Article

Set Up A Continuous Integration Build Definition

By **Abdul Rasheed Feroz Khan** on **Aug 06 2016**

Introduction

This article will help you to work with the code where you can set up a continuous integration build definition.

What is continuous integration build definition?

A continuous integration build definition is the best way to assure that the changes made to the code don't break anything, enabling us to detect the problems, caused by a check in the operation at the earliest possible moment.

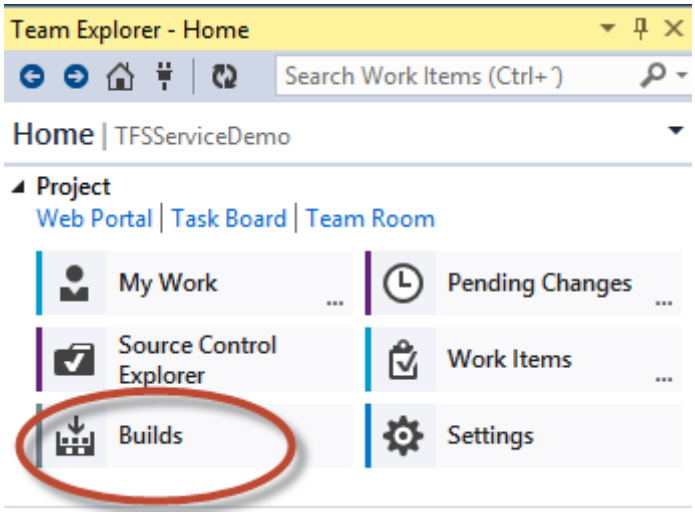
Note: Make sure, you have worked on my previous articles before you surf this.

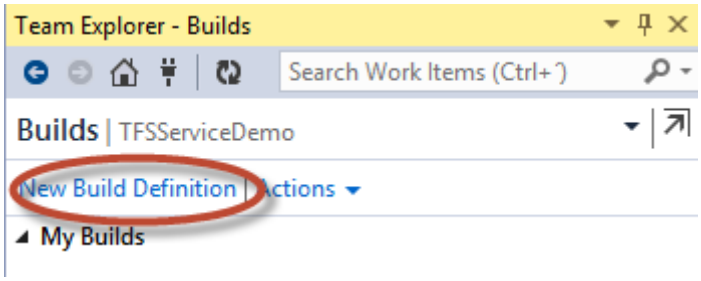
Links

- Click here to create a free Microsoft account (Outlook/Hotmail).
- Click here to learn about how to create a Visual Studio Online account and creating a new team project with TFVC added with inviting the members to work on it.
- Creating a team project with Git
- Connected IDE Experience in Visual Studio Online
- Customize your Visual Studio Online Project
- Work with Sprints at your Visual Studio Online Team Project
- Managing your project Work Flow with Kanban Board at Visual Studio Online
- Creating and Uploading a Code to TFVC Version Control in Visual Studio Online

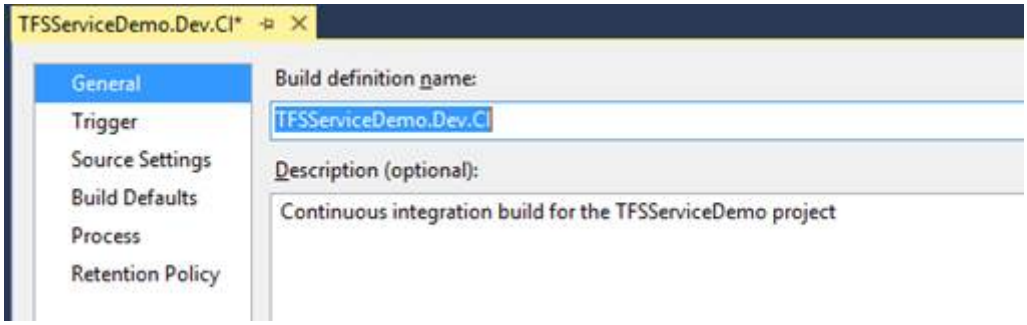
Follow the below steps now,

Step 1: From the Team Explorer, click Builds and then on New Build Definition.



