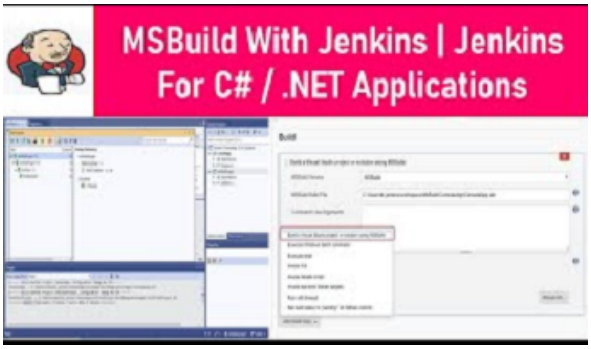
Menu



Build A Visual Studio Project In Jenkins

November 3, 2022 by Amandeep Singh

Spread The Knowledge

Hello everyone, we are back with one more DevOps topic. Jenkins is a very popular CI/CD tool in the DevOps world. Today we will talk about how to build a visual studio project in Jenkins. As it is open source, you may face some issues when you try to build the visual studio project in Jenkins first time. If you are also facing that issue, you are at the right place. I will explain step by step procedure to build a visual studio project in Jenkins.

1. Jenkins server is already installed on Linux/Windows VM and up and running.
2. Configure a windows agent under Jenkins Nodes
3. Install MS Build software on the Windows machine.

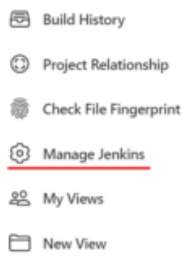
Once the above three prerequisites are in place, we are good to move ahead with our topic. Let's just dig in. Below is the main list of tasks that we need to complete to build a visual studio project in Jenkins.

- Install MS Build Plugin in Jenkins
- Configure MS Build Plugin
- Create a Free Style Project
- Restrict the job to Windows Agent
- Add Git Repository
- Add Build Steps
- Archive the Artifacts.
- Build the project.

Install MS Build Plugin in Jenkins

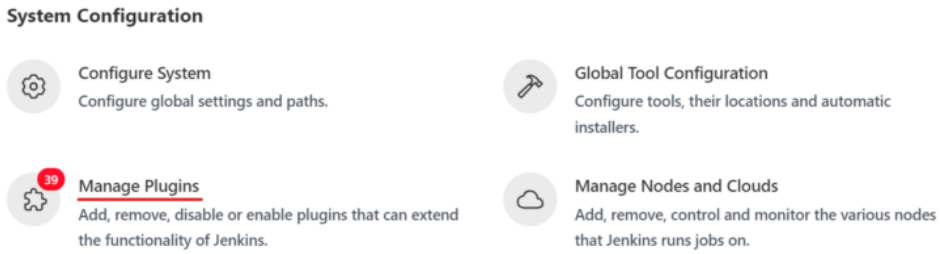
The First step is to install the MS Build plugin in Jenkins.

1. Click on **Manage Jenkins**



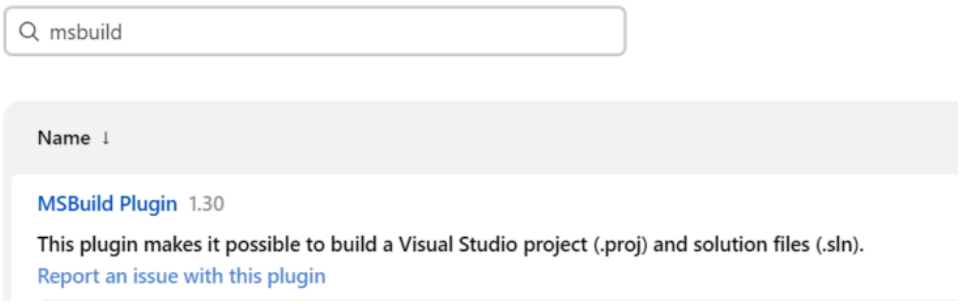
Manage Jenkins

2. Select **Manage Plugins**



Manage Plugins

3. Click on **Available**
4. Search with the term: "msbuild"



msbuild search plugin

5. Click on **Install without restart**
6. This plugin makes it possible to build a Visual Studio project (***.proj***) and solution files (***.sln***).

Configure MS Build Plugin

After installing the msbuild plugin, next step is to configure it so that we can use i our jenkins pipeline.

5. Click on **Add MSBuild**. This is basically telling jenkins where exactly the msbuild has been installed on our agent.
6. Give a name to the installation. **Name** can be any. You can specify the Visual Studio version also here if any specific version has been installed.
7. Paste the msbuild installation path in the **Path to MSBuild** box.

