A grayscale photograph of a laboratory setting. In the center is a compound microscope. To its left are two petri dishes, one of which contains a liquid sample. Below the microscope are two surgical instruments, possibly scalpels or forceps. In the background, there is a notebook or ledger with some writing on it.

Team Foundation Server Lab Management Guide

Visual Studio ALM Rangers

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft grants you a license to this document under the terms of the Creative Commons Attribution 3.0 License. All other rights are reserved.

© 2013 – 2014 Microsoft Corporation.

Microsoft, Active Directory, Excel, Internet Explorer, SQL Server, Visual Studio, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

Foreword	4
Introduction	5
Planning, Setting Up and Configuring Test Lab Environments	6
Identifying Test Environments	6
Environment Awareness	8
Define the Test Approach	9
Preparing the Test Environment	9
Creating the Test Environment	10
Cloning Environments	10
Testing	14
Build, Deploy and Test Workflow	18
How to configure various forms of diagnostic data collection	20
Understand how to use a multi-machine environment for debugging	20
Upgrading Lab Management Environments	22
Test Controllers and Agents	22
System Center Support	22
Upgrading System Center Virtual Machine Manager	24
Maintaining Lab Management Environments	26
Back Up and Restore	27
Virtual Machine Patching	31
Lab Management Infrastructure Planning	35
Hyper-V Servers	36
Library Servers	38
SCVMM Server	38
Team Foundation Server	39
Test and Build Controllers	39
Planning Checklist	40
Automating the Build-Deploy-Test Process	42
Enhanced Lab Template	44
Parallel Lab Template	45
Other Deployment Solutions	53
Support for VM-Ware Environments	54
Support for Cloud Environments	57
Hands-on Lab	60
Lab Management Standard Environment using Azure IaaS	60
Appendix	65

Foreword

In order to increase their release cadence, software delivery organizations are looking for ways to rapidly deploy and test their applications. Visual Studio Lab Management puts that power in the hands of the software development teams by providing them with tools to easily create test environments and to run automated tests on them. It allows teams to setup post-build automation in the form of continuous deployment of builds to a test environment.

A typical usage of Visual Studio Lab Management brings together many different technologies for a software team - platform virtualization, virtual machine management, application deployment, build integration, and automated testing. Naturally, setting up an effective automated end-to-end process can be time consuming. Since its inception in Visual Studio 2010, Lab Management has been on a journey to make it simpler for teams to easily set up such automated processes. At the same time, it has been our goal to help teams leverage the advancements in platform virtualization.

The Lab Management Rangers Guide is a great way for teams - new and old to Visual Studio Lab Management - to understand how to set up automated build-deploy-test processes easily. Unlike MSDN documentation, this guide focuses on all aspects of Lab Management including the System Center Virtual Machine Manager and Windows Hyper-V. Based on feedback from the earlier versions of the Rangers Guide, this version has been condensed significantly making it easy to read without compromising on the quality of guidance. The guide specially focuses on the improvements made in Visual Studio Lab Management 2012 and 2013. The default process templates that ship with Visual Studio Lab Management are not always sufficient to meet the customer's requirements. The Rangers guide has a great example on a customized lab process template that helps customers build their own templates.

We hope that this guidance will help you in improving your software processes, and we welcome your feedback on how to improve it.

Vijay Machiraju - Principal Program Manager
Visual Studio ALM

Introduction

Welcome to the 3rd edition of the Lab Management Guide. This release includes the following changes:

- Content updated to reflect Team Foundation Server 2012 and 2013.
- Content includes information on how to upgrade lab management environments.
- New content on how to leverage Windows Azure IaaS services for lab management.
- Significantly reduced the page count of the guide by removing outdated content.

Intended audience

In this guide, we primarily target the Microsoft “200-300 level” administrators and users of Team Foundation Server. The target group is considered as intermediate to advanced users of Team Foundation Server and has in-depth understanding of the product features in a real-world environment. Parts of this guide may be useful to the Team Foundation Server novices and experts but users at these skill levels are not the focus of this content.

See [ALM Rangers Personas and Customer Profiles](#)¹ for more information on these and other personas.

