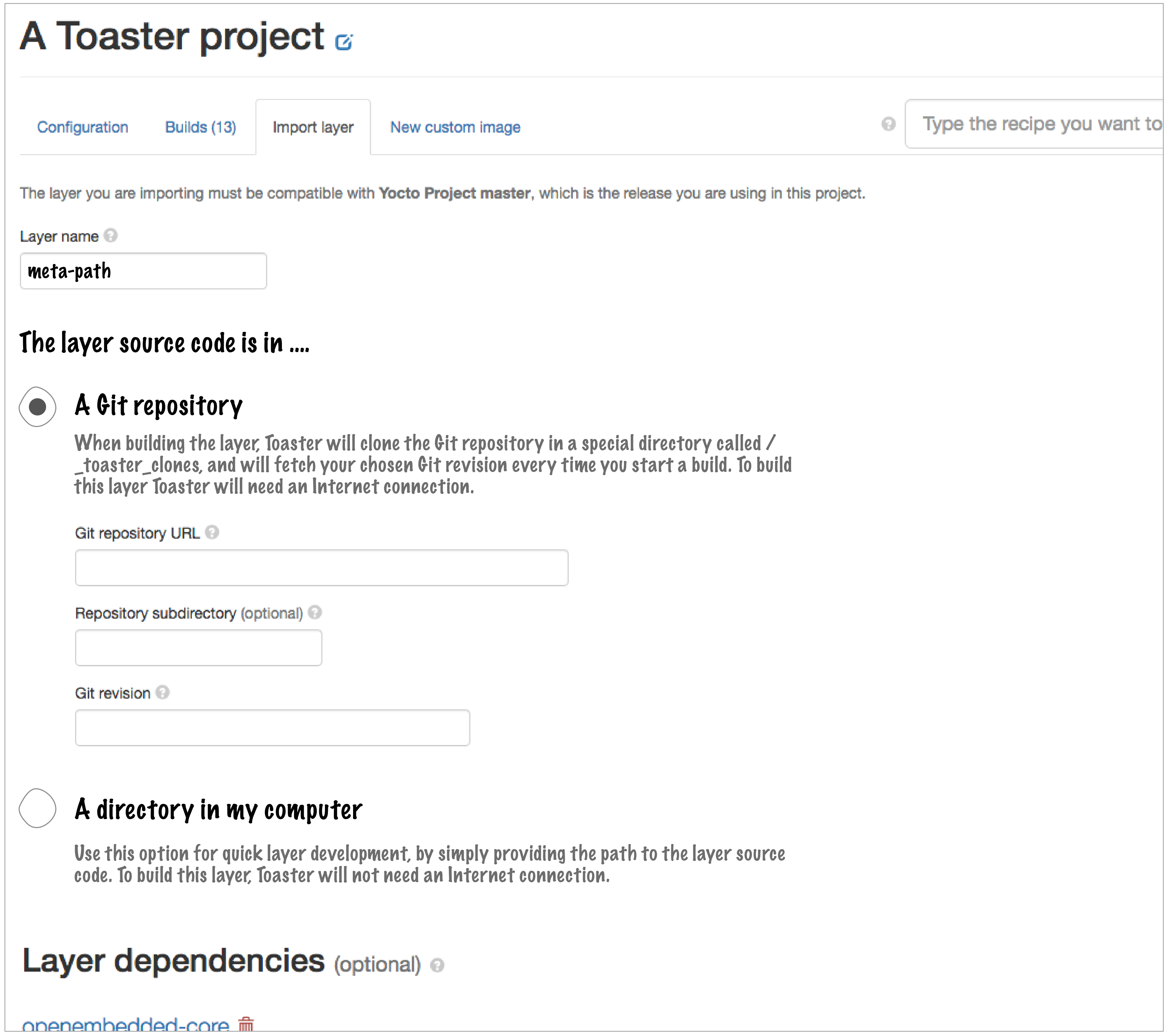
Enabling non-Git layers in Toaster

Design review outcome - 7th July 2016

Reference material:

https://lists.yoctoproject.org/pipermail/toaster/2016-June/004855.html
https://youtu.be/N6gvTtZUP3Y
https://lists.yoctoproject.org/pipermail/toaster/2016-July/004942.html
https://youtu.be/z5wVjBwJDzY

Table of contents



A Toaster project

Configuration | Builds (13) | Import layer | New custom image | Type the recipe you want to

The layer you are importing must be compatible with **Yocto Project master**, which is the release you are using in this project.

Layer name: meta-path

The layer source code is in ....

● A Git repository

When building the layer, Toaster will clone the Git repository in a special directory called /_toaster_clones, and will fetch your chosen Git revision every time you start a build. To build this layer Toaster will need an Internet connection.

Git repository URL

Repository subdirectory (optional)

Git revision

○ A directory in my computer

Use this option for quick layer development, by simply providing the path to the layer source code. To build this layer, Toaster will not need an Internet connection.

Layer dependencies (optional)

openembedded-core

Overview

Currently Toaster can only build layers as Git repositories. These causes problems to some users, who distribute layers as tarballs to customers. It also makes for a cumbersome layer development workflow, where you must push every single change to the corresponding Git repository before running a build from Toaster.

To solve these problems, we have decided to remove the Git constraint, and enable Toaster to build non-Git layers.

High level design

We went through 2 rounds of high level design, which sets the general direction for the web interface of this feature.

