How do I create a nice-looking DMG for Mac OS X using command-line tools?

Asked 16 years, 3 months ago Modified 1 month ago

Viewed 122k times

237

I need to create a nice installer for a Mac application. I want it to be a disk image (DMG), with a predefined size, layout and background image.

I need to do this programmatically in a script, to be integrated in an existing build system (more of a pack system really, since it only create installers. The builds are done separately).

I already have the DMG creation done using "hdiutil", what I haven't found out yet is how to make an icon layout and specify a background bitmap.

macos scripting installation dmg

Share

Improve this question

Follow

edited Feb 22, 2011 at 13:26

community wiki

4 revs, 2 users 100%
Ludvig A Norin

1 Isn't this something for Ask Different? – Lenar Hoyt Aug 30, 2011 at 23:18

2 github.com/LinusU/node-appdmg – Prof. Falken Nov 17, 2014 at 7:45

Since this has so many upvotes, it should be noted that every JDK now comes with jpackage which can create .dmg files. No need for third party tools. – VGR Feb 28, 2023 at 16:21

16 Answers

Sorted by: Highest score (default)

213

After lots of research, I've come up with this answer, and I'm hereby putting it here as an answer for my own question, for reference:

1. Make sure that "Enable access for assistive devices" is checked in System Preferences>>Universal Access. It is required for the AppleScript to work. You may have to reboot after this change (it doesn't work otherwise on Mac OS X Server 10.4).
2. Create a R/W DMG. It must be larger than the result will be. In this example, the bash variable "size" contains the size in Kb and the contents of the folder in the "source" bash variable will be copied into the DMG:

```
hdiutil create -srcfolder "${source}" -
volname "${title}" -fs HFS+ \
```

```
      -fsargs "-c c=64,a=16,e=16" -format
UDRW -size ${size}k pack.temp.dmg
```

3. Mount the disk image, and store the device name (you might want to use sleep for a few seconds after this operation):

```
device=$(hdiutil attach -readwrite -noverify
-noautoopen "pack.temp.dmg" | \
         egrep '^/dev/' | sed 1q | awk
'{print $1}')
```

4. Store the background picture (in PNG format) in a folder called ".background" in the DMG, and store its name in the "backgroundPictureName" variable.

5. Use AppleScript to set the visual styles (name of .app must be in bash variable "applicationName", use variables for the other properties as needed):

```
echo '
   tell application "Finder"
     tell disk "'${title}'"
           open
           set current view of container
window to icon view
           set toolbar visible of container
window to false
           set statusbar visible of
container window to false
           set the bounds of container
window to {400, 100, 885, 430}
           set theViewOptions to the icon
view options of container window
           set arrangement of
theViewOptions to not arranged
           set icon size of theViewOptions
to 72
           set background picture of
theViewOptions to file
```

```
".background:'${backgroundPictureName}'"
           make new alias file at container
window to POSIX file "/Applications" with
properties {name:"Applications"}
           set position of item
"'${applicationName}'" of container window
to {100, 100}
           set position of item
"Applications" of container window to {375,
100}
           update without registering
applications
           delay 5
           close
     end tell
   end tell
' | osascript
```

6. Finialize the DMG by setting permissions properly, compressing and releasing it:

```
chmod -Rf go-w /Volumes/"${title}"
sync
sync
hdiutil detach ${device}
hdiutil convert "/pack.temp.dmg" -format UDZO
-imagekey zlib-level=9 -o "${finalDMGName}"
rm -f /pack.temp.dmg
```

On Snow Leopard, the above applescript will not set the icon position correctly - it seems to be a Snow Leopard bug. One workaround is to simply call close/open after setting the icons, i.e.:

```
..
set position of item "'${applicationName}'" of
container window to {100, 100}
set position of item "Applications" of container
window to {375, 100}
```

```
close
open
```

Share Improve this answer Follow

edited Oct 14, 2014 at 10:56

community wiki
4 revs, 4 users 66%
Ludvig A Norin

1 Excellent. Two questions about that: 1. You recommend `sleep` after mounding the image. How long? Isn't it possible to deterministically wait for the completion of the process? Same for `delay 5` in your AppleScript. I'm always wary of such arbitrary wait times, having had some very bad experience with them. 2. In your step 6 you call `sync` twice – why? – Konrad Rudolph Oct 3, 2009 at 20:05

