

Packaging Guide

Overview

This guide covers how to package the ProtonDrive Sync application for distribution across different platforms and package managers.

Package Formats

1. Python Package (PyPI)

Building for PyPI

```
# Install build tools
pip install build twine

# Build package
python -m build

# Upload to PyPI (maintainers only)
twine upload dist/*
```

Installation from PyPI

```
# Install from PyPI
pip install protondrive-sync

# Run application
protondrive-sync
```

2. Arch Linux / AUR Package

Creating a PKGBUILD

File: PKGBUILD

```
# Maintainer: Your Name <your.email@example.com>
pkgname=protondrive-sync
pkgver=1.0.0
pkgrel=1
pkgdesc="ProtonDrive synchronization client with GUI"
arch=('any')
url="https://github.com/yourusername/protondrive-sync"
license=('MIT')
depends=('python' 'python-pyqt5' 'python-watchdog' 'rclone')
optdepends=(
    'python-pip: for installing additional dependencies'
)
source=("$pkgname-$pkgver.tar.gz::$url/archive/v$pkgver.tar.gz")
sha256sums=('SKIP')

package() {
    cd "$srcdir/$pkgname-$pkgver"

    # Install Python package
    python setup.py install --root="$pkgdir" --optimize=1

    # Install scripts
    install -Dm755 run.sh "$pkgdir/usr/bin/protondrive-sync"
    install -Dm755 install.sh "$pkgdir/usr/share/$pkgname/install.sh"
    install -Dm755 uninstall.sh "$pkgdir/usr/share/$pkgname/uninstall.sh"
    install -Dm755 verify_setup.sh "$pkgdir/usr/share/$pkgname/verify_setup.sh"

    # Install documentation
    install -Dm644 README.md "$pkgdir/usr/share/doc/$pkgname/README.md"
    cp -r docs "$pkgdir/usr/share/doc/$pkgname/"

    # Install license
    install -Dm644 LICENSE "$pkgdir/usr/share/licenses/$pkgname/LICENSE"

    # Install desktop file (if created)
    # install -Dm644 protondrive-sync.desktop "$pkgdir/usr/share/applications/
    protondrive-sync.desktop"
}
```

Testing the PKGBUILD

```
# Build package
makepkg -si

# Test installation
protondrive-sync

# Remove package
sudo pacman -R protondrive-sync
```

Submitting to AUR

```
# Clone AUR repository
git clone ssh://aur@aur.archlinux.org/protondrive-sync.git
cd protondrive-sync

# Add PKGBUILD and .SRCINFO
cp /path/to/PKGBUILD .
makepkg --printsrcinfo > .SRCINFO

# Commit and push
git add PKGBUILD .SRCINFO
git commit -m "Initial commit"
git push
```

3. Debian / Ubuntu Package (.deb)

Creating Debian Package Structure

```
protondrive-sync-1.0.0/
├── DEBIAN/
│   ├── control
│   ├── postinst
│   ├── prerm
│   └── postrm
├── usr/
│   ├── bin/
│   │   └── protondrive-sync
│   ├── lib/
│   │   └── python3/dist-packages/
│   ├── share/
│   │   ├── applications/
│   │   ├── doc/protondrive-sync/
│   │   └── protondrive-sync/
│   └── ...
```

DEBIAN/control

```
Package: protondrive-sync
Version: 1.0.0
Section: utils
Priority: optional
Architecture: all
Depends: python3 (>= 3.8), python3-pyqt5, python3-watchdog, rclone
Maintainer: Your Name <your.email@example.com>
Description: ProtonDrive synchronization client
 A GUI application for synchronizing ProtonDrive with local storage.
 Features selective sync, bandwidth limiting, and more.
Homepage: https://github.com/yourusername/protondrive-sync
```

Building .deb Package

```
# Build package
dpkg-deb --build protondrive-sync-1.0.0

# Test installation
sudo dpkg -i protondrive-sync-1.0.0.deb

# Fix dependencies if needed
sudo apt-get install -f
```

4. RPM Package (Fedora / RHEL)

RPM Spec File

File: protondrive-sync.spec

```

Name:                protondrive-sync
Version:             1.0.0
Release:             1%{?dist}
Summary:             ProtonDrive synchronization client

License:             MIT
URL:                 https://github.com/yourusername/protondrive-sync
Source0:             %{name}-%{version}.tar.gz

BuildArch:           noarch
BuildRequires:       python3-devel
BuildRequires:       python3-setuptools

Requires:            python3 >= 3.8
Requires:            python3-PyQt5
Requires:            python3-watchdog
Requires:            rclone

%description
A GUI application for synchronizing ProtonDrive with local storage.
Features selective sync, bandwidth limiting, and automated synchronization.

%prep
%autosetup

%build
%py3_build

%install
%py3_install
install -Dm755 run.sh %{buildroot}%{_bindir}/protondrive-sync
install -Dm644 README.md %{buildroot}%{_docdir}/%{name}/README.md
cp -r docs %{buildroot}%{_docdir}/%{name}/

%files
%license LICENSE
%doc README.md
%{_bindir}/protondrive-sync
%{python3_sitelib}/*
%{_docdir}/%{name}/

%changelog
* Tue Dec 16 2025 Your Name <your.email@example.com> - 1.0.0-1
- Initial RPM release

```

Building RPM

```
# Setup rpmbuild directory
rpmdev-setuptree

# Copy spec file
cp protondrive-sync.spec ~/rpmbuild/SPECS/

# Copy source tarball
cp protondrive-sync-1.0.0.tar.gz ~/rpmbuild/SOURCES/

# Build RPM
rpmbuild -ba ~/rpmbuild/SPECS/protondrive-sync.spec

# Install
sudo rpm -i ~/rpmbuild/RPMS/noarch/protondrive-sync-1.0.0-1.noarch.rpm
```