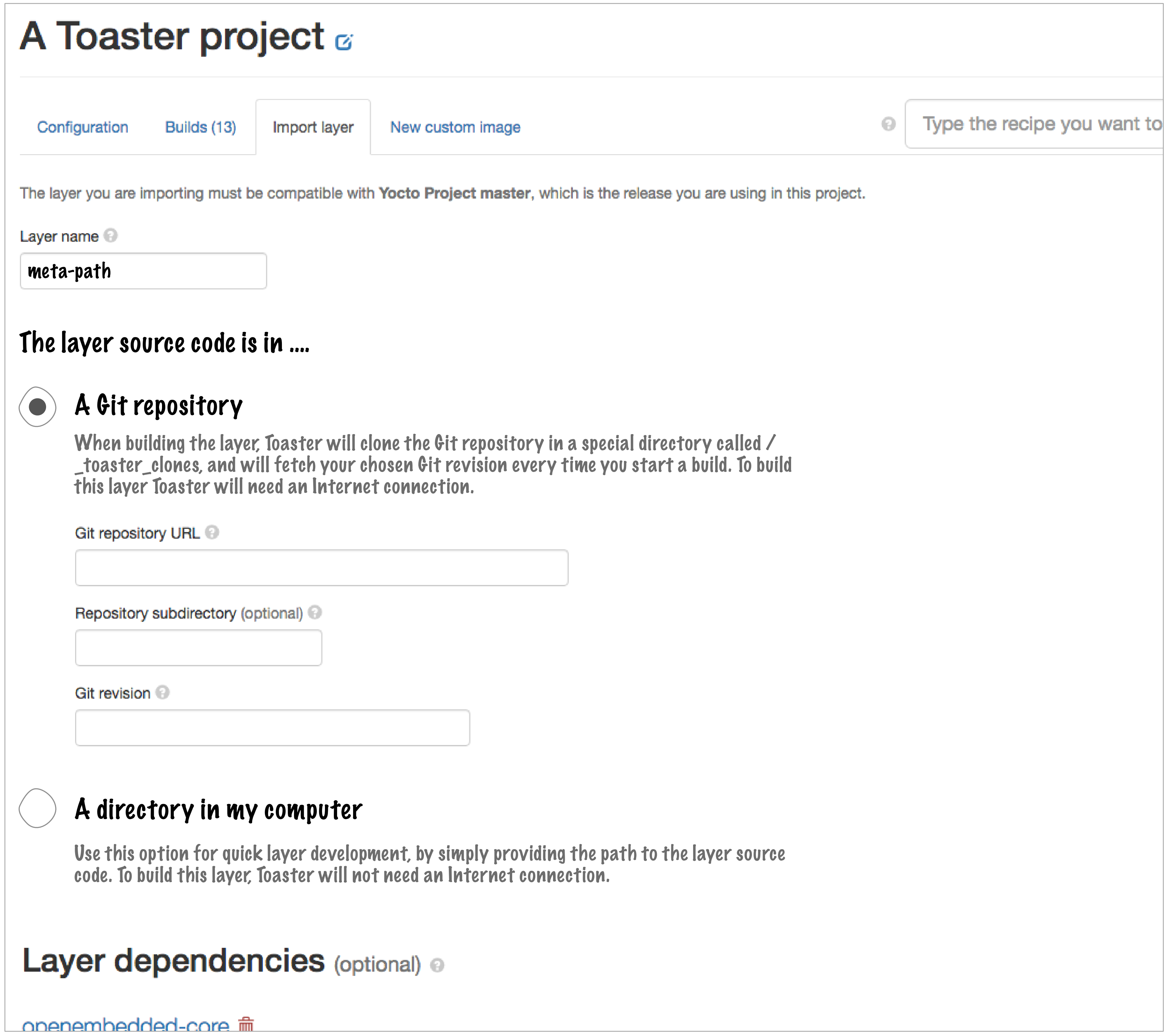
For information on the first round see

https://lists.yoctoproject.org/pipermail/toaster/2016-June/004855.html

and

https://youtu.be/N6gvTtZUP3Y

Overview (contnd.)



A Toaster project

Configuration | Builds (13) | Import layer | New custom image

Type the recipe you want to

The layer you are importing must be compatible with **Yocto Project master**, which is the release you are using in this project.

Layer name

meta-path

The layer source code is in ....

● A Git repository

When building the layer, Toaster will clone the Git repository in a special directory called /_toaster_clones, and will fetch your chosen Git revision every time you start a build. To build this layer Toaster will need an Internet connection.

Git repository URL

Repository subdirectory (optional)

Git revision

○ A directory in my computer

Use this option for quick layer development, by simply providing the path to the layer source code. To build this layer, Toaster will not need an Internet connection.

Layer dependencies (optional)

openembedded-core

For information on the second round see

https://lists.yoctoproject.org/pipermail/toaster/2016-July/004942.html

and

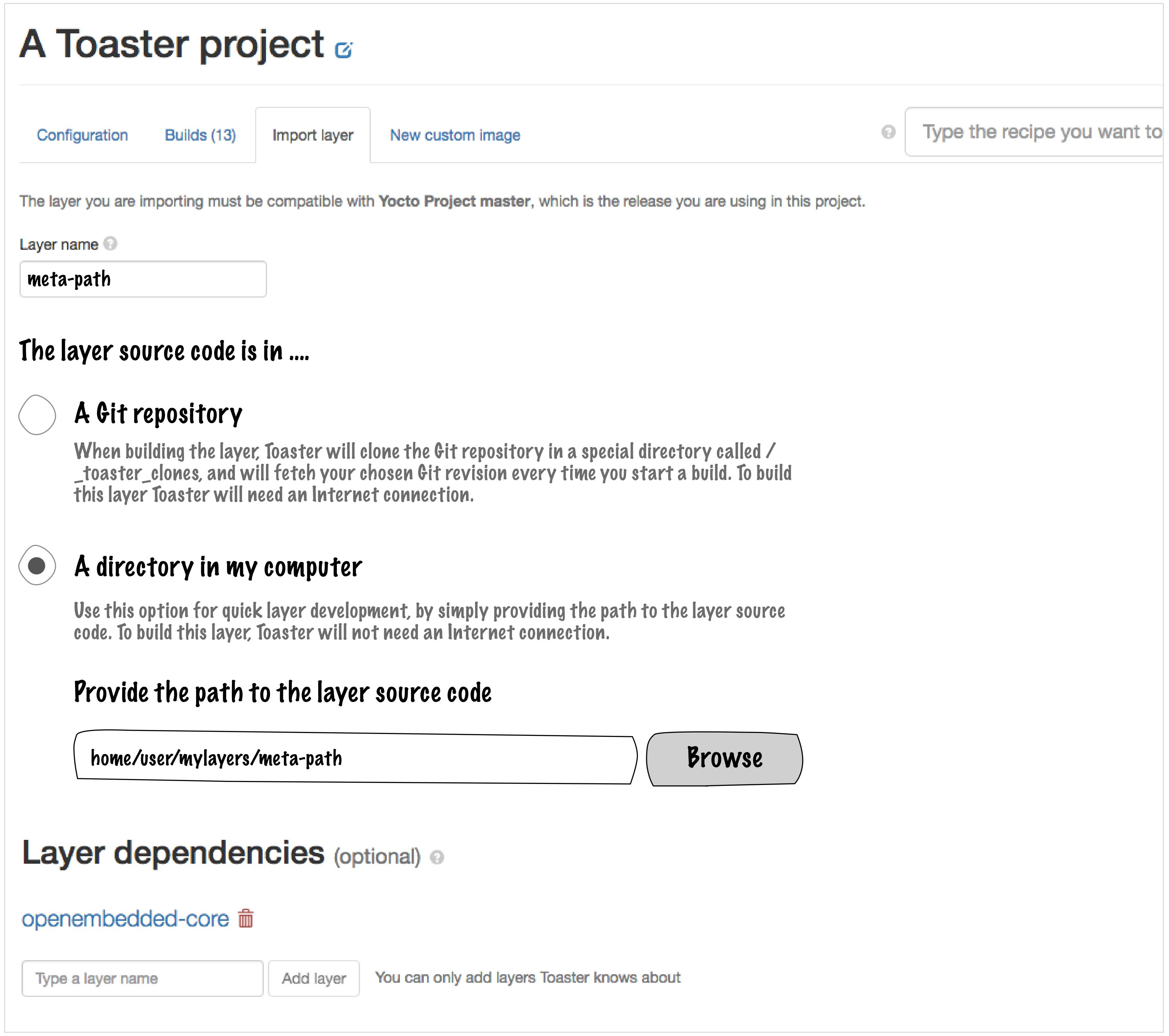
https://youtu.be/z5wVjBwJDzY

We then met on July 7th to review the second round.

