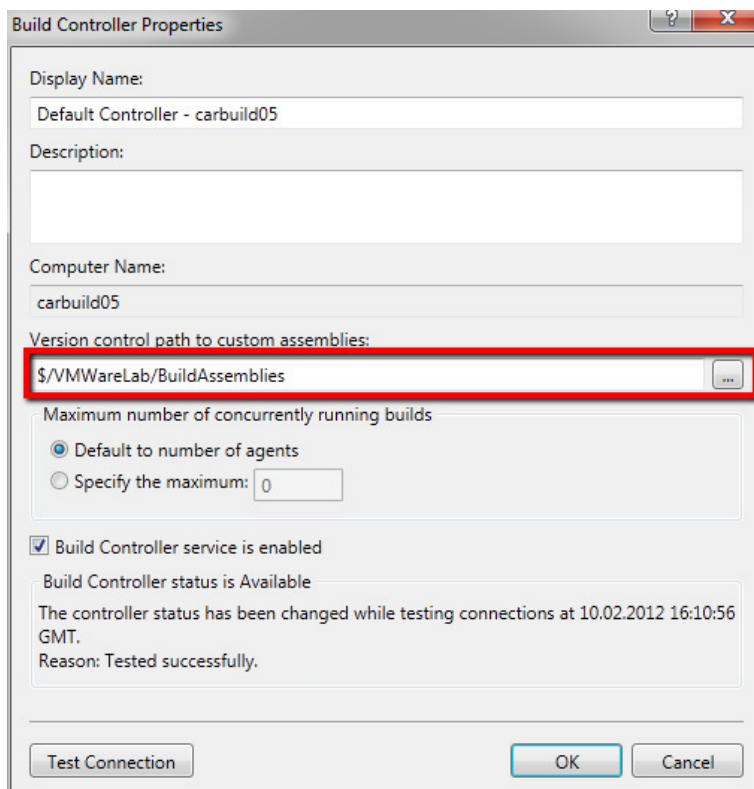


Step by step guide

- 1.) Download the binaries of the VMWare Lab Management project from CodePlex (<http://vmwarelab.codeplex.com>) and extract them to a local folder.
- 2.) Install VMware-PowerCLI SDK 5.0 (or higher) on Buildserver and on the VMware Server or VMware Workstation PC.
(<http://www.vmware.com/support/developer/PowerCLI/index.html>)
- 3.) Set a folder in version control to allow your build controller to read build assemblies from. In Visual Studio go to Build / Manage Build Controllers, select your Build Controller and then click on "Properties". If there is already a folder defined, remember it, otherwise select a folder from your version control structure on TFS.



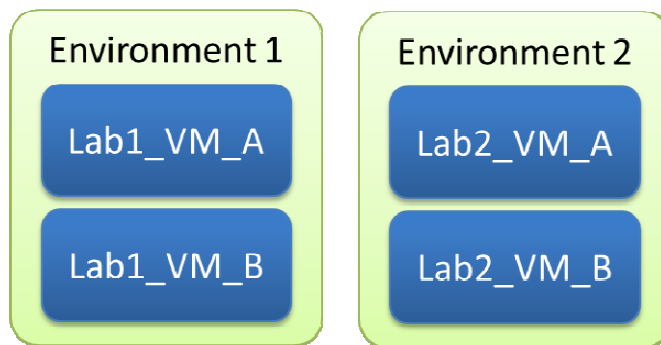
- 4.) Copy the files from the folder BuildAssemblies of the VMWare Lab Management download to the version control folder you have selected in step 2.) and check them in.

VMWareLab	VimService25.XmlSerializers...	Yes	10.02.2012 17:...
BuildAssemblies	VimService40.dll	Yes	10.02.2012 17:...
BuildProcessTemplates	VimService40.XmlSerializers...	Yes	10.02.2012 17:...
CUITDemoApplication	VimService41.dll	Yes	10.02.2012 17:...
	VimService41.XmlSerializers...	Yes	10.02.2012 17:...

- 5.) Copy the VMWareLabDefaultTemplate.xaml file from the subfolder BuildProcessTemplates of the download to the BuildProcessTemplates folder of your team project and check it in.

Preparing test environments

- 1.) Create one or more virtual machines on your VMWare host. The VMWare Lab Management is supported by the following operation systems:
- Windows Server 2008 original release version or later versions
 - Windows Server 2003 SP2 or later versions
 - Windows 7 original release version or later versions
 - Windows Vista SP2 or later versions
 - Windows XP Professional SP3 or later versions



- 2.) Install TFS Build Agent on the machines you want to deploy parts of the application under test to. For more details see <http://msdn.microsoft.com/en-us/library/ms181712.aspx>.
- 3.) Install TFS Test Agent on the machines you want to run automated tests. For more details see <http://msdn.microsoft.com/en-us/library/dd293551.aspx>.

