

For Developers

[Home](#)[SDK](#)

SDK

MY CLOUD OS3 NAS

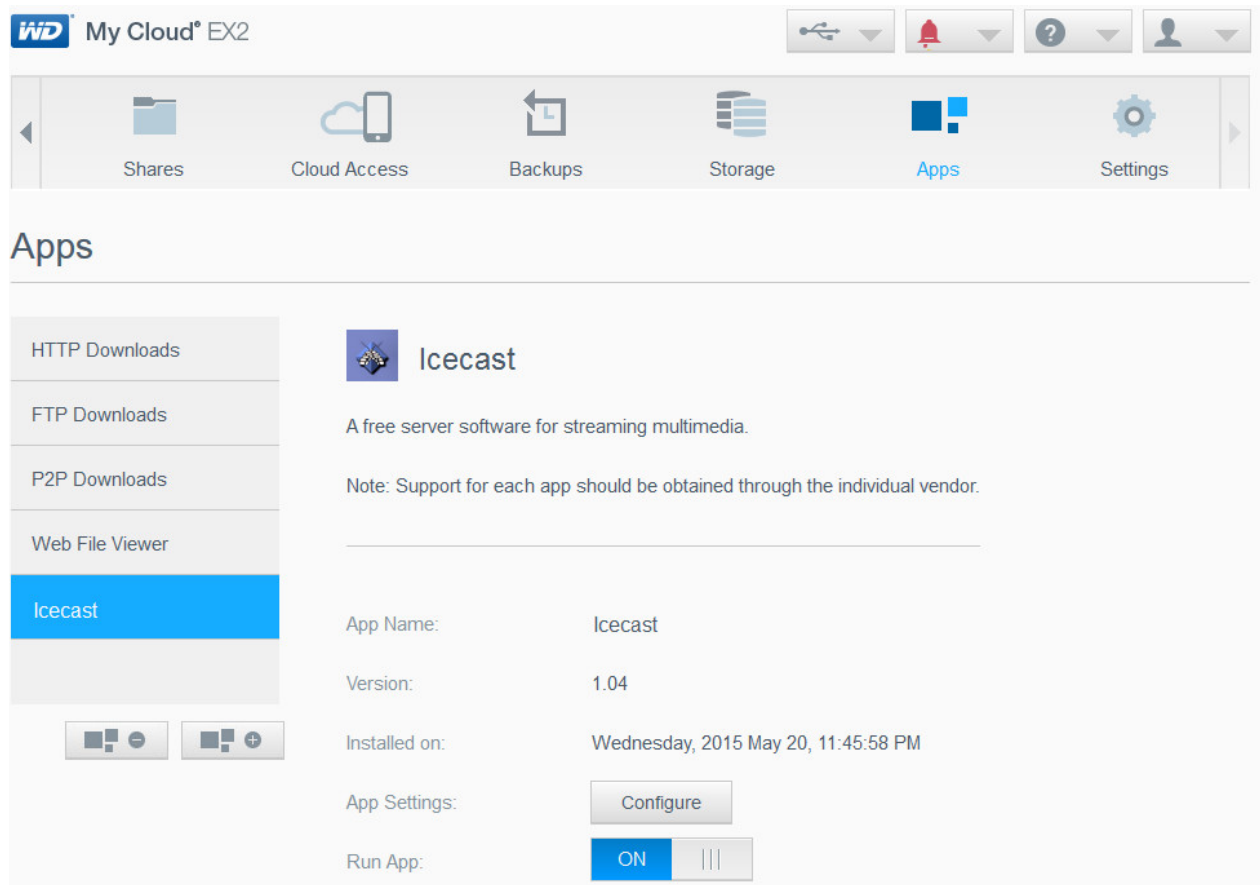
[Introduction](#)[Apps Package SDK v2.0](#)[Getting Started: Using mksapkg](#)[Write Your Own apkg.rc File](#)[Apps Package Naming Rules & Header Definition](#)[When Will Shell Script Files be Called?](#)[App Server](#)[Sample App Package Without Web UI](#)[Create Web Page](#)[Multi-Language Support](#)[Downloads](#)

INTRODUCTION

Apps Package is a small package management system that designed for the My Cloud OS3 NAS devices. We refer to Add-ons in the document as Apps. As you can see by the below image, the UI refers to Add-ons as Apps. The SDK v2.0 provides basic functions to control packages. These controls are: install, remove, start, and stop. You only need to use the templates that are contained in the SDK, which are just simple shell scripts to create an App quickly and easily without writing complex

© 2019 Western Digital Corporation or its affiliates. All rights reserved. Western Digital Technologies, Inc. is the seller of record and licensee in the Americas of SanDisk® products.

programs.



APPS PACKAGE SDK V2.0

The Apps Package SDK v2.0 includes **mksapkg** **, a tool that wraps your app into an Apps Package that can be installed on a My Cloud OS3 NAS device. It also includes a sample that guides you in creating your own app.

