# Part 3: Create your First Repository

If you are new to Continua, It is recommended that you first read Part 1: Create your First Project and Part 2: Create your First Configuration. This tutorial assumes that you already have a project and configuration created as outlined in Part 1 and Part 2.

Without repositories, Continua CI would have no source code to build. Continua CI links directly to your source control repositories and executes builds. For more information on repositories, check out the Repository Types.

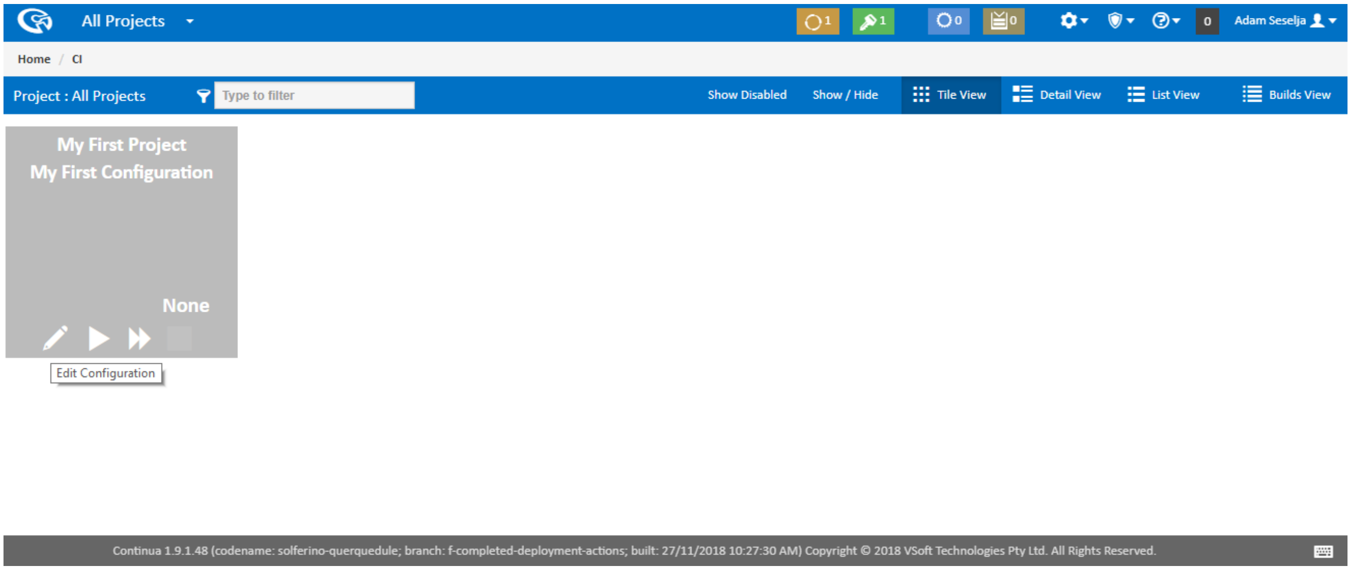
In part 1 and part 2 of this tutorial, we created both our first project and configuration in Continua CI. Following on from this, we are now going to attach a repository to our configuration so that we can get some real data that we can play with!

For this tutorial we are going to use the opensource Fluent Nhiberate GitHub project as our source repository. Fluent NHibernate allows you to write database to object mappings in strongly typed C#, however for this exercise we don't need to worry about what the project actually does. Any project can be used to test out Continua as long as the project can build successfully. We are using Fluent NHibernate for this exercise as it is relatively small and will allow us to get a simple build process up and running.

As Fluent NHibernate uses Git as its Versioning Control Software, you will need to have Git installed on your Continua Server. You can get the latest copy of Git from http://git-scm.com/

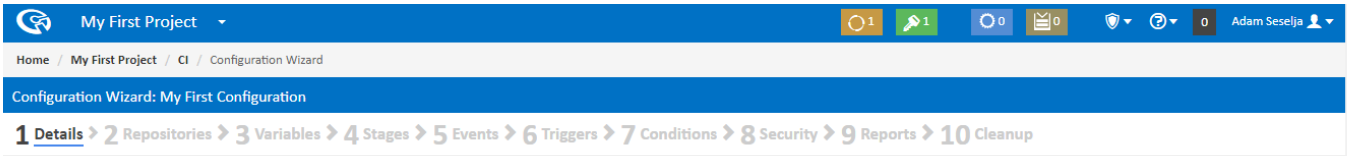
## Adding a Repository to Continua CI

So lets get started. Navigate to the homepage and you should see your project and configuration tile. When you hover over this tile, you will notice that actions appear along the bottom of the tile, as seen below. The action we are interested in is the first action that looks like a pencil. This is the **Edit Configuration** action. Clicking Edit Configuration will take you back to the Configuration Wizard, but this time we will be editing our Configuration rather than creating a new one.



