

## Arvados - Bug #17012

### [versioning] generate correct development version numbers for all packages after a major release

10/15/2020 09:46 PM - Ward Vandewege

<b>Status:</b>	Resolved	<b>Start date:</b>	11/04/2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Ward Vandewege	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2020-11-18		
<b>Description</b>			
<p>The build/version-at-commit.sh script doesn't handle the scenario right after a major release well.</p> <p>For code that hasn't changed since the major release, it generates version numbers without taking the major release tag into account.</p> <p>For instance; as of commit c6c2f3518bc745eed95b5f5b81db5d17db4366ff on master, which is after the fork point of the 2.1-dev branch, these packages were generated for debian10:</p> <pre>arvados-api-server_2.1.0-1_amd64.deb arvados-client_2.1.0~dev20201012021212-1_amd64.deb arvados-controller_2.1.0~dev20201012021212-1_amd64.deb arvados-dispatch-cloud_2.1.0~dev20201012021212-1_amd64.deb arvados-docker-cleaner_2.2.0~dev20201015145956-1_amd64.deb arvados-git-httpd_2.1.0~dev20201012021212-1_amd64.deb arvados-health_2.1.0~dev20201012021212-1_amd64.deb arvados-src_2.2.0~dev20201015145956-1_all.deb arvados-sync-groups_2.2.0~dev20201013141632-1_amd64.deb arvados-workbench_2.1.0~dev20201012021212-1_amd64.deb arvados-ws_2.1.0~dev20201012021212-1_amd64.deb crunch-dispatch-local_2.1.0~dev20201012021212-1_amd64.deb crunch-dispatch-slurm_2.1.0~dev20201012021212-1_amd64.deb crunch-run_2.1.0~dev20201012021212-1_amd64.deb crunchstat_2.1.0~dev20201012021212-1_amd64.deb keep-balance_2.1.0~dev20201012021212-1_amd64.deb keep-block-check_2.1.0~dev20201012021212-1_amd64.deb keep-exercise_2.1.0~dev20201012021212-1_amd64.deb keepproxy_2.1.0~dev20201012021212-1_amd64.deb keep-rsync_2.1.0~dev20201012021212-1_amd64.deb keepstore_2.1.0~dev20201012021212-1_amd64.deb keep-web_2.1.0~dev20201012021212-1_amd64.deb libpam-arvados-go_2.1.0~dev20201012021212-1_amd64.deb python3-arvados-cwl-runner_2.2.0~dev20201015145956-1_amd64.deb python3-arvados-fuse_2.2.0~dev20201015145956-1_amd64.deb python3-arvados-python-client_2.2.0~dev20201015145956-1_amd64.deb python3-crunchstat-summary_2.2.0~dev20201015145956-1_amd64.deb</pre> <p>You can see that the python packages have a version number that takes the release into account (2.2.0~dev20201015145956), which is because there are some commits in master (post the 2.1-dev branch fork) that touch the Python part of the codebase.</p> <p>For other parts of the codebase, a version string is generated that starts with 2.1.0~dev, which is not ideal.</p> <p>For the API server package, it even generated the version string 2.1.0, because the calculated version for that package and its dependencies happens to be the commit that is tagged with the 2.1.0 version.</p>			
<b>Subtasks:</b>			
Task # 17084: Review comment #6			<b>Resolved</b>
Task # 17107: Python packages also depend on timestamp of build/version-at-commit.sh			<b>Resolved</b>

#### Associated revisions

Revision b7e1983d - 11/17/2020 10:43 PM - Ward Vandewege

Merge branch 'master' into 17012-fix-python-version-calculation

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

**Revision 5d23eb93 - 11/17/2020 10:44 PM - Ward Vandewege**

Merge branch '17012-fix-python-version-calculation' into master

closes #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

**Revision 84bf8b98 - 11/23/2020 02:20 PM - Ward Vandewege**

Now that `calculate_python_sdk_cwl_package_versions` returns python-style version string suffixes (.dev/rc), make sure the 2 scripts that use that function are adapted accordingly.

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

**Revision 41a8a689 - 11/23/2020 04:49 PM - Ward Vandewege**

Remove now-unneeded code from `sdk/cwl/test_with_arvbox.sh`, add comment to clarify that `calculate_python_sdk_cwl_package_versions` now outputs python-style package suffixes.