GETTING STARTED: USING MKSAPKG

mksapkg is a tool that helps you create your own app for My Cloud OS3 NAS devices. It is included in the Apps Package SDK v2.0, and we provide the executable binary to run your app on our system. Users must install libxml2 and GNU “tar” in their Linux system to run mksapkg.

```
Usage: mksapkg -E -s -m [module_name]
```

The parameters you can use when running mksapkg in a command line are as follows:

Parameter	Description
-E	auto-start application after install
-s	create apkg.sign for security check
-m	create application for specific NAS device

Note: each device requires its own app file. Please create a separate apps package for each device if your app is supported on multiple OS3 NAS devices. When you specify the [module_name] while packaging the app, use the following values for each supported NAS device:

module_name	NAS device
MyCloudEX2Ultra	WD My Cloud EX2 Ultra
WDMYCloudEX4100	WD My Cloud EX4100
MyCloudPR2100	WD My Cloud PR2100
MyCloudPR4100	WD My Cloud PR4100

If you are missing the xml2 library, use the following for Fedora:

```
$ yum -y install libxml2 libxml2-devel
```

WRITE YOUR OWN APKG.RC FILE

All string characters cannot contain spaces except the description field:

```
Package: utelnetd (the max length is 64)
Section: Apps
Version: 10.10 (the max length is 32)
Packager: WD (the max length is 32)
Email: support@wdc.com (the max length is 64)
Homepage: http://support.wdc.com (the max length is 64)
Description: This is a simple demonstration to wrapped telnet daemon. (the max length is 256)
AddonShowName: utelnetd (the max length is 64)
Icon: utelnetd.png
AddonIndexPage:
AddonUsedPort:
InstDepend:
InstConflict:
StartDepend:
StartConflict:
CenterType:
UserControl:
MinFWVer:
MaxFWVer:
IndividualFlag:
```

Note: the string values you add to apkg.rc will be used to build your app's entry in the App Catalog when you submit your app:

APKG.RC File	App Pages
AddonShowName	App name
Version	App version
Description	App description as it will appear in the App Catalog

Here are the definitions of variables in apkg.rc:

ABOUT USERCONTROL

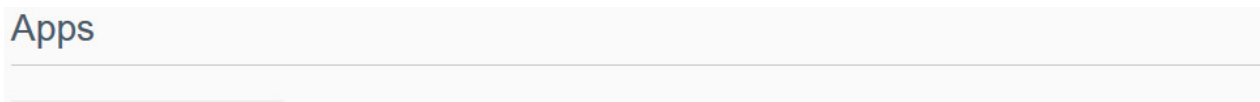
UserControl is used to control the App visibility. Based on the parameter, the App could be visible on admin account only or both admin and user accounts.

If the UserControl value is "1", only the admin can see the app.

Example:

Dropbox is set to "1", and normal user can't see the app.

Login by Admin:



HTTP Downloads


FTP Downloads

P2P Downloads

Web File Viewer

Icecast

Dropbox

 **Dropbox**

Dropbox is a file-hosting service that syncs the content between your device and your Dropbox account. This Dropbox integration is built by Western Digital for WD My Cloud. Dropbox and the Dropbox logos are trademarks of Dropbox, Inc.

App Name: Dropbox

Version: 1.08

Installed on: Wednesday, 2015 May 20, 11:49:31 PM

App Settings:

Configure

Run App:


ON

|||

Login by User:

Apps

Icecast

 **Icecast**

A free server software for streaming multimedia.

Configure

ABOUT CENTERTYPE

If the CenterType value is "1", and the setting page will build-in.

The CentherType value is "0", and the setting page will pop-out.

Setting page build-in:

Apps

HTTP Downloads

FTP Downloads

P2P Downloads

Web File Viewer

Icecast

Dropbox

Cancel Save

Settings

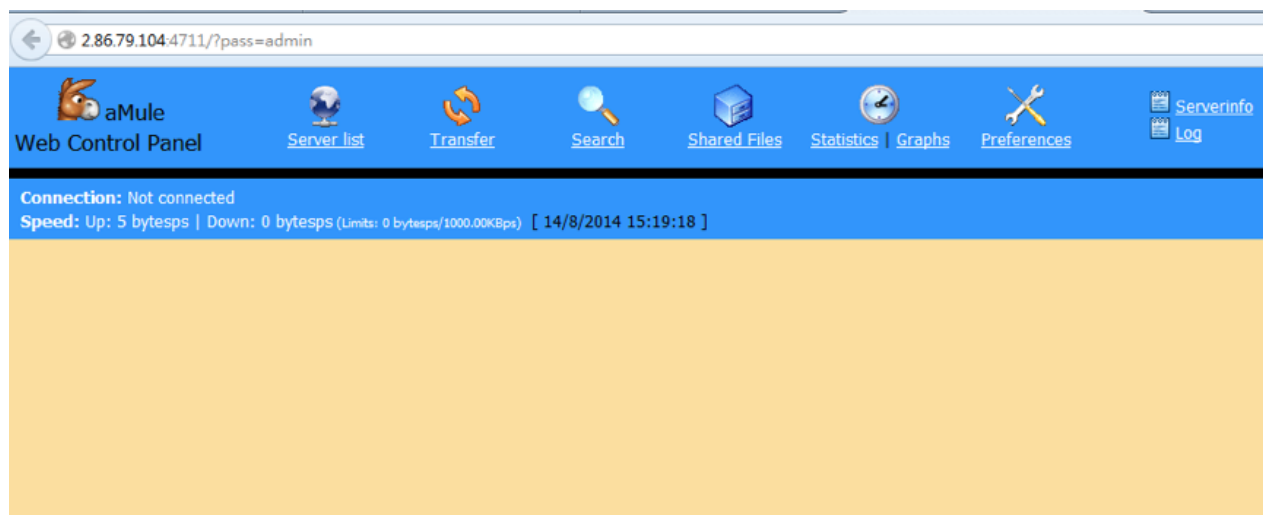
A free server software for streaming multimedia.

