App Central:

Developer's Guide

For APKG 2.0

Revision: 4.0.0

Update: April 9, 2021

Change log:

Date	Update Section	Description	Note
2021/04/09	1.1, 1.2 4.2	1. Add ADM 4.0 information for supported models. 2. Add new variables for network IPs.	ADM 4.0 and above

Note: For the change log on ADM 3.5 and previous version, please refer here.

asustor

Table of Content

1 System Requirements

1.1 Build Machine

1. Ubuntu 18.04 (recommended), 64-bit
2. ASUSTOR cross compiler toolchain for ADM 4.0 (Intel x86-64, ARM 64-bit, ARM 32-bit)
3. Bourne Shell
4. Python 2.7

1.2 Target Machine

Product Series	CPU	Arch
AS71	Intel 9th Xeon E	Intel x86-64
AS70	Intel Core i3 4330 Intel Core i5 i5-4590S Intel Xeon E3	Intel x86-64
AS65	Intel Atom (Denverton)	Intel x86-64
AS66	Intel Celeron (Gemini Lake-Refresh)	Intel x86-64
AS63/64	Intel Celeron (Apollo Lake)	Intel x86-64
AS61/62/31/32	Intel Celeron (Braswell)	Intel x86-64
AS52/53	Intel Celeron (Gemini Lake)	Intel x86-64
AS50/51	Intel Celeron (Bay trail)	Intel x86-64
AS40	Marvell ARMADA 7K	ARM 64-bit
AS33/11	Realtek RTD1296	ARM 64-bit
AS10	Marvell ARMADA 385	ARM 32-bit

ADM information	ADM 4.0		
Shell	BusyBox v1.31.1		
LAMP	Apache 2.4.43 (app) MariaDB 10.5.8 (app) PHP 7.3.12 (app)		
Linux Kernel **AS10** **AS31/32/61/62** **AS50/51** **AS63/64**	 3.10.70[1] 5.4.x 5.4.x 5.4.x		

[1] Supports hardware floating point.

AS70 **AS52/53** **AS66** **AS65** **AS71** **AS40** **AS33/11**	5.4.x 5.4.x 5.4.x 5.4.x 5.4.x 4.4.52 4.9.119		
The GNU C Library (in Toolchain)	GCC 7.4.0 glibc 2.27		
App Central system path	/usr/local/[2]		
App repository	/usr/local/AppCentral/[3]		
App home	/usr/local/AppCentral/$APP_NAME[4]		

Note: For the information on ADM 3.5 and previous version, please refer here.

[2] Default system folder: /usr/local/{/usr/local/{bin,etc,lib,lib64,sbin,tmp}

[3] Files should only be copied to the $APP_NAME directory. If you need to place files in other locations, you may use a soft link and link the files under /usr/local/.

[4] $APP_NAME is the "package" name defined in config.json, please refer to section 2.3 for detail.

2 About APKG

APKG is a package management system for developing and managing ASUSTOR NAS apps. Different APKG versions may have different APK (ASUSTOR application package file) formats, therefore, using the correct build tool/script is very important before building any apps.

2.1 App Central Guidelines

We review all Apps to ensure they are reliable, perform as expected, and are free of offensive material.

Before submitting new or updated Apps for review, please ensure your Apps comply with these guidelines:

1. Ensure the network port(s) used by the application is not taken.
   What network ports are used by ASUSTOR services?
2. If the application creates working folders and files, please set the owner as “admin”, group as “administrator” and access permission as “766”.
3. If the application contains functions that can browse or select directories, please only allow the user to choose directories from the data volumes. (volume1~n)

2.2 Getting Started

Before building your own apps, there is one thing you must know – config.json. It is the fundamental of each app which contains the necessary information about your app and the required environment for installation. The configurations have been divided into three categories (app, desktop and install) in config.json, and we will explain them in the next three sections respectively.

Please see section 4.3 Best Practice for the examples of config.json.

Name	Description
general	This section contains the information about the package and will be displayed in App Central.
adm-desktop	Optional. This section defines the type of this app. The App without GUI does not need to define this section. ADM will not create desktop App icon for the App.
register	Optional. This section is used for installation.

2.3 config.json: general

This section defines the basic information of this app.