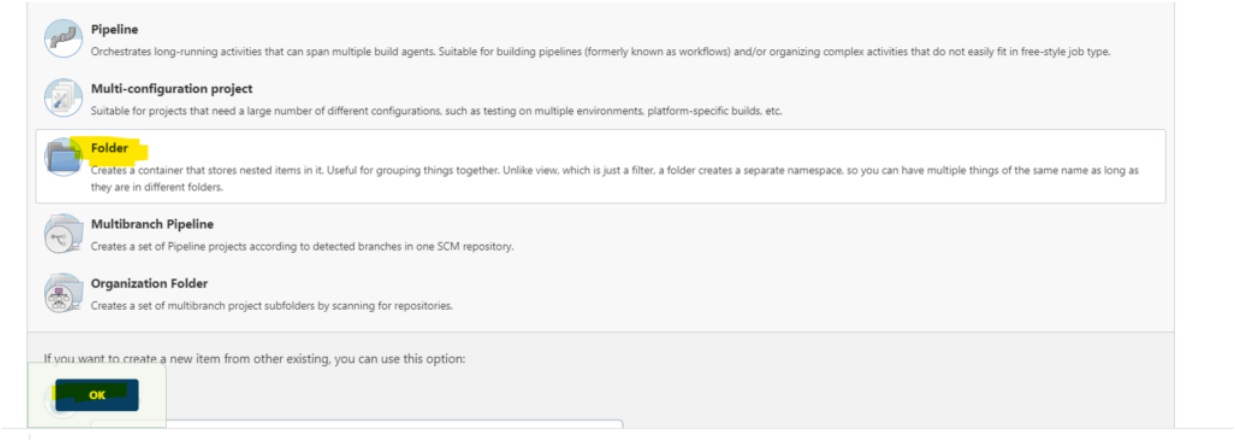
msbuild installation path

8. Click on **Save**

Create A Folder and A Project

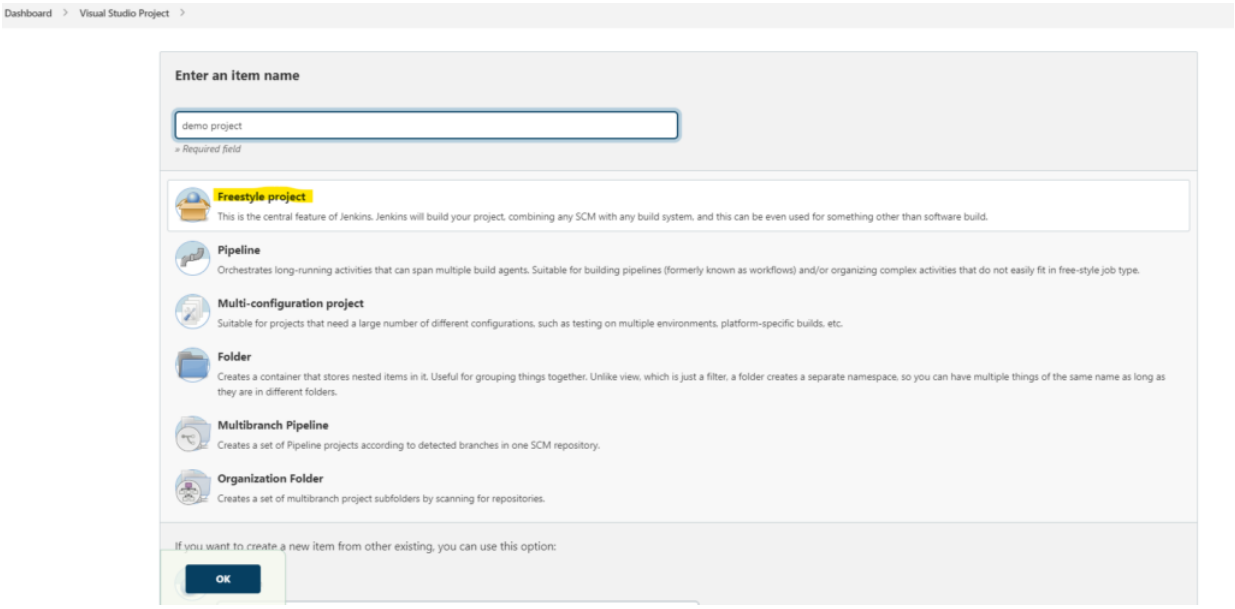
Our MSBuild plugin has been configured now and ready to use in our project. Let's create a project now:

1. Click on New Item.
2. Enter "Visual Studio Project" (any name) and Select Folder.



jenkins new folder

3. Click on OK
4. In next Screen, just click on Save
5. Now go into the newly created folder “Visual Studio Project” and Click on New Item.
6. Now give this project a name and select Freestyle project.