Port 8000

Icecast Playlist

Create

Setting page pop-out:



ABOUT MINFWVER & MAXFWVER

MinFWVer <= Current firmware <= MaxFWVer

Format: 1.xx.xx.xx (support 6 token)

If the MaxFWVer or MinFWVer is empty and it won't check the min version or max version. The version limitation will check the current firmware version when installing apps.

Example:

```
MinFWVer: 1.03.00  
MaxFWVer: 1.04.00
```

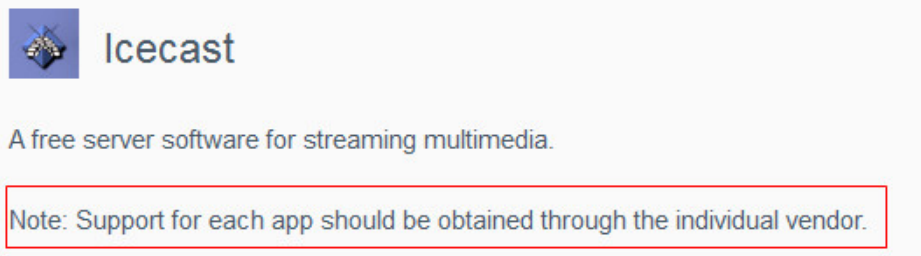
If the current firmware version is 1.02.00 or 1.05.00, App installation will return a failure.

ABOUT INDIVIDUALFLAG

WebUI will show some notice information for Apps. If the App is completed by ALPHA, we will hide the information.

Example:

Note: Support for each App should be obtained through the developer directly. The App's description in the WebUI shows support contact information.

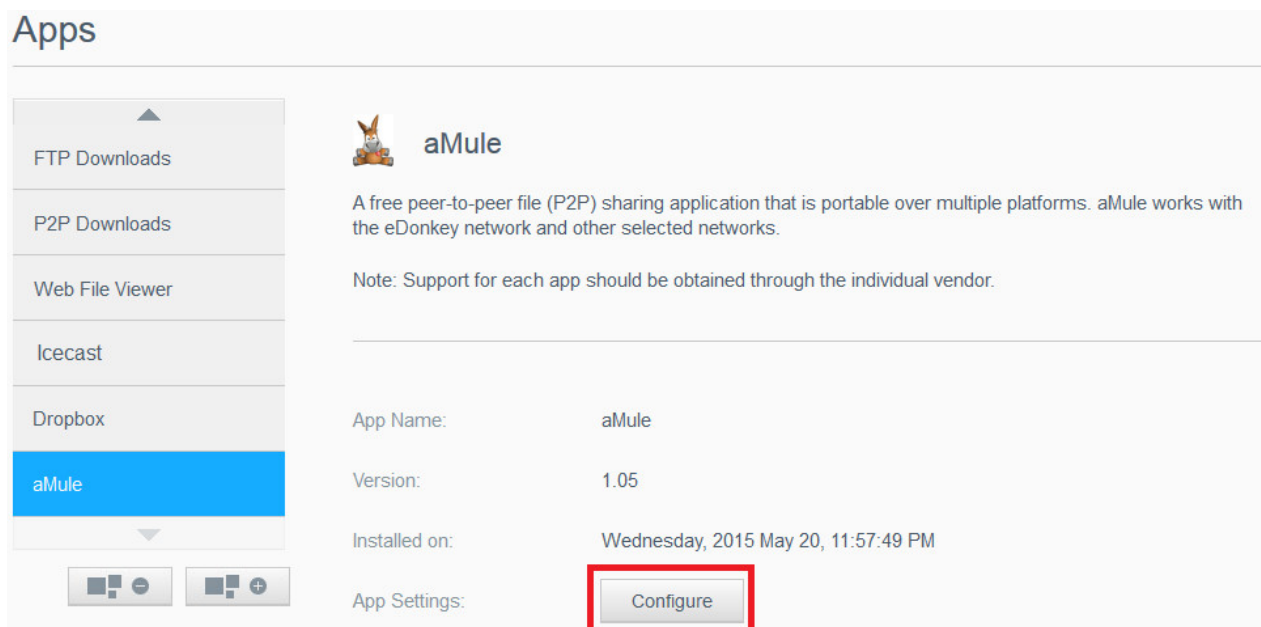


The image shows a card for the Icecast application. It features the Icecast logo (a blue square with a white 'i' and a small figure) and the text 'Icecast'. Below the logo, it says 'A free server software for streaming multimedia.' At the bottom, there is a red-bordered box containing the text: 'Note: Support for each app should be obtained through the individual vendor.'

ABOUT ADDONINDEXPAGE

If the "AddonIndexPage" is empty, the App will not provide a setting page.

The WebUI will still show the "configure" button, but it will not work.



