Packaging Considerations for Simple_Scope

PyVISA Backend Requirements

PyVISA requires backend libraries to communicate with hardware. Depending on which PyVISA backend you're using, you may need to include additional files:

For PyVISA-py:

- For USB connections: `libusb` or `pyusb` dependencies
- For GPIB connections: `linux-gpib` or equivalent libraries
- For serial connections: Additional COM port libraries

For NI-VISA Backend:

If using the National Instruments VISA backend, users will need to have NI-VISA installed on their system. This cannot be bundled with your PyInstaller package.

Runtime Dependencies

Your application relies on several runtime dependencies that need special consideration:

1. **Python DLLs**: PyInstaller should automatically include these, but verify they're present in the final package.
2. **TCL/TK for GUI**: Since you're using Tkinter, ensure that TCL/TK libraries are correctly bundled. PyInstaller should handle this automatically.
3. **Local Data Storage**: Your application saves configuration in the user's home directory. Make sure your app has appropriate permissions to create and modify this directory.

Troubleshooting Common Issues

Missing Modules

If you encounter "ModuleNotFoundError" when running the packaged application, you may need to add additional hidden imports to your spec file:

python

Copy

```
hiddenimports=['pyvisa', 'pyvisa_py', 'tkinter', 'tkinter.filedialog', 'tkinter.messagebox']
```

File Access Issues

Make sure your application's file access methods work correctly within the packaged environment. PyInstaller changes how paths work when the application is frozen.

Device Access Issues

Access to hardware devices often requires special permissions. On Windows, this is typically handled by driver installation, but on Linux or macOS, you might need to add udev rules or other configuration.

Testing Your Package

Before distributing, test your packaged application on a "clean" system that doesn't have your development environment installed. This helps identify missing dependencies.

Version Information

Consider adding version information to your executable. On Windows, you can use a version info file:

python

Copy

```python
# In your spec file
exe = EXE(
    # other parameters
    version='file_version_info.txt',
)
```

Distribution Concerns

Single-File vs. Directory Structure

- **Single-File**: Easier distribution, but slower startup (needs to extract files to temp directory)
- **Directory Structure**: Faster startup, but more complex distribution

Platform-Specific Considerations

Remember that PyInstaller creates packages for the platform it runs on. If you need builds for multiple platforms (Windows, macOS, Linux), you'll need to run PyInstaller on each platform.