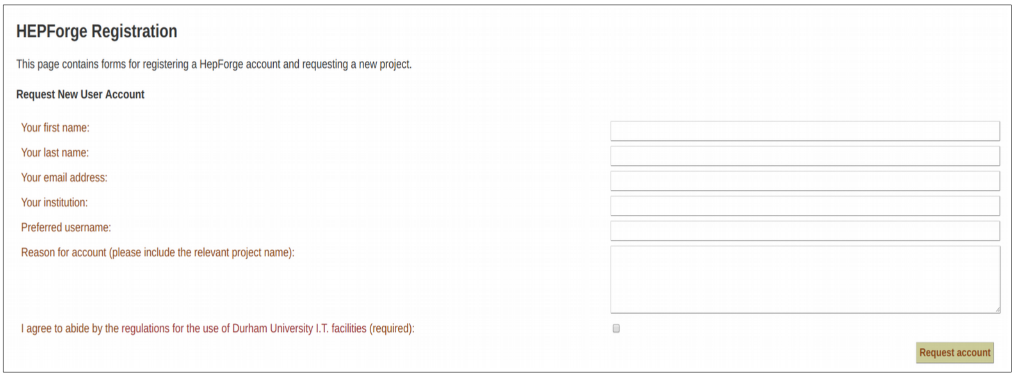
General Users

If you’re looking to create a new project please refer to section 5.0 on page, this guide is aimed at getting you started as a new Hepforge user looking to join an existing project and/or create a Hepfroge repository. This section covers how to join an existing project and getting started with the Hepforge system. This includes all new features and utilities available to you.

1.0 Creating a New User Account

If a new Hepforge account is required please proceed to the Hepforge Registration page (Figure 1.0) at: https://www.hepforge.org/register
Simply fill in the form and one of the administration team will process your account as soon as possible.



HEPForge Registration

This page contains forms for registering a HepForge account and requesting a new project.

Request New User Account

Your first name:
Your last name:
Your email address:
Your institution:
Preferred username:
Reason for account (please include the relevant project name):

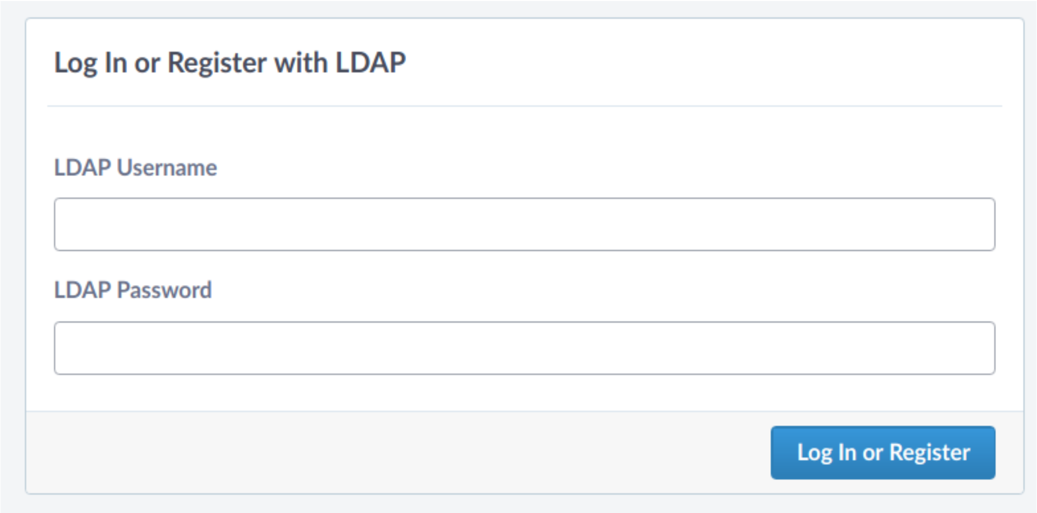
I agree to abide by the regulations for the use of Durham University I.T. facilities (required): □

Fig 1.0 – Hepforge Account Registration

1.1 New User - Logging in

To login to your new account proceed to the following address: https://phab.hepforge.org/

You will be greeted with the login screen shown in figure 1.1, enter your newly generated account details and hit the login button.



Log In or Register with LDAP

LDAP Username

LDAP Password

Fig 1.1 – Hepforge Login

This will take you to your newly generated user dashboard, from here you can access all available system features. For more information on these features see section 3.0.

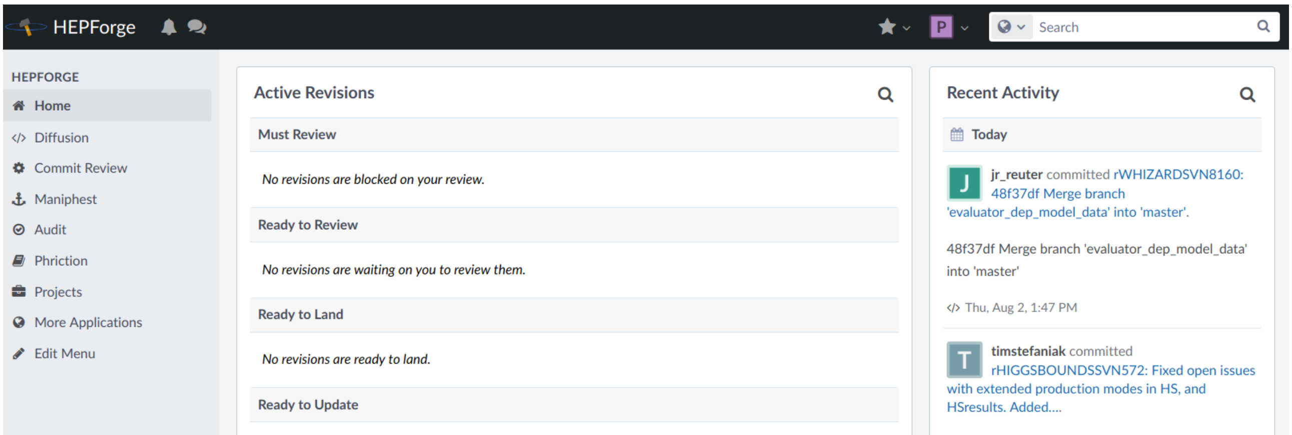
2.0 Joining an Existing Project

If you wish to join an existing development project and already have a Hepforge account please liaise with project administrator for the project you wish to join. The administrator will review your request and add your account to the project.

For a full list of the projects available please visit: https://hepforge.org/projects

3.0 Hepforge Navigation and Features

Upon successful login you will be forwarded to your new Hepforge dashboard as shown in figure 3.0. Your dashboard is where you are able to view and modify any of your currently active projects, view recent activity, action any assigned tasks and view any active repositories. Navigation is accomplished through the navigation bar located on the left hand portion of fig 3.0.



No revisions are blocked on your review.

No revisions are waiting on you to review them.

No revisions are ready to land.

jr_reuter committed rWHIZARDSVN8160: 48f37df Merge branch 'evaluator_dep_model_data' into 'master'.

