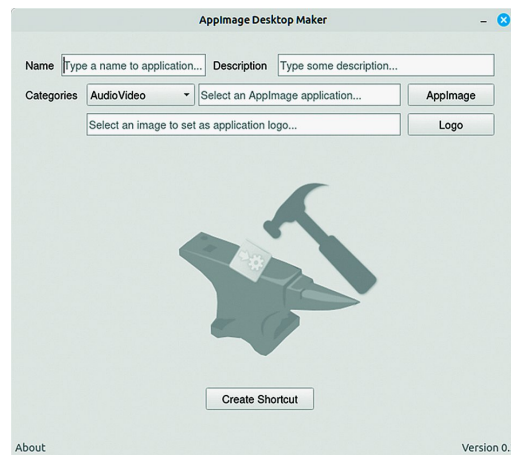
3. Zo kreeg ik bij een poging tot installatie van Scilab-2024.0.0-x86_64.Applmage de foutmelding: 'NoneType' object has no attribute 'getExec'.

De interface van applmageHelper oogt wat rommelig, zo is het niet direct duidelijk wat je moet doen. Het blijkt dus dat je eerst aan de rechterkant de gewenste locatie en de categorie moet selecteren, en daarna kun je door klikken op de knop 'Create Shortcut' aan de linkerkant de gewenste snelkoppeling aanmaken.

applmageHelper is beschikbaar als Applmage. De laatste versie is 2.35 van november 2022. Zijn voertaal is Engels of Zweeds.

Applmage Desktop Maker (AIDM)

AIDM is een simpel programma om op basis van enkele ingevoerde gegevens een desktopbestand voor het systeemmenu van de gebruiker aan te maken. Selecteer een Applmage en een bijbehorend logo, vul naam en beschrijving in en kies de meest geschikte categorie uit een lijst. Dat is alles. In feite kan dit programma worden gebruikt voor elke applicatie waarvoor niet automatisch een desktopbestand is aangemaakt.



Afbeelding 10: Applmage Desktop Maker

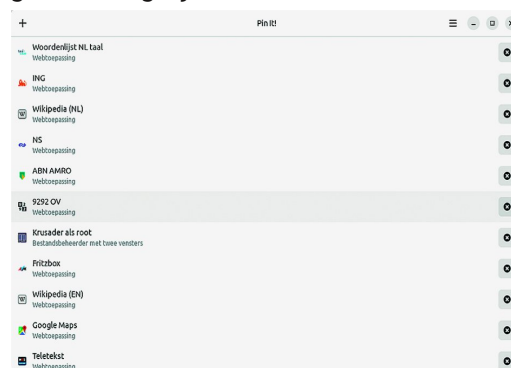
Ook AIDM heeft geen specifieke locatie voor Applimages; dat wordt aan de gebruiker zelf overgelaten.

AIDM is beschikbaar als Applmage en als deb. De laatste versie is 0.3 van juni 2023. Het is een simpel maar doeltreffend programma.

Pin It!

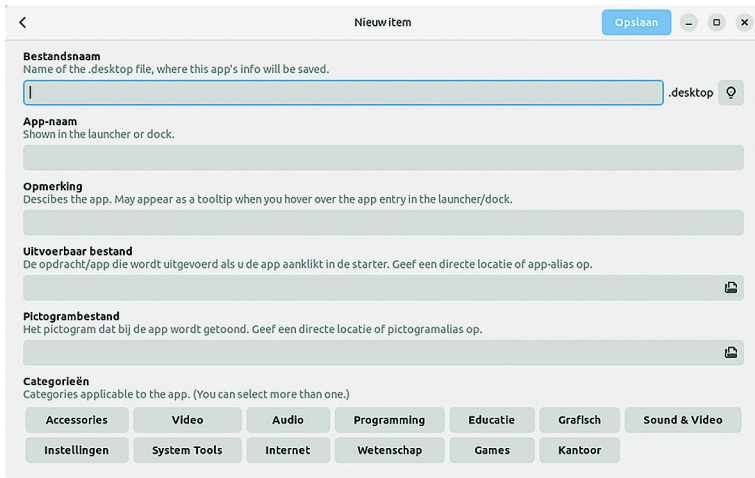
Pin It! doet hetzelfde als ApplmageHelper en AIDM, namelijk opnemen van een applicatie in het systeemmenu van de gebruiker door op basis van enkele in te voeren gegevens een desktopbestand in $\$HOME/.local/share/applications$ aan te maken. Het programma is niet alleen bedoeld voor Applimages, maar voor elke applicatie die niet automatisch in het menu is opgenomen.

Het hoofdenster van Pin It! toont alle desktopbestanden en geeft de mogelijkheid om ze te veranderen of weg te gooien.



