

Pip download prefers newer package version even when local package exists #5500

New issue

(!) Open

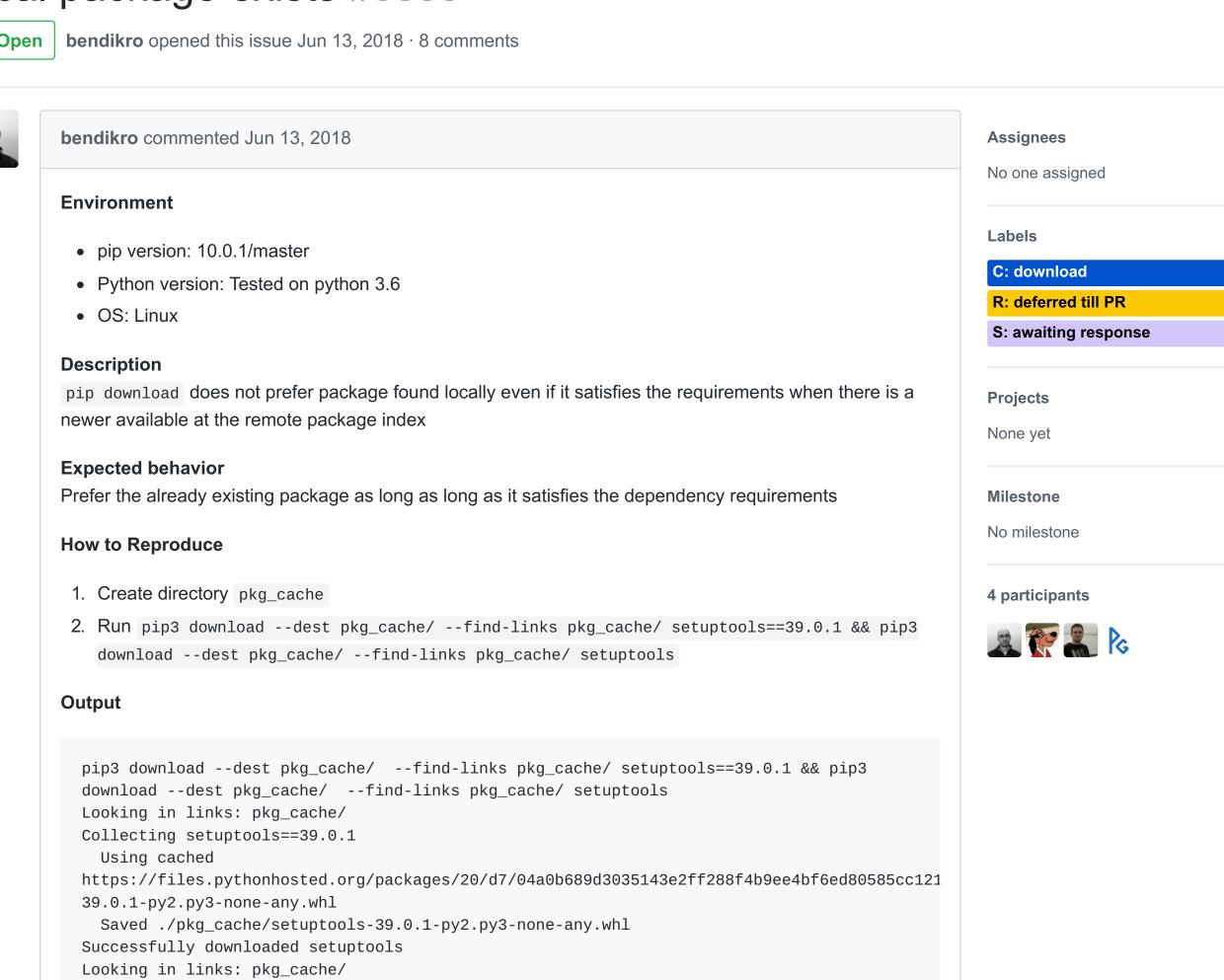
Collecting setuptools

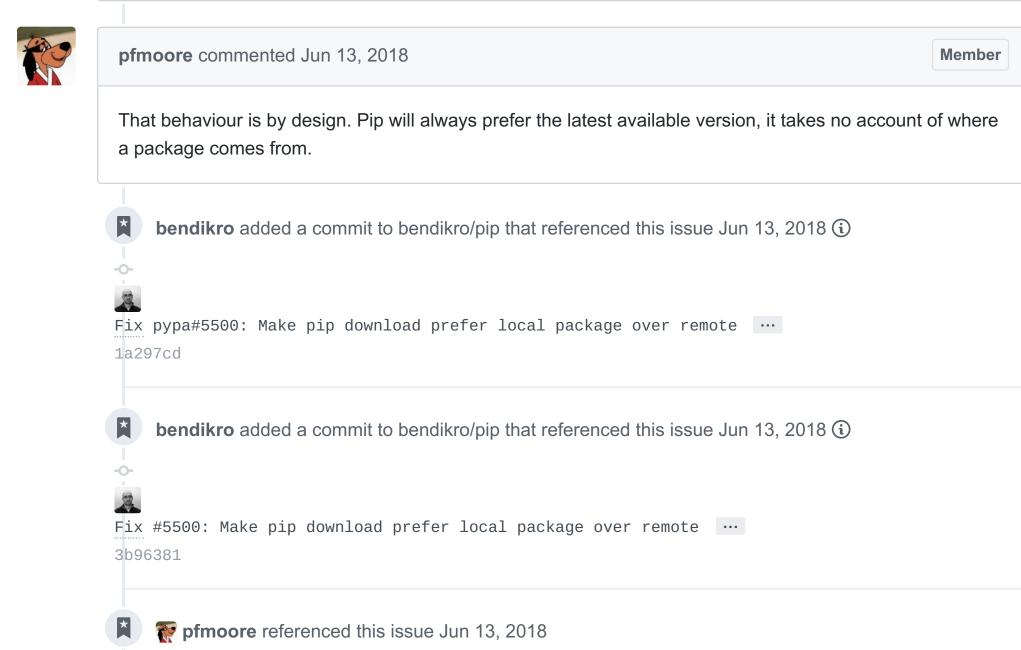
39.2.0-py2.py3-none-any.whl

Successfully downloaded setuptools

Using cached

will be downloaded.





https://files.pythonhosted.org/packages/7f/e1/820d941153923aac1d49d7fc37e17b6e73bfbd2904959

Saved ./pkg_cache/setuptools-39.2.0-py2.py3-none-any.whl

Fix #5500: Make pip download prefer local package over remote #5501 ☼ Open bendikro commented Jun 13, 2018 @pfmoore I see. We have multiple requirement files, and since pip does not handle double requirements it is

necessary to do multiple calls to pip download, one for each requirements file. With the current behavior

of pip, where one file has setuptools and another has setuptools==39.0.1, both 39.0.1 and 39.2.0

pfmoore commented Jun 13, 2018 Member So? That's the point of pip download. I don't know if I'm missing something here but I can't see what the

problem is. What exactly do you use the files downloaded via pip download for? As per the docs the

intention is that you use pip download to populate a directory from which you can later use pip install --find-links to do an install while offline. The pip install command is perfectly capable of handling a --find-links directory with multiple versions of the same package in it, so why are you bothered that this is happening? **§** 1

bendikro commented Aug 21, 2018 The point is that consistency is useful. Things that behave differently all the time is less useful than things

with the dependencies, this wouldn't be a problem though.

pfmoore commented Aug 21, 2018

bendikro commented Aug 21, 2018

above:

installed depending on what was present locally.

c: download S: awaiting response labels Jun 14, 2018

With two requirement files, as explained earlier, you never actually know exactly what package versions will be downloaded. The pip install command is perfectly capable of handling a --find-links directory with

multiple versions of the same package in it, so why are you bothered that this is happening?

Depending on the order of the requirement files you provide, different package versions are installed.

that do the same thing every time. Had pip supported handling multiple requirement files and dealt properly

Consistency is key. Second reason is speed. By looking locally and finding a package that satisfies the dependencies, there is no need to check remotely. Therefore, a call to pip download would be blazing fast if the packages are already downloaded. Currently it's very slow.