48f37df Merge branch 'evaluator_dep_model_data' into 'master'

Thu, Aug 2, 1:47 PM

timstefaniak committed rHIGGSBOUNDSSVN572: Fixed open issues with extended production modes in HS, and HSresults. Added....

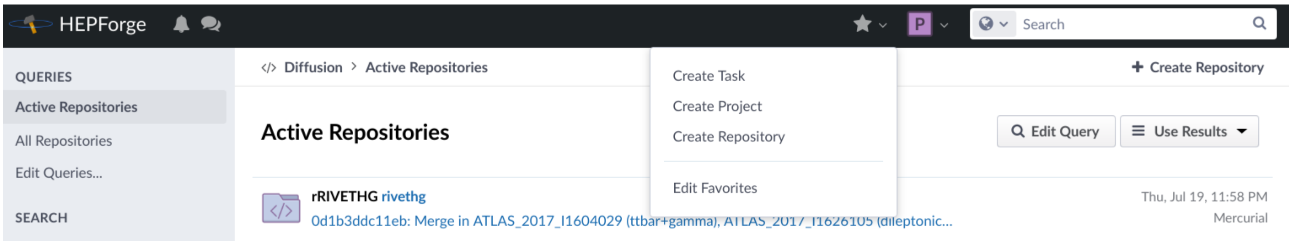
Fig 3.0 – *Hepforge Dashboard*

3.1.0 Diffusion

Diffusion is used to create and manage your repositories, the following looks at the creation of a new repository and how it can be managed.

3.1.1 Repository Creation

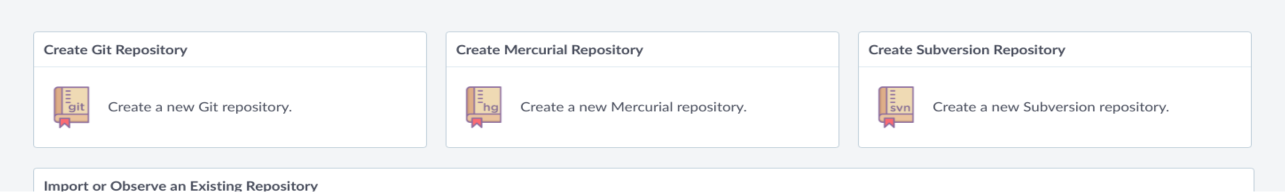
To create a new Repository select the Diffusion option from the Hepforge dashboard, doing so will display a list of all the currently active repositories. In the top right hand section is the "+Create Repository" button(Figure 3.1). Pressing this will take you to the repository selection page (To shortcut this select the star icon from the top navigation bar and select "Create Repository").



rRIVETHG rivethg
0d1b3ddc11eb: Merge in ATLAS_2017_I1604029 (ttbar+gamma), ATLAS_2017_I1626105 (dileptonic...
Thu, Jul 19, 11:58 PM
Mercurial

Fig 3.1 – Active Repositories

From here you have three choices of version control software to use as shown in figure 3.2. When you have chosen one that meets you're requirements click on its tab and you will be forwarded to the repository creation details page.



Create Git Repository

Create a new Git repository.

Create Mercurial Repository

Create a new Mercurial repository.

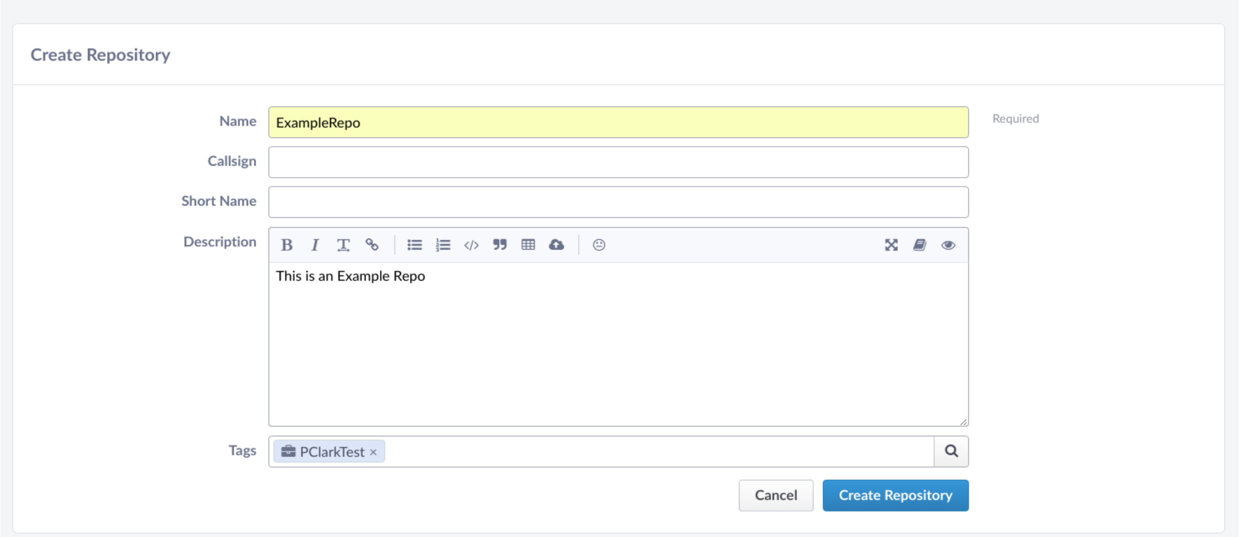
Create Subversion Repository

Create a new Subversion repository.

Import or Observe an Existing Repository

Fig 3.2 – Repository Creation

Once on the Respoitory Creation Page firstly give your new repository a name, description and anything else you want to add . Once you’re happy with the details enter simply hit the Create Reposutiry button and your new Repo will be automaticly generated (Figure 3.3).



Create Repository

Name: ExampleRepo (Required)

Callsign:

Short Name:

Description: This is an Example Repo

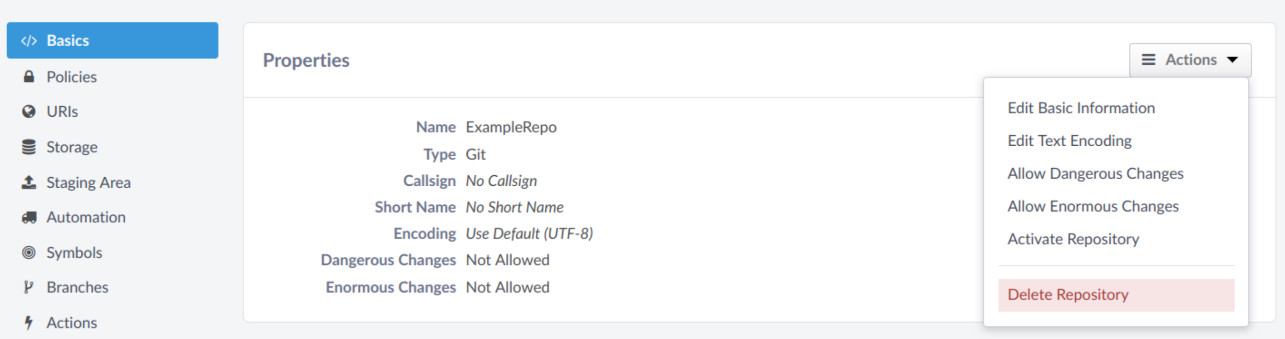
Tags: PClarkTest

Fig 3.3 – Repository Details Page

3.2 Repo Activation and Policies

Your new Repository has now been created but is not yet active on the system. To activate the repository from you Repository page action the following steps:

- Select Actions from your Repository Page (Use the search bar to locate your repository if required)
- From the drop down-list select “Manage Repository”
- Select “basics” from the left hand menu
- Locate and press the Actions button
- From the drop-down list select “Activate your Repository” as shown in Figure 3.4.