I haven't found any way to deterministically wait for the completion of the 'update without registering applications' command. I am not sure sleep is needed after "hdiutil attach", you'll have to check the documentation (man hdiutil). Sync should only be needed to be called once, I do it twice out of old habit since the good old SunOS days. – Ludvig A. Norin Oct 3, 2009 at 21:51

1 It allows applescript to simulate mouse and keyboard input and a bunch of other things. – Ludvig A. Norin Jul 28, 2011 at 9:43

1 If the script hangs until it times out at the "update without registering applications" line, then step 1 hasn't been done (or has become undone). I just found this out the hard way. – Jay Maynard K5ZC May 1, 2012 at 4:25

I think I found a slight glitch - there is an eject in the AppleScript, which is done before you try to run chmod.

chmod will fail because the disk is now ejected. – Hakanai Nov 21, 2013 at 6:40

65

There's a little Bash script called create-dmg that builds fancy DMGs with custom backgrounds, custom icon positioning and volume name.

I've built it many years ago for the company that I ran at the time; it survives on other people's contribution since then, and reportedly works well.

There's also node-appdmg which looks like a more modern and active effort based on Node.js; check it out as well.

Share Improve this answer Follow

edited Feb 10, 2016 at 11:51

community wiki
6 revs, 3 users 79%
Andrey Tarantsov

7 The link to your bash script is broken. Can you put it in a gist on github.com? Thanks – dquimper Jan 16, 2013 at 21:32

2 @laike9m It's been more than 5 years since I've last touched it. I don't really consider DMGs the best way to distribute Mac software any more. So it survives on contributions of others — hopefully someone will figure out the icons issue too. – Andrey Tarantsov Sep 10, 2015 at 15:40

1 @AndreyTarantsov Which way do you think is the best for Mac apps distribution? – Zmey Apr 7, 2016 at 14:47

4 @Zmey Well as a user, I prefer zip files by far. Bonus points for auto-copying to `/Applications` if the app detects it's running from `~/Downloads`. See also this old article by John Gruber. – Andrey Tarantsov Apr 7, 2016 at 14:52

1 @Zmey Also see this article with a screenshot of the UI involved. – Andrey Tarantsov Apr 7, 2016 at 14:55

42

Don't go there. As a long term Mac developer, I can assure you, no solution is really working well. I tried so many solutions, but they are all not too good. I think the problem is that Apple does not really document the meta data format for the necessary data.

Here's how I'm doing it for a long time, very successfully:

1. Create a new DMG, writeable(!), big enough to hold the expected binary and extra files like readme (sparse might work).
2. Mount the DMG and give it a layout manually in Finder or with whatever tools suits you for doing that. The background image is usually an image we put into a hidden folder (".something") on the DMG. Put a copy of your app there (any version, even outdated one will do). Copy other files (aliases, readme, etc.) you want there, again, outdated versions will do just fine. Make sure icons have the right sizes and positions (IOW, layout the DMG the way you want it to be).

3. Unmount the DMG again, all settings should be stored by now.
4. Write a create DMG script, that works as follows:

- It copies the DMG, so the original one is never touched again.
- It mounts the copy.
- It replaces all files with the most up to date ones (e.g. latest app after build). You can simply use *mv* or *ditto* for that on command line. Note, when you replace a file like that, the icon will stay the same, the position will stay the same, everything but the file (or directory) content stays the same (at least with ditto, which we usually use for that task). You can of course also replace the background image with another one (just make sure it has the same dimensions).
- After replacing the files, make the script unmount the DMG copy again.
- Finally call hdiutil to convert the writable, to a compressed (and such not writable) DMG.

This method may not sound optimal, but trust me, it works really well in practice. You can put the original DMG (DMG template) even under version control (e.g. SVN), so if you ever accidentally change/destroy it, you can just go back to a revision where it was still okay. You can add the DMG template to your Xcode project, together with all other files that belong onto the DMG

(readme, URL file, background image), all under version control and then create a target (e.g. external target named "Create DMG") and there run the DMG script of above and add your old main target as dependent target. You can access files in the Xcode tree using ${SRCROOT} in the script (is always the source root of your product) and you can access build products by using ${BUILT_PRODUCTS_DIR} (is always the directory where Xcode creates the build results).

