Subject: UppHub - new package registration
Posted by Klugier on Sat, 23 Jan 2021 21:13:59 GMT
View Forum Message <> Reply to Message

Hello,

Let's assume I created package "MyFirstUppHubPackage". What should be the process of registering it in global registry? Should it be done via pull request or we assume that we will add by adding appropriate message on the forum with UPPHUB_BEGIN and UPPHUB_END. What will happen in such case when site will be offline? We had several outages in the past, so the continuity can not be guarantee.

Backing to UppHub declaration:

```
 { "name": "URR",
      "packages": [ "Urr" ],
      "description": "Simple UDP Request-Response protocol",
      "repository": "https://github.com/mirek-fidler/urr.git",
      "status": "stable",
      "category": "networking",
      "readme": "https://raw.githubusercontent.com/mirek-fidler/urr/master/README.md"
    }
```

Should this manifest be somehow provided in the source files, so it can be modified independently. Here is the modification:

```
UPPHUB_BEGIN
{
  "nests": [
      "https://raw.githubusercontent.com/mirek-fidler/Skylark/upp-hub/manifest.json",
  ]
}
UPPHUB_END
```

upp-hub is single branch with manifest file. Thanks to that, we will be able to connect packages that are not in global registry. I mean would be good to extend UppHub by "Add custom" and then passing the link " https://raw.githubusercontent.com/mirek-fidler/Skylark/upp-hub/manifest.json". In such case it should be registered locally. Moreover, we will be able to add new "registry" package basing on simple PR (I suggest separate repository for it within ultimatepp organization). Mirror, host of this file is welcome just in case.

This is my concerns about current implementation. We need to be very caution here, because one selected pattern will stay with us for the very long time...

Klugier

Subject: Re: UppHub - new package registration
Posted by mirek on Wed, 27 Jan 2021 17:04:22 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 23 January 2021 22:13Hello,

Let's assume I created package "MyFirstUppHubPackage". What should be the process of registering it in global registry?

There will be list maintainer(s) responsible for editing these lists. This cannot be automated. Think of malware...

Quote:
Should it be done via pull request or we assume that we will add by adding appropriate message on the forum with UPPHUB_BEGIN and UPPHUB_END. What will happen in such case when site will be offline? We had several outages in the past, so the continuity can not be guarantee.

The list in the forum is temporary, mostly to test it is possible.

Quote:
Backing to UppHub declaration:

```
{ "name": "URR",
    "packages": [ "Urr" ],
    "description": "Simple UDP Request-Response protocol",
    "repository": "https://github.com/mirek-fidler/urr.git",
    "status": "stable",
    "category": "networking",
    "readme": "https://raw.githubusercontent.com/mirek-fidler/urr/master/README.md"
  }
```

Should this manifest be somehow provided in the source files, so it can be modified independently.

Yes, something like that is already half-implemented.

Quote:
upp-hub is single branch with manifest file. Thanks to that, we will be able to connect packages that are not in global registry. I mean would be good to extend UppHub by "Add custom" and then passing the link " https://raw.githubusercontent.com/mirek-fidler/Skylark/upp-hub/manifest.json". In such case it should be registered locally. Moreover, we will be able to add new "registry" package basing on simple PR (I suggest separate repository for it within ultimatepp organization). Mirror, host of this file is welcome just in case.

I am really concerned about making this automated. BTW, you can already use the list on your harddrive if you activate "Verbose" mode, there is a setting in UppHub dialog. This is mostly meant for testing though.

Subject: Re: UppHub - new package registration
Posted by mirek on Thu, 28 Jan 2021 12:29:27 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 23 January 2021 22:13

```
 { "name": "URR",
     "packages": [ "Urr" ],
     "description": "Simple UDP Request-Response protocol",
     "repository": "https://github.com/mirek-fidler/urr.git",
     "status": "stable",
     "category": "networking",
     "readme": "https://raw.githubusercontent.com/mirek-fidler/urr/master/README.md"
   }
```

Should this manifest be somehow provided in the source files, so it can be modified independently. Here is the modification:

```
UPPHUB_BEGIN
{
  "nests": [
     "https://raw.githubusercontent.com/mirek-fidler/Skylark/upp-hub/manifest.json",
  ]
}
UPPHUB_END
```

Check how Skylark is now linked:

https://www.ultimatepp.org/forums/index.php?t=msg&goto=5 5542&#msg_55542

Subject: Re: UppHub - new package registration
Posted by Oblivion on Sat, 30 Jan 2021 11:18:50 GMT
View Forum Message <> Reply to Message

Hello Mirek,
I think the package licenses (BSD/MIT/GPL, etc.) should also be displayed on the packages list. It

is an important info.