The review was attended by Brian Avery, Michael Wood, David Reyna, Sujith H., Kathy Tufto and Belén Barros.

There was general agreement that the second round was an improvement over the first, and can be used as the baseline for the design of this feature.

We also discussed specific aspects of the design. A summary of the discussion follows.



A Toaster project

The layer you are importing must be compatible with **Yocto Project master**, which is the release you are using in this project.

Layer name: meta-path

The layer source code is in ....

( ) A Git repository

When building the layer, Toaster will clone the Git repository in a special directory called /_toaster_clones, and will fetch your chosen Git revision every time you start a build. To build this layer Toaster will need an Internet connection.

(●) A directory in my computer

Use this option for quick layer development, by simply providing the path to the layer source code. To build this layer, Toaster will not need an Internet connection.

Provide the path to the layer source code: home/user/mylayers/meta-path

Layer dependencies (optional)

openembedded-core

You can only add layers Toaster knows about

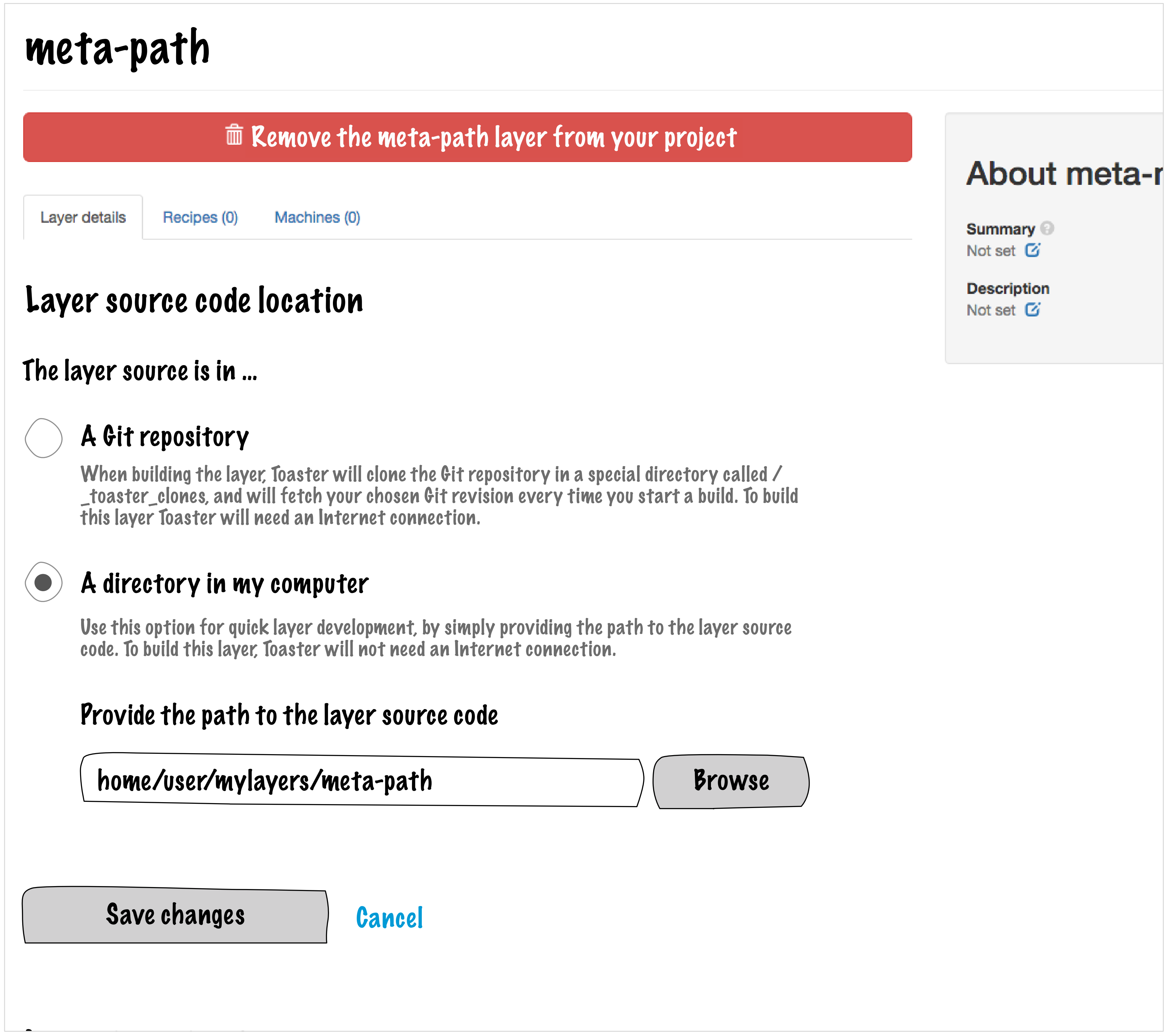
Non-Git layers and remote Toaster instances

To import a non-Git layer into Toaster, you will need to specify the path to the layer source code. Most of the time, the source code will be in a directory in the developer's computer.

This is no problem when Toaster is running in that computer. But if Toaster is running on a different machine, we would need to provide a way to upload the layer source code to the Toaster server.

This was deemed valuable, but not required for a first version of the feature, which is mostly targeting layer development with Toaster running locally.

It was also highlighted that the path does not need to point to a local directory, but to any location accessible to Toaster (for example, a directory in an NFS volume).



meta-path

Remove the meta-path layer from your project

Layer details | Recipes (0) | Machines (0)

Layer source code location

The layer source is in ...

( ) A Git repository

When building the layer, Toaster will clone the Git repository in a special directory called /_toaster_clones, and will fetch your chosen Git revision every time you start a build. To build this layer Toaster will need an Internet connection.

(●) A directory in my computer

Use this option for quick layer development, by simply providing the path to the layer source code. To build this layer, Toaster will not need an Internet connection.

Provide the path to the layer source code

home/user/mylayers/meta-path [Browse]

[Save changes] Cancel

About meta-r

Summary
Not set

Description
Not set

Switching between paths and repos

The design allows users to edit the details of an imported layer, including switching its source code location between a directory and a Git repository.