Properties

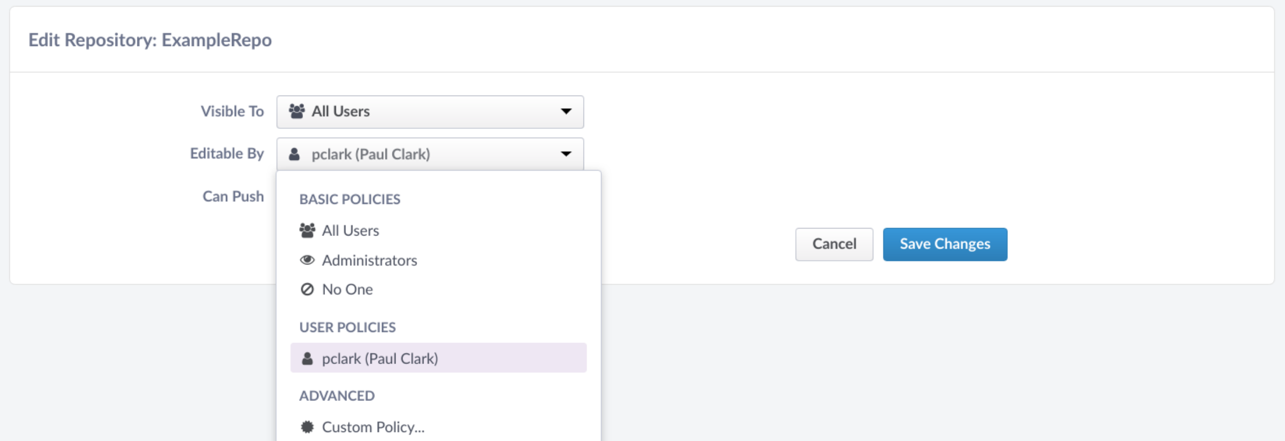
Name	ExampleRepo
Type	Git
Callsign	*No Callsign*
Short Name	*No Short Name*
Encoding	*Use Default (UTF-8)*
Dangerous Changes	Not Allowed
Enormous Changes	Not Allowed

Fig 3.4 – Repository Activation

This will activate your repository so it can be modified and managed. However before activating your repository it is advised that the default access policies be changed. Upon creation your repository is set by default to be editable by all users, for security purposes it is recommended this be changed. To do so select the Policies option from the Repository menu bar (Figure 3.4), here you are shown the following three options to change:

- Visible To – Default: All Users
- Editable By – Default: All Users
- Can Push – Default: All Users

By default repository visibility is set to “All Users”, this allows any other users to view and clone your repository. If you wish to keep your repository private or instead viewable by the public tthis can be changed here by modifying the the “Visible To” drop down box as shown in figure 3.5. The same goes for who you wish to be capable of editing the repository in Hepforge and who can push code to the repository. Change these settings until you are happy with the result, once done remember to activate the repository as show in figure 3.4. Note that these settings can be modified at any time and only apply to this newly created repository.



Edit Repository: ExampleRepo

Visible To: All Users

Editable By: pclark (Paul Clark)

Fig 3.5 – Repository Policies

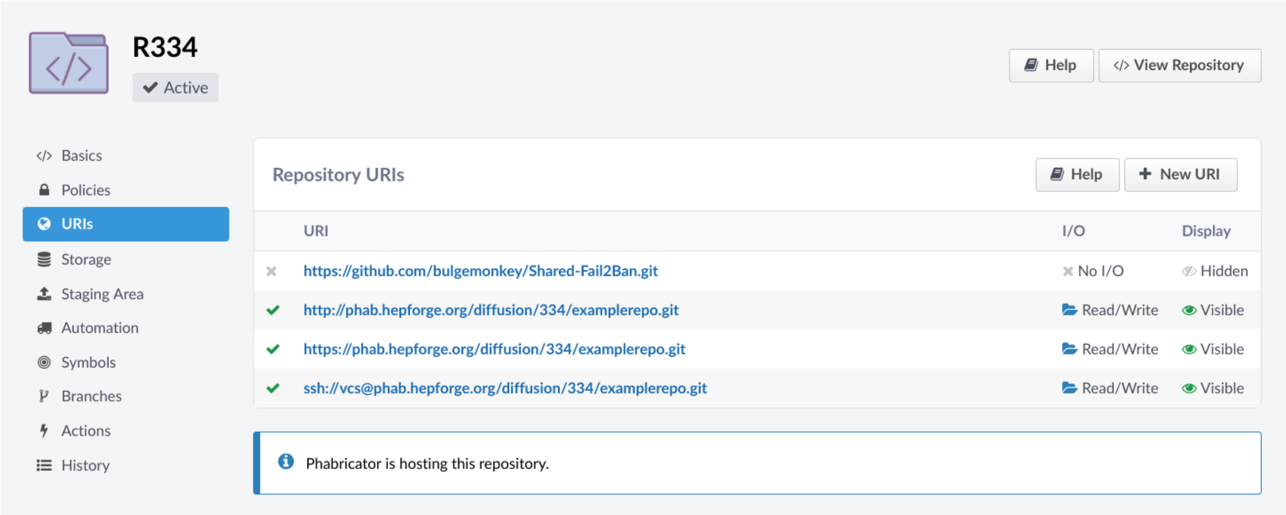
Once you’re happy with the polocies and your Repository is activate it is now time to start uploading code. The following section looks at how to import code from a pre-existing repository, import a fresh code structure plus how to push and pull code form your repository.

4.0 Importing, Pushing code and Cloning your repository

The following section looks at importing code to your new repository from a preexisting reposito

4.1 Importing an existing Repository

To automatically import an external repository from git or svn create a new repository on hepforge but do not set it to active, enter the repository management area (Fig 3.4) and locate the URIs(Universal Resource Identifier) section in the left hand navigation menu. This will display all current URIs associated with this repository (Figure 4.1). To import our external git/svn resource we first need to change the settings of our existing URIs I/O from Read/Write to Read. To do so first select the URI you would like to change by clicking its URI link under the URI Column .