Visual Studio ALM Rangers

The Visual Studio ALM Rangers are a special group with members from the Visual Studio Product group, Microsoft Services, Microsoft Most Valuable Professionals (MVP) and Visual Studio Community Leads. Their mission is to provide out-of-band solutions to missing features and guidance. A growing Rangers Index is available [online](#)².

Contributors

Richard Fennell, Mathias Olausson, Brian Minisi, Dave McKinstry, John Bergman, Mike Douglas, Mike Fourie, Vijay Machiraju, Willy-Peter Schaub.

Using the sample source code, Errata and support

All source code in this guide is available for download via the [Visual Studio Lab Management Guide](#) home page.

Additional ALM Rangers Resources

Understanding the ALM Rangers – <http://aka.ms/vsarunderstand>

Visual Studio ALM Ranger Solutions – <http://aka.ms/vsarsolutions>

¹ <http://vsarguidance.codeplex.com/releases/view/88001>

² <http://aka.ms/vsarindex>

Planning, Setting Up and Configuring Test Lab Environments

Identifying Test Environments

A typical software development team requires and works with different environments during the course of their project.

Environment	Description
Local	Developer's desktop
Virtual Machine	VM ³ hosted on developers desktop or possibly development server
Development	Development server (also known as sandbox)
Integration	CI ⁴ build target, or for developer testing of side effects
Test/QA	For functional ⁵ , performance testing ⁶ , Quality Assurance ⁷ etc.
UAT	User acceptance testing ⁸
Staging/Pre-production	Mirror of production environment
Production/Live	Serves end-users/clients

Table 1 – Environment

Among these environments, the integration, test, and staging environments are more representative of how the application will be deployed in production. Lab Management helps to create such environments. An environment in Lab Management is a collection of virtual machines, with each virtual machine serving a specific role, such as web, database, client, etc. Each role could be running on one or more computer. Multiple agents are installed on

³ http://en.wikipedia.org/wiki/Virtual_machine

⁴ http://en.wikipedia.org/wiki/Continuous_integration

⁵ http://en.wikipedia.org/wiki/Software_testing

⁶ http://en.wikipedia.org/wiki/Software_performance_testing

⁷ http://en.wikipedia.org/wiki/Software_quality_assurance

⁸ http://en.wikipedia.org/wiki/User_acceptance_testing

each of the virtual machines, and these agents help to run tests, collect diagnostic data, deploy a build, or make an environment safe for cloning.

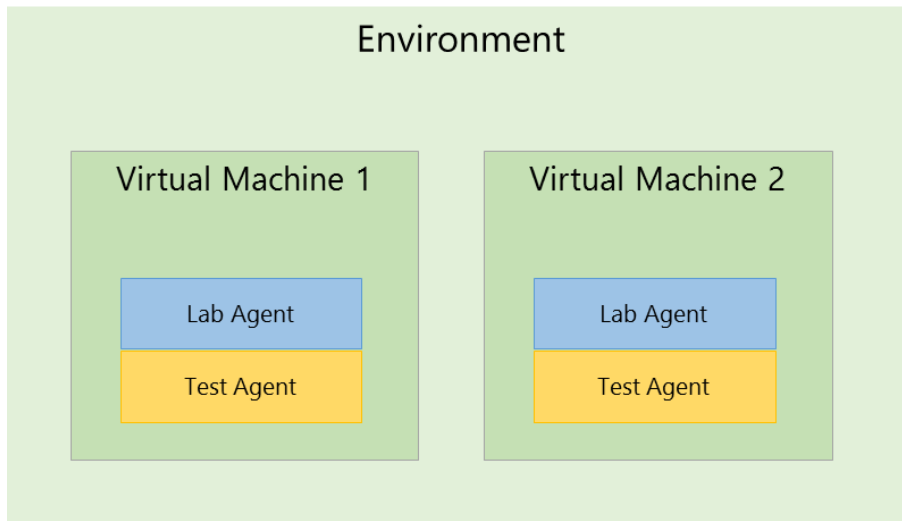


Figure 1 – Lab Management Logical Environment

If you are a team member who is responsible for creating test environments, you need to understand the following process for starting from a production-like environment to deploying a build onto it.

