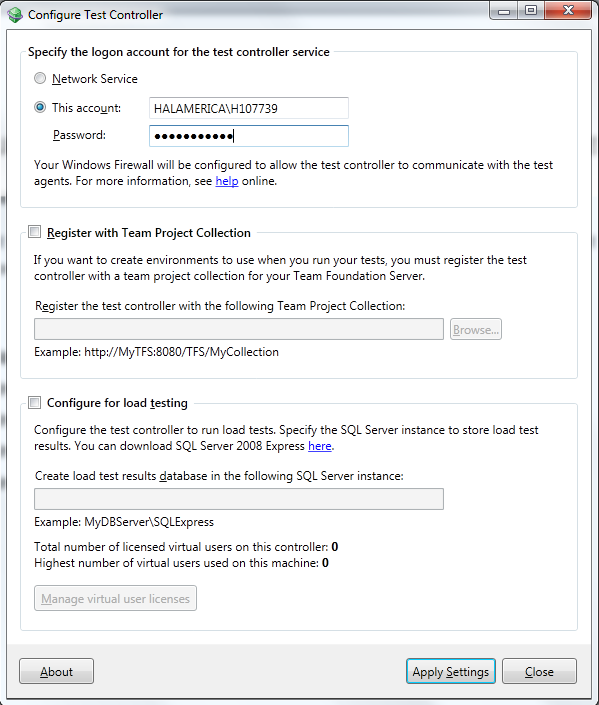
**GUIDELINE TO CREATE UNATTENDED MODE ENVIRONMENT**

1. **Configure Test Controller**

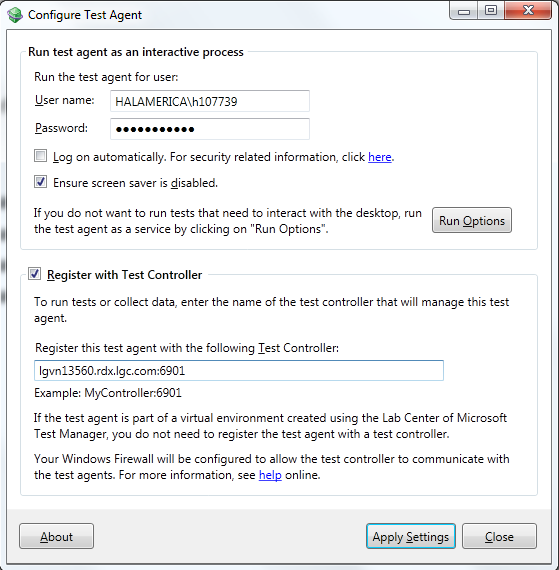
At Unattended mode, we just need to enter Username and Password into Configure Test Controller window. Then, click Apply Settings button.



*Pic 1.1*

1. **Configure Test Agent**

* Open Configure Test Agent window
* Enter User name and Password
* Check on “Ensure screen saver is disable”
* Check on “Register with Test Controller and enter Test controller name with port number
* Click apply setting button.



*Pic 2.1*

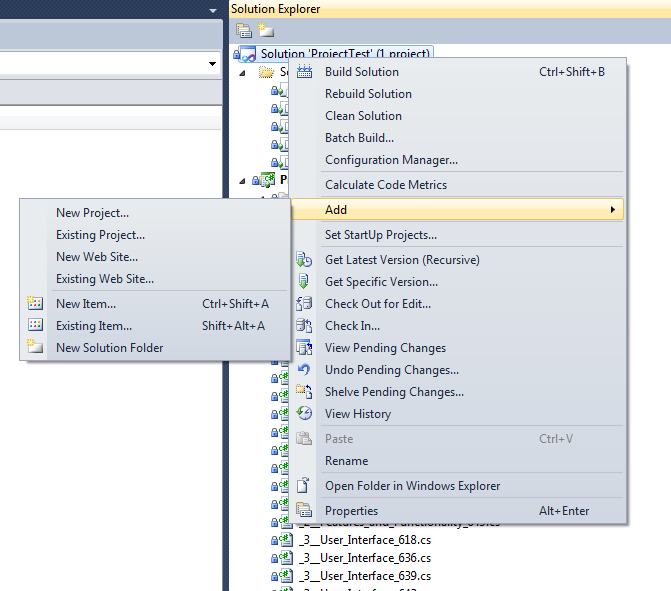
*Note: Make sure test agent is set with interactive process mode. To change to this mode, click Run Options. It should need to restart machine to make Test Agent online in some case.*

1. **Create and configure Test Setting**

Before doing this part, make sure you have TA-MTM integration project in your data source in TFS.