Continua 1.9.1.48 (codename: solferino-querquedule; branch: f-completed-deployment-actions; built: 27/11/2018 10:27:30 AM) Copyright © 2018 VSoft Technologies Pty Ltd. All Rights Reserved.

Along the top of the Configuration Wizard you will see a whole bunch of steps that will add various elements to your Configuration. These steps are a quick way to navigate between the various aspects of a configuration. Once a Configuration has been created, you are free to move between these sections in any order. The order presented is just a representation of the steps needed to get a build up and running.

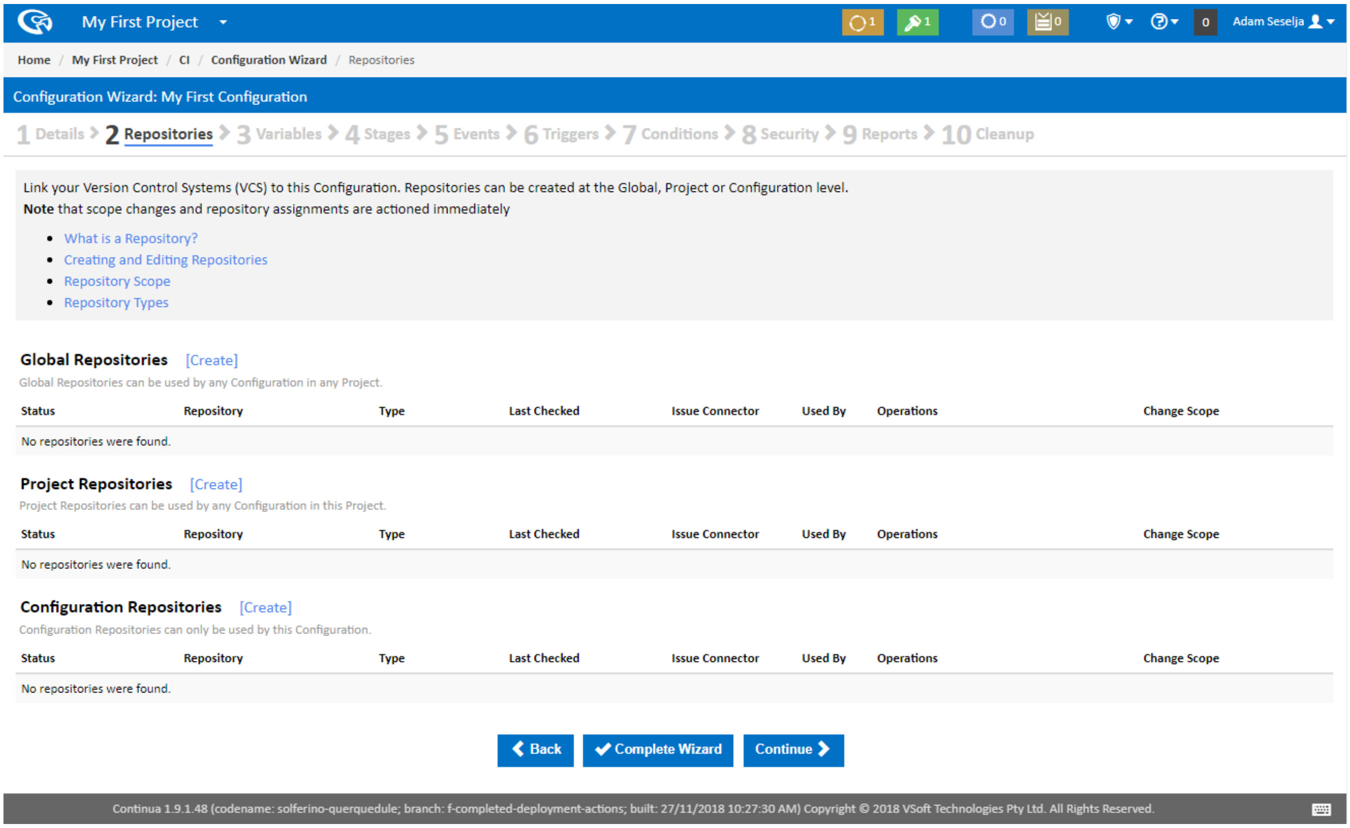


For now, Let's look at the **Repositories** section.