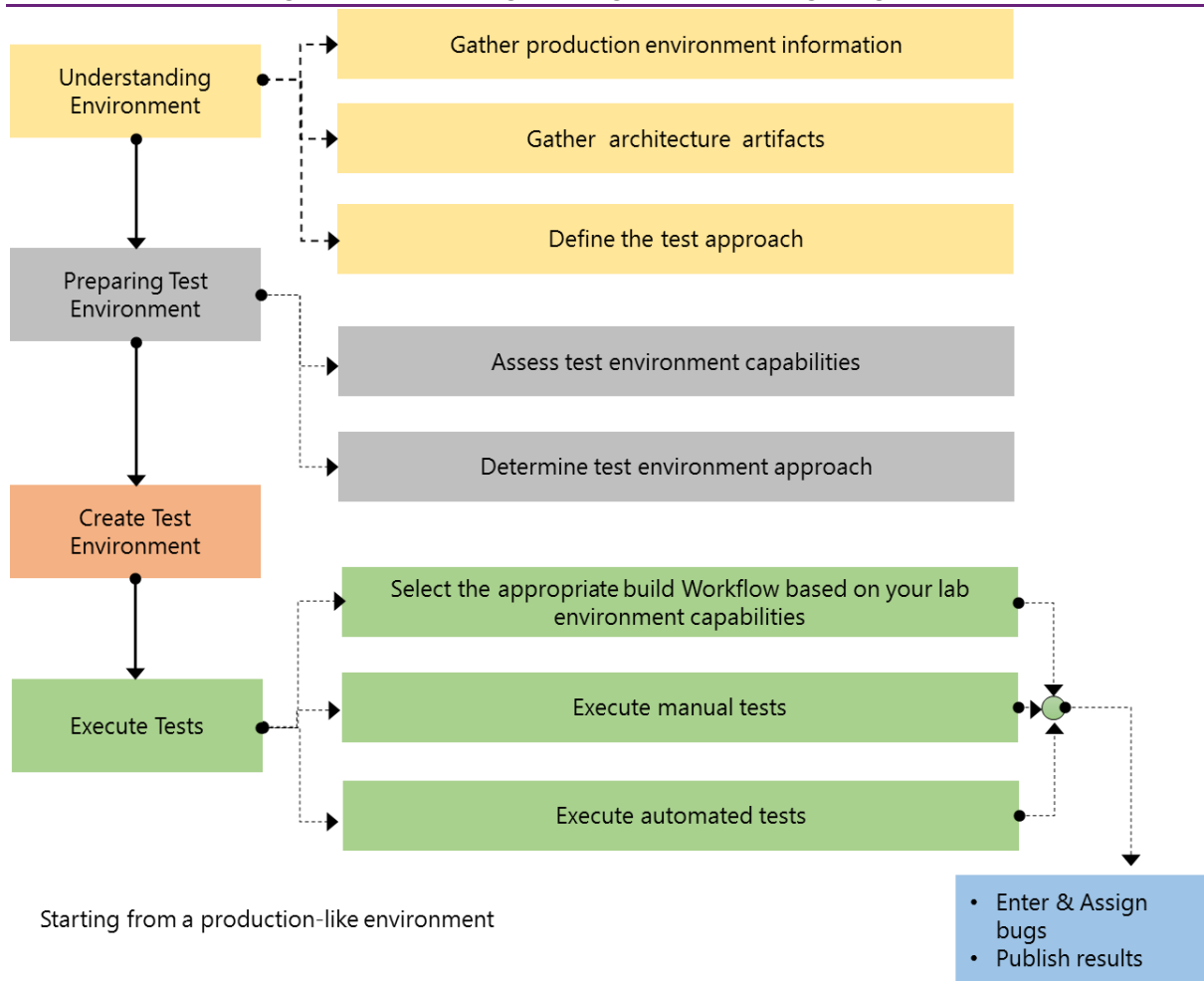


Figure 2 - Lab creation process

Environment Awareness

The first step of the process for starting from a production-like environment is to understand the production context, application architecture deployment, and the test approach.

Gather production environment information

Obtaining information about the target production environment from your IT department is very important if you want to create a test production-like environment. SCVMM allows you to move physical computers to virtual machines if you want to avoid rebuilding the operating system and reinstalling applications. Depending on the following criteria, you can determine the P2V (online or offline) or V2V conversion.