Key	Description	Type	Note
general	App information section.	Object	
package	This is the package name. It is used to distinguish between different apps. It must be a unique name.	String	Unique. Only allow [a-z][A-Z][0-9][+-.] characters.
name	The app name which will be displayed in App Central.	String	
version	The version of this package.	String	
depends	The dependent package list of this package. Before a package is being installed or upgraded, these packages must be installed first.	Array(String)	1. A 2. B (>= 1.0) 3. C (<= 2.0) 4. D (= 1.5) 5. E (>= 1.0, <= 2.0)
conflicts[5]	(RESERVED) Not used.	Array(String)	
developer	Name of developer.	String	
maintainer	Name of maintainer.	String	
email	The email address of maintainer or developer.	String	Please provide validate email for users.
website	The website URL or any associated links for the package.	String	
architecture[6]	This is used to identify the CPU platform that the package can be installed and used.	String	x86-64 / arm64 / arm / any
firmware	Required minimum ADM firmware version.	String	4.0.0
model	This is used to identify the NAS models which the package can only be installed. **Please keep this key empty, if the App is not for specific model only. Just set correct key of "architecture" as well.**	Array(String)	10xx; 11xx; 33xx; 40xx; 31xx; 32xx; 50xx; 51xx; 52xx; 53xx; 61xx; 62xx; 63xx; 64xx; 65xx; 66xx; 70xx; 71xx

[5] Leave the field empty.

[6] If your App is common web applications (php, html), use 'any' for architecture.

default-lang	To specific the default language of license agreement which will be shown in the installation process.	String	cs / da/ de/ en-US / es /fi /fr-FR/ hu/ it-IT/ ja-JP/ ko-KR/ nl-NL/ no/ pl/ pt/ ru-RU/ sv/ tr/ zh-TW/ zh-CN
memory-limit	Minimum memory size requirement. The app will be disabled when the installed memory is lower than this size.	Integer(MB)	
memory-advice	Memory size advice.	Integer(MB)	
privacy-statement	The privacy statement URL path.	String	Ex: (https://xxx.xx)

Note:

1. All words are case sensitive.

2.4 config.json: adm-desktop

This section is used to define the type of this app and its default privileges. There are two major objects in this section: app & privilege

Key	Description	Type	Note
adm-desktop	App icon and privilege settings section.	Object	
app[7]	ADM desktop icon settings.	Object	
privilege[8]	App privilege settings.	Object	

2.4.1 app

This object defines the type of the app and currently there are four types of apps. They are: ***internal***, ***external***, ***webserver***, and ***custom***. Please note that both "***internal***" & "***external***" are reserved for ASUSTOR in-house development. All 3rd party developers will use "***webserver***" & "***custom***".

[7] There are four types so far, they are: internal, external, webserver, customize. Each type has its own format.
[8] This is used to define the default permission of this app.

2.4.1.1 Type: Web Apps

Web Apps are for common web applications such as **phpMyAdmin** and **WordPress**. It runs on the system built-in Apache web server. This will potentially be used by 3rd party maintainers or developers. Here is an example of this kind of App:

```
"app":{
    "type":"webserver"
},
```

Key	Description	Type	Note
app	ADM desktop icon settings	Object	
type	The type for this app	String	webserver
session-id	Send current session id when launching application	String	true / false

Note:
Please also setup "register: prerequisites" to enable httpd (Web server) if the app type is set as "webserver". Please refer to section 2.5 config.json: register for more detail information.

```
"register":{
"prerequisites":{
        "enable-service":["httpd"]
    }
}
```

2.4.1.2 Type: Custom Apps

Most 3rd party developer will use this. You can run your own web server, define the protocol, port and URL. Here is an example of this kind of App:

```
"app":{
    "type":"custom",
    "protocol":"http",
    "port": 39876,
    "url": "/"
}
```

Key	Description	Type	Note
app	ADM desktop icon settings	Object	
type	The type for this app	String	custom
protocol	The network protocol to launch this App	String	http / https
port	The port number	Integer	
url	The postfix of the URL for your web page.	String	The default URL will be the [NAS IP/Host Name]:port

2.4.2 privilege

Key	Description	Type	Note
accessible	You can define the group(s) which will be able to use this app by default. [users] represents all system users while [administrators] represents the specific user(s) who have the administration rights.	String	Administrators / users
customizable	This determines if the access rights to this app can be modified in [Access Control] -> [App Privilege]. For example, if “accessible” is set to “administrators”, but you would like to allow another non-administrator user to access the app, then you should use “true” here.	Boolean	true / false

Note:

1. All words are case sensitive.
2. “privilege” will not be able to restrict access to web applications since most of them have their own account system. It only can be used to determine whether the ADM desktop icon is visible or invisible to users.