## Repositories Section

The Repositories section of the Configuration Wizard allows you to create and edit any repositories that the current configuration can access. You will notice that on this page there are 3 repository levels which are **Global, Project and Configuration Repositories**. These levels determine how many configurations can access a repository. For more information on this, visit the Global Settings, Projects & Configurations section.

For this tutorial we will be adding our repository as a **Configuration Repository**. This means that only this configuration will know about this repository and no other configurations will be able to access it.


My First Project
1
1
0
0
0
Adam Seselja
Home / My First Project / CI / Configuration Wizard / Repositories
1 Details 2 Repositories 3 Variables 4 Stages 5 Events 6 Triggers 7 Conditions 8 Security 9 Reports 10 Cleanup
Back
Complete Wizard
Continue

Configuration Wizard: My First Configuration

Link your Version Control Systems (VCS) to this Configuration. Repositories can be created at the Global, Project or Configuration level.
**Note** that scope changes and repository assignments are actioned immediately

- What is a Repository?
- Creating and Editing Repositories
- Repository Scope
- Repository Types

**Global Repositories** [Create]

Global Repositories can be used by any Configuration in any Project.

| Status | Repository | Type | Last Checked | Issue Connector | Used By | Operations | Change Scope |
|---|---|---|---|---|---|---|---|
| No repositories were found. | | | | | | | |

**Project Repositories** [Create]

Project Repositories can be used by any Configuration in this Project.

| Status | Repository | Type | Last Checked | Issue Connector | Used By | Operations | Change Scope |
|---|---|---|---|---|---|---|---|
| No repositories were found. | | | | | | | |

**Configuration Repositories** [Create]

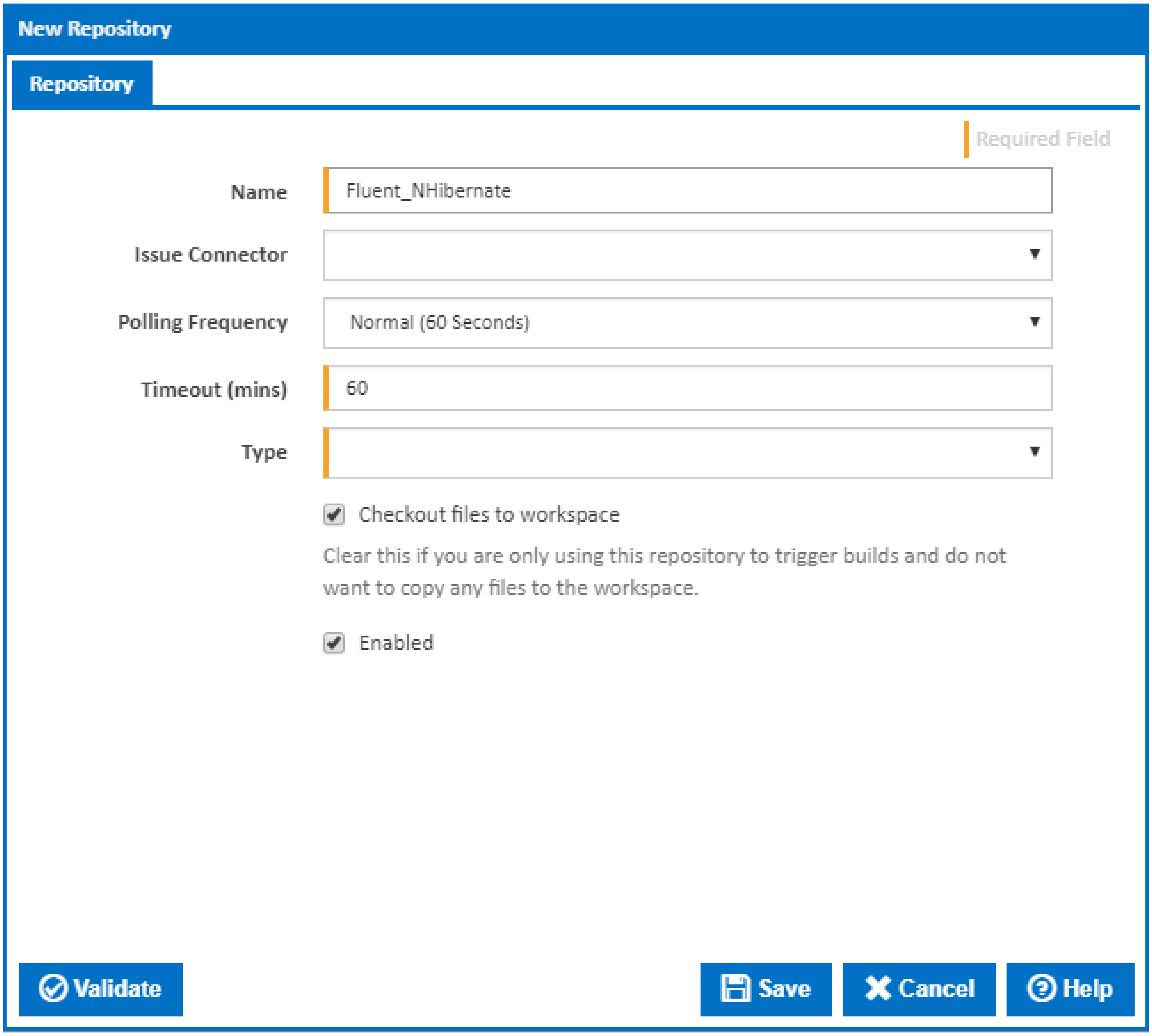
Configuration Repositories can only be used by this Configuration.