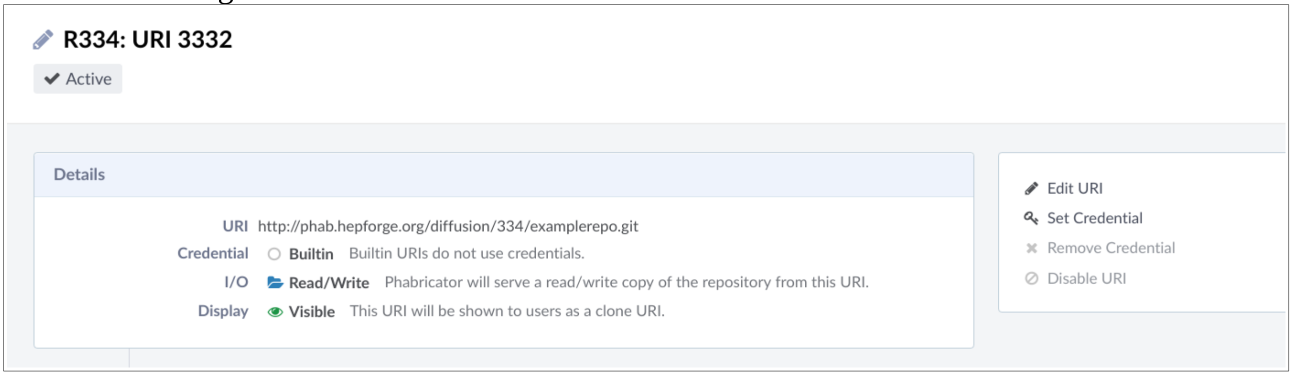
Repository URIs

	URI	I/O	Display
✖	https://github.com/bulgemonkey/Shared-Fail2Ban.git	✖ No I/O	Hidden
✓	http://phab.hepforge.org/diffusion/334/examplerepo.git	Read/Write	Visible
✓	https://phab.hepforge.org/diffusion/334/examplerepo.git	Read/Write	Visible
✓	ssh://vcs@phab.hepforge.org/diffusion/334/examplerepo.git	Read/Write	Visible

Phabricator is hosting this repository.

Fig 4.1 – URI Management Page

This will take you to the repositories URI details, here you can view various information associated with the selected URI along with its current status (Figure 4.2). To make changes click the “Edit URI” from the right hand menu.

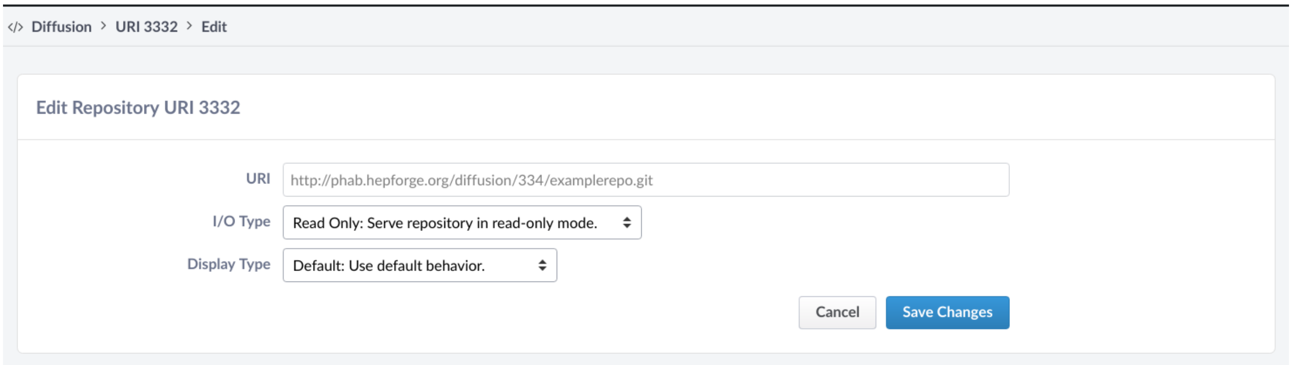


Details

URI	http://phab.hepforge.org/diffusion/334/examplerepo.git
Credential	○ Builtin Builtin URIs do not use credentials.
I/O	Read/Write Phabricator will serve a read/write copy of the repository from this URI.
Display	Visible This URI will be shown to users as a clone URI.

Fig 4.2 – URI Details

Set all pre-existing URI I/O types to “Read Only” before attempting to import an external resource (Figure 4.3). Attempting an import without doing so will result in failure.

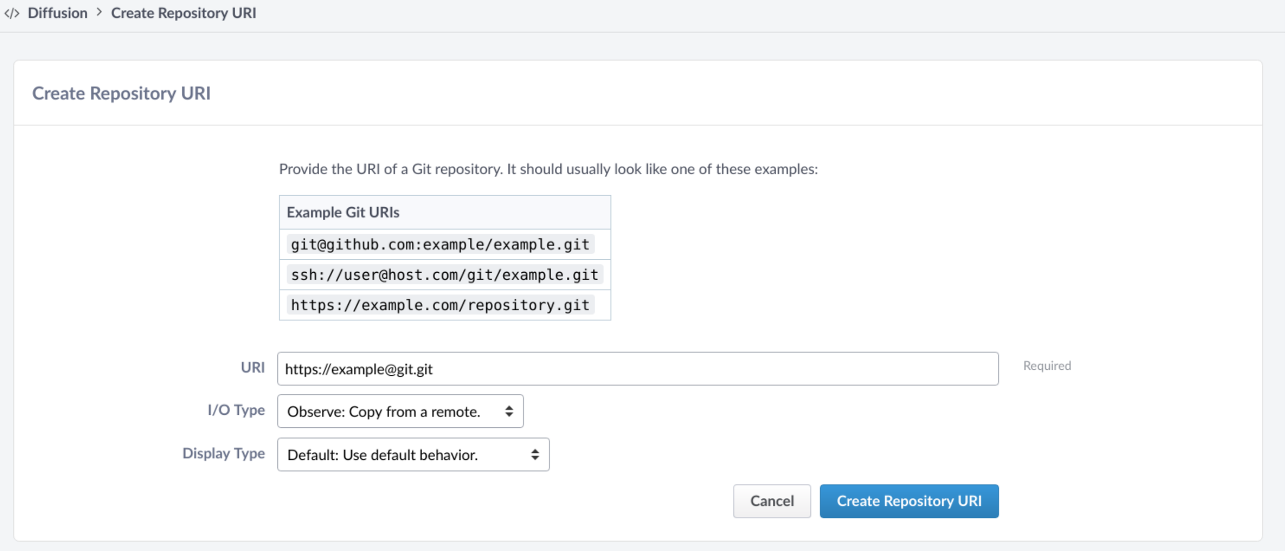


Edit Repository URI 3332

URI	http://phab.hepforge.org/diffusion/334/examplerepo.git
I/O Type	Read Only: Serve repository in read-only mode.
Display Type	Default: Use default behavior.

Fig 4.3 – Edit URI

Once all existing URIs are set to read only it is now time to import your external resources, firstly return to your repositories URI menu and hit the “ + New URI” button. This will allow you to create a new URI that we will use to import the external repository.