2.5 config.json: register

This section is used to define some specific ADM settings/configuration the App needed, and ADM will help to set the correct configuration for the App while installing the App.

Example:

```
"register":{
        "symbolic-link":{
        },
        "share-folder":[
            {
                "name":"Download",
                "description":"Download default shared folder"
            }
        ],
        "port":[
            "9999",
            "55555"
        ],
        "boot-priority":{
            "start-order":20,
            "stop-order":80
        },
        "prerequisites":{
            "enable-service":[],
            "restart-service":[]
        }
    }
```

Key	Description	Type	Note
register	Install settings section	Object	
symbolic-link	The link used for create soft link to /usr/local folder in this App.	Object	/bin, /etc, /lib, /lib64, /sbin, /var
share-folder	This is where you can define the default directories (shared folders) for this app. These directories will be created automatically while installing this app, and if the specified directory already exists, the app will ignore this and just use the directory.	Array(Object)	
name	The name of this shared folder.	String	

description	The description of this shared folder.	String	
port	Port numbers of the App used. The App cannot register the port numbers which are defined for specific services (For ex: SSH, Web server, FTP, etc.). If one of the register port numbers is same with other App, users need to remove the Apps which using same port number from App Central.	Array(Integer)	
boot-priority	Priority of service start-stop.	Object	
start-order	Service start with script: S{$PRIORITY}{$APP_NAME}.	Integer	00 ~ 99
stop-order	Service stop with script: K{$PRIORITY}{$APP_NAME}.	Integer	00 ~ 99
prerequisites	After the package was installed or upgraded, these services must start or restart.		1. samba 2. afp 3. nfs 4. ftp 5. webdav 6. httpd 7. ~~mysql~~(deprecated)[3]
enable-service	These services must be started or enabled.	Array(String)	
restart-service	These services must be restarted.	Array(String)	

Note:

1. All words are case sensitive.
2. All keys of above are optional.
3. Please using "depends":["mariadb"] in config.json:general instead of using mysql in prerequisites:enable-service here. (ADM 3.5.0 and above)

3 Building Your App

In this section, we will introduce the package source structure, utilities for building apps, and final app structure.

3.1 Prepare Your Package Source

Your package source should contain at least one folder – CONTROL. You can also add self-defined folders to store other files, such as www, lib, etc.

Folder Name	Description
CONTROL[9]	This folder is used to store some necessary files, such as config.json, icons and scripts. Please refer to 3.1.1 CONTROL for more details.
www	Optional. This folder is for common web applications (php, html) which need to be run on a web server such as phpMyAdmin and Joomla!. (Apache comes with ADM and can be found under [Services] -> [Web Server])This is where the source files will be placed.
(OTHERS) [10]	Self-defined folders, such as bin, etc, lib, lib64, etc. You are free to define any new folders here.

Note:

1. The CONTROL folder name is case sensitive.

3.1.1 CONTROL

This folder is used to store app information, configuration, icons and other hook scripts. Please see the chart below.

File Name	Description	File Type
config.json	This file contains the information displayed in App Central and setting environment in the installation process.	JSON file
icon.png	MUST be 90 x 90 pixels in PNG format, which is used for manually installed packages to show in App Central and used for ADM[11] desktop Icon. Please upload 256x256 PNG format icon in	PNG image, transparent

[9] This folder is used to store app information, configuration, icons and other hook scripts.
[10] There are also other default folders such as bin, etc, lib, lib64, etc. You are free to define any new folders here.
[11] ASUSTOR Data Master, Web Desktop UI.