Some people argued that this functionality was unlikely to be used frequently. Those who wanted to do such things could simply import a new version of the layer.

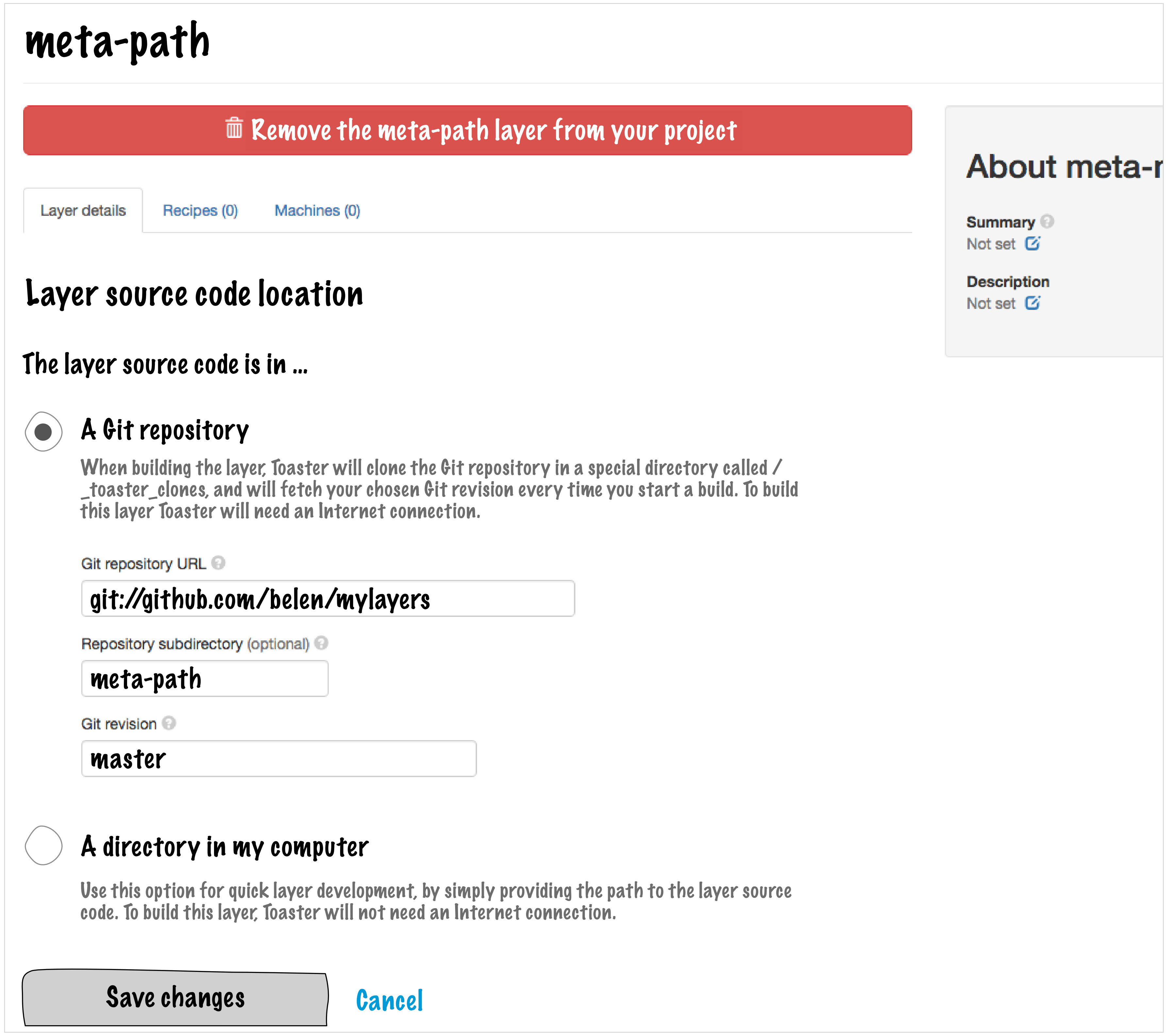
However, during the design review itself all attendees agreed this was a useful feature. We have therefore decided to keep it.

Before the design review, David Reyna sent the following comment about this functionality:

```
“2) When you switch a layer from a
local path to a git path (or the other
way), does Toaster remember the other
values so that you can switch back
and forth without reentering all the
data?
```

Enabling non-Git layers in Toaster - Design review outcome

Switching between paths and repos (contnd.)



meta-path

Remove the meta-path layer from your project

Layer details | Recipes (0) | Machines (0)

Layer source code location

The layer source code is in ...

● A Git repository

When building the layer, Toaster will clone the Git repository in a special directory called /_toaster_clones, and will fetch your chosen Git revision every time you start a build. To build this layer Toaster will need an Internet connection.

Git repository URL: git://github.com/belen/mylayers

Repository subdirectory (optional): meta-path

Git revision: master

○ A directory in my computer

Use this option for quick layer development, by simply providing the path to the layer source code. To build this layer, Toaster will not need an Internet connection.