Provide the URI of a Git repository. It should usually look like one of these examples:

Example Git URIs
`git@github.com:example/example.git`
`ssh://user@host.com/git/example.git`
`https://example.com/repository.git`

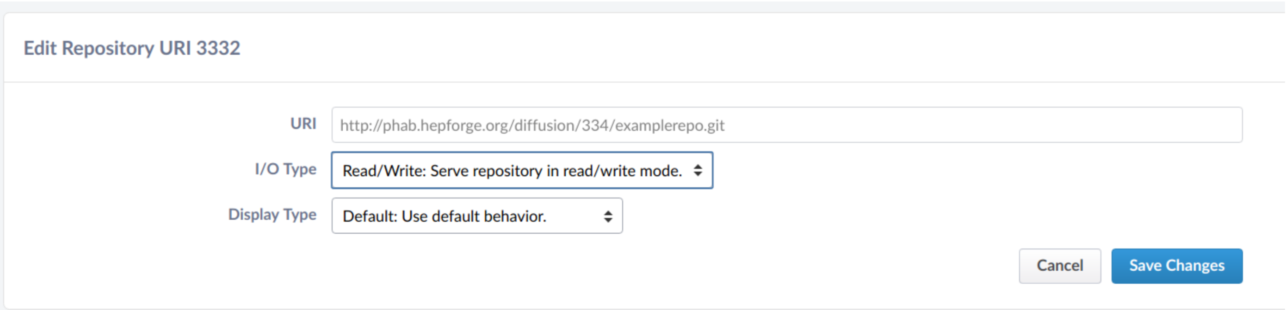
URI: https://example@git.git (Required)

I/O Type: Observe: Copy from a remote.

Display Type: Default: Use default behavior.

Fig4.4 – Creating a URI

From here add the link to your external resource into the URI section, some examples are given to help you. Change the I/O Type to “Observe” as shown in figure 4.4, the observe mode will watch the external resource for changes and import them directly to the Hepforge system. You may also change the display type if required. After you’re done hit the “Create Repository URI” button, this will create the new URI with the information provided. Once done you must now set your new repository to active (See Section 3.2), this will bring the repository up and the external resource will be imported into the Hepforge system.



URI: http://phab.hepforge.org/diffusion/334/examplerepo.git

I/O Type: Read/Write: Serve repository in read/write mode.

Display Type: Default: Use default behavior.

Fig 4.5 – URI Read/Write

Once the external repository has been imported you must change the URI pointing at the external resource from “Observe” to “No I/O” and change all other URI’s back to “Read/Write”. This is done by returning to the URI menu and selecting each URI and making the required changes. An example of this is show in figure 4.5.

Your external repository is now imported into the hepforge system, the following section looks at how this repository can be used to push/pull code and many any changes required.

SVN USERS!!!

Please remove the :22 part of the URI displayed i.e.

svn+ssh://vcs@phab.hepforge.org:22/source/testsvn

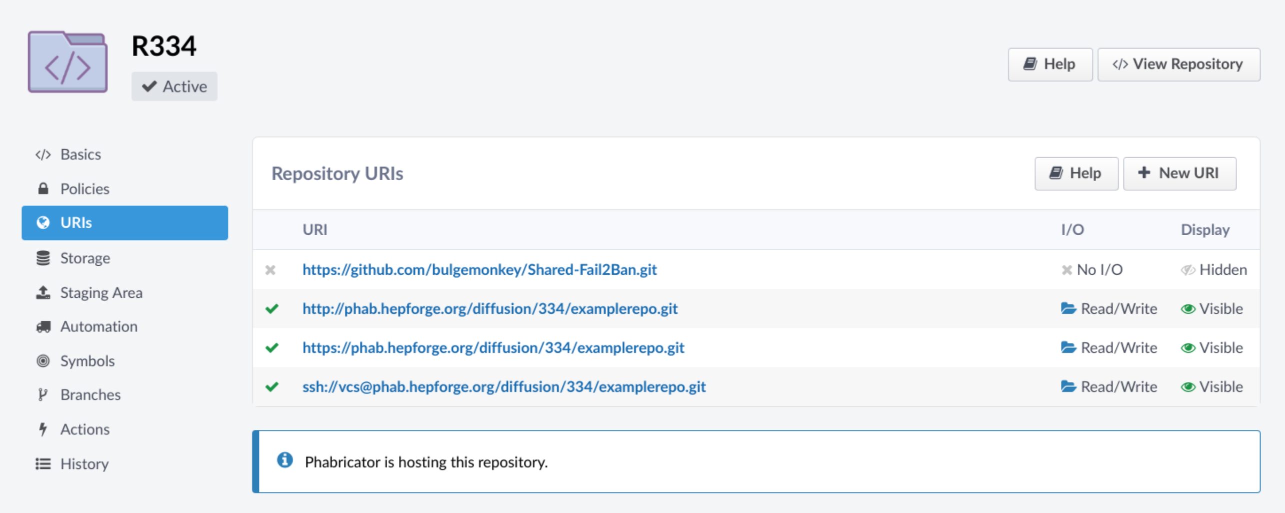
to

svn+ssh://vcs@phab.hepforge.org/source/testsvn

4.2 Setting up your Repository for editing

In the previous section we looked at how to import from an existing respoitory. In this section we look at how to push and pull to your new/current repositry on the Hepforge system.

To push code to your repository you can use the URI links located in your Repository URI properties menu to both pull and push code. To make use these you will need to use the Version Control System Toolset such as git, mercurial or svn depending on the type of repository you originally created or project you're a member of. Figure 4.6 shows the previous example from section 4.1, the URI links shown that are set to Read/Write can be used to help push and pull code using the Git toolset.



	URI	I/O	Display
✕	https://github.com/bulgemonkey/Shared-Fail2Ban.git	✕ No I/O	Hidden
✓	http://phab.hepforge.org/diffusion/334/examplerepo.git	Read/Write	Visible
✓	https://phab.hepforge.org/diffusion/334/examplerepo.git	Read/Write	Visible
✓	ssh://vcs@phab.hepforge.org/diffusion/334/examplerepo.git	Read/Write	Visible

Phabricator is hosting this repository.

Fig 4.6 – URI Links

If you plan on pushing using HTTPS you must set a Version Control Password, this is to help enforce security and applies to all repositories available over HTTP.

To set the password first select the Settings option from your user tab located across the top navigation bar, this is show in figure 4.7.