	Developer Corner for better quality icons.	
description.txt	The general description of the app.	Text file
changelog.txt	The change log of this revision.	Text file
license	Optional. This **folder** contains the license agreement files which will be shown in the installation process.	license text files should put into this folder
pre-install.sh	Optional. This hook script which is executed before installation.	Bourne shell script
pre-uninstall.sh	Optional. This hook script which is executed before uninstalling / upgrading.	Bourne shell script
post-install.sh	Optional. This hook script which is executed after installation / upgrading.	Bourne shell script
post-uninstall.sh	Optional. This hook script which is executed after uninstalling.	Bourne shell script
start-stop.sh	Optional. The init.d script to start and stop an app. This script is for daemon App, it will be executed automatically after booting or before power off. Ex: . /etc/script/lib/apkg_path.sh	Bourne shell script
pre-snapshot-restore.sh	Optional. This hook script which is executed before snapshot restore. (ADM 3.3 and above)	Bourne shell script
post-snapshot-restore.sh	Optional. This hook script which is executed after snapshot restore. (ADM 3.3 and above)	Bourne shell script

Note:

1. All file names are case sensitive and fixed.
2. These files are necessary:
   a. config.json
   b. icon.png
3. The hook scripts will be executed if available, or you can just ignore this.

3.2 Utilities for Building Apps

Here is the Linux script which can help you to build your own app. Please download the apkg-tool here.

3.2.1 Packing an App

Usage:

apkg-tool.py create <pkg_directory> [<destination_directory>]

Example:

root@build-machine:/as_build/apk# apkg-tool.py create download-center

3.3 Final App Structure

After executing the apkg-tool.py script, all source folders and files (as in 3.1 Prepare Your Package Source) will be compressed, and an app with the name ***PACKAGE_VERSION_ARCHITECTURE.apk*** will be generated automatically.

To check your app structure, you can decompress the apk file with uzip. You should then be able to see the following files in the apk.

File Name	Description	File Type	Mandatory
apkg-version	This file specifies the version of the apk format.	Text file	Yes
control.tar.gz	This is a compressed file in .tar.gz format containing all the files that are required for configuration and display, such as configuration files, icons, license, daemon control file, or hook scripts.	Gzipped tarball	Yes
data.tar.gz	This is a compressed file in .tar.gz format containing all source files, such as executable binary, library, or UI files.	Gzipped tarball	Yes

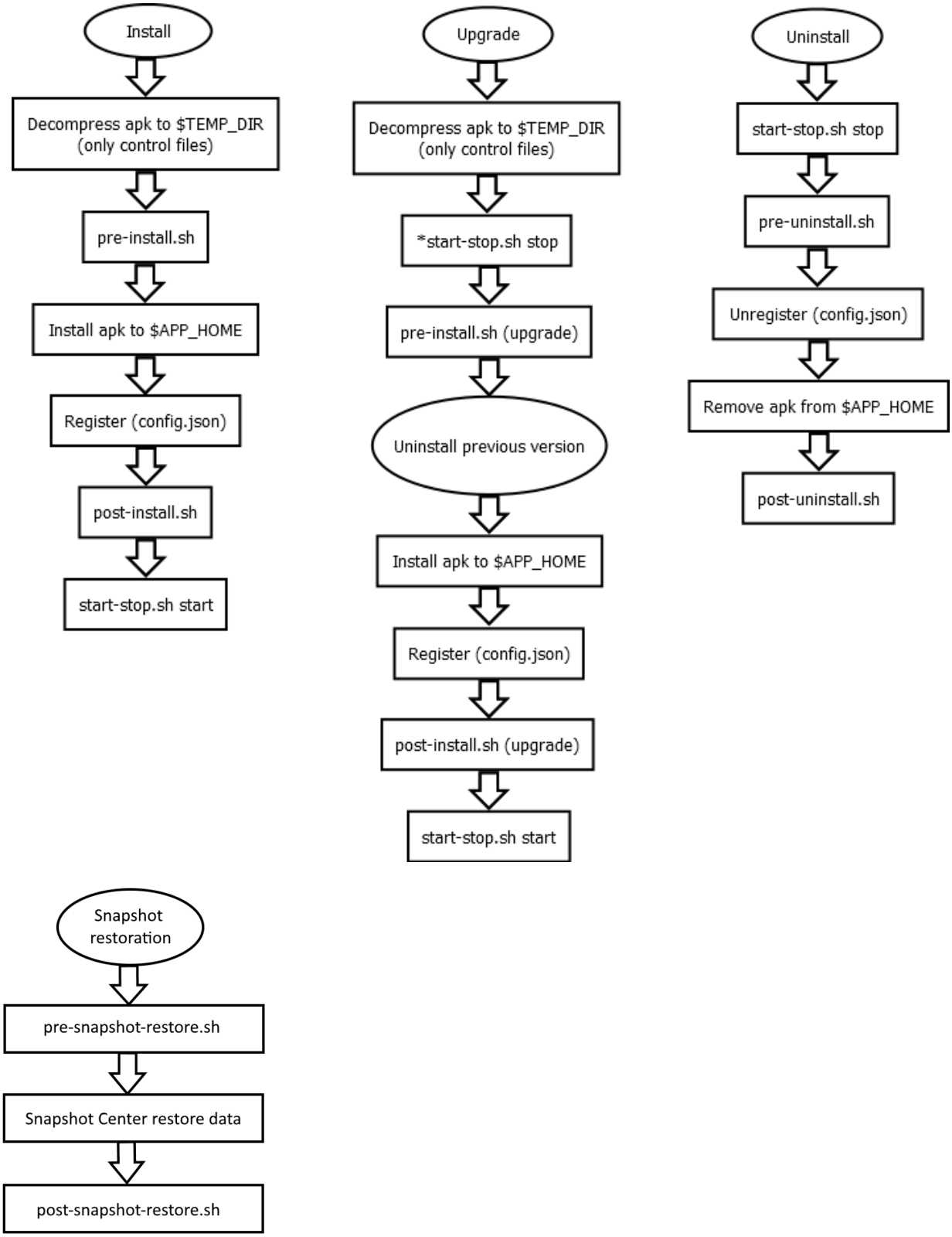
Note:

1. apkg-version file contents[12]:
   2.0
2. Package name format:
   PACKAGE_VERSION_ARCHITECTURE.apk

[12] Apkg version 2.0 is currently in use.

4 Appendix

4.1 APKG State Transition Diagram



*comes from the currently installed package.

4.2 Script Environment Variables

Several variables are exported by App Central and can be used in the scripts. Descriptions of these variables are given below:

Variable Name	Description
AS_NAS_ARCH	The type of CPU architecture.
AS_NAS_KERNEL	The version of NAS kernel.
AS_NAS_MODEL	The name of NAS model.
AS_NAS_FIRMWARE	The version of NAS firmware.
AS_NAS_HOSTNAME	The hostname of NAS.
AS_NAS_TIMEZONE	The time zone setting of NAS.
AS_NAS_INET4_IP1	The IPv4 address of first available network.
AS_NAS_INET4_IP2	The IPv4 address of second available network.
AS_NAS_INET4_ADDR_0	The IPv4 address of first network interface. (eth0)
AS_NAS_INET4_ADDR_1	The IPv4 address of second network interface if there. (eth1)
AS_NAS_INET4_ADDR_2	The IPv4 address of third network interface if there. (eth2)
AS_NAS_INET4_ADDR_3	The IPv4 address of 4th network interface if there. (eth3)
AS_NAS_INET4_ADDR_4	The IPv4 address of 5th network interface if there. (eth4)
AS_NAS_INET4_ADDR_5	The IPv4 address of 6th network interface if there. (eth5)
AS_NAS_INET4_ADDR_6	The IPv4 address of 7th network interface if there. (eth6)
AS_NAS_INET4_ADDR_7	The IPv4 address of 8th network interface if there. (eth7)
AS_NAS_INET4_BOND_0	The IPv4 address of first link aggregation network if there.
AS_NAS_INET4_BOND_1	The IPv4 address of second link aggregation network if there.
AS_NAS_INET4_BOND_2	The IPv4 address of third link aggregation network if there.
AS_NAS_INET4_BOND_3	The IPv4 address of 4th link aggregation network if there.
AS_NAS_INET4_PPPOE	The IPv4 address of PPPOE if there.
AS_NAS_INET4_WIFI	The IPv4 address of WIFI if there. (USB Wifi dongle is needed)
APKG_BASE_DIR	App Central system root.
APKG_REPO_DIR	Apps repository directory.
APKG_PKG_NAME	The package name of App which is defined in config.json.
APKG_PKG_VER	The version of App which is defined in config.json.
APKG_PKG_INST_VER	The version of App which has been installed on App Central.
APKG_PKG_DIR	App directory in which the package is stored.
APKG_PKG_STATUS	Package status can be represented by these values: install, upgrade, uninstall.