Save changes | Cancel

About meta-r

Summary: Not set

Description: Not set

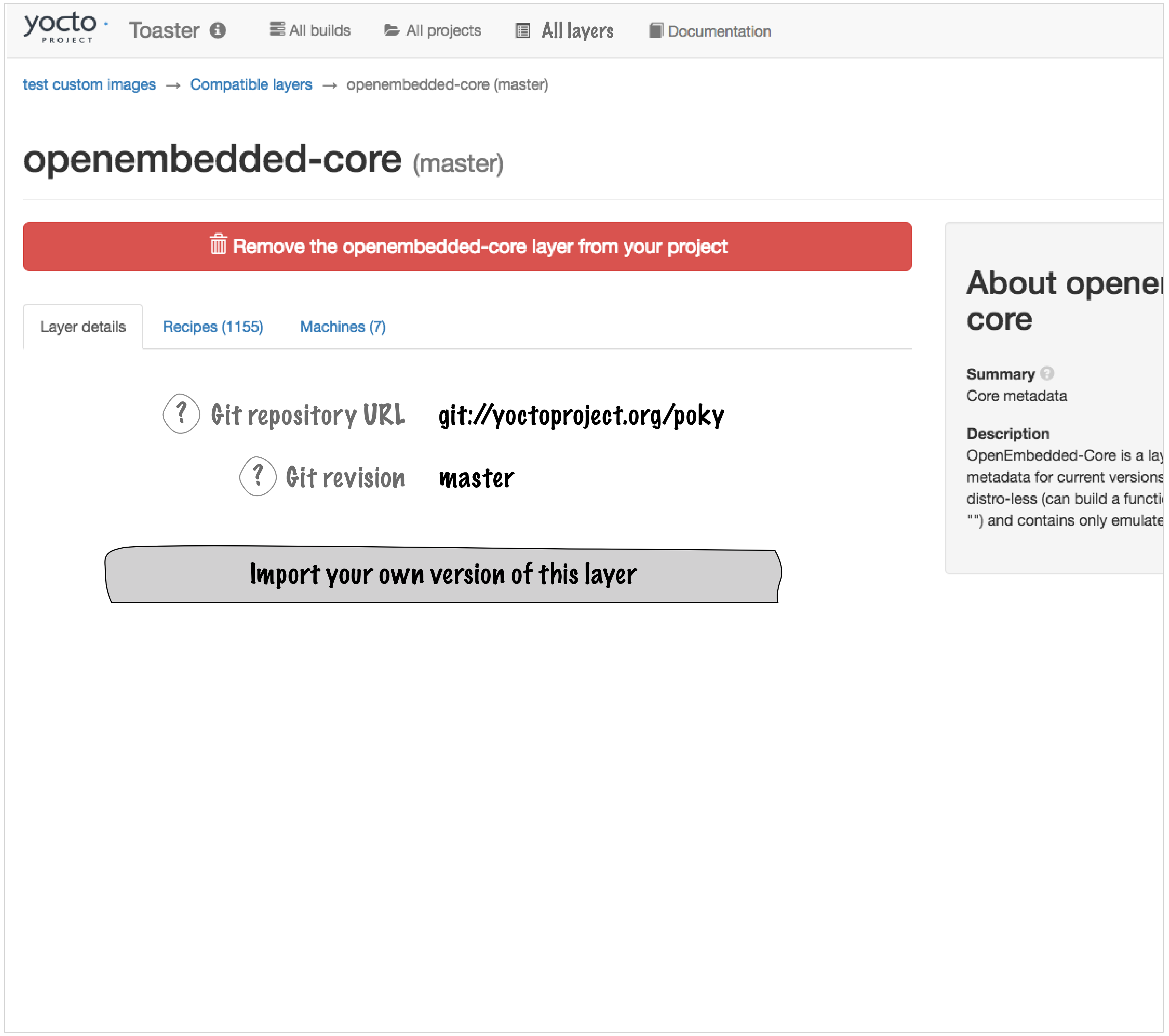
```
That would be handy if you are testing
a local development layer versus
the formal git layer and are switching
back and forth, plus that hidden
persistent effectively provides the
feature from your previous version
without the visual overhead that this
second version is avoiding.”
```

Brian Avery agreed that storing the previous value to save users from entering it again would be useful.

Based on how easy or hard would be to implement, we could aim to include it in the version 1 of the non-Git layers functionality or add it later on.

Enabling non-Git layers in Toaster - Design review outcome

Import your own version of a non-editable layer



test custom images → Compatible layers → openembedded-core (master)

openembedded-core (master)

Remove the openembedded-core layer from your project

Layer details | Recipes (1155) | Machines (7)

Git repository URL git://yoctoproject.org/poky

Git revision master

Import your own version of this layer

About opene
core

Summary
Core metadata

Description
OpenEmbedded-Core is a lay
metadata for current versions
distro-less (can build a functi
"") and contains only emulate

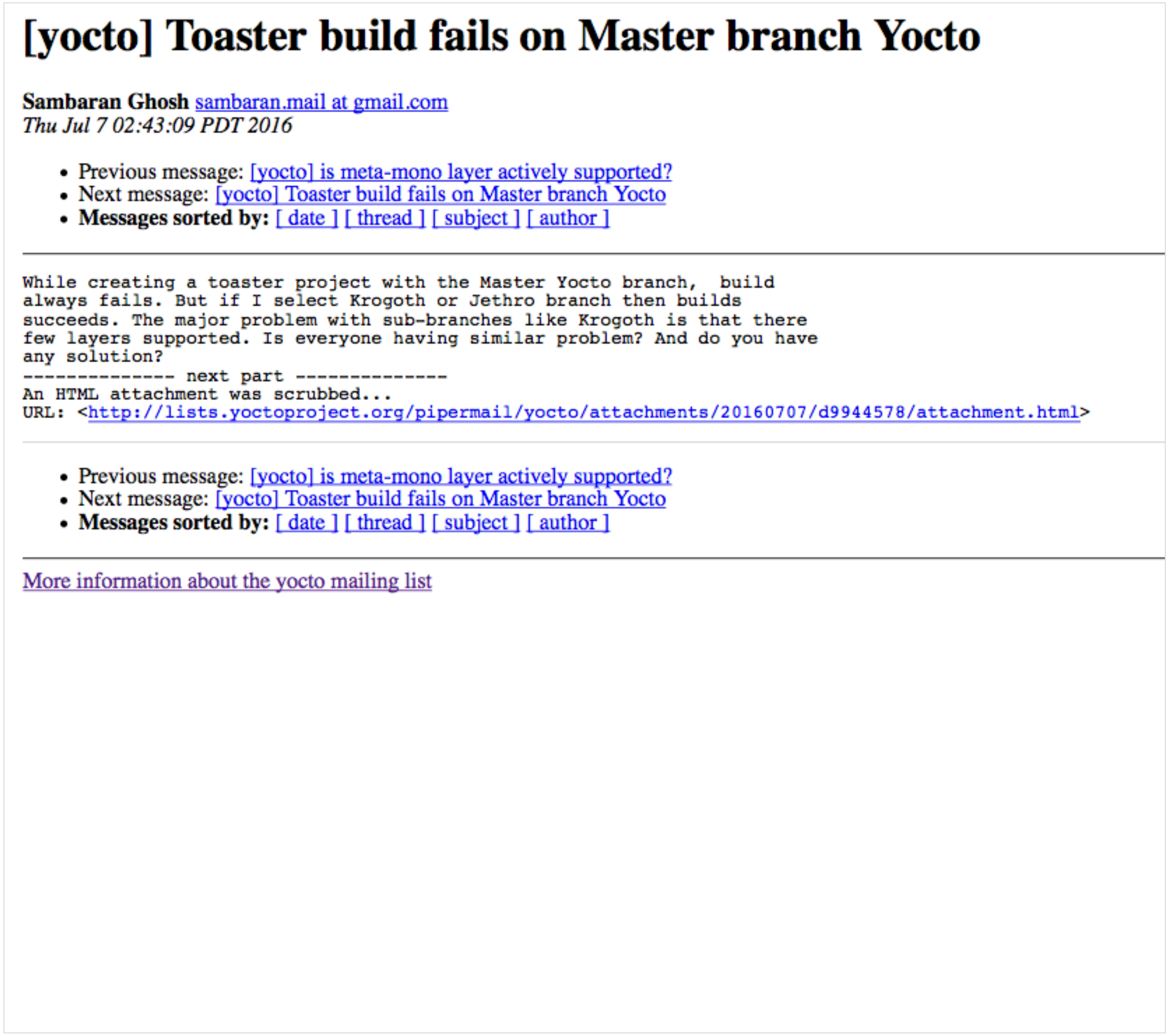
Import you own version of a non-editable layer

The design also added a new option to the layer details pages of non-editable layers. Those are the layers whose data comes from the OpenEmbedded layer index, or is pre-populated in the Toaster database.