The image shows the 'Apps' section of the WebUI. On the left, there is a sidebar with a list of apps: FTP Downloads, P2P Downloads, Web File Viewer, Icecast, Dropbox, and aMule. The aMule app is highlighted in blue. Below the sidebar, there are two buttons: a minus sign and a plus sign. The main content area shows the details for the aMule app. It includes the aMule logo (a small orange figure) and the text 'aMule'. Below the logo, it says 'A free peer-to-peer file (P2P) sharing application that is portable over multiple platforms. aMule works with the eDonkey network and other selected networks.' At the bottom, there is a red-bordered box containing the text: 'Note: Support for each app should be obtained through the individual vendor.' Below this, there is a table with the following information:

App Name:	aMule
Version:	1.05
Installed on:	Wednesday, 2015 May 20, 11:57:49 PM
App Settings:	Configure

Run App:

ON



Dependency & conflict rules defined by the App SDK v2.0:

```
Package:      A
...
InstDepend:   B
InstConflict: C
StartDepend:  D
StartConflict: E, F

_ App A can't install when App B is not installed
_ App A can't install when App C is installed
_ App A can't enable when App D is not enabled
_ App A can't enable when App E or F is enabled
```

ABOUT VERSION

Please set your version string value as **xx.xx** (ex: 0.0, 10.1, 1.10, 99.99 in whole numbers)

When running mksapkg, it will parse apkg.rc to create apkg.xml, and apkg.xml will be used by the App Server to get information about the installed App.

Sample:


```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <apkg>
    <item>
      <procudt_id>0</procudt_id>
      <custom_id>20</custom_id>
      <model_id>0</model_id>
      <user_control>0</user_control>
      <center_type>0</center_type>
      <name>utelnetd</name>
      <show>utelnetd</show>
      <enable>0</enable>
      <version>10.10</version>
      <date>20130301</date>
      <inst_date></inst_date>
      <path></path>
      <ps_name></ps_name>
      <url></url>
      <url_port></url_port>
      <apkg_version>WD</apkg_version>
      <packager>WD</packager>
      <email>support@wdc.com</email>
      <homepage>http://support.wdc.com</homepage>
      <inst_depend></inst_depend>
      <inst_conflict></inst_conflict>
      <start_depend></start_depend>
      <start_conflict></start_conflict>
      <description> This is a simple demonstration to wrapped telnet daemon.</description>
      <icon>utelnetd.png</icon>
      <MinFWVer></MinFWVer>
      <MaxFWVer></MaxFWVer>
    </item>
  </apkg>
</config>
```

```
roy@swtest5 utelnetd (git:master) $ ../mksapkg -E -s -m WDMYCloudEX4
```

```
=====
mksapkg version: v1.1
=====
```

```
utelnetd/
utelnetd/remove.sh
utelnetd/apkg.rc
utelnetd/clean.sh
utelnetd/apkg.sign
utelnetd/bin/
utelnetd/bin/utelnetd
utelnetd/utelnetd.png
utelnetd/init.sh
utelnetd/apkg.xml
utelnetd/install.sh
utelnetd/stop.sh
utelnetd/start.sh
```

```
utelneta/preinst.sh
***[../WDMycloudEX4_utelneta_1.00.bin(06192013)]

NAS type:           WDMycloudEX4
module name:        utelneta
module version:     1.00
packager:           WD

header length:      26659
header checksum:    E40297BE

Add-ons "../WDMycloudEX4_utelneta_1.00.bin(06192013)" create!
```

Notes:

- The name of the directory where mksapkg is run from must match the app name. If not, app installation will fail.
- The app name is from "package name" in the apkg.rc file.
- Because mksapkg will compress the current directory when making the app, apkg will get app name which is from apkg.rc and use the name to check. If the app name doesn't match with the directory name, app installation will fail.

SHELL SCRIPT FILES

preinst.sh: This script is used to pre-run some commands if needed, or for user to back up their configuration files to other places such as: /tmp, /var/tmp or any directory on your hard drive before installing an App.

install.sh: Will copy files and install App to an appropriate folder.

remove.sh: Will remove the installed App from hard drive.

init.sh: Will create necessary symbolic links of installed App before being executed (we suggest creating the symbolic link to /usr/bin or /usr/sbin). If necessary, restore the configuration files backed up through preinst.sh back to the App installation directory.

clean.sh: Will remove all links or files that created by init.sh.

start.sh: Will start App daemon.

stop.sh: Will stop App daemon.

APPS PACKAGE NAMING RULES AND HEADER DEFINITION

APPS PACKAGE NAMING RULE

Ex: WDMMyCloudEX4 utelnetd Package v1.00_03012013

APPS PACKAGE FILE BLOCKS

apkg header
Module tar file

HEADER STRUCTURE DEFINITION

```
typedef struct _APKG_HEADER_
{
    char          ah_magic_num[MAGIC_NUM_LEN];
    char          ah_module_name[APKG_64_LEN];
    char          ah_version[APKG_32_LEN];
    int           ah_apkg_version;
    int           ah_product_id;
    int           ah_custom_id;
    int           ah_model_id;
    char          ah_reserve[64];
    unsigned long ah_checksum;
    unsigned long ah_length;
} APKG_HEADER, *APKG_HEADER_ID;
```

Header magic number definition: the last byte of magic number must be **0x5a** in App SDK v2.0

```
WDMMyCloudEX4: {0x4c,0x69,0x67,0x68,0x74,0x6e,0x69,0x5a}
```

WHEN WILL SHELL SCRIPT FILES BE CALLED?