APKG_TEMP_DIR	App Central randomly generates a directory name for a script to store the configuration.

Note:

1. All words are case sensitive.
2. AS_NAS_INET4_IP1, AS_NAS_INET4_IP2, AS_NAS_INET4_ADDR_2 ~ AS_NAS_INET4_ADDR_7, AS_NAS_INET4_BOND_0 ~ AS_NAS_INET4_BOND_3, AS_NAS_INET4_PPPOE and AS_NAS_INET4_WIFI are available on ADM 4.0 and above.

4.3 Best Practice – LooksGood

LooksGood is in house application which allows user to directly stream videos from NAS to Web browser and mobile devices. Below are the samples of its config.json, source layout and package layout.

4.3.1 App Configuration – config.json

```
{
    "general":{
        "package":"looksgood",
        "name":"LooksGood",
        "version":"1.0.0.r2016",
        "depends":[
            "python(>=2.7.3.r13)",
            "xorg(>=10.14.6.377)"
        ],
        "developer":"ASUSTOR",
        "maintainer":"ASUSTOR",
        "email":"support@asustor.com",
        "website":"https://www.asustor.com/",
        "architecture":"x86-64",
        "firmware":"4.0.0"
    },
    "adm-desktop":{
        "app":{
            "type":"webserver",
            "session-id":true
        },
```

```
        "privilege":{
            "accessible":"administrators",
            "customizable":true
        }
    },
    "register":{
        "share-folder":[
          {
            "name":"Video",
            "description":"Default shared folder for LooksGood"
          }
        ],
        "prerequisites":{
            "enable-service":["httpd"],
            "restart-service":[]
        },
        "boot-priority":{
            "start-order":95,
            "stop-order":5
        },
        "port":[9900, 9901, 9902, 9903, 9904]
    }
}
```

4.3.2 Package Source Layout

Below is the package layout of Download Center. You can use the downloaded **apkg-tool.py** Linux script to build your own app automatically. Please download the apkg-tool here.

```
root@build-machine:/as_build/apk/download-center# tree
├── bin
│   └── dlcd
├── CONTROL
│   ├── config.json
│   ├── icon.png
│   ├── license.txt
│   ├── decription.txt
│   ├── changlog.txt
```

```
├── post-install.sh
├── post-uninstall.sh
├── pre-install.sh
├── pre-uninstall.sh
└── start-stop.sh
├── pre-snapshot-restore.sh
└── post-snapshot-restore.sh
├── etc
│   ├── plugin
│   │   └── rss.json
│   │   └── search.json
│   └── settings.json
├── lib
│   ├── libdlcenter.so -> libdlcenter.so.0.0
│   ├── libdlcenter.so.0 -> libdlcenter.so.0.0
│   ├── libdlcenter.so.0.0
│   ├── libevent-2.0.so.5 -> libevent-2.0.so.5.1.1
│   └── libevent-2.0.so.5.1.1
└── webman
    ├── dlcenter.cgi
    ├── downloadCenter.js
    ├── images
    │   ├── icon-app-task.png
    │   ├── icon.png
    │   ├── icon-title.png
    │   ├── left_icon1.png
    │   └── left_icon2.png
    └── langs
        └── lang-en-US.js
```

4.4 SSL certificate files path

Users can apply valid certificate with defined domain name of the NAS system, and ADM will store the certificate files on specific path. Please just "***READ***" the certificate files if the App need to use/import or check the certificate of the host domain while connecting via HTTPS.

PLEASE NOT TO UPDATE OR MODIFY THE CERTIFICATE FILES AND FILE NAMES, otherwise broken certificate files may cause abnormal behavior and malfunction of ADM.

The path of the certificate files:

/usr/builtin/etc/certificate/ssl.crt

/usr/builtin/etc/certificate/ssl.key

/usr/builtin/etc/certificate/ssl.pem