Building & distributing Linux desktop apps

Using Electron framework
About me

- Creator of Developer Diary
- Founded Invide
  Invite-only community of remote developers
- Produced Git Commit Show
- Open-Source Contributor and Trainer
- Chess Lover
Distraction-Free Diary

Get things out of your head to focus on the present moment

# Day 0
This is your private diary. It will help you switch off from work and be present for other important things in your life.

These notes are:
- Offline: No need of internet to work
- Privacy-focused: Saved locally on your computer only
- Auto-saved: Will be available when you come back, no action required

**Start writing your notes**
Have you used Electron before?
Yes/No

Which Linux distro have you used in last 2 years?
Debian
Fedora
ArchLinux
OpenSuse
Other
Introduction to Electron
Electron

A framework to build cross-platform desktop apps

Chromium rendering engine + Node.js runtime

Code using web technologies
```javascript
const { app, BrowserWindow } = require('electron')

app.whenReady().then(() => {
  createWindow()
})

function createWindow () {
  const win = new BrowserWindow({
    width: 800,
    height: 600
  })

  win.loadFile('index.html')
}
```

```html
<html>
<head>
  <title>Hello World!</title>
</head>
<body style="background: white;">
  <h1>Hello World!</h1>
</body>
</html>
```
Packaging your desktop app
Packaging the app

- Supported package targets
  AppImage, snap, deb, rpm, freebsd, pacman, p5p, apk, 7z, zip, tar.xz, tar.lz, tar.gz, tar.bz2, dir

- Build package for all distributions on one platform
Distributing your app
Direct distribution
Auto updates

- Only for AppImage
- Needs a release server
  - Electron-release-server
  - Hazel
  - Nuts
  - Public Github releases

```javascript
autoUpdater.on('update-downloaded', (event, releaseNotes, releaseName) => {
  const downloadAlertOptions = {
    type: 'info',
    buttons: ['Restart', 'Later'],
    title: 'Application Update',
    detail: 'New version downloaded. Apply updates now?'
  }

dialog.showMessageBox(downloadAlertOptions).then((returnValue) => {
  if (returnValue.response === 0) autoUpdater.quitAndInstall()
})
})
```
External distribution
Classic Distribution

- deb
- rpm
- pacman
- freebsd

Universal distribution

- AppImage
- Snap
- Flatpak
Classic distribution
deb

dpkg -i DeveloperDiary.deb

**Distribute via your own repository**

```shell
sudo add-apt-repository your.repo
apt install developerdiary
```

**Distribute via Debian Package Repository**

*License restriction:* Free, allows to modify and distribute

```
apt install developerdiary
```
rpm

```bash
rpm -i DeveloperDiary.rpm
```

**Distribute via yum**

Yum install developerdiary

pacman

```bash
pacman -S DeveloperDiary
```

freebsd

```bash
pkg install developerdiary
```
Universal distribution
AppImage

Just click to run

- No desktop integration (gnome)
- No auto-start(gnome)
- Security
- No extra step to publish
- Supports auto-update out of the box
Snapcraft

snap install developerdiary

- Sandboxing
- Delta updates
- Client is OSS, server is not
- **Effort to publish is low**

Developer Diary

A minimal diary for developers available right in time with shortcut Ctrl+Shift+i
Flatpak

flatpak install flathub
org.invidelabs.DEVELOPERDIARY

- Sandboxing
- Delta updates
- **Effort to publish** is non-trivial
Comparison of universal distribution channels
## Application Distribution

<table>
<thead>
<tr>
<th>Feature</th>
<th>AppImage</th>
<th>Snap</th>
<th>Flatpak</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Repo / Directory</td>
<td>AppImageHub</td>
<td>Snap Store</td>
<td>FlatHub</td>
</tr>
<tr>
<td>Fully decentralized without central gatekeepers</td>
<td>Yes</td>
<td>No (one dominant app store) [1]</td>
<td>Yes</td>
</tr>
<tr>
<td>Individual App Repositories</td>
<td>No (not stored in repositories)</td>
<td>No (you can only have one repo per device)</td>
<td>Yes</td>
</tr>
<tr>
<td>Can have multiple versions in parallel (including historical ones)</td>
<td>Yes (unlimited number of arbitrary versions)</td>
<td>Yes (one per channel)</td>
<td>Yes (any version available in OSTree can be installed in parallel)</td>
</tr>
<tr>
<td>Once the application is installed, it still can be easily copied to another machine (e.g., share with a friend locally)</td>
<td>Yes (one app=one file; there is no “installation” so the app is in the same form at all times)</td>
<td>Yes (but also need to copy snaps it depends on)</td>
<td>Yes (you can use <code>flatpak create</code> <code>usb</code> to copy to USB drive)</td>
</tr>
</tbody>
</table>

