NAME

bundle−pristine − Restores installed gems to their pristine condition

SYNOPSIS

bundle pristine

DESCRIPTION

pristine restores the installed gems in the bundle to their pristine condition using the local gem cache from RubyGems. For git gems, a forced checkout will be performed.

For further explanation, **bundle pristine** ignores unpacked files on disk. In other words, this command utilizes the local **.gem** cache or the gem´s git repository as if one were installing from scratch.

Note: the Bundler gem cannot be restored to its original state with **pristine**. One also cannot use **bundle pristine** on gems with a ´path´ option in the Gemfile, because bundler has no original copy it can restore from.

When is it practical to use **bundle pristine**?

It comes in handy when a developer is debugging a gem. **bundle pristine** is a great way to get rid of experimental changes to a gem that one may not want.

Why use **bundle pristine** over **gem pristine −−all**?

Both commands are very similar. For context: **bundle pristine**, without arguments, cleans all gems from the lockfile. Meanwhile, **gem pristine −−all** cleans all installed gems for that Ruby version.

If a developer forgets which gems in their project they might have been debugging, the Rubygems **gem pristine [GEMNAME]** command may be inconvenient. One can avoid waiting for **gem pristine −−all**, and instead run **bundle pristine**.