

WINLIBRE OVERVIEW

GETTING STARTED

WinLibre is a package management system for the Windows operating system. Its goal is to provide a functional and flexible system for managing software on the system. WinLibre is split up into three sections:

- 1) Package Manager – This is the application that the end user installs on their system to manage the software being installed with WinLibre
- 2) Website – This provides the Package Manager a server to get the list of the latest package versions, updates, and the actual packages themselves. The websites are also known as “repositories” that contain all the available packages and meta-data for that specific site.
- 3) Package Creator – This gives developers a full set of tools for creating a package easily without manually creating the package to the specifications and uploading to the website.

WINLIBRE FROM AN END USER’S POINT OF VIEW

Since WinLibre aims to be an easy to use application, much of the background work is hidden from the average user. What really happens when you select to download updates or a new application? Each time a user chooses to do something with WinLibre, it follows a simple set of tasks:

- 1) Check the local database and ask it, if we want to install or remove this, is there anything else we need to get first? Is there anything that will be left over that we can get rid of after we remove this? WinLibre makes note of all these tasks and performs them with the required task, only notifying the user that these tasks need to happen as well as what they chose to do.
- 2) If new software is being downloaded, then grab them.
- 3) Prepare the packages. Whether this is for removal, check to see if anything needs to be done before we do the actual task.
- 4) Install or remove the packages.
- 5) Clean up and post installation and removal. This is for any clean up or optional code, such as creating shortcuts which does not pertain to the actual installed state of the package.

These are the five basic steps of each task that is done through WinLibre. Some of them differ slightly from others so the next few sections

UPDATING THE LOCAL LIST OF AVAILABLE PACKAGES

This is used to update the list of available packages on the user’s machine. It downloads the list of available packages for each repository that the user has enabled.

- 1) WinLibre parses a file containing a list of repositories to check. By default this will only contain one repository: the official WinLibre repository. Others may be added at the user’s discretion.
- 2) With this list of repositories, WinLibre next goes to each repository and downloads the latest package list from the website. This will be served through the GET method on the Website end.

- 3) The package lists will be parsed and stored locally in the database. The user is able to manage any of these packages that are stored in the local database.

DOWNLOADING PACKAGES AND UPDATES

The user will be able to choose to download the updates or new packages. Either way the following set of tasks will happen:

- 1) WinLibre will go through each package marked to download (which may be only updates for existing packages or for packages the user chose) and will check for their dependencies. All of these will be marked to be downloaded as well.
- 2) All the packages listed will be downloaded from the web server using the URL of the server they are hosted on and the filename which points to the path on that server where the file is located.
- 3) Once the packages have been downloaded, package install scripts will begin executing. For each package preinstall.py, install.py and postinstall.py will be executed in that order if they exist.
- 4) Upon completion of installation of a package, its local database entry will be updated with the status, whether it be “install ok” or “install incomplete” in case of failure during one of the install scripts.

UNINSTALLING PACKAGES

Uninstalling a package is performed in the following manner after the user selects to remove the package:

- 1) The package is marked to be uninstalled by setting its status to “remove”. Each dependency is looked at to see if it is required by another package, if not, then those dependencies are removed as well because they serve no purpose being installed on that machine.
- 2) Once this has been completed, packages are uninstalled in the following manner: preremove.py is executed for each package if it exists, remove.py is run for each package and postremove.py is executed if it exists.
- 3) The status will be changed to “remove ok” in the local database if the package is successfully removed. If it is unsuccessful during removal the package will be marked “remove incomplete”.

WINLIBRE FROM A DEVELOPER’S POINT OF VIEW

Developers, while still end users in the long run have a slightly different perspective on WinLibre. These are the dedicated men and women writing the software that we are helping to distribute. As more and more adopters start releasing their software using WinLibre, we need to be able to accommodate anything they would need to be able to do. Let’s look at a use cases for developers:

- A small project wishes to release on WinLibre. Normally, downloads are just a single zip file and the user just extracts it and runs the executable. Nothing special, just straight forward and simple installation.
- A large project using many outside libraries wishes to release their application on Windows like they do on Linux. With releases on Linux, they just mark their packages as dependent upon those external libraries but with Windows they have always had to include them in their installer. They like the idea of using WinLibre for dependency resolution so they no longer have to package their app with the dependencies and can use it for easy release.

- A project would like to provide packages that are licensed under separate, yet incompatible license. They would like to use WinLibre to release software from their own repository so the user would have to accept their licensing and add their repository in order to the software they provide.

What does all this mean? WinLibre must be capable of installing packages that may possibly require many methods of installation and should be able to be done from many different repositories to provide the best flexibility for everyone.

In order to do this, each package is allowed 6 scripts: preinstall, install, postinstall, preremove, remove, and postremove. With Windows software, some need to simply just be extracted, while others require things such as registering DLLs. Package install scripts need to provide all this and more. WinLibre accomplishes this by providing a plethora of helper functions to do common tasks as writing to the registry, extracting archives, copying files and creating short cuts.

The website runs on Django means that it can store its information into and provides package information to WinLibre. It is capable of running on any host that supports Django which also several database formats on the web host as well. It provides full search capabilities for users, administrative tools for developers and downloads for packages and the list of packages. This allows other groups to host repositories of different types of packages and the added flexibility of mirroring repositories in other locations.

CREATING AND DEPLOYING A PACKAGE

Developers may create a package by hand, following the rules in the WinLibre Packaging Policy if they like, however most will prefer to use the Package Creator. This tool provides both a command-line and graphical interface for package creation. Developers will set the package meta-data and create the install scripts. Once the package has been finalized and tested to the developer's approval, the package will then be submitted to a repository for inclusion.

Package submission is only allowed by accepted developers. This ensures the authenticity of each package as well as adds a layer of trust to the packages which you are downloading. The package creator will allow the user to login and upload the package through a POST request to the website.

From here the website will parse the newly uploaded package doing the following actions:

- 1) Add a new package entry to the website's database of available packages.
- 2) Store the package in its appropriate directory on the server.

As soon as this is completed, the package is available for any WinLibre users who have added this repository as a source. Users will be able to download packages immediately upon successful upload.