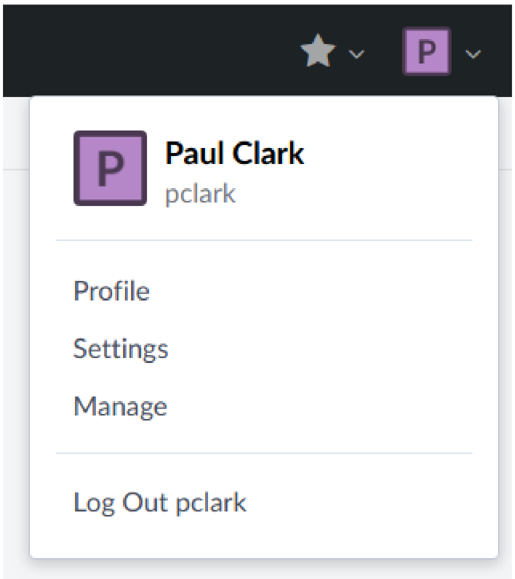
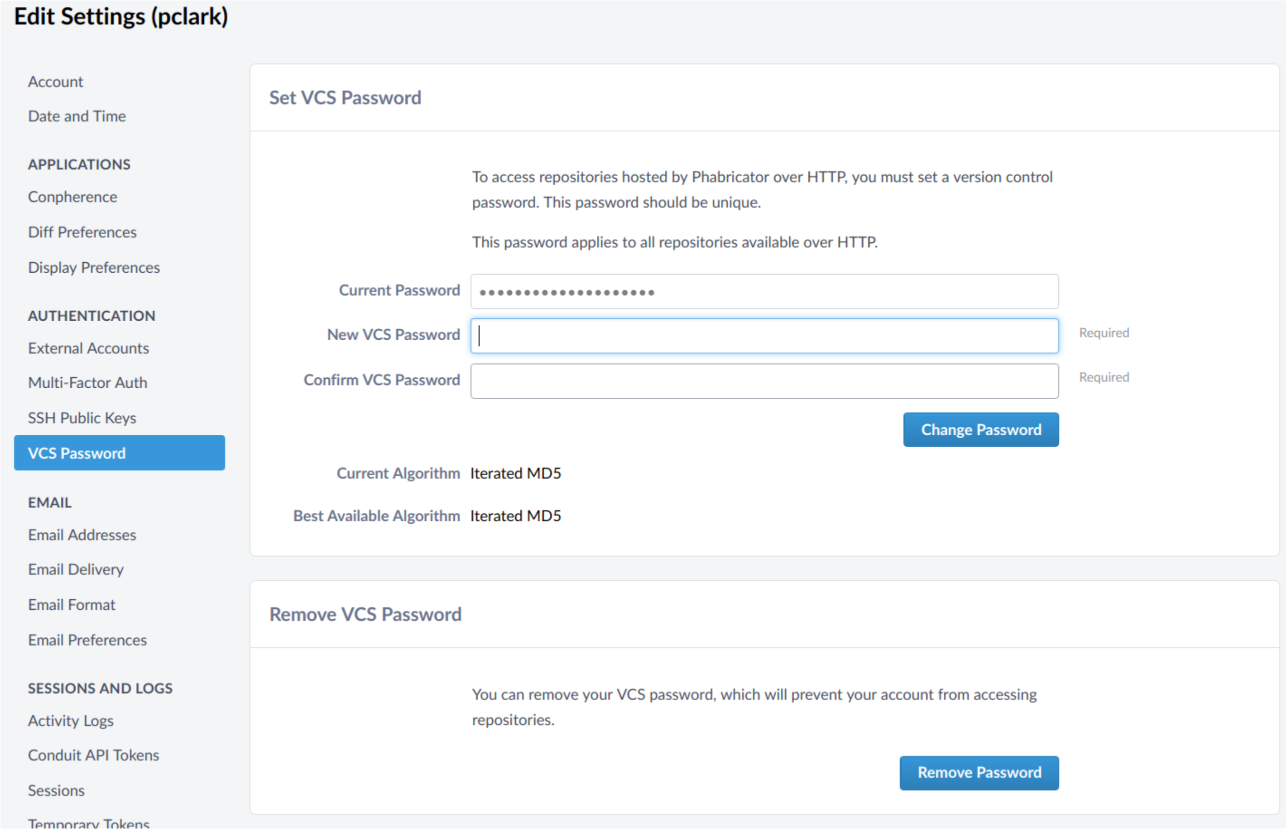


Fig 4.7 - Settings

From the menu that appears (See Figure 4.8) select “VCS password” from the left hand navigation bar and enter your new VCS password and press the Change Password button. If wanting to remove an older previously used password simply click the “Remove Password” button.



Edit Settings (pclark)

Set VCS Password

To access repositories hosted by Phabricator over HTTP, you must set a version control password. This password should be unique.

This password applies to all repositories available over HTTP.

Current Password ••••••••••••••••••••

New VCS Password | Required

Confirm VCS Password Required

Current Algorithm Iterated MD5

Best Available Algorithm Iterated MD5

Remove VCS Password

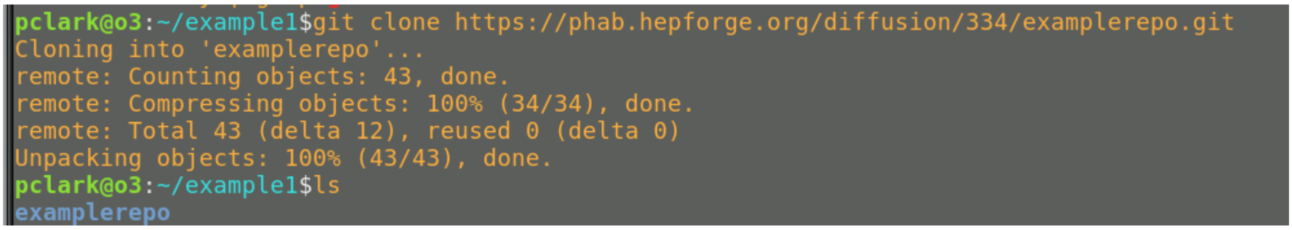
You can remove your VCS password, which will prevent your account from accessing repositories.

Fig 4.8 – Setting your VCS Password

With the VCS Password set you are now able to make changes to your Hepforge Repository using the HTTP URIs.

4.3 Cloning, pushing and pulling from your Repository

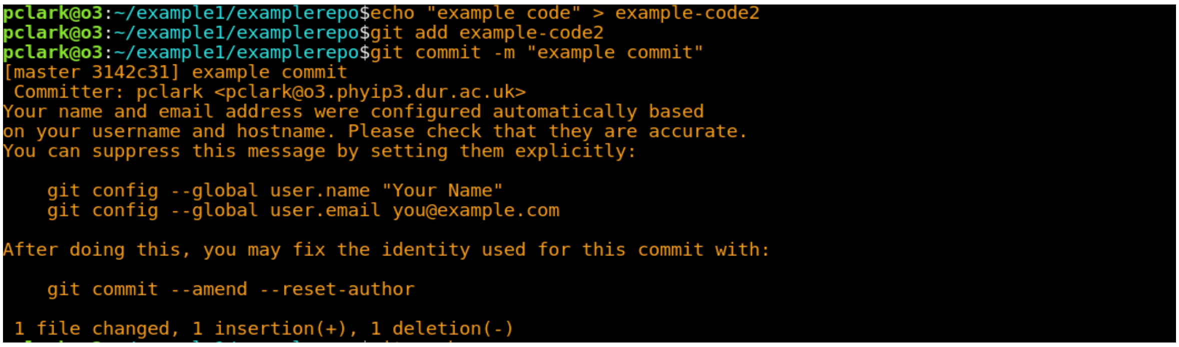
Cloning a copy of your repository can be done using a number of different methods and tools. The following example shows the examplerepo being cloned from the Hepforge system using the git toolset (Figure 4.9). If you have chosen a mercurial or svn repository please make use of those tool sets for cloning the repository to your local/development system.



```
pclark@o3:~/example1$git clone https://phab.hepforge.org/diffusion/334/examplerepo.git
Cloning into 'examplerepo'...
remote: Counting objects: 43, done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 43 (delta 12), reused 0 (delta 0)
Unpacking objects: 100% (43/43), done.
pclark@o3:~/example1$ls
examplerepo
```

