

Part 6: Automate your Builds

This tutorial is the last in our beginners tutorial section. It assumes you have read and completed the following tutorials:

- [Part 1: Create your First Project](#)
- [Part 2: Create your First Configuration](#)
- [Part 3: Create your First Repository](#)
- [Part 4: Create your First Action](#)
- [Part 5: Using Builds](#)

In our previous tutorials we setup our build environment which retrieves the latest source code, compiles the code and moves it to our ContinuaTutorial folder. The only problem is that we still need to manually start a build, which is where build triggers come into play.

Triggers

Triggers allow you to automate how configurations are built and when they are built. In Continua, there are several types of triggers:

- **Repository Triggers:** These triggers will begin a new build any time a source change is detected in a repository.
- **Time-base Triggers:** These triggers execute at a specific time.
- **Build Completed Triggers:** These will fire off a build when another build has finished. This trigger allows you to daisy-chain builds.

In this tutorial we will create a repository trigger that will execute a build whenever the source code changes in our [GitHub Fluent NHibernate](#) repository. So lets head back into our **Configuration Wizard** and navigate to the **Triggers** section.

The screenshot shows the 'Triggers' section of the Configuration Wizard. At the top, there is a navigation bar with 'My First Project' and various icons. Below it, a breadcrumb trail shows 'Home / My First Project / CI / Configuration Wizard / Triggers'. The main content area has a blue header 'Configuration Wizard: My First Configuration' and a breadcrumb trail '1 Details > 2 Repositories > 3 Variables > 4 Stages > 5 Events > 6 Triggers > 7 Conditions > 8 Security > 9 Reports > 10 Cleanup'. The text explains that triggers allow you to start a build based on a particular event and lists three types: Time based triggers, Repository triggers, and Build Completed triggers. Below this is a table with columns 'Trigger Type', 'Name', 'Description', 'Enabled', 'Last Executed', and 'Last Modified'. The table is currently empty, with the text 'No triggers were found.' below it. At the bottom, there are three buttons: 'Back', 'Complete Wizard', and 'Continue'.

Once you arrive at the Triggers page, you will notice that we do not have any Triggers associated with this configuration yet. So lets add a trigger by clicking the **Create** link which will bring up the Trigger dialog.

New Trigger

Trigger

Variables

Required Field

Name

Type

Enabled

Build Priority

This determines which build should run first when there are more builds queued than available agents.

Force repository check

Check for changes in all repositories associated with this configuration every time a build is started by this trigger.

Override repository branch mappings

Display a tab for selecting the branch to build for each repository. This will override the default repository branch and any repository branch mappings.

 Save

 Cancel

 Help

Similarly to the repository dialog, the options you can set for this trigger will change depending on the **Repository Type** that is set. Lets go ahead and set this trigger to be a **Repository trigger**.

New Trigger

Trigger Repository Comment Filters Variables

Required Field

Name

Type

Enabled

Build Priority

This determines which build should run first when there are more builds queued than available agents.

Force repository check
Check for changes in all repositories associated with this configuration every time a build is started by this trigger.

Override repository branch mappings
Display a tab for selecting the branch to build for each repository. This will override the default repository branch and any repository branch mappings.

Trigger specific properties can be set via the tabs above.

For this tutorial, the default settings are all we need to automate our builds. You should note that the **Repository** property should point to our repository **Fluent_NHibernate** and that this property auto populates with all the repositories that this configuration can access. So make sure **Repository** is pointing to our **Fluent_NHibernate** repository and that this trigger **is enabled** and save our new trigger.

New Trigger

Trigger **Repository** Branch Tags File Filters Comment Filters Variables

Required Field

Repository

Quiet Period 

Length of time that a build will wait on the queue accumulating changesets before starting. Set this to 0 if you want one build per changeset.

Associate Changesets 

Which changesets to list on the Changes tab of the Build view for builds triggered by this trigger

Only notify users who caused the build

 Save  Cancel  Help

Once your trigger has been created it will appear in the Triggers list which means that Continua will be constantly checking the Fluent NHibernate GitHub project for any code changes! However, as this is a GitHub project and we cannot easily change the source of the project, lets also create a daily trigger that will build our project at the same time every day. This will allow us to see automated builds in action. So lets create another trigger but this time around lets set the repository type to **Daily**.

New Trigger

Trigger

Daily

Variables

Required Field

Name

My First Daily Trigger

Type

Daily

Enabled

Build Priority

Normal

This determines which build should run first when there are more builds queued than available agents.

Force repository check

Check for changes in all repositories associated with this configuration every time a build is started by this trigger.

Override repository branch mappings

Display a tab for selecting the branch to build for each repository. This will override the default repository branch and any repository branch mappings.

 Trigger specific properties can be set via the tabs above.

 Save

 Cancel

 Help

When creating this trigger, change the time so that it will run in a couple of minutes. Once everything is configured correctly, save the trigger and head back to the Configuration homepage and wait for a build to be triggered!

New Trigger

Trigger **Daily** Variables

Required Field

Time
When to trigger the build.

Repeat
▼

Associate Changesets
▼
Associate only the latest changeset from all repositories, or associate all changesets from all repositories that have been committed since the last successful build.

 Save  Cancel  Help

Reviewing your Automated Build

Once your build has begun to execute, you should see that the **Started By** value is no longer set to your name, but rather your triggers name. Triggered builds are executed in the exact same way as manual builds and if you check your ContinuaTutorial folder, you will see that this daily trigger is updating the folder too.

Running Builds (1)

Build #	State	Agent	Stages	Started By	Branches	Started	Duration	Changesets	
1.0.0.20	Stage Executing	tornado	<div style="width: 20px; height: 10px; background-color: #4a86e8; border: 1px solid #ccc;"></div>	My First Daily Trigger	master	6 seconds ago	6 seconds	1 changesets	Stop

Queued Builds

Build #	State	Stages	Started By	Branches	Started	Elapsed	Changesets
No queued builds were found.							

Recently Completed Builds (10)

Build #	State	Stages	Started By	Branches	Finished	Duration	Changesets	Artifacts	Build Tags	Actions
1.0.0.19	[Pin] Build Completed	<div style="width: 20px; height: 10px; background-color: #27ae60; border: 1px solid #ccc;"></div>	Adam Seselja	master	2 minutes ago	22 seconds	1 changesets	2 files	None	
1.0.0.18	[Pin] Build Completed	<div style="width: 20px; height: 10px; background-color: #27ae60; border: 1px solid #ccc;"></div>	Adam Seselja	master	Yesterday at 6:06 PM	21 seconds	1 changesets	2 files	None	
1.0.0.17	[Pin] Build Completed	<div style="width: 20px; height: 10px; background-color: #27ae60; border: 1px solid #ccc;"></div>	Dave Sparks	master	Yesterday at 5:37 PM	38 seconds	1 changesets	2 files	None	

Tutorial Completed

A final congratulations for successfully completing our Continua tutorial. Feel free to leave all of the components created in this tutorial running in Continua to use as a reference. If you come back to this configuration in a few weeks and there has been some activity on GitHub then you will see some builds that were executed from our repository trigger.

You should now have a basic understanding of how Continua works and hopefully you are now contemplating how Continua can be used within your own development environment.

There are still many more aspects of Continua that were not discussed in this tutorial. If you would like to get more out of Continua then try out one of our other, more advanced tutorials.