The option aimed to highlight the fact that, although these layers are not editable, users can still build their own versions by importing them as separate layers.

There was general agreement that this should not be needed, and will be scrapped from the design going forward.

An alternative that came up during the review was providing documentation, or help text within the page.



[yocto] Toaster build fails on Master branch Yocto

Sambaran Ghosh sambaran.mail at gmail.com
Thu Jul 7 02:43:09 PDT 2016

- Previous message: [yocto] is meta-mono layer actively supported?
- Next message: [yocto] Toaster build fails on Master branch Yocto
- **Messages sorted by:** [date] [thread] [subject] [author]

```
While creating a toaster project with the Master Yocto branch,  build
always fails. But if I select Krogoth or Jethro branch then builds
succeeds. The major problem with sub-branches like Krogoth is that there
few layers supported. Is everyone having similar problem? And do you have
any solution?
-------------- next part --------------
An HTML attachment was scrubbed...
URL: <http://lists.yoctoproject.org/pipermail/yocto/attachments/20160707/d9944578/attachment.html>
```

- Previous message: [yocto] is meta-mono layer actively supported?
- Next message: [yocto] Toaster build fails on Master branch Yocto
- **Messages sorted by:** [date] [thread] [subject] [author]

More information about the yocto mailing list

Showing all layer versions available

I have received complaints about the fact that Toaster only shows you layers with branches that are compatible with the selected project release. For example, if you create a Toaster project and you choose the "krogoth" release, the list of layers Toaster shows you in that project will only include layers that, in the OpenEmbedded layer index, have a krogoth branch.

If the layer you want to use doesn't have a krogoth branch, but it has a master branch (meta-raspberrypi being currently an example of this), you will not find the layer anywhere in your krogoth project, and you might assume it doesn't exist.

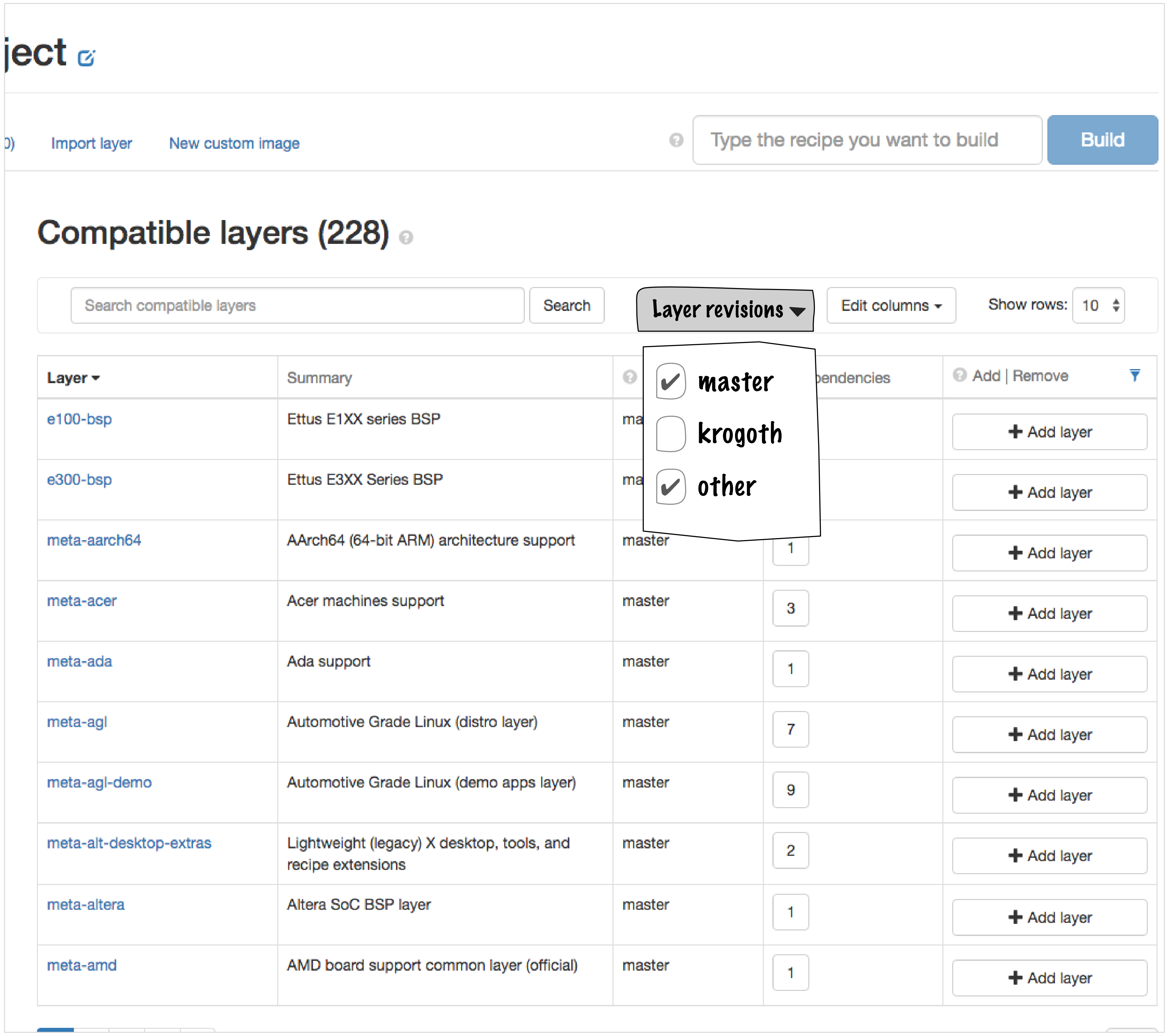
Emails like this

https://lists.yoctoproject.org/pipermail/yocto/2016-July/031013.html

are evidence of this being an usability problem, since users do not realise that they can import the layers to work around the release constraint.