Also, I an eager to move at lease some components to UppHub, if not the whole upp-components repo.
I am currenty assesing a viable strategy to do that. (I need to keep upp-componetns repo intact, possibly synced with upphub pckages, as I can't assume that all users use TheIDE or Upp >= 2021.1)

Best regards,
Oblivion.

Subject: Re: UppHub - new package registration
Posted by mirek on Sat, 30 Jan 2021 19:41:55 GMT
View Forum Message <> Reply to Message

Have a good strategy for moving packages from bazaar (basically, import upp, clone it, delete everything except the package(s) you want to move, move package to repo root) - I will post the recipe soon in bazaar.

Good point about the licenses.

Mirek

Subject: Re: UppHub - new package registration
Posted by Oblivion on Sun, 31 Jan 2021 22:37:44 GMT
View Forum Message <> Reply to Message

By the way,

Do we really want to use seemingly arbitrary categories in the list? I am almost sure that those tags will be bloom after the takeoff, and we'll probably get redundant category tags like "GUI", "user-interface", etc.

Wouldn't it be better to aling the categories with upp's main directory structure, ie.a functional categorization, like "Core", "Core/Plugin", "CtrlLib", "Draw", "Examples" (users may want to provide more example code with documentation), "Application", etc.
Package readme, if provided, and short summary that appears on the list, already give the idea as to what the package does.

Best regards,
Oblivion

Subject: Re: UppHub - new package registration
Posted by mirek on Sun, 31 Jan 2021 23:56:37 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Sun, 31 January 2021 23:37By the way,

Do we really want to use seemingly arbitrary categories in the list? I am almost sure that those tags will be bloom after the takeoff, and we'll probably get redundant category tags like "GUI", "user-interface", etc.

Wouldn't it be better to aling the categories with upp's main directory structure, ie.a functional categorization, like "Core", "Core/Plugin", "CtrlLib", "Draw", "Examples" (users may want to provide more example code with documentation), "Application", etc.
Package readme, if provided, and short summary that appears on the list, already give the idea as to what the package does.

Best regards,
Oblivion

That is why we have list maintainers, right?

Subject: Re: UppHub - new package registration
Posted by Oblivion on Mon, 01 Feb 2021 10:26:48 GMT
View Forum Message <> Reply to Message

Ok, then.

I have attempted to register my first upphub package (MessageCtrl) and called it a "widget" (for the time being, until we come up with a better categorization scheme)

Best regards,
Oblivion.

Subject: Re: UppHub - new package registration
Posted by Novo on Mon, 01 Feb 2021 21:24:02 GMT
View Forum Message <> Reply to Message

IMHO, it makes sense to add sections "supported"/"unsupported" to the JSON package registration format because, for example, gdal cannot be compiled with cpp17, Turtle cannot be compiled on Mac, e.t.c.
At least, this should help to understand where a project is supposed to work.
And secondly, this information will be machine-readable and potentially can be used by tools like a build automation system.

Just my two cents.

Subject: Re: UppHub - new package registration
Posted by mirek on Mon, 01 Feb 2021 22:10:44 GMT
View Forum Message <> Reply to Message

Novo wrote on Mon, 01 February 2021 22:24IMHO, it makes sense to add sections "supported"/"unsupported" to the JSON package registration format because, for example, gdal cannot be compiled with cpp17, Turtle cannot be compiled on Mac, e.t.c.
At least, this should help to understand where a project is supposed to work.
And secondly, this information will be machine-readable and potentially can be used by tools like a build automation system.

Just my two cents.

Yes. Good point. So licenses, supported OSes. I am also thinking about linking documentation root....

Mirek

Subject: Re: UppHub - new package registration
Posted by Novo on Tue, 02 Feb 2021 00:16:40 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 01 February 2021 17:10
Yes. Good point. So licenses, supported OSes. I am also thinking about linking documentation root....

Mirek
I'd say supported OSes, compilers (a lot of code won't compile with msvc), version of C++ (gdal won't compile for cpp17, and a lot of code require cpp17 these days).