1. **Create Test Setting file:**

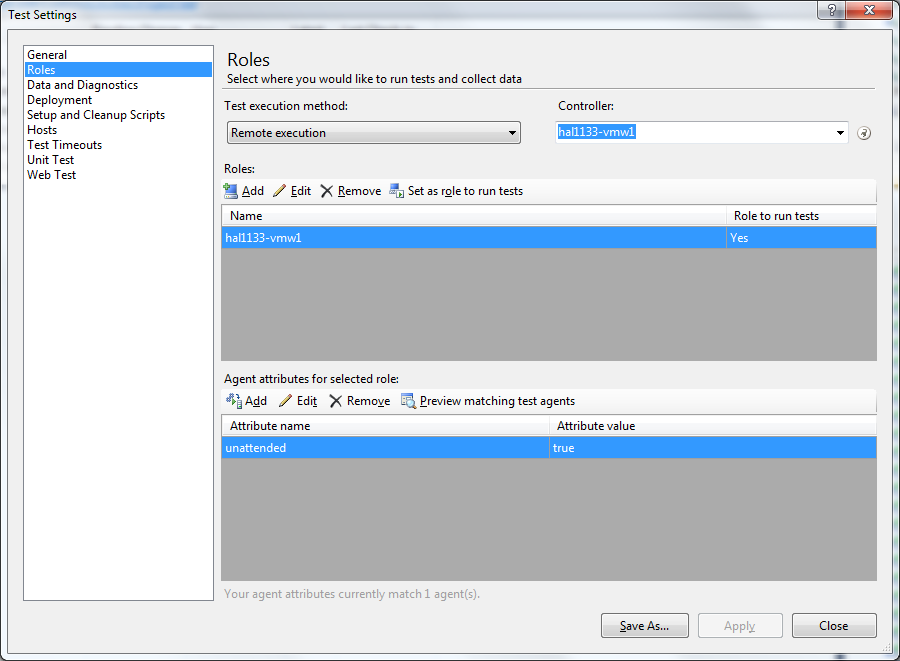
* Open TA-MTM integration project solution
* Right click on project solution and choose Add->New item
* Search and select test setting file. Rename file if you want



*Pic 3.1*

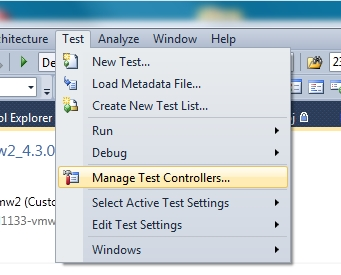
1. **Configure Test Setting file:** *Pic 3.1*

* Select Roles tab
* Select “Test execution method” is Remote execution
* Enter Test controller name into Controller textbox
* Add Roles name and enter any name
* Add Agent attributes:
  + Enter Attribute name is “unattended” (This is used to be a keyword for identifying which test agent is executed. You could use any name)
  + Enter Attribute value is “true” (default value, must be “true”)
* Select “Test Timeouts” tab and enter suitable value timeouts