Enabling non-Git layers in Toaster - Design review outcome

Showing all layer versions available (contnd.)



ject

Import layer | New custom image | Type the recipe you want to build | Build

Compatible layers (228)

Search compatible layers | Search | Layer revisions | Edit columns | Show rows: 10

- [✓ master]
- krogoth
- [✓ other]

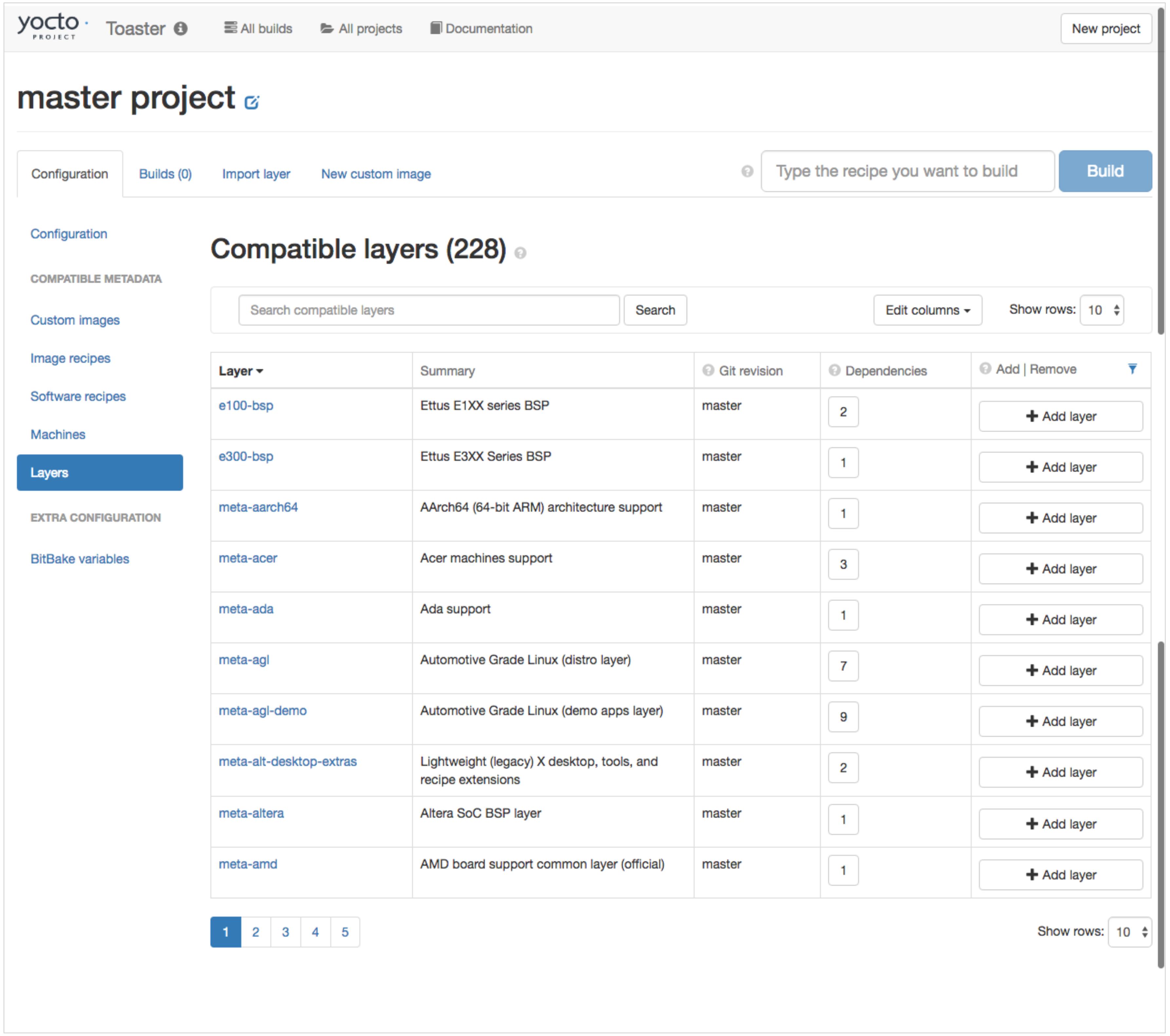
Layer	Summary		Dependencies	Add | Remove
e100-bsp	Ettus E1XX series BSP	ma		Add layer
e300-bsp	Ettus E3XX Series BSP	ma		Add layer
meta-aarch64	AArch64 (64-bit ARM) architecture support	master	1	Add layer
meta-acer	Acer machines support	master	3	Add layer
meta-ada	Ada support	master	1	Add layer
meta-agl	Automotive Grade Linux (distro layer)	master	7	Add layer
meta-agl-demo	Automotive Grade Linux (demo apps layer)	master	9	Add layer
meta-alt-desktop-extras	Lightweight (legacy) X desktop, tools, and recipe extensions	master	2	Add layer
meta-altera	Altera SoC BSP layer	master	1	Add layer
meta-amd	AMD board support common layer (official)	master	1	Add layer

The problem could be solved by adding an option to show all layer versions supported by Toaster (normally master, plus one or two stable releases). Such option could take the form of an additional menu in the “compatible layers” table, labelled “Layer revisions” or similar.

We ran out of time before we could properly discuss this functionality, but Brian Avery is particularly fond of it.

Sujith H. and David Reyna had questions regarding the "Other" categorgy included in that ”Layer revisions" menu.

The design is still a bit rough around the edges, but since this is not strictly part of the non-Git layers functionality, we can refine it at a later stage.



master project

Compatible layers (228)

Layer	Summary	Git revision	Dependencies	Add | Remove
e100-bsp	Ettus E1XX series BSP	master	2	Add layer
e300-bsp	Ettus E3XX Series BSP	master	1	Add layer
meta-aarch64	AArch64 (64-bit ARM) architecture support	master	1	Add layer
meta-acer	Acer machines support	master	3	Add layer
meta-ada	Ada support	master	1	Add layer
meta-agl	Automotive Grade Linux (distro layer)	master	7	Add layer
meta-agl-demo	Automotive Grade Linux (demo apps layer)	master	9	Add layer
meta-alt-desktop-extras	Lightweight (legacy) X desktop, tools, and recipe extensions	master	2	Add layer
meta-altera	Altera SoC BSP layer	master	1	Add layer
meta-amd	AMD board support common layer (official)	master	1	Add layer