Result: Actually Xcode can produce the DMG at the end of the build. A DMG that is ready to release. Not only you can create a release DMG pretty easy that way, you can actually do so in an automated process (on a headless server if you like), using xcodebuild from command line (automated nightly builds for example).

Share Improve this answer Follow

edited Nov 12, 2021 at 10:42

answered Sep 18, 2008 at 21:11

Mecki

133k ● 34 ● 259 ● 269

3 I've already discarded the idea of doing it this way, for several reasons. Here are two of them: the contents of the installers will vary with product, and we want to rely only on software installed on the pack machines and scripts - a single, minimal, manual routine for adding new products.
– Ludvig A. Norin Sep 19, 2008 at 11:20

This is the same scenario as we have. We have more than a dozen of products; each has an entirely different DMG.

Creating one template DMG per product is one time only task and takes you a couple of minutes. And what do you mean by "installer"? PKG/MPKG install packages? – Mecki Sep 19, 2008 at 11:44

It's not the same scenario. We add products often, with short notice. The minimal manual routine means running a script giving the product a name and a few other attributes. There are reasons beyond this as well that made us make the decision not to use that kind of solution. – Ludvig A. Norin Sep 19, 2008 at 21:39

We have separated the pack process from the build process, because it is done by different people at different times. The pack process creates installers for Windows, Windows CE, Symbian, AIX, Linux and Solaris as well. – Ludvig A. Norin Sep 19, 2008 at 22:25

You are probably refering to hdiutil, not hdutil. – Ivan Vučica Aug 18, 2009 at 12:45

25

Bringing this question up to date by providing this answer.

`appdmg` is a simple, easy-to-use, open-source command line program that creates dmg-files from a simple json specification. Take a look at the readme at the official website:

https://github.com/LinusU/node-appdmg

Quick example:

1. Install appdmg

```
npm install -g appdmg
```

2. Write a json file (`spec.json`)

```
{
  "title": "Test Title",
  "background": "background.png",
  "icon-size": 80,
  "contents": [
    { "x": 192, "y": 344, "type": "file",
"path": "TestApp.app" },
    { "x": 448, "y": 344, "type": "link",
"path": "/Applications" }
  ]
}
```

3. Run program

```
appdmg spec.json test.dmg
```

(disclaimer. I'm the creator of appdmg)

Share Improve this answer Follow

answered Jan 23, 2015 at 15:12

community wiki
Linus Unnebäck

4 simple and effective. The only downside is it requires npm to be installed – Jack James Jul 21, 2015 at 21:32

@Creator: can you please make it advance like this one offer GUI without having to drag and drop? s.sudre.free.fr/Software/Packages/about.html – user285594 Jan 14, 2017 at 21:41

If the .app file isn't in the same folder as the json file, this doesn't work, it gives "file not found" error when you specify a

relative path in "path" – Joey Mar 23, 2017 at 22:22

@Joey, could you open an issue on the Github repository if it doesn't work for you? – Linus Unnebäck Apr 3, 2017 at 7:51

@Joey: It is not required to be in the same folder, I have background, app all in different path, just pass relative paths as ../../../ABC – Anoop Vaidya Jan 11, 2019 at 13:21

24

For those of you that are interested in this topic, I should mention how I create the DMG:

```
hdiutil create XXX.dmg -volname "YYY" -fs HFS+ -
srcfolder "ZZZ"
```

where

```
XXX == disk image file name (duh!)
YYY == window title displayed when DMG is opened
ZZZ == Path to a folder containing the files that
will be copied into the DMG
```

Share Improve this answer

Follow

answered Sep 18, 2008 at 21:13

Ludvig A. Norin

5,325 ● 7 ● 33 ● 34

1 That's fine, but it doesn't address the actual need (background image, positioning of items in the folder, etc.) – Tom Bogle Jul 28, 2017 at 6:10

created DMG but again i have to run my Script(.sh) using CMD i need its automatically run the sh after creating DMG – SANTOSH Jul 16, 2019 at 11:56

14

My app, DropDMG, is an easy way to create disk images with background pictures, icon layouts, custom volume icons, and software license agreements. It can be controlled from a build system via the "dropdmg" command-line tool or AppleScript. If desired, the picture and license RTF files can be stored under your version control system.