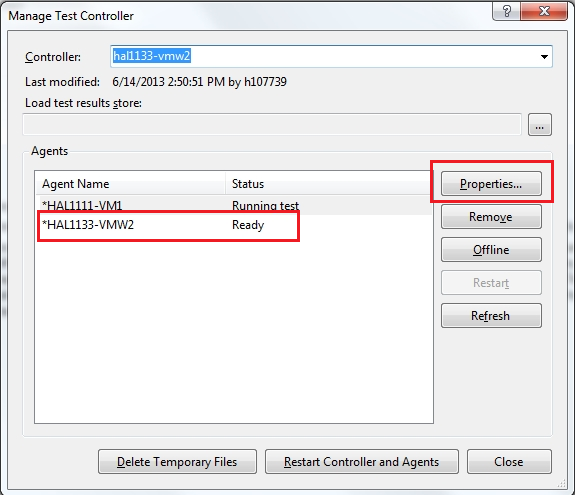


*Pic 3.2*

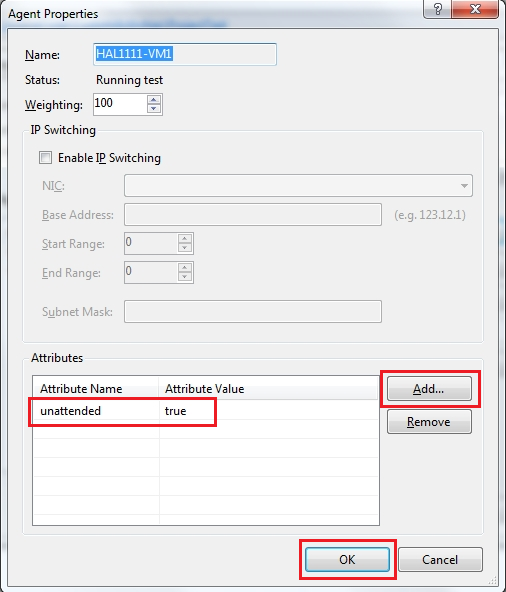
* Click Save As button
* In toolbar of Visual Studio, select Test->Manage Test Controllers



* Select test agent which you want to run on.
* Click Properties button



* Click Add button
* Enter Attribute Name and Attribute Value exactly in Role tab of test setting (Pic 3.1)

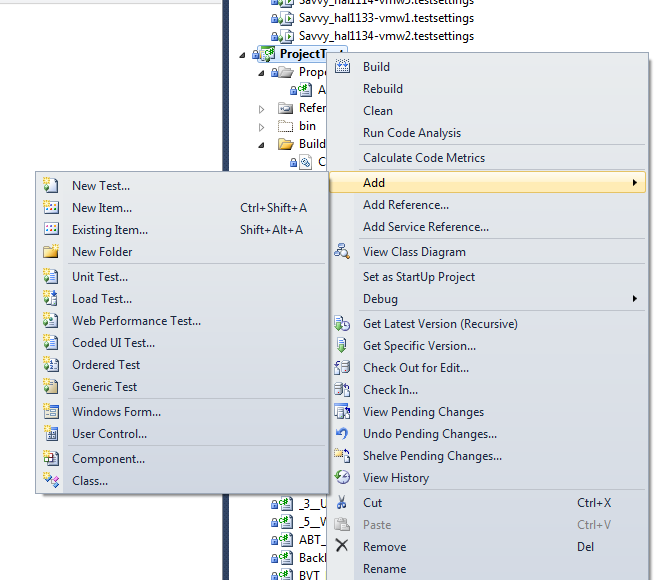


* Click Ok button.

1. **Create Test List**

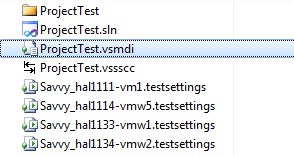
Create order test file:

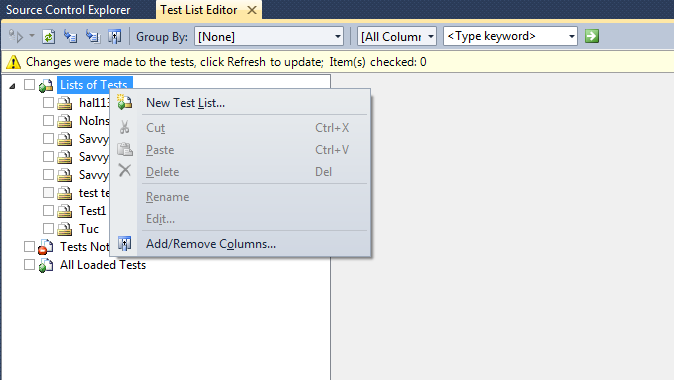
* Right click on TA-MTM Integration project and select Add->Ordered Test (pic
* Rename ordered test file
* Select test cases and add to order test