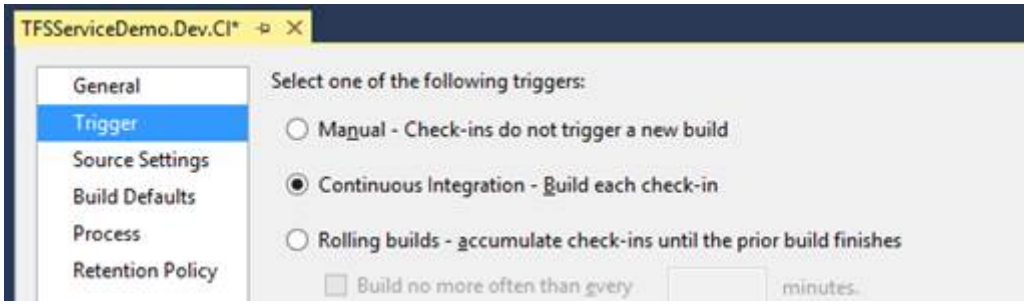


Step 2: Give the build definition; a meaningful name and a description.



Step 3: Switch to the Trigger section on the left side and select Continuous Integration.

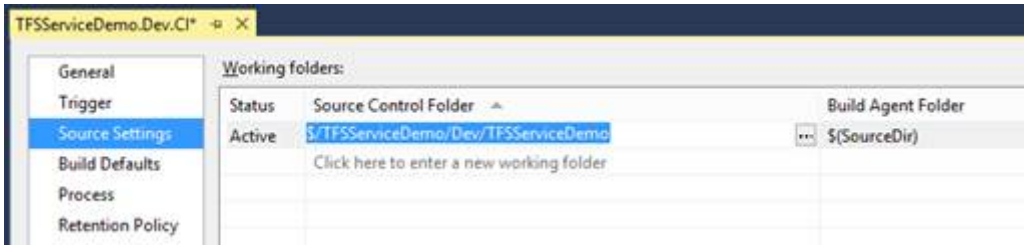
Note: This kind of trigger will make the build run, whenever anyone, checks in any changes to the version control.



Step 4: In the Source Settings section, map the appropriate workspace for the build.

Note: The Workspace mappings define the code (files) that the build will get, when triggered. Make sure that the folder contains the code , you want to build when mapped. Try to make the build. Get only the files it requires. This way, it will run faster and it's important for a CI build that can be triggered, many

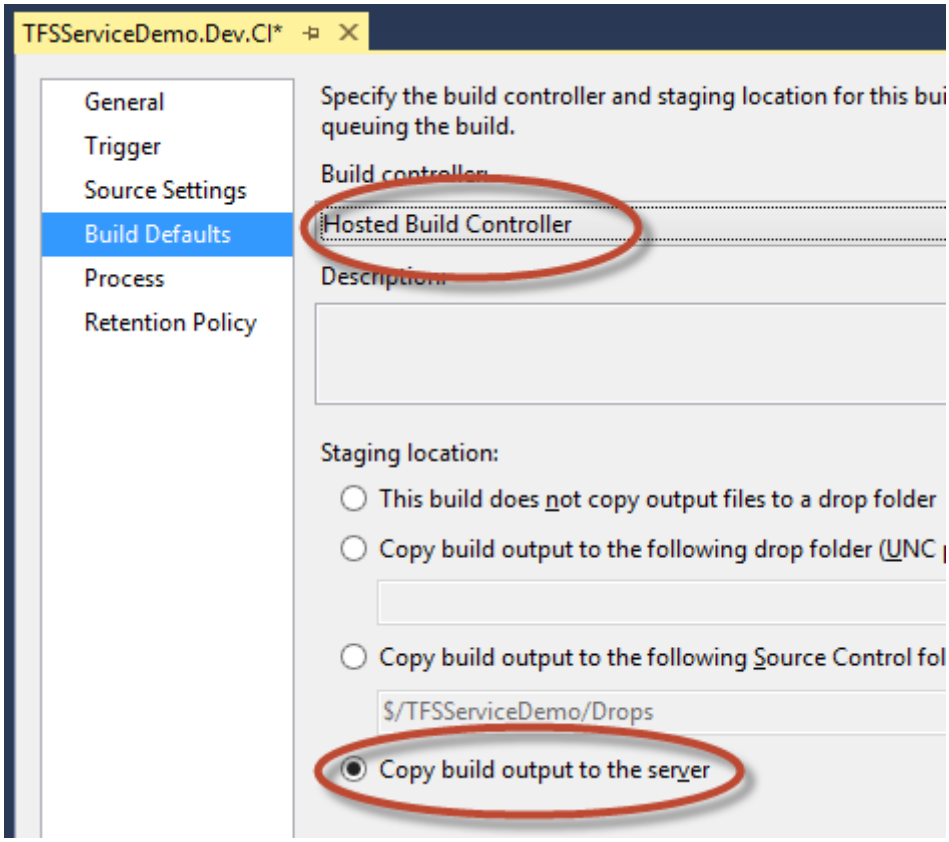
times a day.