Fig 4.9 – git clone of example

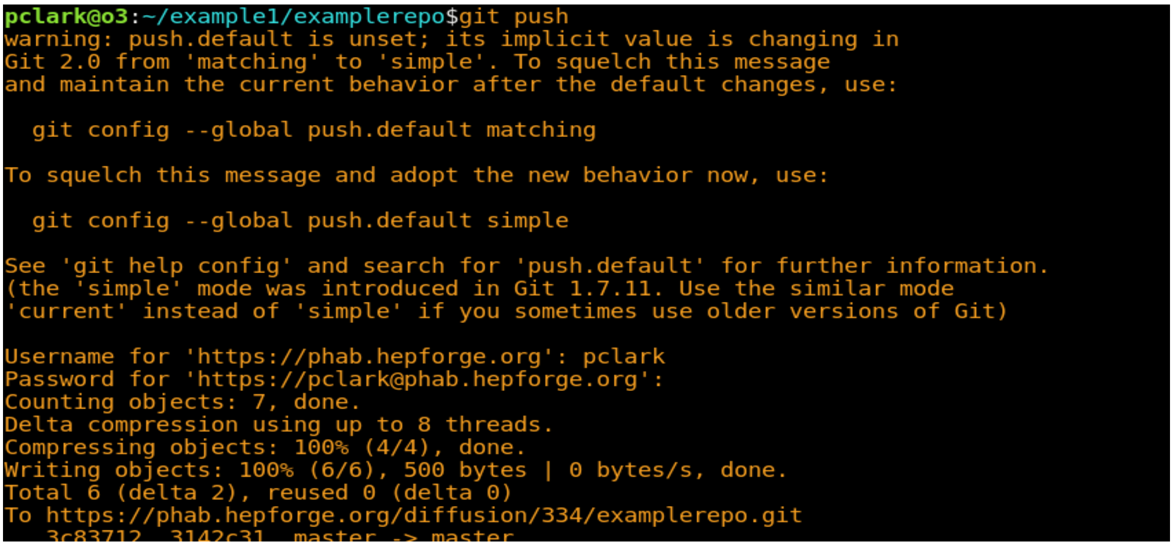
Now that the example repository has been cloned to our local system we can begin pushing code to it. The simplest method is to review the example shown in figures 4.10 and 4.11. Firstly we create an example code file, we add and commit this so that it is ready to be pushed to Hepforge.



```
pclark@o3:~/example1/examplerepo$echo "example code" > example-code2
pclark@o3:~/example1/examplerepo$git add example-code2
pclark@o3:~/example1/examplerepo$git commit -m "example commit"
[master 3142c31] example commit
 Committer: pclark <pclark@o3.phyip3.dur.ac.uk>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+), 1 deletion(-)
```

Fig 4.10 – Example code file

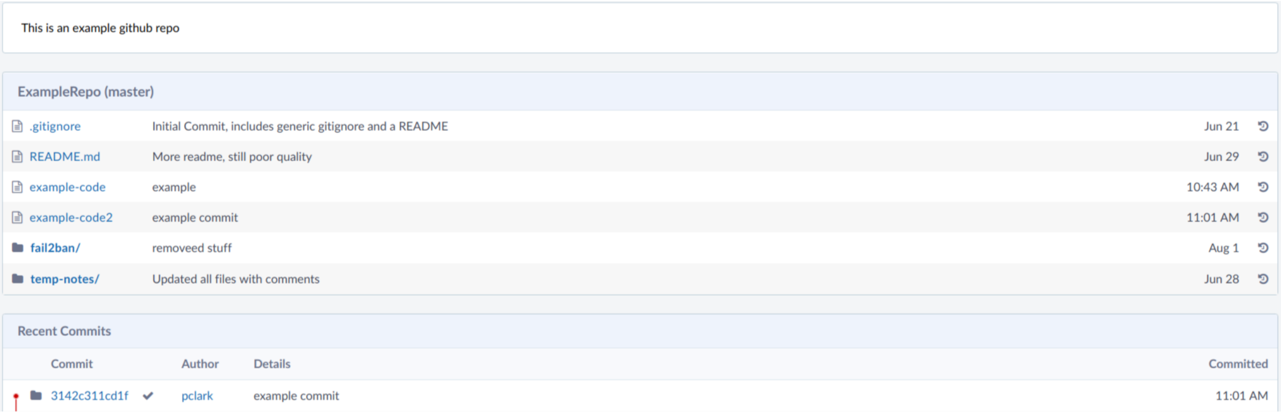
Now we push the changes to hepforge into our example repository, when we run the git push command we are asked to authenticate, this will be include your hepforge username and password, if however you're using the HTTP URIs ensure that you use the VCS password that was set in section 4.2 Figure 4.8. Figure 4.11 shows the the example code file being pushed to example repository.



```
pclark@o3:~/example1/examplerepo$git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://phab.hepforge.org': pclark
Password for 'https://pclark@phab.hepforge.org':
Counting objects: 7, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 500 bytes | 0 bytes/s, done.
Total 6 (delta 2), reused 0 (delta 0)
To https://phab.hepforge.org/diffusion/334/examplerepo.git
   3c83712..3142c31  master -> master
```

Fig 4.11 – Pushing to Git

To verify that the file now exists we can return to our web browser and navigate to your repository on Hepforge. Figure 4.12 verifies that the file example-code2 was sucessfuly pushed to the hepforge system.



This is an example github repo

ExampleRepo (master)

.gitignore	Initial Commit, includes generic gitignore and a README	Jun 21
README.md	More readme, still poor quality	Jun 29
example-code	example	10:43 AM
example-code2	example commit	11:01 AM
fail2ban/	removeed stuff	Aug 1
temp-notes/	Updated all files with comments	Jun 28

Recent Commits

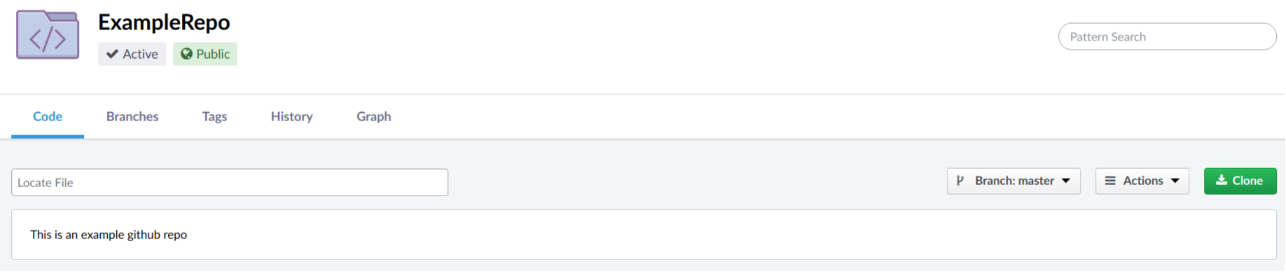
Commit	Author	Details	Committed
3142c311cd1f ✓	pclark	example commit	11:01 AM

Fig 4.12 – Pushed Successfully to Hepforge

In the next section we look at how to associate your new/imported repository with an existing Hepforge project.