Share Improve this answer Follow

answered Jun 30, 2010 at 15:38

community wiki
Michael Tsai

3 My team has this working beautifully and automatically on a Jenkins CI build server, complete with background, drag to Applications alias. Run, don't walk, to DropDMG for making disk images. – Paul Collins Dec 6, 2013 at 18:43

Good App, I'm going to purchase it after my trial expires. – kilik52 Feb 2, 2014 at 9:16

Looks nice, but it does not seem to have option to resize the window. – avi Mar 31, 2015 at 5:09

@avi DropDMG automatically sizes the window to the background picture that you set (or inset into that picture if desired). – Michael Tsai Apr 1, 2015 at 13:08

Great, thanks! Is it possible to resize without background image? – avi Apr 1, 2015 at 13:38

9

For creating a nice looking DMG, you can now just use some well written open sources:

- create-dmg
- node-appdmg
- dmgbuild

Share Improve this answer Follow

edited Sep 27, 2017 at 11:31

community wiki
2 revs, 2 users 95%
Anni S

Might be they moved it. You can use create-dmg and node-appdmg. I'm using create-dmg. Its good. – Anni S Jun 19, 2016 at 8:01

1 @PamelaCook-LightBeCorp In case you are still interested. The link to dmgbuild has been fixed. – SubOptimal Sep 27, 2017 at 11:31

6

I found this great mac app to automate the process - http://www.araelium.com/dmgcanvas/ you must have a look if you are creating dmg installer for your mac app

Share Improve this answer

Follow

answered Feb 22, 2012 at 14:05

community wiki
Saurabh

1 Note that this is not a free program, so it's not suitable for many environments. – Jay Maynard K5ZC May 1, 2012 at 4:14

5

If you want to set custom volume icon then use below command

```
/*Add a drive icon*/
cp "/Volumes/customIcon.icns"
"/Volumes/dmgName/.VolumeIcon.icns"


/*SetFile -c icnC will change the creator of the
file to icnC*/
SetFile -c icnC /<your path>/.VolumeIcon.icns
```

Now create read/write dmg

```
/*to set custom icon attribute*/
SetFile -a C /Volumes/dmgName
```

Share Improve this answer

Follow

edited Jul 15, 2014 at 4:43

community wiki

2 revs
Parag Bafna

Can you explain what does the "your path" mean here? Can I use any icons file on the disk, and SetFile would copy it, or would I need to use a file that is inside .dmg? I only have one Mac, so it's hard to test if stuff will work on other machines.
– Milan Babuškov Jul 14, 2014 at 17:56

"your path" is DMG name. (/Volumes/dmgName)
– Parag Bafna Jul 15, 2014 at 4:40

You should copy icns file. cp "/Volumes/customIcon.icns" "/Volumes/dmgName/.VolumeIcon.icns" – Parag Bafna Jul 15, 2014 at 4:42

5

I finally got this working in my own project (which happens to be in Xcode). Adding these 3 scripts to your build phase will automatically create a Disk Image for your product that is nice and neat. All you have to do is build your project and the DMG will be waiting in your products folder.

Script 1 (Create Temp Disk Image):

```
#!/bin/bash
#Create a R/W DMG

dir="$TEMP_FILES_DIR/disk"
dmg="$BUILT_PRODUCTS_DIR/$PRODUCT_NAME.temp.dmg"

rm -rf "$dir"
mkdir "$dir"
cp -R "$BUILT_PRODUCTS_DIR/$PRODUCT_NAME.app"
"$dir"
ln -s "/Applications" "$dir/Applications"
```

```
mkdir "$dir/.background"
cp "$PROJECT_DIR/$PROJECT_NAME/some_image.png"
"$dir/.background"
rm -f "$dmg"
hdiutil create "$dmg" -srcfolder "$dir" -volname
"$PRODUCT_NAME" -format UDRW

#Mount the disk image, and store the device name
hdiutil attach "$dmg" -noverify -noautoopen -
readwrite
```