## Application Updates

<table>
<thead>
<tr>
<th>Feature</th>
<th>AppImage</th>
<th>Snap</th>
<th>Flatpak</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update mechanism</td>
<td>AppImageUpdate</td>
<td>From Repo</td>
<td>From Repo</td>
</tr>
<tr>
<td>Binary delta updates</td>
<td>Yes (using zs同步 with no need to generate deltas in advance)</td>
<td>Yes (Only if using a private server-side service that needs to generate the deltas)</td>
<td>Yes (using OSTree to provide atomic updates)</td>
</tr>
<tr>
<td>Applications can be self-updating</td>
<td>Yes (using embedded information)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Source
### Linux Distribution Support

<table>
<thead>
<tr>
<th>Feature</th>
<th>Appimage</th>
<th>Snap</th>
<th>Flatpak</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earliest Ubuntu Supported</td>
<td>Ubuntu 10.04</td>
<td>Ubuntu 14.04</td>
<td>Ubuntu 16.04</td>
</tr>
<tr>
<td>Earliest OpenSUSE Supported</td>
<td>OpenSUSE 11.3</td>
<td>Leap 42.2</td>
<td>Leap 42.1</td>
</tr>
<tr>
<td>Earliest Fedora Supported</td>
<td>Fedora 12</td>
<td>Fedora 24</td>
<td>Fedora 23</td>
</tr>
<tr>
<td>Earliest Debian Supported</td>
<td>Debian 6</td>
<td>Debian 9</td>
<td>Debian 9</td>
</tr>
<tr>
<td>Earliest CentOS Supported</td>
<td>CentOS 6</td>
<td>CentOS 7.6</td>
<td>CentOS 7</td>
</tr>
<tr>
<td>Runs on Ubuntu out-of-the-box</td>
<td>✔ Yes</td>
<td>✔ Yes</td>
<td>✗ No</td>
</tr>
<tr>
<td>Runs on OpenSUSE out-of-the-box</td>
<td>✔ Yes</td>
<td>✗ No</td>
<td>tbc</td>
</tr>
<tr>
<td>Runs on Fedora out-of-the-box</td>
<td>✔ Yes</td>
<td>✗ No</td>
<td>✔ Yes</td>
</tr>
<tr>
<td>Runs on Debian out-of-the-box</td>
<td>✔ Yes</td>
<td>✗ No</td>
<td>tbc</td>
</tr>
<tr>
<td>Runs on CentOS out-of-the-box</td>
<td>✔ Yes</td>
<td>✗ No</td>
<td>✔ Yes</td>
</tr>
<tr>
<td>Live systems (e.g., Live ISO, Live USB, Live CD, Live network boot)</td>
<td>✔ Full</td>
<td>▲ Partial (starting with 18.04, but it is limited by a kernel limitation and &quot;a pain to work with, we spend almost zero time with that&quot; according to a Canonical developer)</td>
<td>▲ Partial (Session must be restarted for exports to be picked up)</td>
</tr>
<tr>
<td>Can run on Chrome OS (Crostini)</td>
<td>✔ Yes (Chrome OS 73)</td>
<td>✔ Yes (Chrome OS 73)</td>
<td>✔ Yes</td>
</tr>
</tbody>
</table>
Maintain your app
Maintenance

- Upgrade electron and node versions every quarter
- Tests before packaging
- Use typescript
- Managing different versions
  - How will you test?
  - Can you test all the channels on the same system?
    - Different product names
    - Test server for test channels
    - Flag for client version
Analyze and improve
Analytics

- Linux users expect privacy at the core
- Do not use external analytics services such as Google Analytics
- Anonymous analytics using update request logs
  - # of users
  - # of returning users
  - Avg. time spent
calcDownloads() {
    matchPeriodL=$1
    cat ${sourceFile} | awk '0~matchPeriod && /GET /download/*/\.(exe|\.(deb)|\.(dmg))/{print $1}'
    matchPeriod="${matchPeriodL}" | sort -u | wc -l
}
THANK YOU

pradeep-io.medium.com