Afbeelding 11: Pin It! hoofdenster

De volgorde waarin de desktopbestanden worden getoond is helaas onnavolgbaar. Niet alfabetisch in elk geval. Een nieuw desktopbestand kan worden toegevoegd door op de '+' links boven te klikken. Dan kunnen alle gegevens voor een desktopbestand worden ingevuld: selecteer een applicatie en een bijbehorend logo, vul naam en beschrijving in en kies de meest geschikte categorieën uit een lijst (afbeelding 12). Mooi is dat meerdere categorieën kunnen worden gekozen.



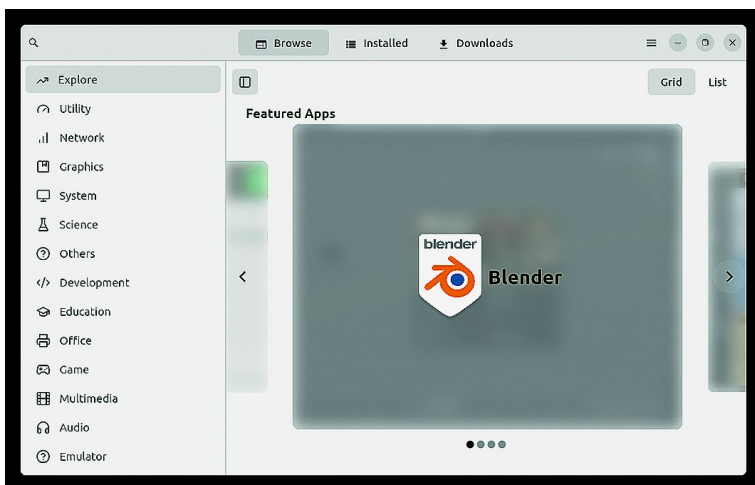
Afbeelding 12: Pin It! venster voor nieuw item

Een belangrijk verschil met ApplmageHelper en AIDM is ook dat de aangemaakte desktopbestanden naderhand gemakkelijk kunnen worden gewijzigd of verwijderd in het geval de applicatie is verplaatst, danwel verwijderd.

Pin It! is beschikbaar als flatpak. De laatste versie is 2.0.3 van januari 2024. Het is een heel handig programma voor het beheer van al die eigen desktopbestanden in je systeemmenu.

appimagepool

appimagepool is een eenvoudige ApplmageHub-client, die Applmages na download in het systeem kan integreren door ze via een desktopbestand toe te voegen aan het systeemmenu.



Afbeelding 13: appimagepool

Het programma kan in ApplmageHub opgenomen Applmages direct van de bron (meestal github.com) downloaden en installeren. De installatiemap wordt in de Preferences - lichtelijk verwarrend - 'Download Path' genoemd. Standaard is dat `$HOME/Applications`, dus gelijk aan de standaardmap voor ApplmageLauncher. Als je ApplmageLauncher al gebruikt is het misschien handig om voor appimagepool een andere map te kiezen, bijvoorbeeld `$HOME/appimagepool`. In de

Preferences kan verder de map met applicaties (in de vorm van desktopbestanden) en de map met pictogrammen worden opgegeven, standaard `$HOME/.local/share/applications` resp. `$HOME/.local/share/icons`. Het is het beste dat zo te laten.

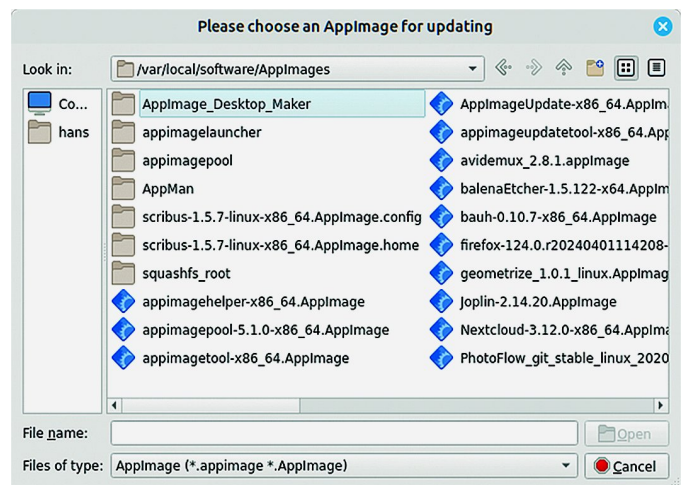
Een gedownloade Applmage verschijnt in de lijst van Downloads, maar wordt pas na herstarten van appimagepool in de lijst van geïnstalleerde Applmages getoond. Eenmaal daar kan gekozen worden voor integratie van de Applmage in het systeem. Dat houdt in dat er een desktopbestand voor wordt gemaakt in de map met applicaties, zodat hij vanuit het systeemmenu kan worden aangeroepen. Daarbij wordt zijn naam uitgebreid met een nummer van 32 hexadecimale cijfers, net zoals ApplmageLauncher doet. Het desktopbestand krijgt dezelfde uitgebreide naam voorafgegaan door 'aip_' (van appimagepool).