Script 2 (Set Window Properties Script):

```
#!/usr/bin/osascript
#get the dimensions of the main window using a
bash script

set {width, height, scale} to words of (do
shell script "system_profiler
SPDisplaysDataType | awk '/Main Display:
Yes/{found=1} /Resolution/{width=$2; height=$4}
/Retina/{scale=($2 == \"Yes\" ? 2 : 1)} /^ {8}
[^ ]+/{if(found) {exit}; scale=1} END{printf
\"%d %d %d\\n\", width, height, scale}'")
set x to ((width / 2) / scale)
set y to ((height / 2) / scale)

#get the product name using a bash script
set {product_name} to words of (do shell script
"printf \"%s\", $PRODUCT_NAME")
set background to alias
("Volumes:"&product_name&":.background:some_image.

tell application "Finder"
    tell disk product_name
        open
        set current view of container window to
icon view
        set toolbar visible of container window
to false
        set statusbar visible of container
window to false
```

```
        set the bounds of container window to
{x, y, (x + 479), (y + 383)}
        set theViewOptions to the icon view
options of container window
        set arrangement of theViewOptions to
not arranged
        set icon size of theViewOptions to 128
```

The above measurement for the window work for my project specifically due to the size of my background pic and icon resolution; you may need to modify these values for your own project.

Script 3 (Make Final Disk Image Script):

```
#!/bin/bash
dir="$TEMP_FILES_DIR/disk"
cp
"$PROJECT_DIR/$PROJECT_NAME/some_other_image.png"
"$dir/"

#unmount the temp image file, then convert it to
final image file
sync
sync
hdiutil detach /Volumes/$PRODUCT_NAME
rm -f "$BUILT_PRODUCTS_DIR/$PRODUCT_NAME.dmg"
hdiutil convert
"$BUILT_PRODUCTS_DIR/$PRODUCT_NAME.temp.dmg" -
format UDZO -imagekey zlib-level=9 -o
"$BUILT_PRODUCTS_DIR/$PRODUCT_NAME.dmg"
rm -f "$BUILT_PRODUCTS_DIR/$PRODUCT_NAME.temp.dmg"

#Change the icon of the image file
sips -i "$dir/some_other_image.png"
DeRez -only icns "$dir/some_other_image.png" >
"$dir/tmpicns.rsrc"
Rez -append "$dir/tmpicns.rsrc" -o
"$BUILT_PRODUCTS_DIR/$PRODUCT_NAME.dmg"
SetFile -a C
```

```
"$BUILT_PRODUCTS_DIR/$PRODUCT_NAME.dmg"

rm -rf "$dir"
```

Make sure the image files you are using are in the $PROJECT_DIR/$PROJECT_NAME/ directory!

Share Improve this answer Follow

edited Jul 24, 2015 at 21:13

community wiki
2 revs
P.M.

It is creating blank disk image on my project . Any suggestion. – ManiaChamp Jan 14, 2016 at 8:54

Shell script part works only, but how can I add Apple script with Shell Script in RunScript under Build Phase it is showing error ,every apple script statement is symbol not found. – ManiaChamp Jan 28, 2016 at 6:52

3

I used dmgbuild.

- Installation: `pip3 install dmgbuild`
- Mount your volume
- Create a settings file:

```
{
    "title": "NAME",
    "background": "YOUR_BACKGROUND.png",
    "format": "UDZO",
```

```
    "compression-level": 9,
    "window": { "position": { "x": 100, "y": 100 },
                "size": { "width": 640, "height": 300
    "contents": [
        { "x": 140, "y": 165, "type": "file", "path":
"/Volumes/YOUR_VOLUME_NAME/YOUR_APP.app" },
        { "x": 480, "y": 165, "type": "link", "path":
    ]
}
```

- The width value is the width of the background.
- The height value should be the background height + 20 for the window bar.
- In a terminal: `dmgbuild -s settings.json "YOUR_VOLUME_NAME" output.dmg`

Share Improve this answer Follow

answered Apr 14, 2021 at 13:43

community wiki
mmerle

2

.DS_Store files stores windows settings in Mac. Windows settings include the icons layout, the window background, the size of the window, etc. The .DS_Store file is needed in creating the window for mounted images to preserve the arrangement of files and the windows background.

Once you have .DS_Store file created, you can just copy it to your created installer (DMG).