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

**Revision c85b48eb - 11/23/2020 04:51 PM - Ward Vandewege**

Update error checking in `run-build-docker-jobs-image.sh`

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

**Revision 12be280e - 11/23/2020 07:49 PM - Ward Vandewege**

Fix OS package iteration number in `run-build-docker-jobs-image.sh`

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

**Revision 56816842 - 12/10/2020 08:27 PM - Ward Vandewege**

Now that `calculate_python_sdk_cwl_package_versions` returns python-style version string suffixes (.dev/rc), make sure the 2 scripts that use that function are adapted accordingly.

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

**Revision d5d15b19 - 12/10/2020 08:27 PM - Ward Vandewege**

Remove now-unneeded code from `sdk/cwl/test_with_arvbox.sh`, add comment to clarify that `calculate_python_sdk_cwl_package_versions` now outputs python-style package suffixes.

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

**Revision f7499eb0 - 12/10/2020 08:27 PM - Ward Vandewege**

Update error checking in `run-build-docker-jobs-image.sh`

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

## Revision a3165699 - 12/10/2020 08:27 PM - Ward Vandewege

Fix OS package iteration number in run-build-docker-jobs-image.sh

refs #17012

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <[ward@curii.com](mailto:ward@curii.com)>

### History

---

#### #1 - 10/15/2020 09:46 PM - Ward Vandewege

- Status changed from New to In Progress

#### #2 - 10/15/2020 09:51 PM - Ward Vandewege

- Description updated

#### #3 - 10/15/2020 10:04 PM - Ward Vandewege

- Description updated

#### #4 - 10/16/2020 12:24 AM - Ward Vandewege

- Description updated

#### #5 - 10/16/2020 12:44 AM - Ward Vandewege

- Description updated

#### #6 - 10/16/2020 01:24 AM - Ward Vandewege

After some reflection, I think I've convinced myself that this behavior - counterintuitive as it is - is actually OK. The packages that are being generated do actually correspond with the source code from that version.

There is, however, a different bug. This all came up because a branch was merged that changed the package version numbers for dev packages from `.dev` to `~dev`.

The version number generation code did not recognize that this change (obviously!) was a reason to generate new package version numbers. That's because that code never took the 'build' directory into account when calculating the in-tree dependencies of each of our pieces of software, the git hashes and timestamps of which are used to generate an appropriate version number.

So, the real fix here is to add the 'build' directory to the list of directories that is checked as part of the version number calculation.

This is easy for the Go and Rails packages, and I've pushed a fix along those lines in [e94ddba9d544b173e5b56b41c6ac76ea0b072a26](#)

For the Python packages, the version calculation is done in 'arvados\_version.py', and that happens inside a virtualenv. It's more complicated to take the 'build' directory into account there, so I'm going to punt on that for now.

#### #7 - 10/21/2020 03:59 PM - Ward Vandewege

- Target version changed from 2020-10-21 Sprint to 2020-11-04 Sprint

#### #8 - 11/04/2020 05:07 PM - Peter Amstutz

- Target version changed from 2020-11-04 Sprint to 2020-11-18

#### #9 - 11/12/2020 05:53 PM - Peter Amstutz

Ward Vandewege wrote:

After some reflection, I think I've convinced myself that this behavior - counterintuitive as it is - is actually OK. The packages that are being generated do actually correspond with the source code from that version.

There is, however, a different bug. This all came up because a branch was merged that changed the package version numbers for dev packages from `.dev` to `~dev`.

The version number generation code did not recognize that this change (obviously!) was a reason to generate new package version numbers. That's because that code never took the 'build' directory into account when calculating the in-tree dependencies of each of our pieces of software, the git hashes and timestamps of which are used to generate an appropriate version number.

So, the real fix here is to add the 'build' directory to the list of directories that is checked as part of the version number calculation.

This is easy for the Go and Rails packages, and I've pushed a fix along those lines in [e94ddba9d544b173e5b56b41c6ac76ea0b072a26](#)

For the Python packages, the version calculation is done in 'arvados\_version.py', and that happens inside a virtualenv. It's more complicated to

take the 'build' directory into account there, so I'm going to punt on that for now.

I don't think there's any problem taking the 'build' directory into account for versioning. The python code already does `curdir+'../../build/version-at-commit.sh'` to get the version-at-commit script. The packages are always built from the source tree. It writes the version number to `"_version.py"` and during installation it just uses the version from `"_version.py"`.