When the Web UI uploads an App package, it is installed to:

```
Example : module name is (MNAME) = utelnetd
default install path: (INST_PATH) = /mnt/HD/HD_a2/Nas_Prog/$MNAME
default upload path: (UPLOAD_PATH) = /mnt/HD/HD_a2/Nas_Prog/_install
```

The install and upload path is scanned dynamically by the App server when daemon is initiated.

When installing a new App, the following shell script will be called:

```
$UPLOAD_PATH/install.sh $UPLOAD_PATH $INST_PATH  
$INST_PATH/init.sh $INST_PATH
```

When removing an App, the following shell script will be called:

```
$INST_PATH/stop.sh  
$INST_PATH/clean.sh  
$INST_PATH/remove.sh $INST_PATH
```

When enabling an App, the following shell script will be called:

```
$INST_PATH/start.sh
```

When disabling an App, the following shell script will be called:

```
$INST_PATH/stop.sh
```

Apps

The screenshot shows a web interface for managing applications. On the left is a sidebar with a list of apps: FTP Downloads, P2P Downloads, Web File Viewer, Icecast, Dropbox, and aMule (which is highlighted in blue). Below the list are two icons: a square with a minus sign and a square with a plus sign. The main content area is titled 'aMule' with a small mule icon. It contains a description: 'A free peer-to-peer file (P2P) sharing application that is portable over multiple platforms. aMule works with the eDonkey network and other selected networks.' Below this is a note: 'Note: Support for each app should be obtained through the individual vendor.' Further down, there are fields for 'App Name: aMule', 'Version: 1.05', and 'Installed on: Wednesday, 2015 May 20, 11:57:49 PM'. At the bottom, there is a section for 'App Settings:' with a 'Configure' button (highlighted with a red box) and a 'Run App:' section with an 'ON' button and a pause icon.

When re-installing an App, the following shell script will be called:

```
$INST_PATH/stop.sh
$INST_PATH/clean.sh
$INST_PATH/preinst.sh $INST_PATH
$INST_PATH/remove.sh $INST_PATH
$UPLOAD_PATH/install.sh $UPLOAD_PATH $INST_PATH
$INST_PATH/init.sh $INST_PATH
```

When your app has its own configure file settings, you need to backup these files to other place in preinst.sh script file and copy files back in init.sh. Because remove.sh will remove all configuration files of this App on the hard disk and reinstall it in install.sh. Beware of incorrect shell script files which may cause the system to crash. For example, without the remove link from hard drive correctly, it may cause the system to format hard drive failure due to hard drive can't be un-mounted.

APP SERVER

- The App Server will scan all volume to find out the installed Apps and enable it, if necessary
- The App Server will control and maintain installed database of Apps
- Apps Server will provide interface to run shell script file from Web GUI

The main purpose of Apps SDK v2.0 is implemented the App dependencies and configuration check.

SAMPLE APP PACKAGE WITHOUT A WEB UI

Prepare the necessary binaries of your application, and create a folder name that is the same name as the Package field in the apkg.rc file.

```
$ ls -R
.:
apkg.rc  bin  clean.sh  init.sh  install.sh  preinst.sh  remove.sh  start.sh  stop.sh

./bin:
utelneta
```

Create your own apkg.rc file

```
Package:          utelnetd
Version:          10.10
Packager:         WD
Email:            support@wdc.com
Homepage:         http:// support.wdc.com
Description:      This is a simple demonstration to wrapped telnet daemon.
Icon:             utelnetd.png
AddonShowName:    utelnetd
AddonIndexPage:
AddonUsedPort:
InstDepend:
InstConflict:
StartDepend:
StartConflict:
CenterType:
UserControl:
MinFWVer:
MaxFWVer:
```

Write preinst.sh

```
#!/bin/sh

path_src=$1
path_des=$2
#we do nothing here for utelnetd
```

Write install.sh (most apps will use this example)

```
#!/bin/sh

path_src=$1
path_des=$2
mv $path_src $path_des
```

Write remove.sh (most apps will use this example)

```
#!/bin/sh

path=$1
rm -f /usr/bin/utelnetd 2> /dev/null
rm -f /var/www/utelnetd >/dev/null
rm -rf $path
```

Write init.sh

```
#!/bin/sh

path=$1

#create link
ln -s $path/sbin/utelnetd /usr/bin/utelnetd
ln -s $path/var/www/utelnetd /var/www

#cmd on pre-install
idx=0
while [ $idx -lt 7 ]; do
    mknod /dev/ptyp$idx c 2 $idx 2>/dev/null
    mknod /dev/ttyp$idx c 3 $idx 2>/dev/null
    idx=`expr $idx + 1`
done
```

Write clean.sh

```
#!/bin/sh

#remove link
rm -f /usr/bin/utelnetd 2> /dev/null
rm -f /var/www/utelnetd >/dev/null
rm -f /dev/ttyp*
rm -f /dev/ptyp*
```

Write start.sh

```
#!/bin/sh

#start daemon
utelnetd -d
```

Write stop.sh

```
#!/bin/sh

#stop daemon
killall utelnetd
```

Create Apps package using mksapkg

```
[hostname ~ /mksapkg/utelneta]$ ls
apkg.rc bin clean.sh init.sh install.sh preinst.sh remove.sh start.sh stop.sh
[hostname ~ /mksapkg/utelneta]$ mksapkg -E -s -m WDMYCloudEX4
=====
mksapkg version: v1.1
=====
utelneta/
utelneta/remove.sh
utelneta/apkg.rc
utelneta/clean.sh
utelneta/apkg.sign
utelneta/bin/
utelneta/bin/utelneta
utelneta/utelneta.png
utelneta/init.sh
utelneta/apkg.xml
utelneta/install.sh
utelneta/stop.sh
utelneta/start.sh
utelneta/preinst.sh
***[../WDMYCloudEX4_utelneta_1.00.bin(06192013)]

NAS type:          WDMYCloudEX4
module name:       utelneta
module version:    1.00
packager:          WD

header length:     26659
header checksum:   E40297BE

Add-ons "../WDMYCloudEX4_utelneta_1.00.bin(06192013)" create!
```

What happens when the installed App doesn't work?