- Physical server topology
- Operating system version and service pack
- Server type (physical, virtual)
- Server role
- Lab Management capacity planning

Define the Test Approach

After you determine the production environment information and the application architecture, you must define your test approach. First, determine how many servers you will require in your test environment to meet your test mission. Base this number on how many test cases you have to run. Next, based on the testing techniques that you will use in the execution test step, you can determine the test type and the test components that you will need in your environment.

- Obtain the following information from your test team.
- Determine the test mission
- Assess possible testing techniques
- Define the test metrics
- Define the test plan from the requirements
- Determine the test type (manual or automated)
- Determine if you need to test in parallel
- Identify the test area and environment
- Identify the test case flow details

Preparing the Test Environment

Creating Standard Environments using existing test rigs

If your project has existing test rigs then these can easily be integrated into Lab Manager and used for testing. This is done by adding the existing servers into Standard Environments. Lab Manager will then install the Test Agent on all the servers and allow all Lab Management features to be used on these rigs.

To create a standard environment, follow the steps in [Creating a Standard Environment](#) on MSDN.

Assess test environment capabilities

The test lab environment is dependent on the application types and your choice of consolidating the server and client test environments. There is no recommendation for the golden templates that are used to run the test lab environment. However, you should consider the following criteria:

- The test environment should resemble the production environment as closely as possible.
 - When you work with a Web Application or Client Server test lab environment, the environment should be split into a server VM and a client VM. In other words, you should define and implement a server and a separate client golden template, which are hosted as separate virtual environments.
 - When working with server or client environments that will be supported and maintained on different platforms or operating systems, each environment should be defined and implemented in a separate golden template. For example, when you are developing and testing a solution that has to run on Windows XP, Windows 7, and Windows Server 2008 R2, you would define and implement three different templates, which are hosted as separate virtual environments.

Determine the test environment approach

Before you create an environment, you need to create some basic assets in SCVMM. These assets will serve as the starting point for creating environments. In particular, you have to create either virtual machines or virtual machine templates. In SCVMM, you can create new virtual machines from any of the following production sources:

- Existing virtual production hard disks (VHDs)
- Existing production virtual machines
 - Hyper-V virtual machines

- Virtual Server virtual machines
- VMware virtual machines
- Existing physical computers

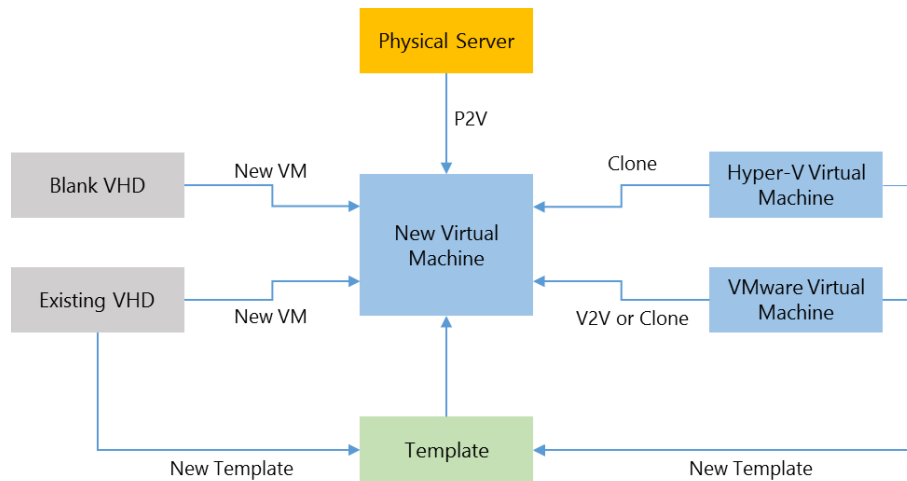


Figure 3 - Virtual Machine architecture

Your choice of virtual machine sources will depend largely on your needs and existing infrastructure.

- **Existing Hyper-V or Virtual Server Virtual Machine**
Used when you want to create a clone of a virtual machine for testing, UAT, or Stage/Preproduction Environment.
- **VMware to Virtual Machines Conversion**
See this conversion approach and considerations in [V2V: Converting Virtual Machines in VMM](#)⁹.
- **Physical to Virtual Machines Conversion**
See this conversion approach and considerations in [P2V: Converting Physical Computers to Virtual Machines in VMM](#)¹⁰.