| Status | Repository | Type | Last Checked | Issue Connector | Used By | Operations | Change Scope |
|---|---|---|---|---|---|---|---|
| No repositories were found. | | | | | | | |

Continua 1.9.1.48 (codename: solferino-querquedule; branch: f-completed-deployment-actions; built: 27/11/2018 10:27:30 AM) Copyright © 2018 VSoft Technologies Pty Ltd. All Rights Reserved.

## Create Repository Dialog

So lets create our repository by clicking the **Create link next to Configuration Repositories**. This will bring up the **Create Repository dialog** as shown below.



New Repository

Repository

Required Field

Name: Fluent_NHibernate

Issue Connector:

Polling Frequency: Normal (60 Seconds)

Timeout (mins): 60

Type:

☑ Checkout files to workspace

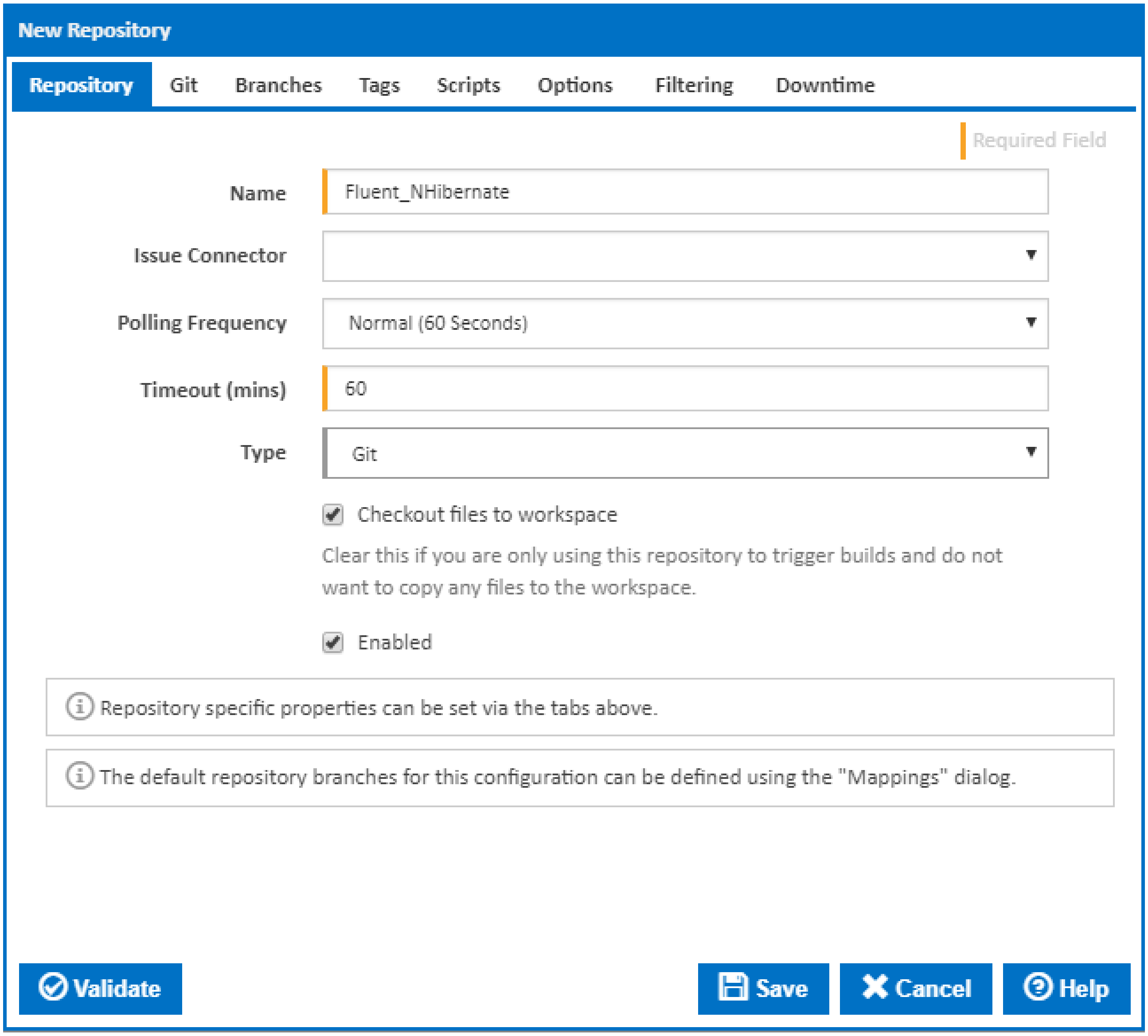
Clear this if you are only using this repository to trigger builds and do not want to copy any files to the workspace.