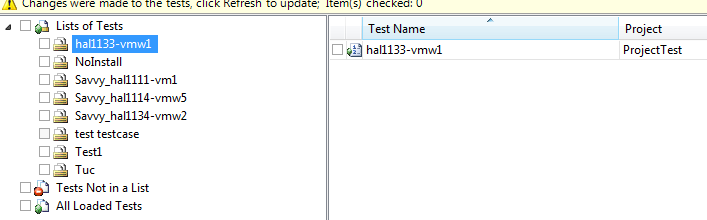


Create Test List:

* Open Test List Editor by click on file .vsmdi
* Right click on “Lists of Tests” and select New Test List…
* Enter Test List name
* Drag file order test to Test List







1. **Create Project file for Build Definition**

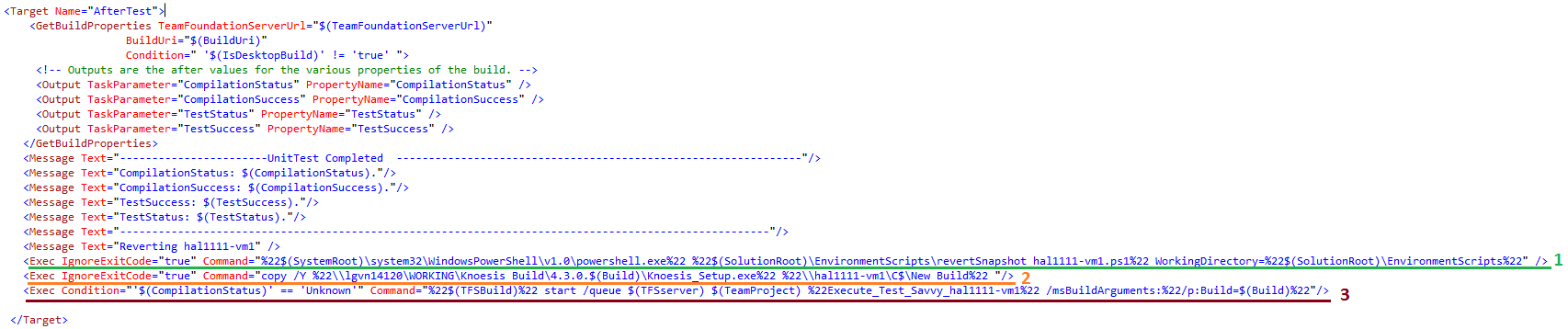
We already have a template project file which was gotten from client and modified due to use on our local environment. The project file’s name is TFSBuild.proj *(In any case, please do not rename this file. This file can use only with this name).*

As client model, they use two kinds of build definitions. One is used to revert and execute. And one is used to execute only. So that we need two kinds of project files, one used to call PowerShell command to revert visual machine and call test build definition, one used to execute test list.

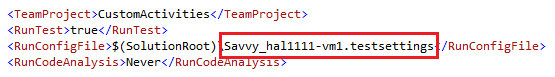
We already have a template of two kind of project files. We need to pay attention in some point:

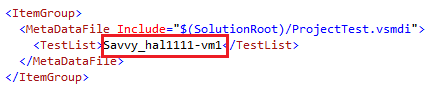
* Revert and execute project file:
  + Call reverting PowerShell file to revert machine in line 1.
  + Copy AUT build to folder in test machine in line 2.
  + Call execution build definition in line 3.