Member

Member

I'm not sure I follow. Pip's current behaviour is perfectly consistent - I described it above: Pip will always prefer the latest available version, it takes no account of where a package comes from. In fact, if we preferred local files, we'd be harming consistency, because you'd get something different

than you've currently offered.

I'm not sure I follow. Pip's current behaviour is perfectly consistent - I described it

you'll need to provide details of what you propose, and you'll probably need more persuasive arguments

I don't see anything actionable here. Pip's current behaviour is by design, if you want to propose a change,

True. It's consistent in that you never know which version it will download in the scenario I describe. In fact, if we preferred local files, we'd be harming consistency, because you'd get something different installed depending on what was present locally. The whole point is to know exactly what will be installed based on the local files. But having pip download the same package versions each time is not possible with multiple requirement files as I describe. I agree that the current default behaviour shouldn't be changed, but an option to be able to prefer local packages over checking remotely would still be useful. What I propose is to have an option that makes pip check locally if a package that satisfies the given

pfmoore commented Aug 21, 2018

dependency already exists locally, and if so, do not check remotely.

and store the downloaded file in the destination directory.

that I find your justification for the behaviour compelling, but that's something that can be debated later, when there's a PR to review. **3**

OK, so what you're suggesting is an option to pip download that says "for each requirement, if it can

already be satisfied from the destination directory, skip it, otherwise download the requirement as normal

I can see the logic in that. If you wanted to create a PR implementing it, I'm not going to object. I can't say

This would also be very useful for HPC clusters on which the staff may build python wheels that are optimized for their CPU architecture. The current behavior requires HPC staff to always be recompiling new versions as soon as they are out, or risk users using dramatically slower python packages in some

mboisson commented Aug 30, 2018

would be very useful to us.

pfmoore added the R: deferred till PR label Aug 21, 2018

situations. Being able to tell pip to favor a local wheelhouse over some minor version increase found online