☑ Enabled

The Create Repository dialog allows you create or edit a repository and apply all settings that you will need to connect to your repository. There are plenty of options that you can include when creating a repository, however these will not be covered in this tutorial. For more information, check out the Repository Types.

As you can see, there is currently not a whole lot of information you can enter regarding your repository. That is because this dialog is designed to adapt to whichever repository type you are creating.

For this tutorial we will be creating a **Git Version Control Repository** so lets select this in the **Type dropdown**. Once Git is selected you will see additional tabs appear at the top of the dialog, as seen below:



Name: Fluent_NHibernate

Issue Connector:

Polling Frequency: Normal (60 Seconds)

Timeout (mins): 60

Type: Git

☑ Checkout files to workspace

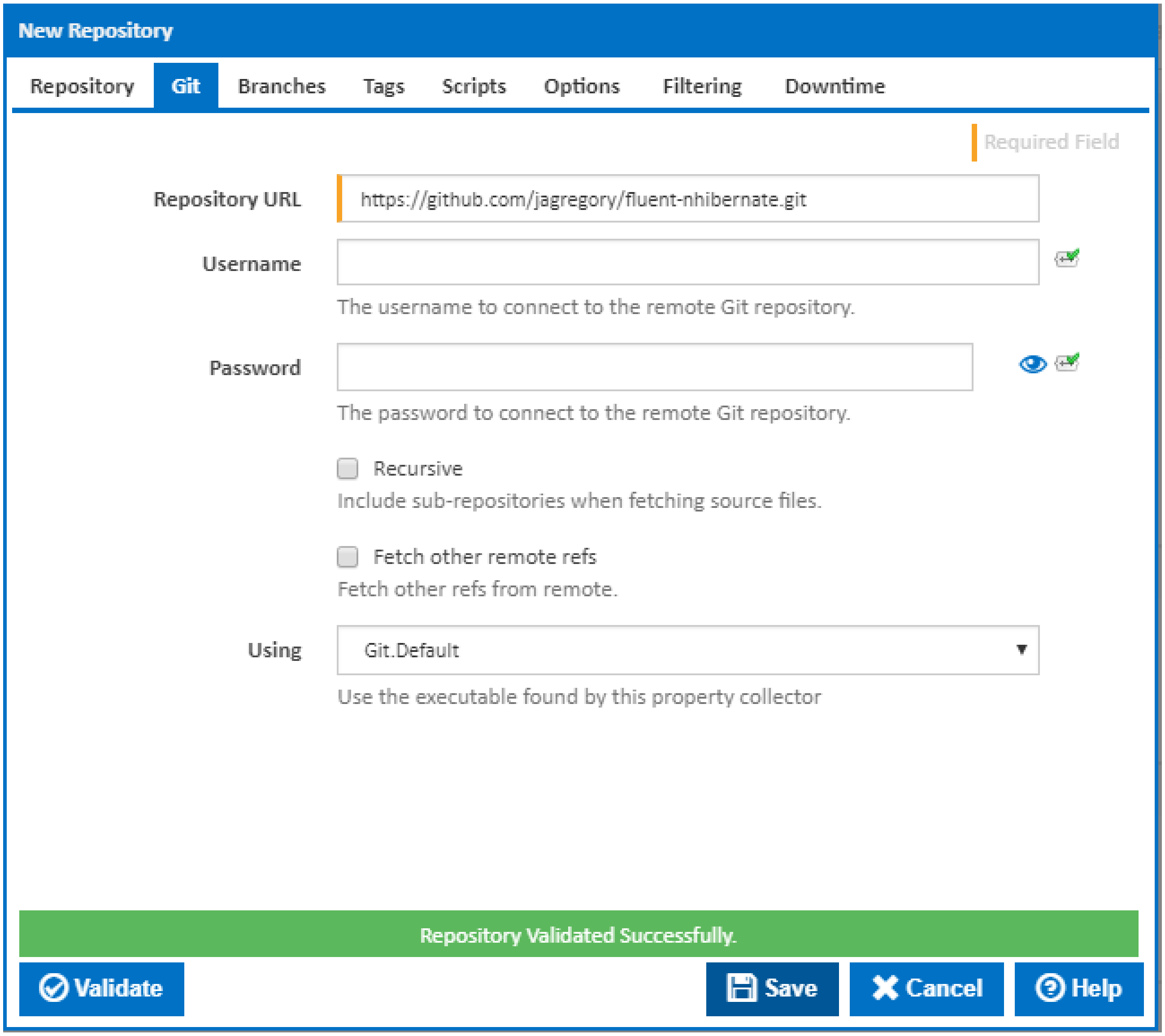
Clear this if you are only using this repository to trigger builds and do not want to copy any files to the workspace.