Step 5: In the Build Defaults section, leave the Hosted Build Controller and Copy build output to the Server options selected.

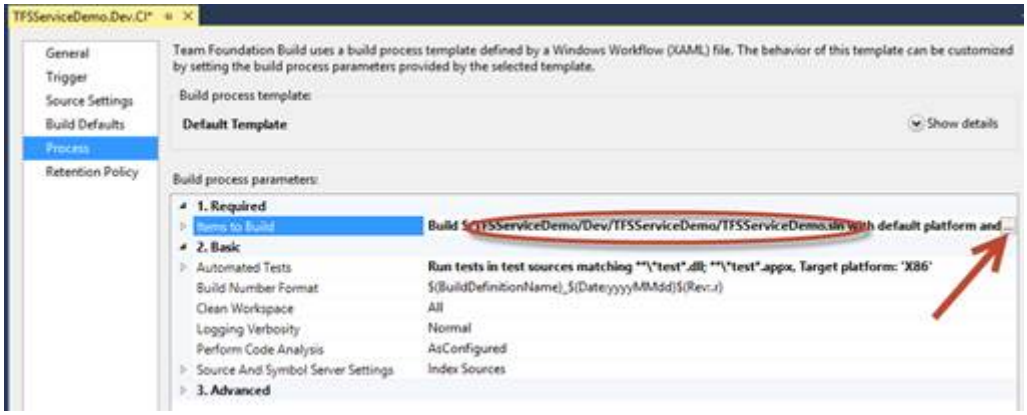
Note: The hosted build controller is the easiest way to have a build set up without the need to reserve and configure any build Servers. The Cloud hosted build controller is already configured. Thus, it is able build most of the project types, available in Visual Studio. Nevertheless, if we have to use additional assemblies, we can upload them to the version control and configure the hosted controller. Thus, it looks for them there. If we want our build to interact with the local resources in our network (for example, to deploy the built assemblies), we can always set up our own build controller and agent(s) and have them connected to the team project in Visual Studio Online.

Using the Copy build output to the Server option, the results of the build will be copied to the TFS Server (in this case, to the Cloud) and we will be able to access them from the build summary, once it has successfully finished.

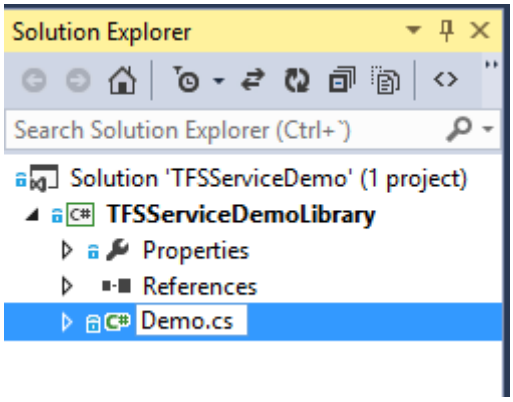


Step 6: In the process section, include the solution, we checked in, in the items to build the parameter, if it is not already present.