We could probably simplify `arvados_version.py` a little bit by separating the list of relative directories from the rest of the code. Unfortunately, consolidating it to one script that's symlinked everywhere is not so easy because Python packaging is weird about symlinks.

#### #10 - 11/12/2020 06:03 PM - Peter Amstutz

The other thing to be aware of is that any other place that computes the version needs to incorporate 'build' as well, e.g. `run-library.sh calculate_python_sdk_cwl_package_versions()`

Maybe that function could be changed so that it calls `arvados_version.py`

#### #11 - 11/17/2020 04:08 PM - Ward Vandewege

Peter Amstutz wrote:

We could probably simplify `arvados_version.py` a little bit by separating the list of relative directories from the rest of the code. Unfortunately, consolidating it to one script that's symlinked everywhere is not so easy because Python packaging is weird about symlinks.

Yeah. I've made a change along those lines.

The other thing to be aware of is that any other place that computes the version needs to incorporate 'build' as well, e.g. `run-library.sh calculate_python_sdk_cwl_package_versions()`

Yes, I've made the corresponding change. This function is only used in a few of our build scripts that make docker images.

Maybe that function could be changed so that it calls `arvados_version.py`

I didn't do that, it would have been much more complicated than this self-contained bash function.

Ready for a look in [185b8af696c553c0978f27e720c6924148af22fd](https://github.com/Arvados/arvados/commit/185b8af696c553c0978f27e720c6924148af22fd) on branch 17012-fix-python-version-calculation. Tests at <https://ci.arvados.org/view/Developer/job/developer-run-tests/2181/>

#### #12 - 11/17/2020 09:37 PM - Peter Amstutz

The Python parts look great.

I don't think `calculate_python_sdk_cwl_package_versions` is quite correct.

`cwl_runner_version` is always going to be `$max`.

`python_sdk_version` is either `python_sdk_version` or `build_version`, whichever is later.

But we could also add something like this to `arvados_version.py`:

```
if __name__ == '__main__':
    print(get_version(SETUP_DIR, "arvados"))
```

Then `calculate_python_sdk_cwl_package_versions()` would just be:

```
calculate_python_sdk_cwl_package_versions() {
    python_sdk_version=$(cd sdk/python && python arvados_version.py)
    cwl_runner_version=$(cd sdk/cwl && python arvados_version.py)
}
```

#### #13 - 11/17/2020 10:38 PM - Ward Vandewege

Peter Amstutz wrote:

The Python parts look great.

I don't think `calculate_python_sdk_cwl_package_versions` is quite correct.

`cwl_runner_version` is always going to be `$max`.

It was actually correct - `cwl_runner_version` was never set to `$max` (which is a timestamp, but not clearly named as such) but rather to the latest of

cwl/build/sdk.

But we could also add something like this to arvados\_version.py:

[...]

Then calculate\_python\_sdk\_cwl\_package\_versions() would just be:

[...]

OK, I've implemented that in [15d64feed20daa4a422ff9092615ac1e295d5ca2](#). I don't like that it's bash calling python calling bash, but the code is a bit simpler.

**#14 - 11/17/2020 10:43 PM - Peter Amstutz**

Ward Vandewege wrote:

Peter Amstutz wrote:

The Python parts look great.

I don't think calculate\_python\_sdk\_cwl\_package\_versions is quite correct.

cwl\_runner\_version is always going to be \$max.

It was actually correct - cwl\_runner\_version was never set to \$max (which is a timestamp, but not clearly named as such) but rather to the latest of cwl/build/sdk.

But we could also add something like this to arvados\_version.py:

[...]

Then calculate\_python\_sdk\_cwl\_package\_versions() would just be:

[...]

OK, I've implemented that in [15d64feed20daa4a422ff9092615ac1e295d5ca2](#). I don't like that it's bash calling python calling bash, but the code is a bit simpler.

LGTM.

**#15 - 11/17/2020 10:46 PM - Ward Vandewege**

- Status changed from *In Progress* to *Resolved*

Applied in changeset [arvados!5d23eb939e64e4cf86dccb33aa24d9be3addf068](#).

**#16 - 05/13/2021 03:03 PM - Peter Amstutz**

- Release set to 38