Jenkins freestyle project

7. Click OK.

Restrict the Job to Windows Agent

As you know that we have visual studio installed on a windows agent, hence in the pipeline project also, we need to configure the pipeline to run on windows agent c

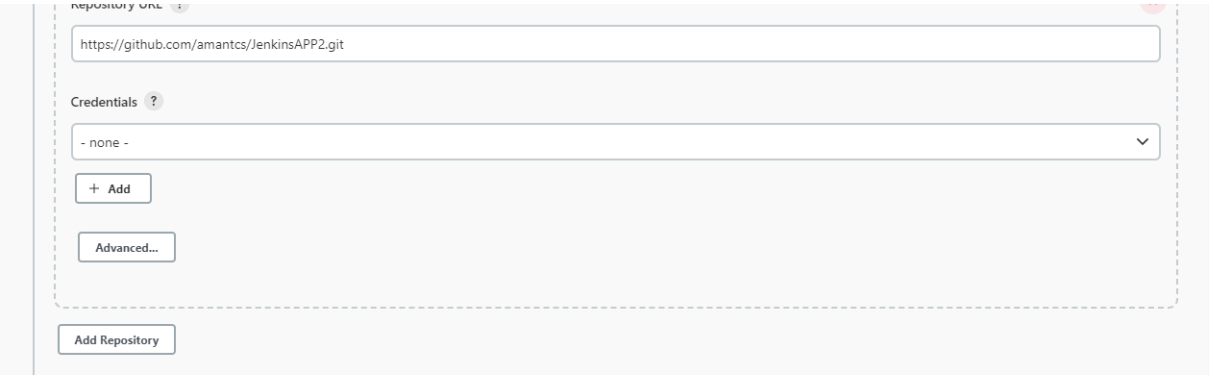
2. After Description box, there would be checkboxes, just check the "**Restrict where this project can be run**"
3. In the **Label Expression**, enter the label name you have given to your windows agent. like below:

jenkins windows node

Add Git Repository

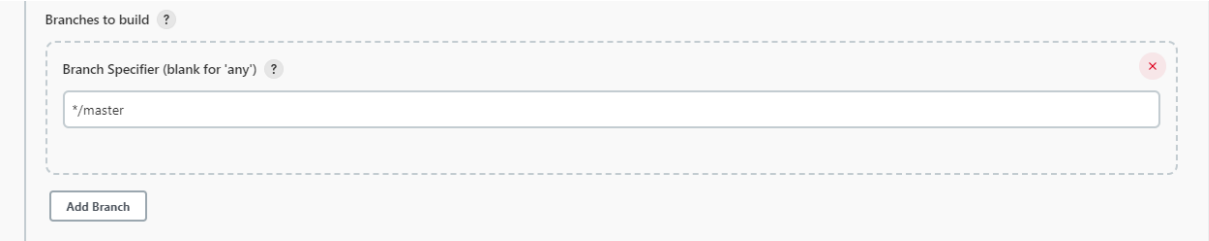
Under Source Code Management, we need to give the source location where we have our code. This code is basically Visual Studio solution files which are stored on a git repo.

1. Under Source Code Management, Select **Git**
2. In **Repository URL**, you can give repo link. If you don't have any, you can use below one with one sample visual studio project:
   https://github.com/amantcs/JenkinsAPP2.git



jenkins repository url

3. In the above step, if you get error "*Failed to Connect to Repository Error Performing Git Command Git Ls-Remote*", then pls check out this link
4. In **Branch to build**, you can leave it as default value "*/master" or specify your own branch which you want to build.

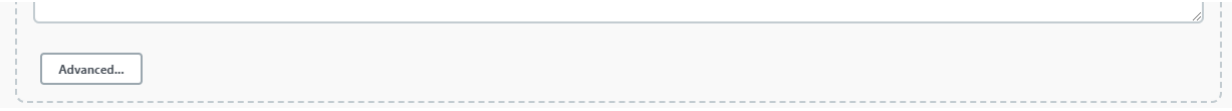


specify git branch

Add Build Steps

This is very important step. Please follow each and every step very carefully in this section.