Django fixtures for Toaster configuration

We also discussed the ongoing work on Django fixtures, which will replace the existing toasterconf.json file for defining Toaster configuration. See Bugzilla

https://bugzilla.yoctoproject.org/show_bug.cgi?id=9582

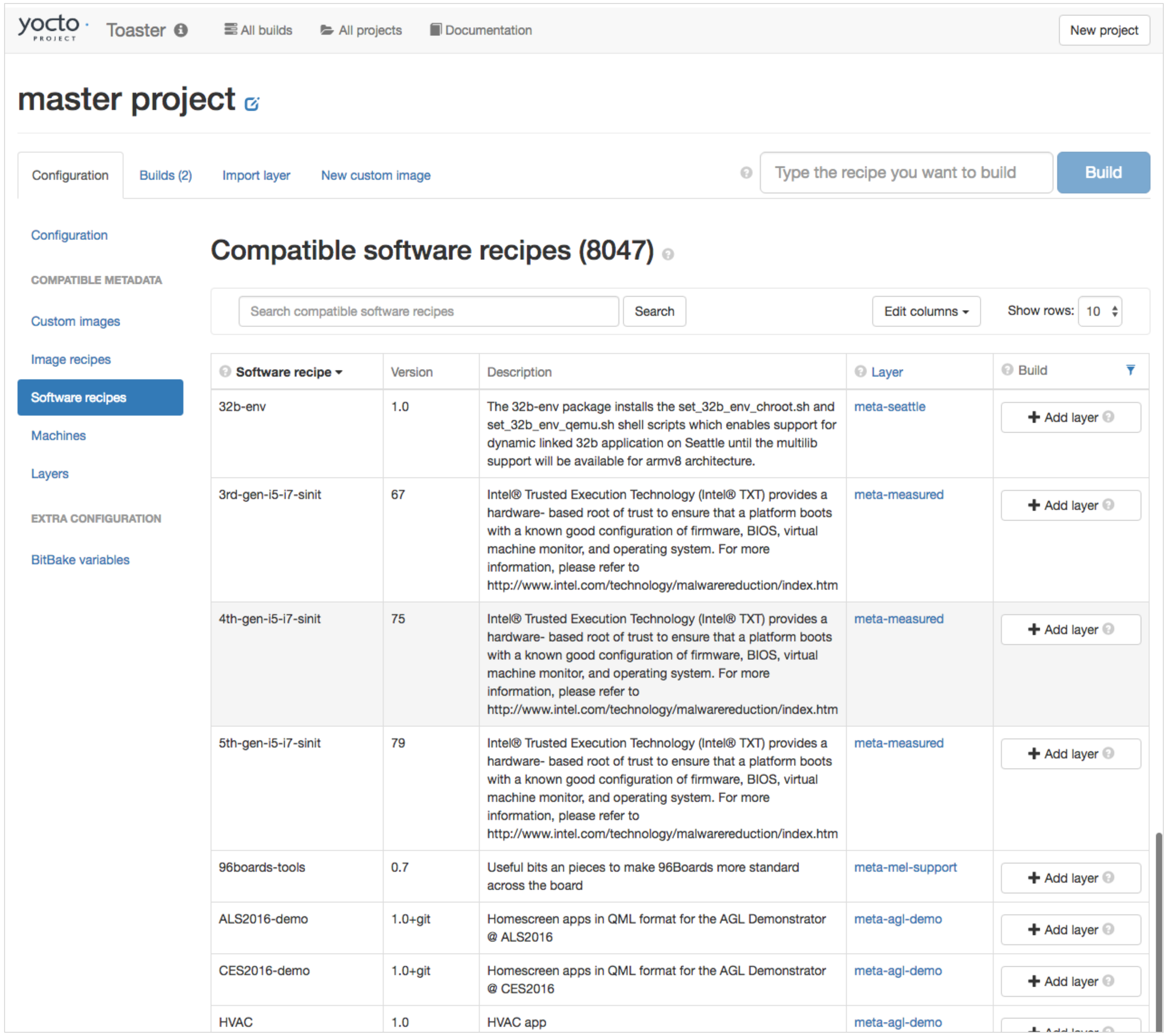
The discussion started from questions about providing bulk addition of layers in Toaster.

The move to fixtures will allow companies to ship Toaster with a pre-populated database containing information about their layer set.

Such layers will appear listed in the “compatible layers” table. Customers can then use the “add layer” buttons to select the ones they want to use.

Enabling non-Git layers in Toaster - Design review outcome

Django fixtures for Toaster configuration (contnd.)



Software recipe	Version	Description	Layer	Build
32b-env	1.0	The 32b-env package installs the set_32b_env_chroot.sh and set_32b_env_qemu.sh shell scripts which enables support for dynamic linked 32b application on Seattle until the multilib support will be available for armv8 architecture.	meta-seattle	Add layer
3rd-gen-i5-i7-sinit	67	Intel® Trusted Execution Technology (Intel® TXT) provides a hardware- based root of trust to ensure that a platform boots with a known good configuration of firmware, BIOS, virtual machine monitor, and operating system. For more information, please refer to http://www.intel.com/technology/malwarereduction/index.htm	meta-measured	Add layer
4th-gen-i5-i7-sinit	75	Intel® Trusted Execution Technology (Intel® TXT) provides a hardware- based root of trust to ensure that a platform boots with a known good configuration of firmware, BIOS, virtual machine monitor, and operating system. For more information, please refer to http://www.intel.com/technology/malwarereduction/index.htm	meta-measured	Add layer
5th-gen-i5-i7-sinit	79	Intel® Trusted Execution Technology (Intel® TXT) provides a hardware- based root of trust to ensure that a platform boots with a known good configuration of firmware, BIOS, virtual machine monitor, and operating system. For more information, please refer to http://www.intel.com/technology/malwarereduction/index.htm	meta-measured	Add layer
96boards-tools	0.7	Useful bits an pieces to make 96Boards more standard across the board	meta-mel-support	Add layer
ALS2016-demo	1.0+git	Homescreen apps in QML format for the AGL Demonstrator @ ALS2016	meta-agl-demo	Add layer
CES2016-demo	1.0+git	Homescreen apps in QML format for the AGL Demonstrator @ CES2016	meta-agl-demo	Add layer
HVAC	1.0	HVAC app	meta-agl-demo	Add layer

It should also be possible to pre-populate the Toaster database with recipe and machine information from those layers.

This greatly enhances the usability of Toaster, since it allows customers to search for recipes and build them easily.

To generate the database content, you could use a custom instance of the layer index to parse the layers. Alternatively, you could build the layers beforehand.

The advantage of the latter is that it will also generate package information, which would make image customisation (adding and removing packages to / from existing images) much easier.

Suggested Bugzilla features

Based on the discussion, the non-Git feature could be broken down into the following Bugzilla features:

1. Import a layer that points to a directory instead of a Git repository.
Target release: 2.2

2. Build a layer that points to a directory instead of a Git repository.
Target release: 2.2

3. Edit the details of an imported layer, so that you can edit the source code location details, and move between a local directory and a Git repository.
Target release: 2.2

4. For imported layers, remember the previous values for directories and Git repositories.
Target release: 2.3

5. Show all layer revisions supported by Toaster, not just the ones compatible with the current project.
Target release: 2.3

We already have a Bugzilla entry for the Django fixtures work, targeting the 2.2. release:

https://bugzilla.yoctoproject.org/show_bug.cgi?id=9582

Detailed designs

From the design perspective, the next step is creating detailed design specification documents for the 5 Bugzilla features, and prototype the web interface using Bootstrap 3.

Questions or comments?

Email them to the Toaster mailing list

toaster@yoctoproject.org

Subscribe at

https://lists.yoctoproject.org/listinfo/toaster