I'd also add "requires"/"depends_on" for external packages and libraries.

Subject: Re: UppHub - new package registration
Posted by Novo on Tue, 02 Feb 2021 14:58:02 GMT
View Forum Message <> Reply to Message

We also need a "branch", because default "master" was changed to "main" and external tools often require an explicit branch name.
That includes my build automation system.
I cannot build MessageCtrl because of that.

Subject: Re: UppHub - new package registration
Posted by Oblivion on Tue, 02 Feb 2021 21:59:45 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I have noticed an issues with the license and readme files: Since these files reside in the root directory, TheIDE is unable to read them (in the license information window or via the package files section.)

Best regards,
Oblivion

Subject: Re: UppHub - new package registration
Posted by mirek on Sat, 13 Feb 2021 07:59:10 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Tue, 02 February 2021 22:59Hello Mirek,

I have noticed an issues with the license and readme files: Since these files reside in the root directory, TheIDE is unable to read them (in the license information window or via the package files section.)

Best regards,
Oblivion

You can still open them with "Open any file", no? And Copying should be inside the package (maybe as duplicate).

Subject: Re: UppHub - new package registration
Posted by mirek on Sat, 13 Feb 2021 08:00:40 GMT
View Forum Message <> Reply to Message

One general suggestion: Let us tone down a bit "Ultimate". I am trying to use U++ instead everywhere.

E.g. package maintainers should suggest replacing Ultimate++ with U++ in readme files and elsewhere.

Mirek

Subject: Re: UppHub - new package registration
Posted by Klugier on Sun, 14 Feb 2021 19:32:44 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I would like to point UppHub users to "release-1.10.0" tag instead of main/master branch. Can we add "branch" to the UppHub nest declaration? Alternatively, we could add branches as an array that will point to all available branches that user can select. It will be up to the nest maintainer which branches they want to expose:

```
  {
    "name": "GoogleTest",
    "packages": [
      "plugin/gtest",
      "plugin/gmock"
    ],
    "branches": [
        "release-1.10.0", // The top is default one
        "release-1.9.0",
        "release-candidate-1.9.0"
        "main" // If you want to risk...
    ]
    "description": "The nest contains GoogleTest and GoogleMock libraries, as U++ packages",
    "repository": "https://github.com/klugier/UppGoogleTest.git",
    "status": "stable",
    "category": "testing",
    "readme": "https://raw.githubusercontent.com/klugier/UppGoogleTest/main/README.md" // Upstream README.md is fine, right now. In the future we could change it to readmes and create map with bracnh readme...
  },
```

Thanks to that you could make experiments on main branch without worrying users.

Klugier

Subject: Re: UppHub - new package registration
Posted by mirek on Sun, 14 Feb 2021 22:56:51 GMT
View Forum Message <> Reply to Message

The branch to be installed is there since the beginning - check the docs

---

Subject: Re: UppHub - new package registration
Posted by Klugier on Sun, 14 Feb 2021 23:41:09 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I checked the doc and I found that tag can be defined. I created PR I would be grateful if you will review it. Here is my change:

"repository": "https://github.com/klugier/UppGoogleTest.git release-1.10.0",
"readme":
"https://raw.githubusercontent.com/klugier/UppGoogleTest/release-1.10.0/README.md"

I understand that this is initial version, so the version management is not must. However, we need to add updates possible and let user even download custom commit/branch/version. I do not ses that ones downloaded branch will stay forever without any updates or possibility to change to certain commit. I know that you can remove and then uninstall, but this is not very handy...

Let's assume that I fixed the problem with my module. Then what? In above case with tag the user can not download anything besides the release-1.10.0. I think option like "Install certain commit/branch/tag.." could work in our case. This will remove whole repo and download repo with certain commit. Clear and simple solution.

I would like to also see if I update global tag like "release-1.10.0" to "release-1.11.0" UppHub detects that there is update and user will have the opportunity to reinstall package. This shouldn't be hard to implement... Background task should do the work.

Klugier

Subject: Re: UppHub - new package registration
Posted by mirek on Sun, 14 Feb 2021 23:51:59 GMT
View Forum Message <> Reply to Message

Klugier wrote on Mon, 15 February 2021 00:41
I understand that this is initial version, so the version management is not must. However, we need to add updates possible and let user even download custom commit/branch/version. I do not ses that ones downloaded branch will stay forever without any updates or possibility to change to certain commit. I know that you can remove and then uninstall, but this is not very handy...