Share Improve this answer Follow

answered Apr 14, 2010 at 8:09

community wiki
hor10zs

> But first add a background image to the dmg by naming it something like .background.png and then using Cmd-J to add that (!) image to the background. – Gijs Jun 23, 2017 at 21:31

2

I also in need of using command line approach to do the packaging and dmg creation "programmatically in a script". The best answer I found so far is from Adium project' Release building framework (See R1). There is a custom script(AdiumApplescriptRunner) to allow you avoid OSX WindowsServer GUI interaction. "osascript applescript.scpt" approach require you to login as builder and run the dmg creation from a command line vt100 session.

OSX package management system is not so advanced compared to other Unixen which can do this task easily and systematically.

R1: http://hg.adium.im/adium-1.4/file/00d944a3ef16/Release

Share Improve this answer Follow

answered Oct 9, 2011 at 12:55

community wiki
T.J. Yang

1

These answers are way too complicated and times have changed. The following works on 10.9 just fine, permissions are correct and it looks nice.

Create a read-only DMG from a directory

```
#!/bin/sh
# create_dmg Frobulator Frobulator.dmg path/to/frobula
Identity' ]
set -e

VOLNAME="$1"
DMG="$2"
SRC_DIR="$3"
CODESIGN_IDENTITY="$4"

hdiutil create -srcfolder "$SRC_DIR" \
  -volname "$VOLNAME" \
  -fs HFS+ -fsargs "-c c=64,a=16,e=16" \
  -format UDZO -imagekey zlib-level=9 "$DMG"

if [ -n "$CODESIGN_IDENTITY" ]; then
  codesign -s "$CODESIGN_IDENTITY" -v "$DMG"
fi
```

Create read-only DMG with an icon (.icns type)

```
#!/bin/sh
# create_dmg_with_icon Frobulator Frobulator.dmg path/
path/to/someicon.icns [ 'Your Code Sign Identity' ]
set -e
```

```
VOLNAME="$1"
DMG="$2"
SRC_DIR="$3"
ICON_FILE="$4"
CODESIGN_IDENTITY="$5"

TMP_DMG="$(mktemp -u -t XXXXXXX)"
trap 'RESULT=$?; rm -f "$TMP_DMG"; exit $RESULT' INT Q
hdiutil create -srcfolder "$SRC_DIR" -volname "$VOLNAM
               -fsargs "-c c=64,a=16,e=16" -format UDR
TMP_DMG="${TMP_DMG}.dmg" # because OSX appends .dmg
DEVICE="$(hdiutil attach -readwrite -noautoopen "$TMP_
'NR==1{print$1}')"
VOLUME="$(mount | grep "$DEVICE" | sed 's/^[^ ]* on //
# start of DMG changes
cp "$ICON_FILE" "$VOLUME/.VolumeIcon.icns"
SetFile -c icnC "$VOLUME/.VolumeIcon.icns"
SetFile -a C "$VOLUME"
# end of DMG changes
hdiutil detach "$DEVICE"
hdiutil convert "$TMP_DMG" -format UDZO -imagekey zlib
if [ -n "$CODESIGN_IDENTITY" ]; then
  codesign -s "$CODESIGN_IDENTITY" -v "$DMG"
fi
```

If anything else needs to happen, these easiest thing is to make a temporary copy of the SRC_DIR and apply changes to that before creating a DMG.

Share Improve this answer Follow

edited Jan 2, 2014 at 10:54

community wiki
6 revs
user246672

This answer is not adding anything to what's been written earlier, and it also does not answer the original question (it's

not about just putting an icon in the dmg, or how to sign it; the question is about adding graphics and positioned icons programmatically to a dmg). – Ludvig A. Norin Jan 2, 2014 at 12:43

1

I've just written a new (friendly) command line utility to do this. It doesn't rely on Finder/AppleScript, or on any of the (deprecated) Alias Manager APIs, and it's easy to configure and use.

Anyway, anyone who is interested can find it on PyPi; the documentation is available on Read The Docs.

Share Improve this answer Follow

answered Feb 17, 2014 at 10:35

community wiki
al45tair

0

Here's another tool which is by far the easiest of the bunch: https://github.com/sindresorhus/create-dmg

Note: it's not the same as the bash-script option of the same name mentioned in other comments.

```
npm install --global create-dmg
create-dmg YourApp.app
```

It's pretty full featured. It adds a background, created a "DMG icon" using your app icon, and signs the dmg.

Share Improve this answer Follow

answered Nov 8 at 17:19

community wiki
scosman

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.