appimagepool heeft de mogelijkheid een portable *home* en dito *config* te maken, mits de Applmage dat ondersteunt (oudere doen dat niet). Het programma herkent alleen de extensie `.Applmage`, dus niet `.appimage`.

appimagepool is beschikbaar als Applmage en als flatpak. De laatste versie is 5.1.0 van februari 2024.

ApplmageUpdate

ApplmageUpdate is bedoeld voor het gemakkelijk bijwerken van Applmages op basis van erin opgenomen update-informatie. Voorwaarde is natuurlijk wel dat de bij te werken Applmage ingebedde update-informatie bevat. Als dat niet zo is meldt ApplmageUpdate dat de update is mislukt omdat er geen update-informatie is gevonden. Een wel geslaagde update kan eindigen met de mededeling 'Signature validation problem: Applmage not signed'. Reden te meer om te controleren of de bron betrouwbaar is.



Afbeelding 14: ApplmageUpdate

Het programma werkt goed, maar het nut ervan is betrekkelijk omdat heel veel Applmages geen ingebedde update-informatie hebben.

Het programma is beschikbaar als Applmage. De laatste versie is `2.0.0-alpha-1-20230526` van mei 2023. Er is ook een versie voor de opdrachtregel met de naam `appimageupdate-tool`, ook beschikbaar als Applmage.

Conclusie

ApplmageLauncher, Gearlever, ApplmageHelper, AIDM, Pin It!, appimagepool en ApplmageUpdate zijn nu de revue gepasseerd. Er zijn drie programma's die Applmages in het systeem integreren door ze middels een desktopbestand in het sys-

teemenu op te nemen, en de Applimages een vaste plaats te geven: ApplimageLauncher, Gearlever en appimagepool. Eveneens drie programma's integreren Applimages in het systeem alleen door een desktopbestand voor ze aan te maken zodat ze in het systeemmenu verschijnen: applimageHelper, AIDM en Pin It! Twee programma's kunnen updates van Applimages uitvoeren, mits er update-informatie in is opgenomen: ApplimageLauncher en ApplimageUpdate. Ze hebben bijna allemaal hun positieve punten en extra mogelijkheden. Alleen applimageHelper vind ik wat zwak: de interface oogt rommelig, en de naam die in het menu aan de Applimages wordt gegeven is onhandig. En AIDM doet wat het moet doen, maar heeft geen extra functionaliteit. Wat die extra functionaliteit betreft: Pin It! kun je uitstekend gebruiken

als algemeen beheerprogramma voor de eigen desktopbestanden van de gebruiker in de map `$HOME/.local/share/applications`. En *appimagepool* is een prima programma om Applimages te zoeken en indien gewenst te downloaden.

Al met al lijkt me een combinatie ApplimageLauncher, Pin It! en appimagepool ideaal, zij het dat ook Gearlever niet is te versmaden.

Een uitgebreide lijst met meer tools voor Applimage is te vinden op de website van de ApplimageCommunity op github.com onder de naam 'awesome Applimage' (Lovingly crafted Applimage tools and resources). Zie onder Links.

Links

- *Applimage*: <https://appimage.org/>, <https://itsfoss.com/use-appimage-linux/>
- *Centrale magazijn van Applimages*: <https://appimage.github.io/>
- *ApplimageHub*: <https://www.appimagehub.com/>
- *Firejail appimage-ondersteuning*: <https://firejail.wordpress.com/documentation-2/appimage-support>
- *Nitrux*: <https://nxos.org/>
- *ApplimageLauncher*: <https://github.com/TheAssassin/AppImageLauncher>, <https://ostechnix.com/integrate-appimages-to-application-menu-using-appimagelauncher/>
- *Gearlever*: <https://github.com/mijorus/gearlever>, <https://ostechnix.com/manage-appimages-with-gear-lever/>
- *applimageHelper*: <https://gitlab.com/posktomten/appimagehelper>
- *Applimage Desktop Maker*: <https://github.com/Alexsussa/AIDM>
- *Pin It!*: <https://github.com/ryonakano/pinit>
- *appimagepool*: <https://github.com/prateekmedia/appimagepool>
- *ApplimageUpdate*: <https://github.com/AppImageCommunity/AppImageUpdate>
- *awesome Applimage*: <https://github.com/AppImageCommunity/awesome-appimage>