You can always create normal nest and git clone anything there.

Quote:

Let's assume that I fixed the problem with my module. Then what? In above case with tag the user can not download anything besides the release-1.10.0. I think option like "Install certain commit/branch/tag.." could work in our case. This will remove whole repo and download repo with certain commit. Clear and simple solution.

a) I think it should be other way around. master is always current stable version. If you want to

experiment, experiment in branch.
b) Well, I actually agree there is an utility in this. But I do not yet see how to do it right. Should e.g. there be some global branch names? How will user know what difference is between branches?
c) Also, UppHub does not need branches listed - that information can be loaded from the git.

There is BTW one other much more serious problem and that is a release of user's application. I have solution for that...

Mirek

Subject: Re: UppHub - new package registration
Posted by Klugier on Mon, 15 Feb 2021 00:44:41 GMT
View Forum Message <> Reply to Message

Hello Mirek,

A) I think for most projects master/main is unstable. If it is stable enough and certain features are present then the master became release via tagging. If it requires more polishing then release branch is created and then tagged if everything is fine. Please look at upp trun (the equivalent of master in git). Sometimes you commit DDUMP and it will not compile. We understand it, because it is trunk. For tagged branch the author guarantees you that it was tested enough, so the trivial problems are not there.

Of course, you could protect master better like running CI/CD (builds and automated tests) on each PR. You can no use direct commits in such case, because you can break compilation. However, you still can sneak some nasty bug that should be eliminated before release (feature freeze phase).

My opinion here is clear. If maintainer provide tagged branch, then it means that users receives more polish and stable product. We should allow that and follow in that direction.

B) The differences between release can be read in releases bookmark on GitHub (Example - similar to Upp Status & Roadmap): https://github.com/klugier/UppGoogleTest/releases). So, the user could be aware of the changes. The use of certain commit/branch could be the situation when you fix certain issue for certain user and would like to give him opportunity to test it from UppHub.

I think we should display in the array "commit/branch" that is actually fetched. On right click with menu we could open dialog where user could specific branch/tag/commit etc... And after accepting he should be re-switched.

C) Agreed!

Anyway, we could discuss it via Skype. For me it is good to clear it up before release.

Klugier

Subject: Re: UppHub - new package registration
Posted by Novo on Mon, 15 Feb 2021 05:12:29 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 14 February 2021 17:56The branch to be installed is there since the beginning - check the docs
Actually, it is hard to spot in the docs.
IMHO, it would be much better (for tooling) to have branch as a separate key-value pair.
I'm currently have a lot of problems because this pair is missing.

Subject: Re: UppHub - new package registration
Posted by Oblivion on Mon, 15 Feb 2021 08:48:23 GMT
View Forum Message <> Reply to Message

Hi,

Quote:I think for most projects master/main is unstable. If it is stable enough and certain features are present then the master became release via tagging. If it requires more polishing then release branch is created and then tagged if everything is fine.

I agree with Mirek here. git-based development is meant to be done via branches. Development should be done in developmental branches and the changes should be merged to master/main after they are streamlined and tested. Only then the package main branch is tagged a milestone or release.

I believe, however, it would be inappropriate to force this scheme to developers.

By the way, maybe the branching issue should be fixed in a more geneal way, via our repo tool. So that we can switch branches without manually cloning the package elsewhere or disrupting the upphub scheme.

Best regards,
Oblivion

Subject: Re: UppHub - new package registration
Posted by mirek on Mon, 15 Feb 2021 14:36:08 GMT
View Forum Message <> Reply to Message

I feel like the whole PR issue for the list is completely missing the point.

The list maintainer should actually test the module before adding it to the list. The list information

is the least important part...

Subject: Re: UppHub - new package registration
Posted by mirek on Wed, 17 Feb 2021 08:07:20 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 02 February 2021 15:58
That includes my build automation system.

BTW, I was thinking: Would it be in perspective possible to commit (automatically) your results to UppHub as some Json in some form? Probably only if changed to keep number of commits low.

It is just an idea, but I think this could definitely allow us to do some thing automatically, like OS support, detecting broken packages etc... UppHub code could refer directly to your files.

Mirek

Subject: Re: UppHub - new package registration
Posted by Novo on Wed, 17 Feb 2021 22:36:49 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 17 February 2021 03:07Novo wrote on Tue, 02 February 2021 15:58
That includes my build automation system.