1. Under Build, click on **Add build step**, A drop down menu will pop up.
2. Select **Execute Windows Batch command**.
3. Under **command** box, enter "*dotnet restore*"



windows batch command

4. Now click again on **Add Build step**.
5. Select **Build a Visual Studio project or solution using MSBuild**
6. Under **MSBuild Version**, select the name you gave to msbuild installation during msbuild configuration step.
7. Next, give the .sln file name under **MSBuild Build File**. for example: JenkinsAPP.sln

msbuild step jenkins

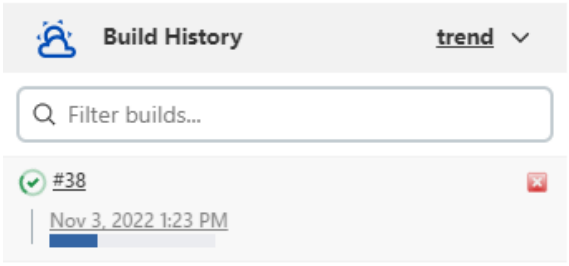
Artifacts

We have successfully added all the build steps required to build a Visual Studio solution. Now it is time to save the artifacts which will be produced as part of build. for that, please follow below:

1. Under Post-build Actions, click on **Add post-build action.**
2. From drop down, select **Archive the artifacts**
3. In **Files to archive** field, add path to the location of dll files which would be generated after build. For example:

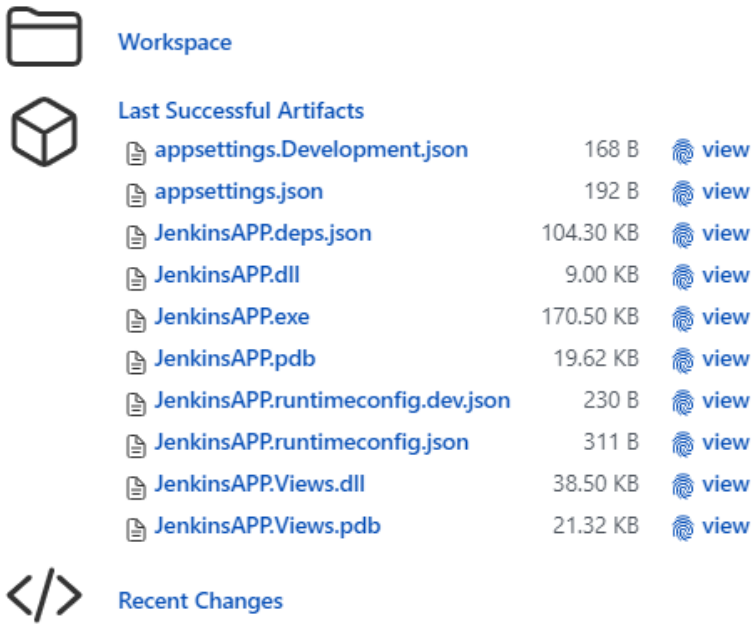
Build The Project

After completing all the above steps, now you can click on Save. Our pipeline is ready now. Click on Build Now to build the project. You will see a progress bar:



jenkins build progress

You can click on console output to see real time processing data from console. Once finished, you will see SUCCESS in the end of console output. On project home page, you will see the artifacts files as below.



msbuild artifacts

Congratulations!! You have successfully built the Visual Studio project in Jenkins CI/CD tool and produced the artifacts. You can use these artifacts for deployment later. If you face any issue while following above steps, please do let us know in the comments, we will try our best to help you out. Goodbye!

Read more about Jenkins here.

Spread The Knowledge

.NET project using MSBuild in Jenkins?, How does Jenkins integrate with Visual Studio?, How to build dotnet project in Jenkins, how to build visual studio project in jenkins, how to configure msbuild in jenkins, jenkins, msbuild, Which plugin can be used to build .NET project?

‹ Failed to connect to repository Error performing git command git ls-remote -https github com jenikis hook git HEAD

› From Manual to Automated: A Windows Server Admin's Guide to DevOps

Leave a Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

I'm not a robot reCAPTCHA Privacy - Terms

Post Comment

From Manual to Automated: A Windows Server Admin's Guide to DevOps March 5, 2023

Build A Visual Studio Project In Jenkins November 3, 2022

Failed to connect to repository Error performing git command git ls-remote -https github com jenikis hook git HEAD August 8, 2022

Terraform Commands Ultimate Cheat Sheet May 9, 2022

[convertful id="131366"]

Archives

March 2023 (2)

November 2022 (1)

August 2022 (1)

May 2022 (1)

September 2021 (1)

July 2021 (11)

June 2021 (2)

February 2021 (3)

January 2021 (12)

December 2020 (16)

November 2020 (21)

October 2020 (33)

September 2020 (11)

August 2020 (10)

July 2020 (27)

June 2020 (9)

May 2020 (1)

April 2020 (20)

Categories

Cloud DevOps Linux PowerShell Tech News VMWare Windows