5. Flatpak

Flatpak Manifest

File: com.github.yourusername.ProtonDriveSync.yml

```
app-id: com.github.yourusername.ProtonDriveSync
runtime: org.freedesktop.Platform
runtime-version: '23.08'
sdk: org.freedesktop.Sdk
command: protondrive-sync

finish-args:
  - --share=ipc
  - --socket=x11
  - --socket=wayland
  - --filesystem=home
  - --share=network

modules:
  - name: rclone
    buildsystem: simple
    build-commands:
      - install -Dm755 rclone /app/bin/rclone
    sources:
      - type: archive
        url: https://github.com/rclone/rclone/releases/download/v1.65.0/rclone-
v1.65.0-linux-amd64.zip
        sha256: CHECKSUM_HERE

  - name: protondrive-sync
    buildsystem: simple
    build-commands:
      - pip3 install --prefix=/app .
      - install -Dm755 run.sh /app/bin/protondrive-sync
    sources:
      - type: git
        url: https://github.com/yourusername/protondrive-sync.git
        tag: v1.0.0
```

Building Flatpak

```
# Build
flatpak-builder build-dir com.github.yourusername.ProtonDriveSync.yml

# Install locally
flatpak-builder --user --install build-dir com.github.yourusername.ProtonDriveSync.yml

# Run
flatpak run com.github.yourusername.ProtonDriveSync
```

6. AppImage

Building AppImage

```
# Create AppDir structure
mkdir -p ProtonDriveSync.AppDir/usr

# Install application
pip install --target=ProtonDriveSync.AppDir/usr/lib/python3/site-packages .

# Copy files
cp -r protondrive_sync ProtonDriveSync.AppDir/usr/lib/python3/site-packages/
cp run.sh ProtonDriveSync.AppDir/usr/bin/protondrive-sync

# Create AppRun
cat > ProtonDriveSync.AppDir/AppRun << 'EOF'
#!/bin/bash
HERE="$(dirname "$(readlink -f "${0}")")"
export PYTHONPATH="$HERE/usr/lib/python3/site-packages:$PYTHONPATH"
exec python3 -m protondrive_sync.main "$@"
EOF
chmod +x ProtonDriveSync.AppDir/AppRun

# Download appimagetool
wget https://github.com/AppImage/AppImageKit/releases/download/continuous/appimagetool-x86_64.AppImage
chmod +x appimagetool-x86_64.AppImage

# Build AppImage
./appimagetool-x86_64.AppImage ProtonDriveSync.AppDir
```

7. Windows Installer (Optional)

Using PyInstaller

```
# Install PyInstaller
pip install pyinstaller

# Create executable
pyinstaller --onefile --windowed --name="ProtonDrive Sync" protondrive_sync/main.py

# Create installer with NSIS or Inno Setup
```

Desktop Integration

Desktop Entry File

File: **protondrive-sync.desktop**

```
[Desktop Entry]
Version=1.0
Type=Application
Name=ProtonDrive Sync
Comment=Synchronize ProtonDrive with local storage
Exec=protondrive-sync
Icon=protondrive-sync
Terminal=false
Categories=Network;FileTransfer;Qt;
Keywords=protondrive;sync;cloud;backup;
```

Application Icon

Create icons in standard sizes:

- 16x16
- 32x32
- 48x48
- 64x64
- 128x128
- 256x256

Install to:

```
/usr/share/icons/hicolor/{size}/apps/protondrive-sync.png
```

Distribution Checklist

Before distributing:

- ☐ Update version in all files (setup.py, PKGBUILD, etc.)
- ☐ Update changelog
- ☐ Test installation on clean system
- ☐ Verify all dependencies are listed
- ☐ Check file permissions
- ☐ Validate desktop integration
- ☐ Test uninstallation
- ☐ Update documentation
- ☐ Create release notes
- ☐ Tag release in git

Release Process

1. Version Bump

```
# Update version in setup.py
sed -i 's/version=".*"/version="1.1.0"/' setup.py

# Update version in other files
# - PKGBUILD
# - README.md
# - Documentation
```

2. Create Git Tag

```
git tag -a v1.1.0 -m "Release version 1.1.0"
git push origin v1.1.0
```

3. Build All Packages

```
# Python package
python -m build

# Arch package
makepkg

# Debian package
dpkg-deb --build protondrive-sync-1.1.0

# ... etc
```

4. Upload to Distribution Channels

- PyPI: `twine upload dist/*`
- AUR: Push to AUR git repository
- GitHub Releases: Upload built packages

5. Update Documentation

- Update installation instructions
- Announce new release
- Update website (if applicable)

Automated Building

GitHub Actions Workflow

File: `.github/workflows/build.yml`

```

name: Build Packages

on:
  push:
    tags:
      - 'v*'

jobs:
  build-python:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-python@v4
        with:
          python-version: '3.10'
      - name: Build package
        run: |
          pip install build
          python -m build
      - name: Upload artifacts
        uses: actions/upload-artifact@v3
        with:
          name: python-package
          path: dist/*

  build-deb:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Build .deb
        run: |
          # Build script here
      - name: Upload artifacts
        uses: actions/upload-artifact@v3
        with:
          name: deb-package
          path: '*.deb'

```

See Also

- [Installation Guide](#) (../setup/INSTALLATION_GUIDE.md)
- [CachyOS Optimization](#) (cachyos-optimization.md)
- [Python Packaging Guide](https://packaging.python.org/) (https://packaging.python.org/)
- [Arch Package Guidelines](https://wiki.archlinux.org/title/PKGBUILD) (https://wiki.archlinux.org/title/PKGBUILD)