If an App is not installed correctly, then the package will not install, remove, enable, or disable due to improper shell script files creation. It is easy to solve this error by removing the installed package folder and sending a rescan command to the APKG daemon.

```
# rm -rf /mnt/HD/HD_a2/Nas_Prog/utelneta
# kill -USR1 `pidof apkg`
```

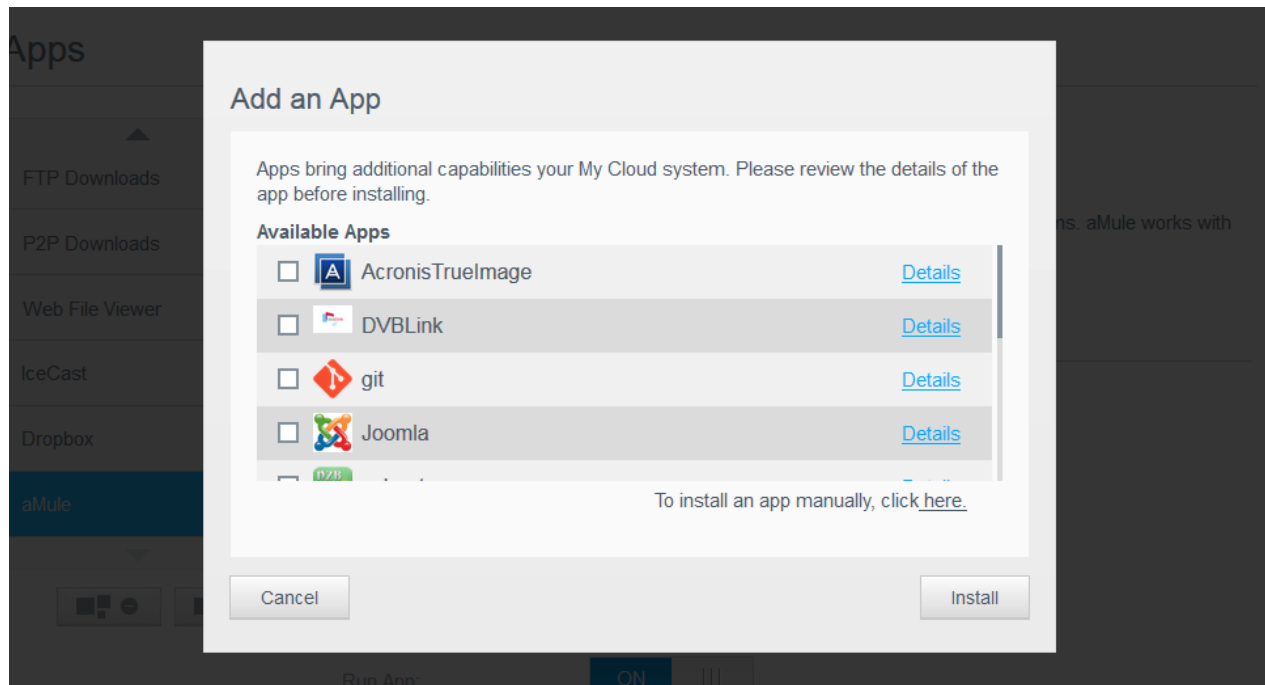
When you are at the stage of development, it is recommended not to install other Apps. This is because a clean environment will help you to reduce development time for debugging.

CREATE WEB PAGE

Check these items in the `apkg.rc` file:

```
Package:      hello
...
Icon:        hello.png
AddonShowName: hello
AddonIndexPage: index.html
AddonUsedPort:
...
```

Note: If your package does not require a GUI, make sure the `AddonIndexPage` field is empty.



In the Application page, we need to have an App icon and the recommended icon size is 128xn or nx128 ($n > 128$).

Icon naming rules: `Package_name.png`.

When package name is hello, you should rename icon file as `hello.png`. And make sure the icon can be found in the `/var/www/hello/hello.png`.

Put your `html`, `jpg`, `css` files in the webpage directory. You should link webpage to `/var/www/hello`, because the root path of web server is `/var/www`.

When you implement `cgi` functions, you should link to `/var/www/cgi-bin`. Just write it down on `init.sh` script file. And remember to add `remove` link command on `clean.sh` script file.