☑ Enabled

Repository specific properties can be set via the tabs above.

The default repository branches for this configuration can be defined using the "Mappings" dialog.

You can see that the dialog now incorporates fields that are specific to Git. For this tutorial, we can leave all the fields to their default values and set the Name to **Fluent_NHibernate** and the **Repository URL** to https://github.com/jagregory/fluent-nhibernate.git. The Repository URL should point to where your Git repository is hosted. In this case we are pointing to the Fluent NHibernate repository on GitHub.



Repository URL: https://github.com/jagregory/fluent-nhibernate.git

Username:

The username to connect to the remote Git repository.

Password:

The password to connect to the remote Git repository.

□ Recursive

Include sub-repositories when fetching source files.

□ Fetch other remote refs

Fetch other refs from remote.

Using: Git.Default ▼

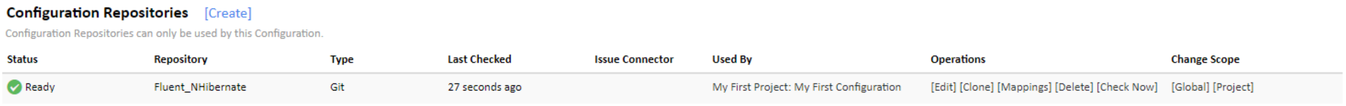
Use the executable found by this property collector

Repository Validated Successfully.

The **Using** property is used to tell Continua which version of Git it should use when retrieving your repository. This Using property uses property collectors to point Continua to the correct application it should use when performing a task. Unless Git was installed in a non-standard directory, the default **Git. Default** Property Collector should find your installed copy of Git. If Continua cannot find your copy of Git then a new property collector will need to be specified in the administration section.

Once you have provided Continua with a Repository URL, you can either **Validate** the repository or just **Save** the repository.

Once saved, the repository will be added to the Configuration Repositories like so:



**Configuration Repositories** [Create]

Configuration Repositories can only be used by this Configuration.

| Status | Repository | Type | Last Checked | Issue Connector | Used By | Operations | Change Scope |
|---|---|---|---|---|---|---|---|
| ✓ Ready | Fluent_NHibernate | Git | 27 seconds ago | | My First Project: My First Configuration | [Edit] [Clone] [Mappings] [Delete] [Check Now] | [Global] [Project] |

Congratulations, you have successfully added a repository to our configuration. Now every time **My First Configuration** is run, Continua CI will get the latest source from GitHub and use this code to run its builds.

There is one final step that needs to be done before we can successfully run a build. In Part 4: Create your First Action, we will create a build run by adding actions to our Configuration.

Continue to Part 4: Create your First Action