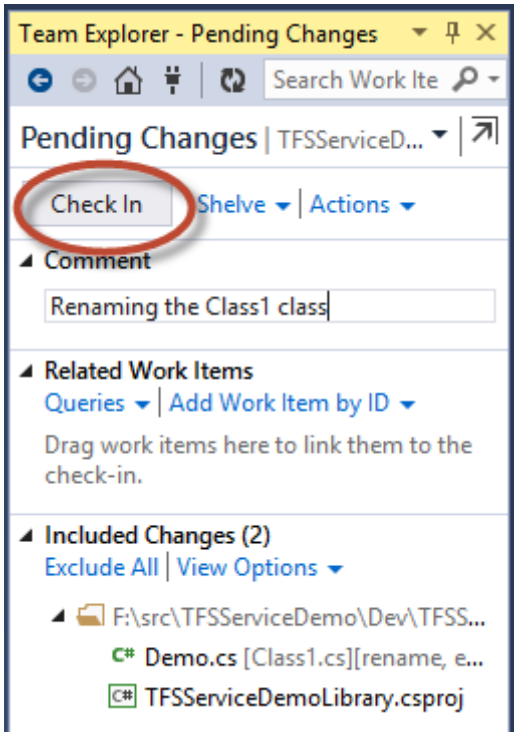
Note: As you can see, the build is already configured to search for and run any automated tests included among our code, although we still don't have any of them written.



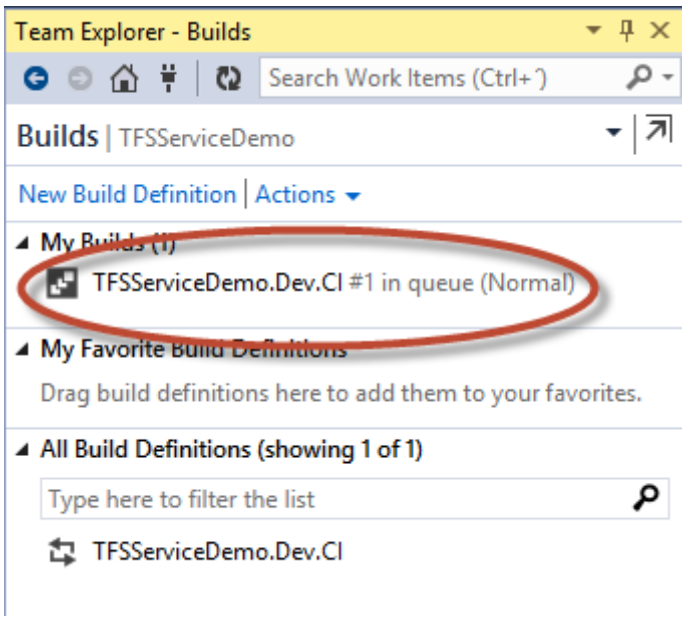
Step 7: Save the build definition. From the Solution Explorer, make any change to the code, such as renaming Class1.cs. Make sure that the solution is built correctly in Visual Studio.



Step 8: Check the pending changes from the Pending Changes view in Team Explorer.