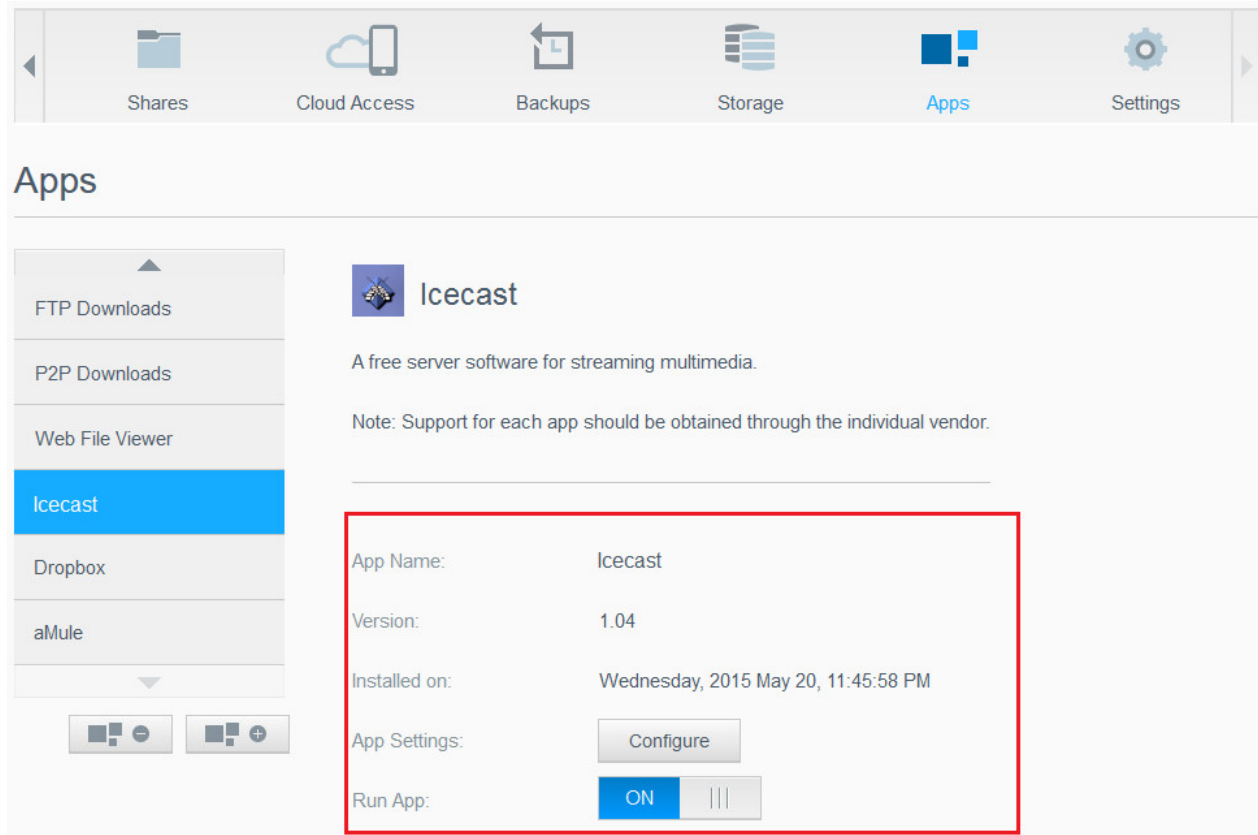
```
ln -s $path/webpage /var/www/hello
ln -s $path/cgi/* /var/www/cgi-bin/
```

Make sure the path of index file is correct. ex:

```

AddonIndexPath:    index.html -> you should have this file in /var/www/hello/index.html
AddonIndexPath:    /web/index.html -> you should have this file in /var/www/hello/web/index.html

```



MULTI-LANGUAGE SUPPORT

We support multiple languages in the app's description. Please put language xml file to Addon_Name/web/desc.xml and link the language file to /var/www/Addon_Name in init.sh when creating the addon package.

1. Put language xml file to the add-on folder.

```

[LIGHTNING-4A] vodka@swtest6 web (git:master) $ pwd
/home/vodka/wdic-all/module/apkg/addons/common/aMule/web
[LIGHTNING-4A] vodka@swtest6 web (git:master) $ ls
aMule_sub.html  css  desc.xml  gui_info.xml  help  aMule.html  images  index.html  js
[LIGHTNING-4A] vodka@swtest6 web (git:master) $