*Note: we could add command copy all test data to test machine.*



* Execution project file:
  + Which test setting file is used on project file? (This is test setting which was mentioned in part 3)
  + Which test list is used on project file? (This is a test list which was mentioned in part 4)



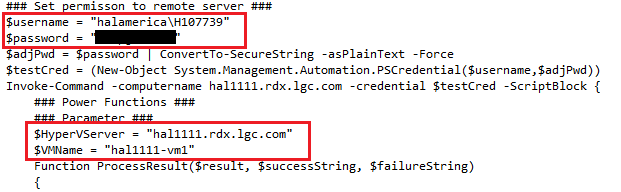


1. **Create reverting PowerShell file**

Reverting PowerShell script depend on what kind of visual machine. In our visual machine, it is Hyper-V, so we could use template PowerShell in our local TFS server.

Fix PowerShell script following this guideline due to your purpose:

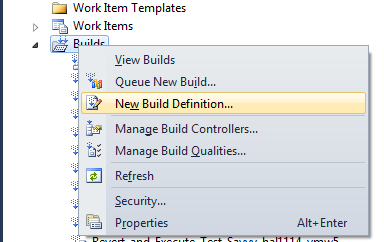
* Username: domain user of host Hyper-V machine. i.e. “halamerica\H107739”
* Password: password of user of host Hyper-V machine. i.e. “abcd”.
* HyperVServer: hostname and domain of host Hyper-V machine. i.e. “hal1111.rdx.lgc.com”.
* VMName: hostname of test machine. i.e. “hal1111-vm1”.



1. **Create and configure Build Definition**

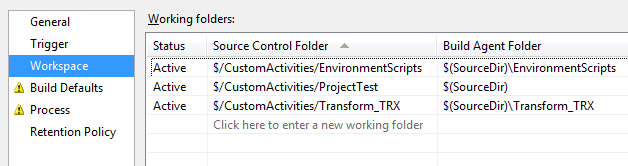
Follow model of client, we also have two kind of build definitions. One is used to revert and execute test. One is only used to execute test.

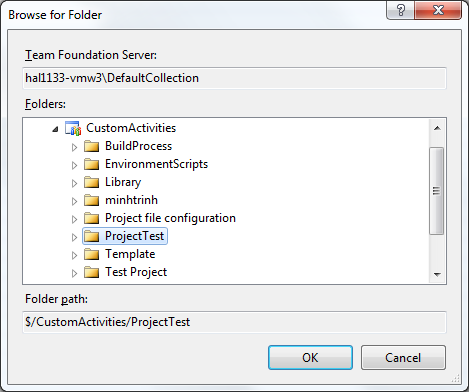
* Go to Team Explorer
* Right click on Builds
* Select New Build Definition



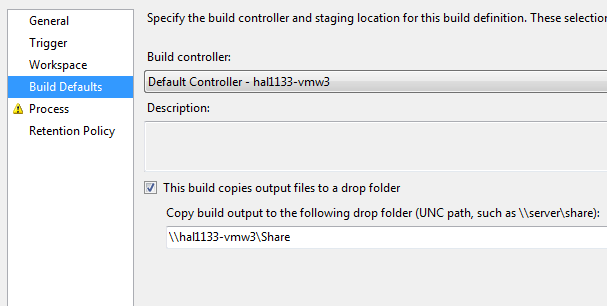
1. **Revert and execute test build definition:**

* In General tab, enter build definition name.
* In Workspace tab, select working folder following this structure:
  + Folder TA-MTM project map with $(SourceDir) in Build Agent Folder
  + Folder Transform contain TRXTransform.xslt file which is template to create final result.
  + Folder EnvironmentScripts contain all PowerShell script.