BTW, I was thinking: Would it be in perspective possible to commit (automatically) your results to UppHub as some Json in some form? Probably only if changed to keep number of commits low.

It is just an idea, but I think this could definitely allow us to do some thing automatically, like OS support, detecting broken packages etc... UppHub code could refer directly to your files.

Mirek

My build automation system is based on Buildbot (for historical reasons).
Buildbot is used by almost all major IT companies. Example: LLVM/Clang.
Buildbot has a concept of "reporters". Most useful reporters in case of U++ are email, Telegram, and GitHubStatusPush.
I've already set up a Telegram bot, which can report results of U++ builds. It is currently down because U++ builds and my own builds share same setup. I need to separate them first.
I guess you do not want to get spammed by emails from Buildbot.
Another useful thing is GitHubStatusPush, which is using GitHub Status API to report build result directly to GitHub.
I do not want to develop custom reporters to produce JSON files or something else. At least right

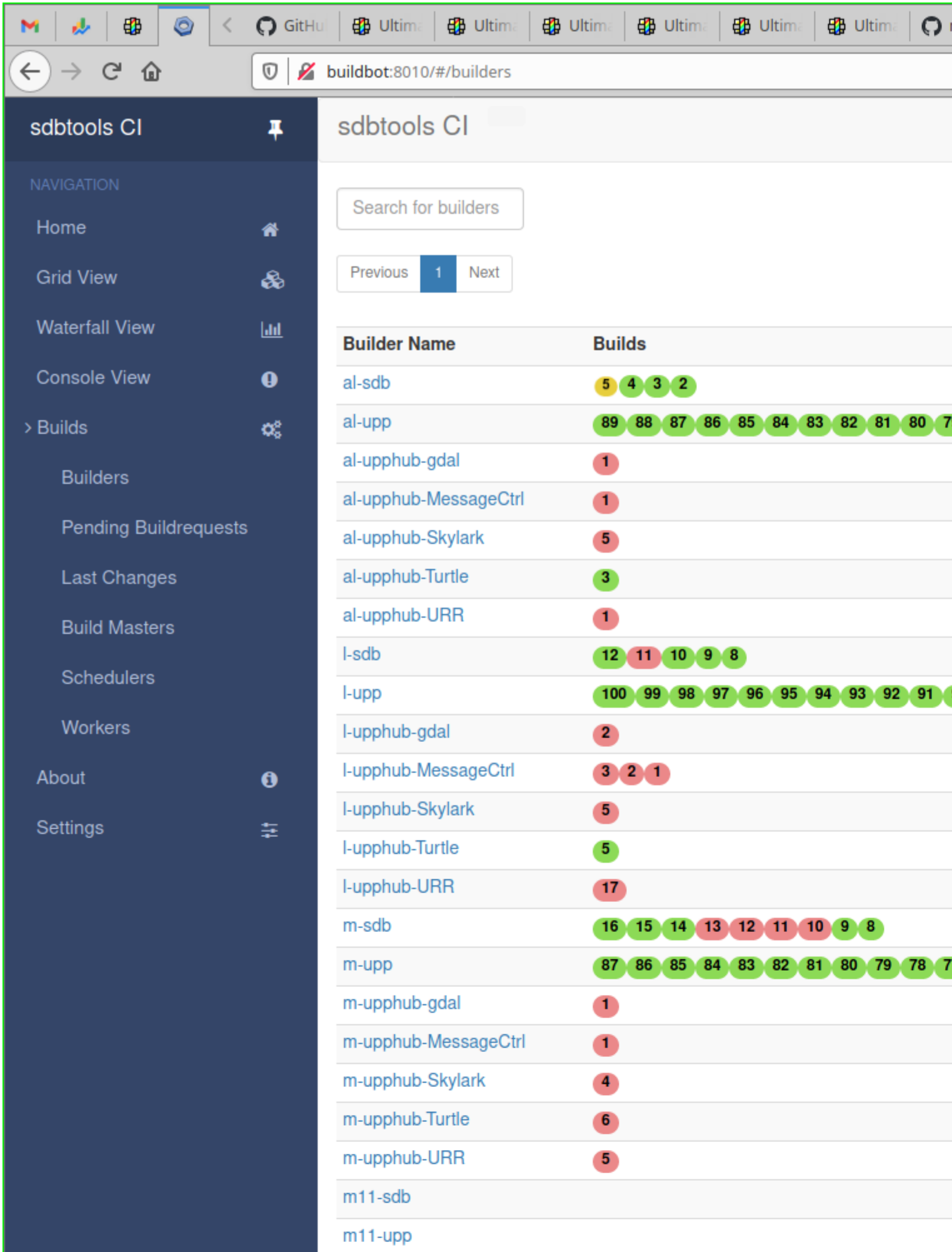
now.
My current goal is to make osxcross work to get rid of an always running MacOS VM, which is using 8GB of RAM and keeps one CPU busy all the time.
And Buildbot requires explicit branch name ...