```

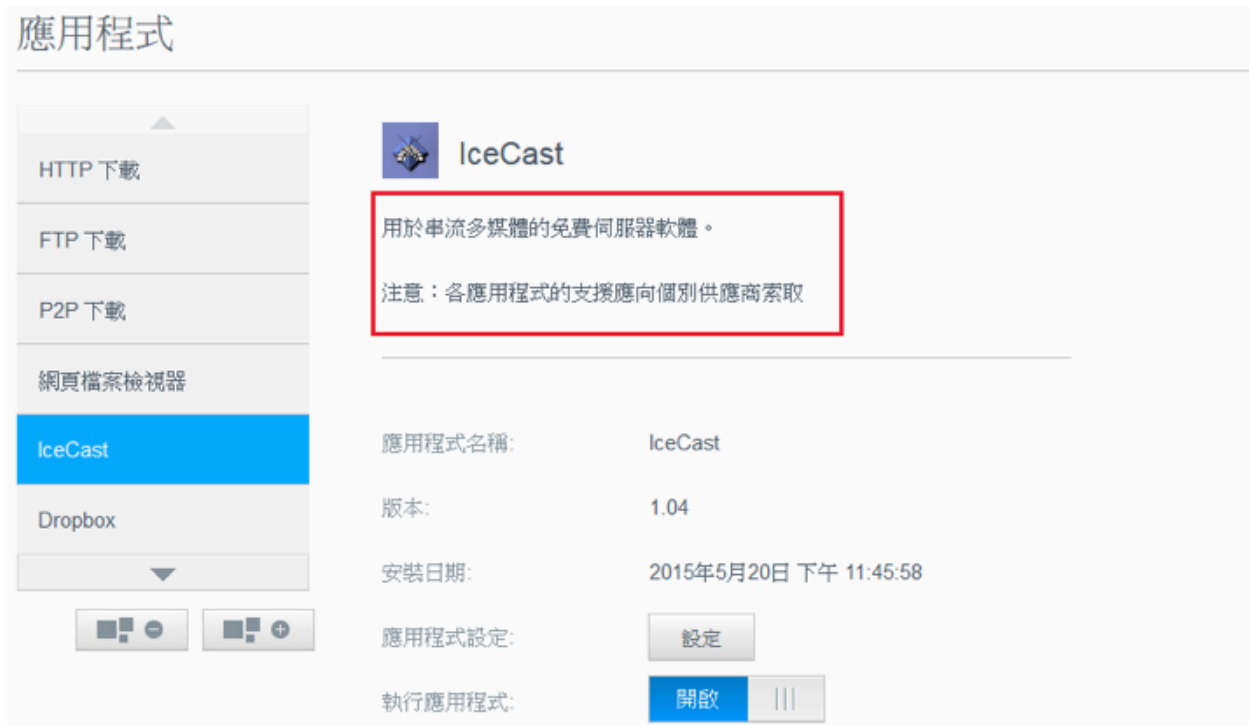
2. Add link command for the desc.xml file to init.sh

```
WEBPATH="/var/www/aMule/"
Path="" /mnt/HD/HD_a2/Nas_Prog/aMule"

ln -s $path/web/desc.xml $WEBPATH
```

You can put "desc.xml" to anywhere in add-on package, but you must link the page to its folder.

The Web UI will check the "desc.xml" file to show the indicated language.



The desc.xml example for Icecast:

```
<?xml version="1.0" encoding="utf-8"?>
<config>
  <en-US>A free server software for streaming multimedia.</en-US>
  <cs-CZ>Bezplatný serverový software pro vysílání multimediálního obsahu.</cs-CZ>
  <de-DE>Eine kostenlose Serversoftware für das Streaming von Multimedia-Inhalten.</de-DE>
  <es-ES>Software gratuito de servidor para transmitir contenido multimedia.</es-ES>
  <fr-FR>logiciel serveur gratuit de diffusion de contenus multimédias.</fr-FR>
  <hu-HU>Ingyenes kiszolgálószoftver a multimédiás tartalmak adatfolyam útján történő továbbításához</hu-HU>
  <it-IT>software server gratuito per lo streaming dei contenuti multimediali.</it-IT>
  <ja-JP>マルチメディア ストリーミング用の無料サーバー ソフトウェア。</ja-JP>
  <ko-KR>멀티미디어 스트리밍을 위한 무료 서버 소프트웨어.</ko-KR>
  <no-NO>Et gratis serverprogram for direkteavspilling av multimedia.</no-NO>
  <nl-NL>Een gratis serversoftware voor het streamen van multimedia.</nl-NL>
  <pl-PL>Bezpłatne oprogramowanie serwerowe do strumieniowych transmisji danych multimedialnych.</pl-PL>
  <pt-BR>Um software de servidor gratuito para fazer transmissão multimídia.</pt-BR>
  <ru-RU>Бесплатное серверное программное обеспечение для потокового воспроизведения мультимедиа.</ru-RU>
  <sv-SE>En gratis serverprogramvara för strömning av multimedia.</sv-SE>
  <tr-TR>Multimedya akılna yönelik ücretsiz bir yazılım.</tr-TR>
  <zh-CN>用于流式传输多媒体的免费服务器软件。</zh-CN>
  <zh-TW>用於串流多媒體的免費伺服器軟體。</zh-TW>
</config>
```

Note: the translated app descriptions you add to desc.xml will be used to build your app's entry in the App Catalog when you submit your app.

As you can see from the desc.xml example, your app can be added to the App Catalog in the following regions & locales:

- en-US - English (United States)
- cs-CZ - Czech (Czechia)
- de-DE - German (Germany)
- es-ES - Spanish (Spain)
- fr-FR - French (France)
- hu-HU - Hungarian (Hungary)
- it-IT - Italian (Italy)
- ja-JP - Japanese (Japan)
- ko-KR - Korean (South Korea)
- no-NO - Norwegian (Norway)
- nl-NL - Dutch (Netherlands)
- pl-PL - Polish (Poland)

pt-BR - Portuguese (Brazil)

ru-RU - Russian (Russia)

sv-SE - Swedish (Sweden)

tr-TR - Turkish (Turkey)

zh-CN - Chinese (China)

zh-TW - Chinese (Taiwan)

DOWNLOADS

Visit the Downloads page to get the Apps Package SDK & compilers **

** Requires a free Developer Portal account - [click here to register](#)

DO MORE WITH THE MY CLOUD OS3 NAS SDK

Click the button below to visit the My Cloud OS3 NAS Developer Home page

DEVELOPER HOME

Get started, access the SDK, build your apps, & learn more

[View Home](#)