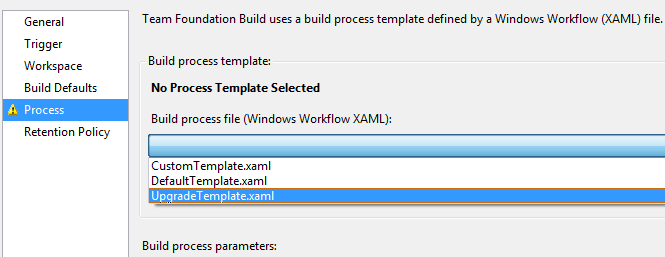




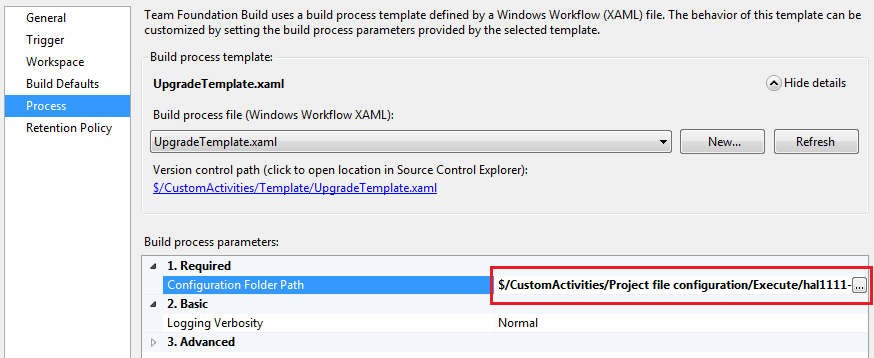
* In Build Defaults tab, enter share folder used to be a drop folder and select default build controller

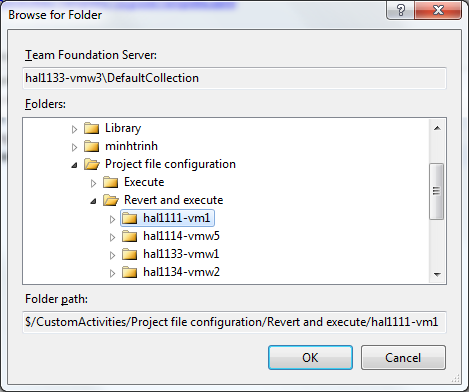


* In Process tab, select Build Process Template is UpgradeTemplate.xaml (if this template is not in your TFS server, add it by clicking on New button)



* At Build Process Parameters, select “Configuration Folder Path” which is a folder contains project file use for build definition.

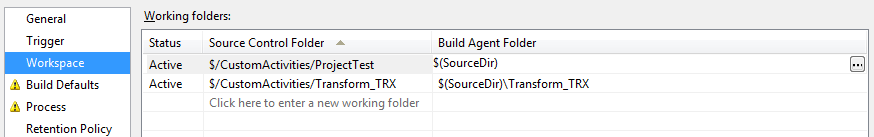


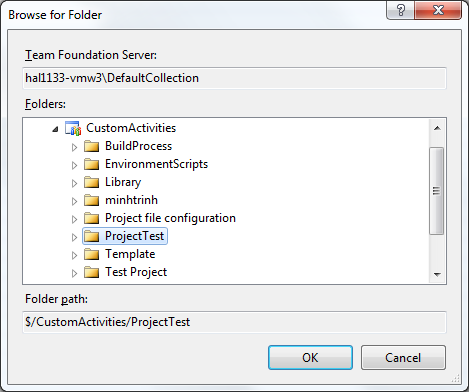


* Save build definition.

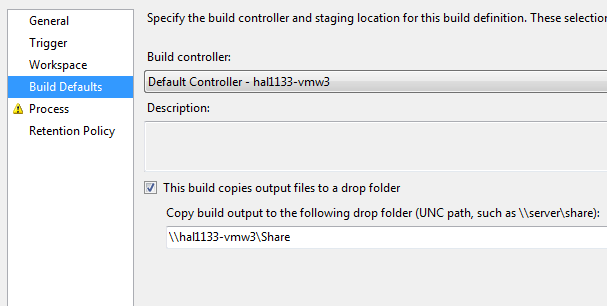
1. **Execution build definition:**

* In General tab, enter build definition name.
* In Workspace tab, select working folder following this structure:
  + Folder TA-MTM project map with $(SourceDir) in Build Agent Folder
  + Folder Transform contain TRXTransform.xslt file which is template to create final result.