Setting up the BDT build workflow

- 1.) Since the BDT workflow does not compile your application under test nor your test projects, you'll have to setup a compile build for them first if you don't have already one. The simplest scenario is if you have one build compiling your application under test and your test projects in one. For more details see <http://msdn.microsoft.com/en-us/library/ms181716.aspx>.

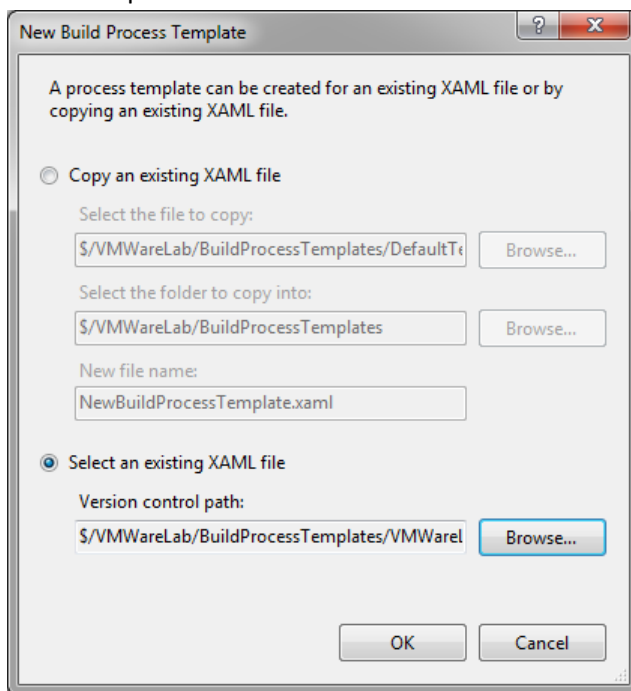
- 2.) Create a new Build Definition by selecting Build / New Build Definition in Visual Studio.
- 3.) Enter a name for the build definition and a description.
- 4.) Select a trigger which will define when your build will be started.

- 5.) The workspace settings can be ignored since we don't use workspaces for the BDT workflow
- 6.) Select your Build Controller and disable the option "This build copies output files to a drop folder" on the "Build Defaults" page.

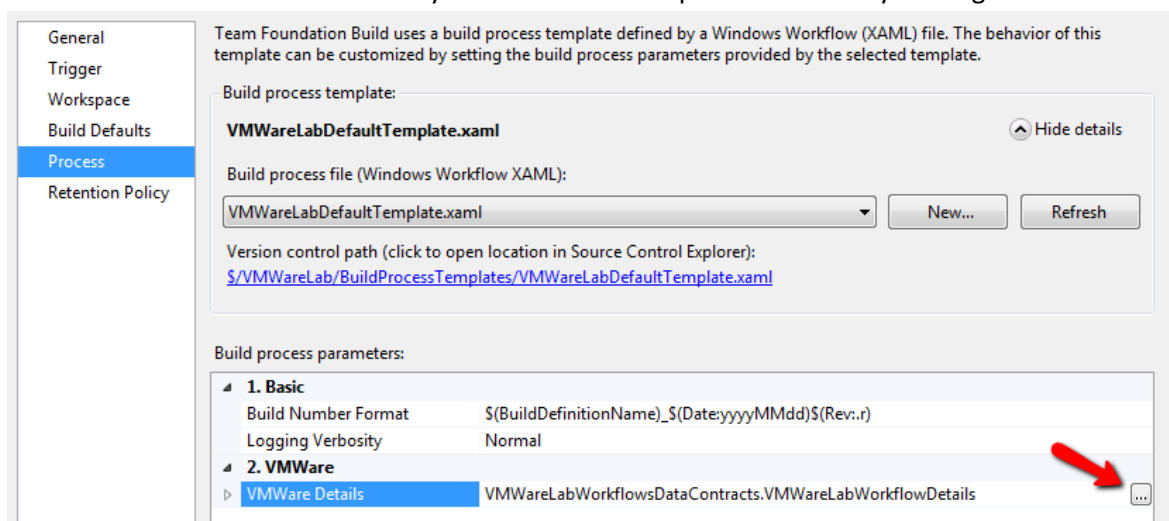
- 7.) On the "Process" page expand the "Build process template" area by clicking the arrow-button next to "Show details"

- 8.) If the VMWareLabDefaultTemplate is not already in the list, click "New..."

- 9.) Select the option “Select an existing XAML file” and then select the VMWareLabDefaultTemplate.xml from the version control path you have copied it to in an earlier step.



- 10.) Now you can set the build process parameters. The most important settings will be the VMWare Details which can be set by a wizard. You can open the wizard by clicking the button



Follow the instructions on the wizard for defining your environment, selecting a build, configuring deployment and pick the tests you want to run.

With this all set, you will be able to run your tests within your VMWare based test environments including revert to snapshot and deployment actions.