



# Introduction

The `pkgmanager` module is currently used, primarily as a top-level import, to determinate which package manager is being used, depending on the import and version.

This causes some circular dependency imports from time to time, as we want to take this module further and improve it to allow more customization and more, eventually, aggregate all things related to package managers in this module.

## What to expected that will be in this module?

Currently, we are aiming to provide capabilities for YUM, DNF and RPM. Each of those package managers will have its own module that will interact with either the Python API, or the CLI call that will correspond with the desired package manager.

The YUM/DNF classes that will be described below, will have one thing in common. They will be part of a base class that will dictate how they should operate, what methods to implement, and what needs to be followed to whoever needs to implement a new package manager.

### YUM

The YUM class will implement all of the Python API used under the hood for Convert2RHEL, we will aim here at gathering all the common operations used here and implemented under this class. One can expect that we will deal and treat all basic operations thorough here, like, look if a package is installed, installing a package, removing package, determinating if the base class needs to exist or not and so on...

We could think about this class being very similar to `yum transaction handler`.

### DNF

The DNF class will be very similar to the YUM one, as we have the same intention to maintain a sane interface between the two.

There is no much to add here since they will be very similar, the only thing that will change is that we will use the DNF Python API instead.

## RPM

The RPM module may be the easiest module to convert, as it consist mostly of binary calls to the RPM binary present on the system. The difficult part will be integrating this class with DNF and YUM classes, as this one will be very different from the rest.

Since the YUM/DNF classes will operate mostly on Python API bindings, and not so much on CLI calls, we can expect a few differences here, and they might not be fully compatible.

Last update: May 23, 2023

Authors: [Rodolfo Olivieri](#)