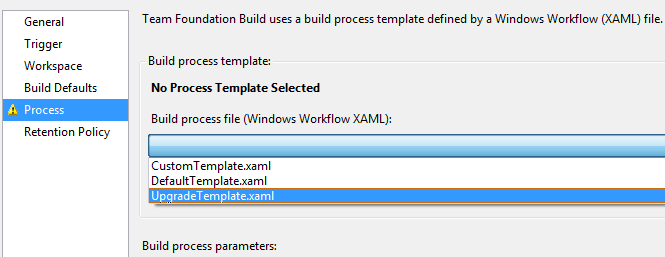




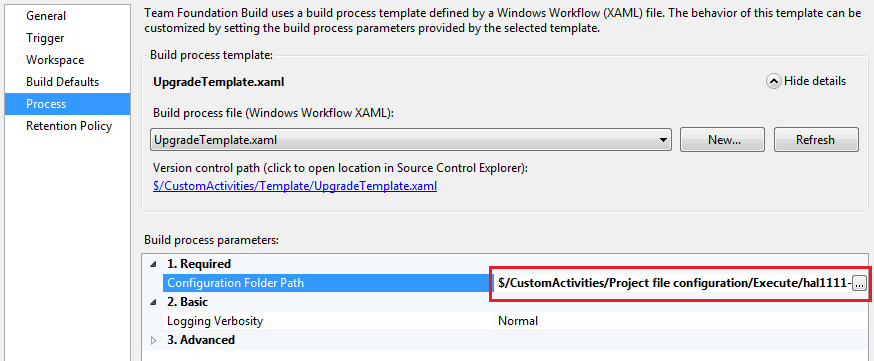
* In Build Defaults tab, enter share folder used to be a drop folder and select default build controller

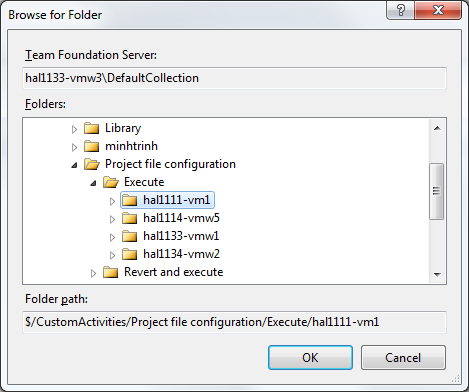


* In Process tab, select Build Process Template is UpgradeTemplate.xaml (if this template is not in your TFS server, add it by clicking on New button)



* At Build Process Parameters, select “Configuration Folder Path” which is a folder contains project file use for build definition.



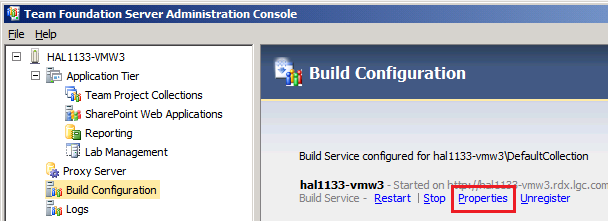


* Save build definition.

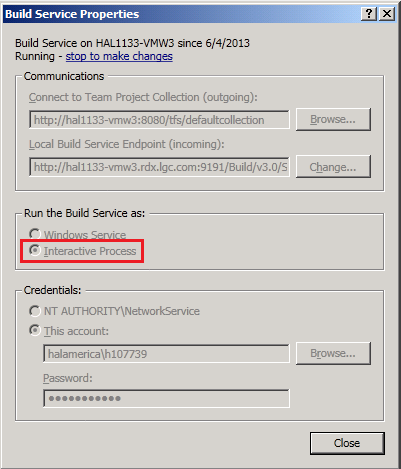
1. **Appendix 1: Configure TFS server**

In TFS server, we have to configure build service to interactive process mode. Following these steps:

* In machine server is used to be TFS server, open Team Foundation Server Administration Console.
* Select Build Configuration.
* Select Properties of build service.



* In Build Service Properties window select Interactive Process (It should require stop build service to change process mode)



1. **Appendix 2: Execute unattended mode**

*When running unattended mode in a first time, it should require enter information in TA Schedule window. You should take a snapshot for test machine after enter these information, or you could create a .ini file for TA Schedule, add in TFS server and use copy command to copy this ini file into test machine after reverting. You could use this method to copy test data into test machine.*