

## openQA Project - action #53546

### Easier dependencies handling for packages, e.g. reduce duplication of build requirements in spec, documentation, Dockerfile

2019-06-27 12:15 - okurz

<b>Status:</b>	New	<b>Start date:</b>	2019-06-27
<b>Priority:</b>	Low	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Feature requests	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	future		
<b>Difficulty:</b>			

#### Description

#### Motivation

For both os-autoinst and openQA "build requirements" are defined in multiple places, e.g.

[https://github.com/os-autoinst/openQA/blob/master/docker/travis\\_test/Dockerfile#L10](https://github.com/os-autoinst/openQA/blob/master/docker/travis_test/Dockerfile#L10) ,

<https://github.com/os-autoinst/openQA/blob/master/cpanfile> , <https://github.com/os-autoinst/openQA/blob/master/openQA.spec#L37>

and correspondingly for os-autoinst . This fact and also the current setup based on the Dockerfile "docker/travis\_test/Dockerfile" which a container is built from in OBS, but only after merge of a PR, makes it hard to understand and hard to introduce new dependencies and test for them properly.

#### Related issues:

Related to openQA Project - action #43619: Improve workflow for dealing with ...	<b>Resolved</b>	<b>2018-11-09</b>
Related to openQA Project - action #55346: packaging test as part of every PR	<b>Resolved</b>	<b>2019-08-11</b>
Related to openQA Project - action #66649: Test GitHub actions for os-autoinst	<b>Resolved</b>	<b>2020-05-08</b>
Related to openQA Project - action #66721: Use GitHub actions for os-autoinst	<b>Resolved</b>	<b>2020-05-12</b>
Related to openQA Project - action #73309: every time a direct dependency is ...	<b>Workable</b>	<b>2020-10-13</b>
Blocked by openQA Project - action #56525: Create rpm requires from cpanfile	<b>Resolved</b>	<b>2019-09-06</b>

#### History

##### #1 - 2019-06-27 16:30 - okurz

What I thought of and partially tried:

- Define a "devel" package in the spec file and let it require all build time dependencies which are needed also for use outside OBS and install that virtual package whenever we want to develop, e.g. as in <https://github.com/os-autoinst/os-autoinst/pull/1171>
- Generate the spec file completely from a template and generated information, e.g. as with cpan2spec. Challenge would be where to store the spec-file. The best I could think of is to use travis CI to generate the spec file for every merge to master and save it in a special non-master git branch which the OBS service can check out. Why not store in master? A: To not mix generated and human-written files
- Extract the dependencies for the container scripts from the spec file, e.g. by parsing it and transforming into a list to give to zypper on-the-fly

All of the above help to define a single place for dependencies. Now to actually solve the problem that dependencies would only be active after merge not before one could use the same dependency list in the downstream "openqa" container (not "travis\_test" aka. "openqa\_dev") and call zypper -n in \$deps again to cover any potential diff in packages, e.g. when adding new dependencies.

Other thing: We have a container "travis\_test" aka. "openqa\_dev" which is built and published by OBS and a container "openqa" based on openqa\_dev which is built and run only during travis CI tests. This is confusing (the names in particular) and not helping us to cover testing of the actual other Dockerfiles we have. Maybe we can get rid of "openqa" by building "openqa\_dev" again but based on cached results? Maybe define the "openqa" container just as here-doc and a less complicated script with more setup-helper-scripts, e.g. as in <https://github.com/okurz/openQA/commit/b255b05939417792580abf8d5e017c28d33d75b1> where I reduce duplication to setup a test database

EDIT: Maybe <https://stackoverflow.com/questions/4339417/is-there-any-syntax-or-trick-to-be-able-to-create-a-multiline-rpm-spec-file-macr> also helps to define lists of dependencies and include in spec file.

##### #2 - 2019-07-01 07:39 - okurz

- Status changed from New to Feedback

- Assignee set to okurz

##### #3 - 2019-07-15 07:31 - okurz

- Related to action #43619: Improve workflow for dealing with openQA's dependencies added

**#4 - 2019-10-15 10:25 - okurz**

- Blocked by action #56525: Create rpm requires from cpanfile added

**#5 - 2019-10-15 10:27 - okurz**

- Related to action #55346: packaging test as part of every PR added

**#6 - 2019-10-15 10:27 - okurz**

- Status changed from Feedback to Blocked

waiting for [#56525](#) first

**#7 - 2020-05-12 09:44 - okurz**

- Related to action #66649: Test GitHub actions for os-autoinst added

**#8 - 2020-05-12 10:07 - okurz**

Discussed with tinita and mkittler and we re-confirmed what I wrote in [#53546#note-1](#) , there is one noteworthy difference. The openqa\_dev container definition is now only used within os-autoinst but stored in the openQA repo. That one can be moved to os-autoinst instead.

**#9 - 2020-05-12 13:37 - tinita**

- Blocked by action #66721: Use GitHub actions for os-autoinst added

**#10 - 2020-05-12 13:37 - tinita**

- Blocked by deleted (action #66721: Use GitHub actions for os-autoinst)

**#11 - 2020-05-12 13:38 - tinita**

- Related to action #66721: Use GitHub actions for os-autoinst added

**#12 - 2020-05-12 13:38 - tinita**

- Related to deleted (action #66721: Use GitHub actions for os-autoinst)

**#13 - 2020-05-12 13:39 - tinita**

- Related to action #66721: Use GitHub actions for os-autoinst added

**#14 - 2020-07-28 11:27 - okurz**

- Target version set to Ready

**#15 - 2020-09-08 13:30 - okurz**

[mkittlertinita](#) see [#53546#note-8](#) , we still have both "openqa\_dev" and "os-autoinst\_dev" in <https://build.opensuse.org/project/show/devel:openQA> , can we delete the first or something?

**#16 - 2020-09-08 13:33 - tinita**

Yes, I think [https://build.opensuse.org/package/show/devel:openQA/openqa\\_dev](https://build.opensuse.org/package/show/devel:openQA/openqa_dev) can now be deleted

Edit: actually it is still used as the documented way to run tests locally.

**#17 - 2020-09-08 14:15 - tinita**

PR to remove Dockerfile in openQA: <https://github.com/os-autoinst/openQA/pull/3372>

**#18 - 2020-10-13 21:43 - okurz**

- Related to action #73309: every time a direct dependency is updated in Factory our CI jobs fail until the package is updated added

**#19 - 2021-02-11 08:57 - okurz**

- Status changed from Blocked to New

- Assignee deleted (okurz)

- Priority changed from Normal to Low

- Target version changed from Ready to future

a major part of making working with dependencies has been done by tinita in [#43619](#) . We don't have a good idea for the other points mentioned. I would still favor a solution that is less openSUSE centric and duplicates less code in our source repo even if it is auto-generated so I will keep the ticket but keep out of our backlog as an idea for later.