4.4 Associating your Repository with a Project

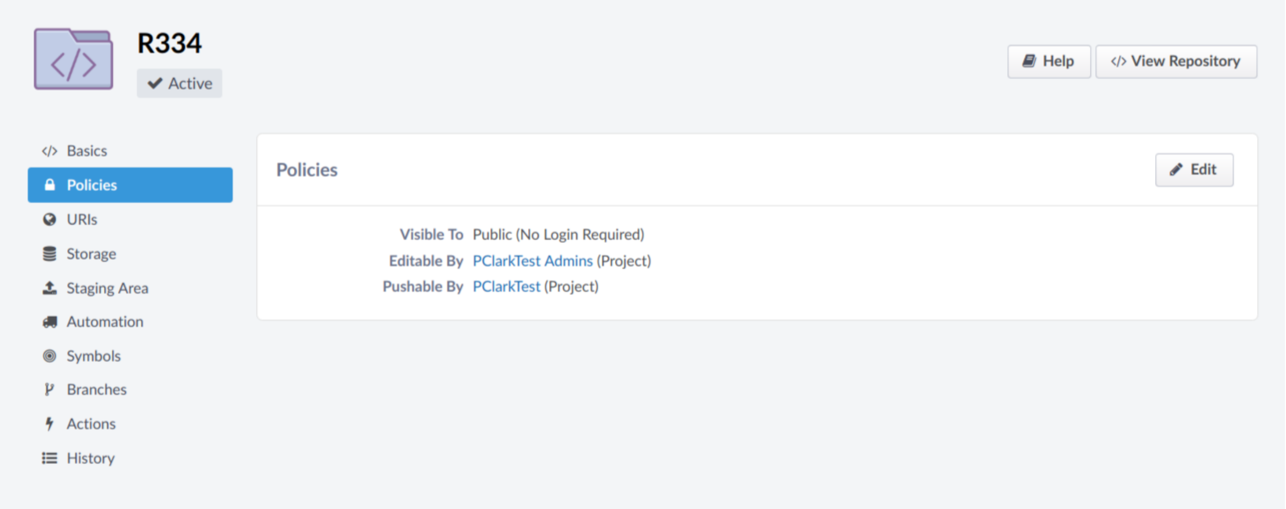
You can associate your hepforge repositories with an a new/existing project to help aid with community based development. Adding your repository to a project is a simple process, firstly navigate to the menu for the repository you wish to associate and press the actions button and select manage repository from the drop down menu (See Figure 4.13).



This is an example github repo

Fig 4.13 – Example Repository

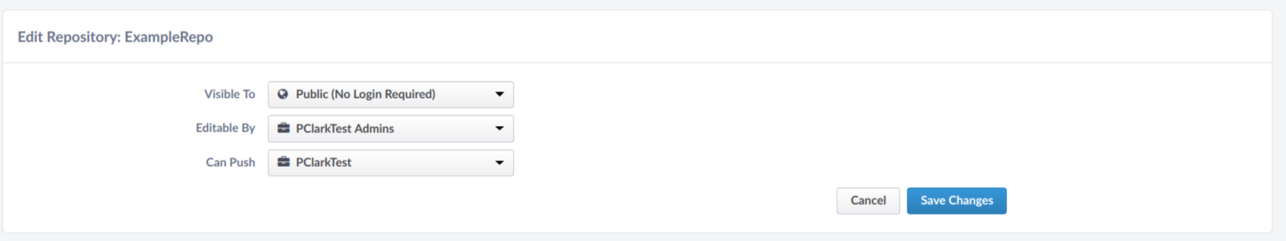
From the left hand menu select Policies(see figure 4.14) and press the Edit button.



Visible To Public (No Login Required)
Editable By PClarkTest Admins (Project)
Pushable By PClarkTest (Project)

Fig 4.14 – Repository Policies

The Edit Repository menu has a number of options, to associate it with a project select who can edit and who can push to the repository. In the following example we will allow the project administrators from the PclarkTest Admins group to edit the example repo and allow general members of the PclarkTest project to push code, these are shown in figure 4.15.



Visible To: Public (No Login Required)
Editable By: PClarkTest Admins
Can Push: PClarkTest

Fig 4.15 – Edit Policies

To edit these select the drop down for each option and select “Other Project” option (Figure 4.16) and enter the name of the project you wish the repository to be associated with (Figure 4.17).

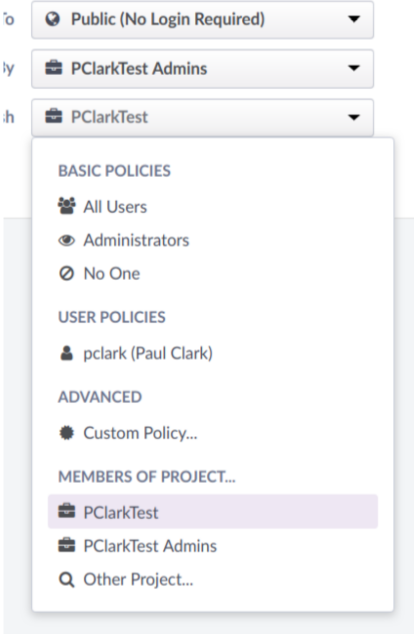


Fig 4.16 – Members of Project

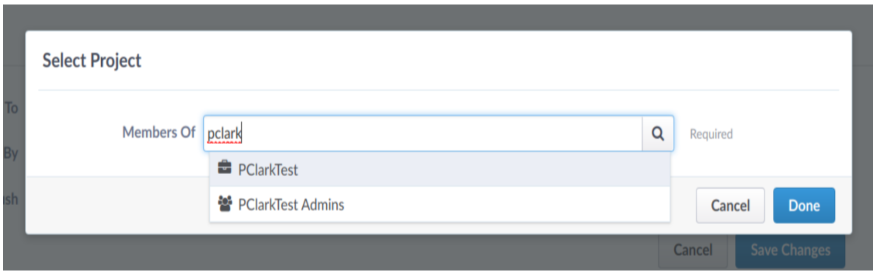


Fig 4.17 – Search for Project

The projects added to the repository policies section now allow members of the selected project to make changes and push to our example repository.

5.0 Project Creation

Creating a project in hepforge is a simple process, to do so please complete the project registration form located at: https://www.hepforge.org/register or contact the Hepforge Administration team at admin@hepforge.org and your request will be processed at the earliest opportunity. Once your Project development space has been created you will be informed via email.

To view your new project login to your Hepforge account at https://phab.hepforge.org and proceed to the Projects section located in the left hand navigation bar. If you require a new user account see section 1.0.

Created: 30th August 2018
Updated: 11th September 2018

Author: Paul Clark
Editors: Adam Boutcher.

www.hepforge.org