Subject: Re: UppHub - new package registration
Posted by Novo on Wed, 17 Feb 2021 23:25:51 GMT
View Forum Message <> Reply to Message

This is how current situation with builds looks like at the moment.
"l" stands for Linux (Ubuntu)
"al" stands for Alpine Linux.
"m" - MacOS 10.13
"m11" - MacOS 11.1
"mingw" - mingw-clang (cross-compiler for Windows)

File Attachments

1) Screenshot_2021-02-17_17-55-48.png, downloaded 657 times


buildbot:8010/#/builders
sdbtools CI
NAVIGATION
Home
Grid View
Waterfall View
Console View
Builds
Builders
Pending Buildrequests
Last Changes
Build Masters
Schedulers
Workers
About
Settings
sdbtools CI
Search for builders
Previous
1
Next
Builder Name
Builds
al-sdb 5 4 3 2
al-upp 89 88 87 86 85 84 83 82 81 80
al-upphub-gdal 1
al-upphub-MessageCtrl 1
al-upphub-Skylark 5
al-upphub-Turtle 3
al-upphub-URR 1
l-sdb 12 11 10 9 8
l-upp 100 99 98 97 96 95 94 93 92 91
l-upphub-gdal 2
l-upphub-MessageCtrl 3 2 1
l-upphub-Skylark 5
l-upphub-Turtle 5
l-upphub-URR 17
m-sdb 16 15 14 13 12 11 10 9 8
m-upp 87 86 85 84 83 82 81 80 79 78
m-upphub-gdal 1
m-upphub-MessageCtrl 1
m-upphub-Skylark 4
m-upphub-Turtle 6
m-upphub-URR 5
m11-sdb
m11-upp

Subject: Re: UppHub - new package registration
Posted by mirek on Fri, 19 Feb 2021 10:32:38 GMT
View Forum Message <> Reply to Message

Problem with branch is now fixed.

I have improved hidden url settings:

These are now activated by F12 or by activating verbose mode. They allow to override the url of upphub list, even allowing to set it to file. In practice, it means you can clone UppHub repo, test the list changes and then push tested list.

Subject: Re: UppHub - new package registration
Posted by Oblivion on Fri, 19 Feb 2021 11:45:48 GMT
View Forum Message <> Reply to Message

Quote:Problem with branch is now fixed.

I have improved hidden url settings:

These are now activated by F12 or by activating verbose mode. They allow to override the url of upphub list, even allowing to set it to file. In practice, it means you can clone UppHub repo, test the list changes and then push tested list.

OK, thanks. This wiil come really handy.

Best regards,
Oblivion

Subject: Re: UppHub - new package registration
Posted by Klugier on Fri, 19 Feb 2021 17:57:30 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I think "Settings" should be visible all the time in UppHub. I think it is nothing wrong to have "alternative" nests.json if you are know what you are doing. This is also fine for security reasons. If somebody wants to host all remotes packages on internal servers and do not want to use the public ones, then fine.

I think we should make it visible for all and add additional prompt that changing the value might break UppHub. Please noticed that you can accidental press F12 and modify and forget that something wrong happend with UppHub. I would like to avoid such situation.

The button "Reset to defaults" should also do the job.

Klugier

Subject: Re: UppHub - new package registration
Posted by mirek on Thu, 04 Mar 2021 14:19:01 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Tue, 02 February 2021 22:59Hello Mirek,

I have noticed an issues with the license and readme files: Since these files reside in the root directory, TheIDE is unable to read them (in the license information window or via the package files section.)

Best regards,
Oblivion

Found a way how to solve this nicely:

https://www.ultimatepp.org/forums/index.php?t=msg&goto=5 6395&#msg_56395

Mirek