**NYC DOT IT&T**

**April 28, 2016**

****

**NYC DOT IT&T**

**Lower Manhattan Permit Management System**

**NYCStreets Permit Search – Business Requirements Document**

**DRAFT**

**NYCStreets**

**Permit Management System**

**TFS Build and Release Guide**

****

**NYCStreets Permit Management System**

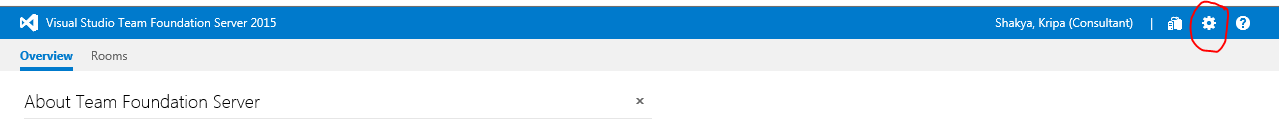
**TFS Build and Release Guide**

|  |  |
| --- | --- |
| **Author:** | **Kripa Shakya** |
| **Project Manager’s Name:** | **Patrick McCambridge** |
| **Date:** | **April 28, 2016** |
| **Document Version Number:** | **V1.0** |
| **Document:** | **TFS Build and Release Guide** |
| **Template Version:** | **V1.0** |

## TFS BUILD and RELEASE GUIDE

#### Set up Agent Pool

* 1. Go to this link <http://dottfs01:8080/tfs/>
  2. Click on the Settings button located on the top right corner as highlighted in red below:



* 1. Then click on Agent Pools.
  2. Once you click on Agent Pools, you will see a ‘Download agent’ option, so click on that.
  3. It’s going to download a zip folder, now extract it wherever you want to configure it.
  4. Open command prompt window and run **ConfigureAgent.cmd** command.

*[Note: The path in which you run the above command should be where you extracted your agent folder]*

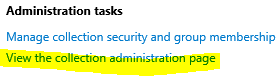
For Example:



/\* Rishi will provide the screenshot to cover this section to include the questions being asked after the above command is being run \*/

#### Create a New Project

* + 1. Go to this link <http://dottfs01:8080/tfs/>
    2. Follow the similar step as mentioned above in section **I.2.**
    3. Once you land in to the Control Panel, click on ‘View the collection administration page’ as highlighted below:



* + 1. You will land into the page where you can create a New Project. Under ‘Projects’ heading, click on the ‘New team project’ link.
    2. You will get a popup modal, enter Project name, Description, Process template as **Scrum** and Version Control as ‘**Team Foundation Version Control’**.
    3. Once the project is created, click on **‘Navigate to Project**’ button. This will direct you to the homepage of your newly created project where you can view all the items related to it.

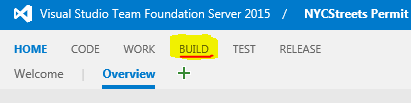
#### BUILD Process

Now that you have the project set up, we are all set to build it!

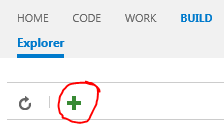
* 1. Check In all your code in TFS from Visual Studio.

*[Note: Please do not include packages folder when you check in the code. This will create an issue when you build the solution in TFS.]*

* 1. Click on “**Build**” tab as highlighted below:



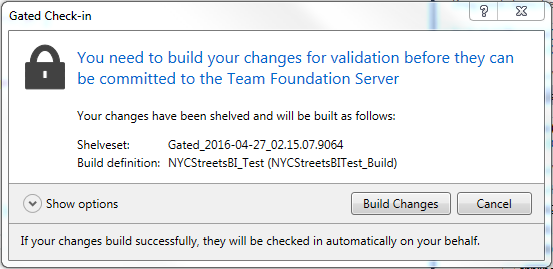
* 1. This is the page where you create/manage all your builds. Let’s go ahead and create a new build.
  2. Click on the Green plus button located on the explorer menu as shown below:



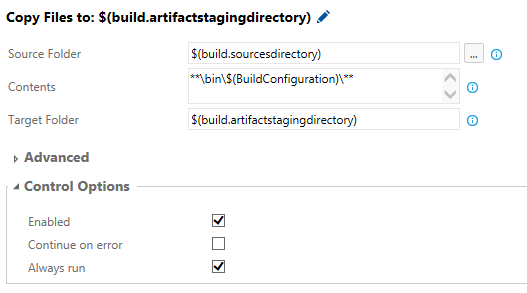
* 1. A popup modal will be opened. Following the following steps:
     1. Select a build template as “Visual Studio” *[Selecting this option will create some basic tasks required to publish the solution; you can also opt to select Empty template and add tasks later].*
     2. Click next Select a repository (this should be your project).
     3. You can also select the checkbox if you want to do the Continuous Integration [run the build whenever the repository is updated – example: this project will build whenever someone checks in].
     4. Click Create.
  2. Since we selected “Visual Studio” as our build template. It created 6 basic tasks as mentioned above in section 5.a.
  3. Useful Features:



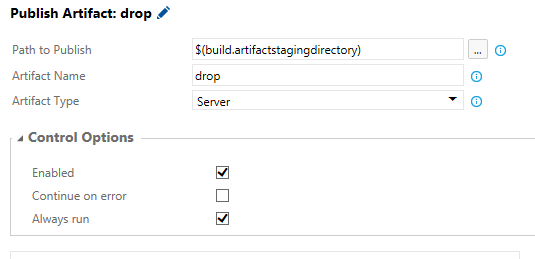
* + 1. Variables: A series of variables can be created in Variables. Username and password is one example where variables can be made use of.
    2. Triggers: This tab has options to choose when we want to do the build.
       - Continuous Integration (CI): Build will run each time there’s an update to the repository that means it’ll build each time someone checks in the code.
       - Scheduled: You can schedule when to build the solution.
       - Gated Check-in: This option accepts check-ins only if the submitted changes merge and build successfully. Following option will be prompted when you try to check in the code in Visual Studio with this feature turned on:



* 1. Let’s edit these tasks now:
     1. NuGet Installer: The build is based on the solution. This is the place where we locate the path to the solution. Example: \*\*\\*.sln [Note: If you Uncheck Enabled option under ‘Control Options’ section, it will disable this task and skip while building as well, and this applies to any of the tasks].
     2. Visual Studio Build:Include MSBuild Arguments: /p:OutDir="$(build.stagingDirectory)" /p:DeployOnBuild=True /p:WebPublishMethod=Package /p:SkipInvalidConfigurations=true
     3. Visual Studio Test: Include Test project here in the Test Assembly if you have one.
     4. Index Sources & Publish Symbols: Edit anything if required.
     5. Copy Files: Add the folder’s names from where you want to copy the files from, on which location(target solution).



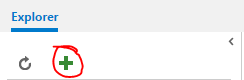
* + 1. Publish: This is where the published files get copied to.



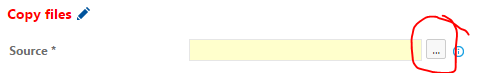
1. **RELEASE PROCESS**
   1. Click on Release Option:



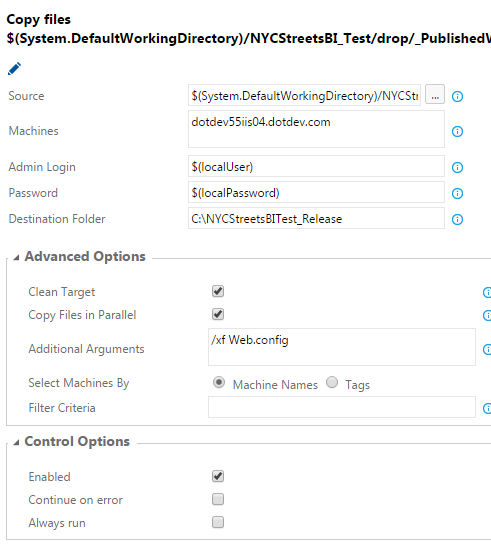
* 1. Click on the little Plus button that’s located on the Explorer menu



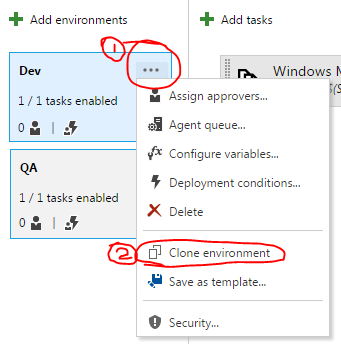
* 1. A popup modal will be presented. Select Empty Template and hit OK button.
  2. A Default Environment will be created. Click on the “Default Environment” name to change the Environment name of your choice. For Example: “DEV Environment”.
  3. Similar to build process, here as well you can add tasks.
     1. Click “Add Tasks”
     2. Select the kind of task you want to perform. Let’s add “Windows Machine File Copy”
     3. Source: After the above task is being added, select the build you want to release by clicking on the highlighted button below:



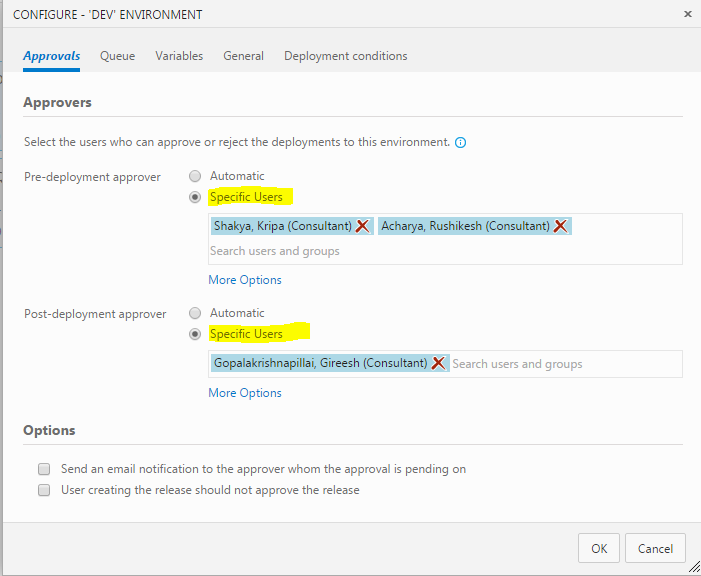
* + 1. Machines: This would be your server name. Similar to build process you can add variables in the release process as well. This is a great feature because if you have multiple tasks you can just define the machine names in your variable.
    2. Admin Login: Login ID for the above mentioned machine if applicable.
    3. Password: Password for the above Login Id.
    4. Destination Folder: Target location where you want to deploy the code.
    5. Advanced Options:
       - Additional Arguments: Use Robocopy command if you wish to exclude any files/folders.
       - For example:
         * To exclude single file: /xf Web.config
         * To exclude a folder: /xd JavaScript
    6. Release Configuration should look something like this:



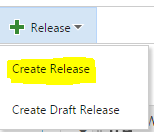
* + 1. If you want to release the code to multiple, you can achieve that by creating multiple tasks with similar configuration, but different destination folder.
    2. Releasing to a multiple environment, example DEV, QA and PROD:
       - Once we set up Dev Environment, we can copy the same configuration to the other environments as well. You can create a new environment from scratch by Clicking on ‘Add environments’ button and following the same method as mentioned above as well.
       - As highlighted below, click on the three dotted button next to an environment already created.
       - Then select ‘Clone Environment’, this will create a duplicate environment.
       - Once this is done you can go ahead and edit the configurations of the duplicated environment.



* + 1. Assign approvers: You can also assign pre-deployment and post-deployment approvers. Click on the ‘Assign approvers’ option as you can see as the first option in the snapshot above. The following configuration popup modal will be displayed:

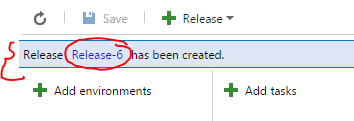


* + 1. Let’s go ahead and release this build.Click on ‘Create Release’ button as shown below

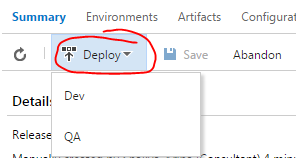


* + 1. A pop up will be displayed, select the successful build you want to release, Click ‘Create’ button.
    2. A notification will be displayed that new release has been created as shown below:

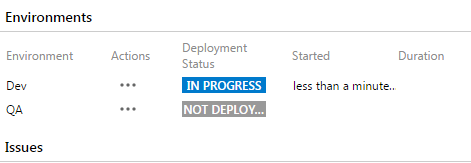
[Click on the release id to view the log, for example “Release-6” as shown below].



* + 1. You’ll land into the summary section, click on the Deploy dropdown button and choose where you want to deploy the code.



Once you deploy, “Deployment Status” changes to “In Progress” of the selected environment:



* + 1. Click on “Logs” to view step by step process of the deployment.