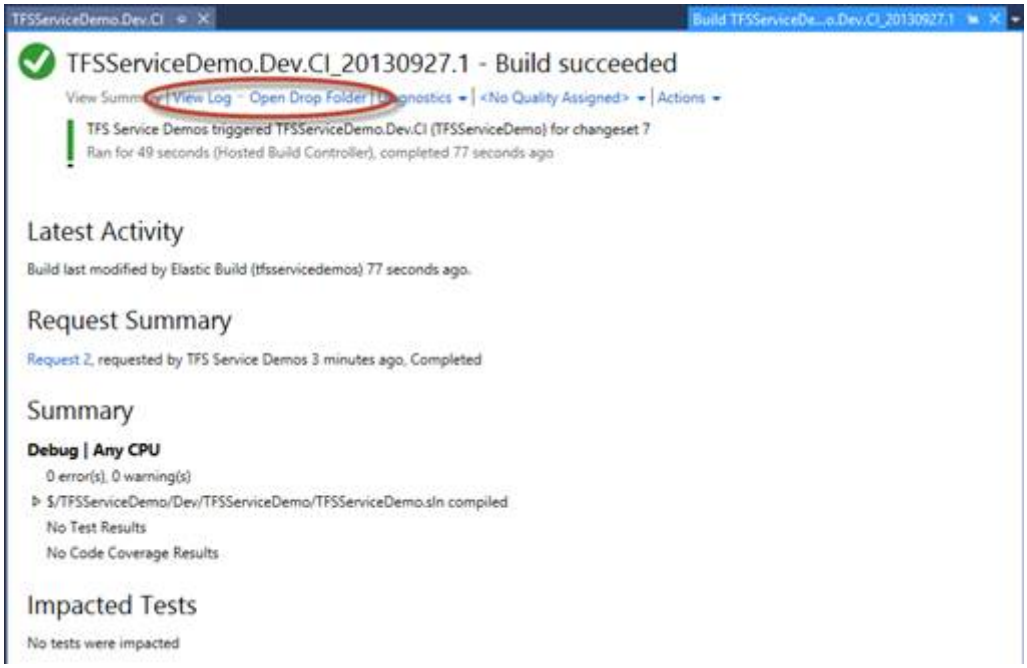
Step 9: Switch to the Builds section in Team Explorer. A new instance of the build we have just configured should have been triggered, because its trigger was set to the continuous integration.



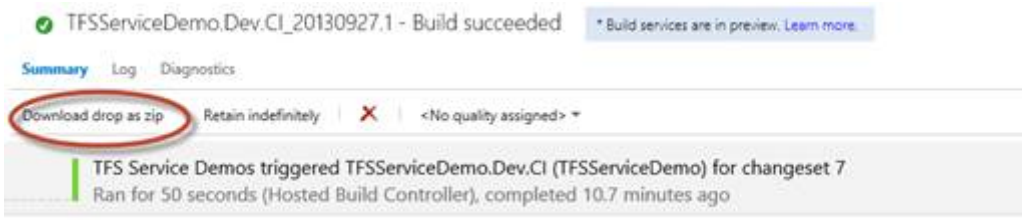
Step 10: Double-click on the running build and show the progress.

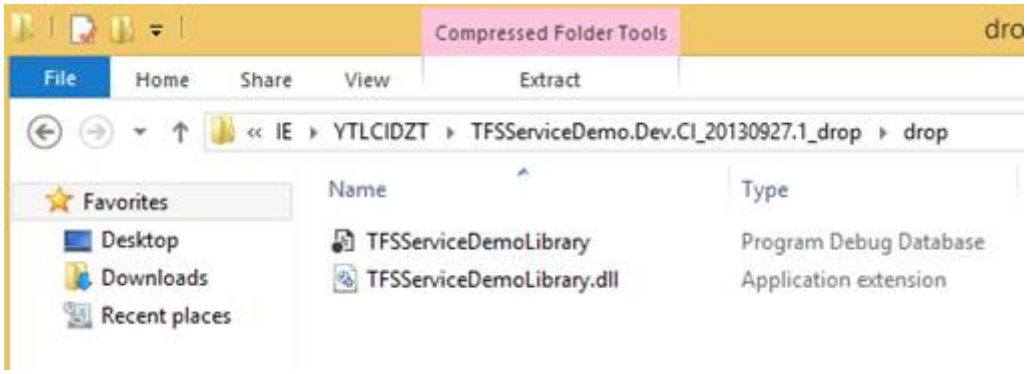
Step 11: Click the Open Drop folder link to check the outcomes of the build.

Note: Once the build has finished, we can view the details of its execution by clicking View Log or get the resulting artifacts by clicking Open Drop folder.



Step 12: You will be redirected to the details of the build in the Web Access site of the team project. Click on the available link to download the drop and show the results.





Follow my future articles for the Team Collaboration on Visual Studio Online.

Thank you for using C# Corner