Creating the Test Environment

There are two ways to create an environment:

- New virtual environment
- Composed virtual environment

To create a new virtual environment (or also referred to as SCVMM environment), follow the steps in [Creating an SCVMM Environment Using Stored Virtual Machines and Templates](#)¹¹ on MSDN.

Cloning Environments

Lab Management enables you to run two exact replicas, or clones, of an environment simultaneously by using a feature named network isolation. Network isolation is one of the capabilities that are available for every environment, and after this capability is enabled, you can create clones of the environment without network conflicts.

⁹ <http://technet.microsoft.com/en-us/library/cc793147.aspx>

¹⁰ <http://technet.microsoft.com/en-us/library/cc764232.aspx>

¹¹ <http://msdn.microsoft.com/en-us/library/ee518915.aspx>

Creating a clone of an environment can be useful in the following situations:

- You have set up an environment by installing the pre-requisites for your application on each virtual machine. From that point forward, you just want to clone that environment every time you install a build of the application. This saves you time because you do not have to re-install the pre-requisites repeatedly.
- You have set up an environment with the latest build of your application. From that point forward, you want all the testers and developers to clone this environment. This saves time for each team member and ensures consistency of the build.
- The system administrator performed a physical-to-virtual machine conversion of the production environment to create the virtual machines. You then created a copy of the environment that mimics the entire production setup, using these virtual machines. All the development and testing effort in the team create clones of this virtual environment. This simplifies the installation and configuration of your application production-like machines.
- A tester has found several bugs in your application and has used snapshots in an environment to capture those bugs. Creating exact clones of that environment lets developers work on the bugs while the tester continues testing.

To create a clone of a stored environment:

- Make sure that the network isolation capability is enabled in the stored environment.
- Deploy as many copies of the stored environment as you need.
- When creating a clone of an environment, ensure you have enough available disk space to hold copies of all associated virtual machines

To create a clone of an active environment:

- Create an active environment from virtual machines or templates and with network isolation capability enabled.
- Make sure that the environment is configured for your needs, and then shut down the environment.
- Store the environment in the library. The active environment remains in its original condition.
- After the environment is stored, deploy as many copies of it as you need.

If you create multiple clones of an environment using the store and deploy operations but without network isolation, there can be several types of network conflicts:

- **Computer name conflicts:** Multiple machines on a common network will have the same computer name. Consequently, only one of the machines will have network connectivity. Depending on the software installed on the individual computers, often times renaming computers can be non trivial.
- **Conflicts in Active Directory:** Multiple computers that use the same identity have one entry in the Active Directory. Those virtual machines might not be able to access domain resources.
- **Conflicts in DNS:** Multiple machines with the same machine name override each other's entries in DNS. As a result, attempts to connect to virtual machines, for example, from Environment Viewer, could take you to the wrong machines.

Network isolation helps prevent these conflicts by creating an internal network between the virtual machines of an environment. Lab Management generates a unique alias name for each virtual machine and registers it in DNS. Each virtual machine in a network isolated environment will have the following:

- Two network connections: one to an internal network for communication within the environment, and one to an external network to communicate with machines outside the environment in the lab network.
- Two IP addresses: an IP address on the internal network assigned by Lab Management, and one on the external network assigned by DHCP server of your lab network.
- Two machine names: the original machine name for communication within the environment, and the alias name for communication outside.

Lab Management - Planning, Setting Up and Configuring Test Lab Environments

The computer name, MAC address, and IP address of a virtual machine in an environment are the same for each clone on the internal network. Application components that run within the environment can use the same computer name to refer to each other in every clone. This makes them seem like exact replicas. The alias name, MAC address, and IP address of a virtual machine are different for each clone on the external network. This allows the co-existence of clones and connectivity from external computers.

NOTE

Network isolation is a misnomer because it does not really mean that the environment is not reachable from computers outside the environment. On the contrary, computers outside the environment can communicate with network isolated virtual machines by using the external IP addresses or alias names.

There are also other ways to prevent network conflicts, although without the ability to create clones:

- Using templates while you are creating environments.
- Fixing the conflicts manually by changing the computer name, SID, and so forth.

For more information about when to use network isolation in comparison with these other techniques, refer to the following tables.

When to Use Network Isolation

You can use the scenarios in the following table to help you determine when to use network isolation.

Scenario	Recommendation
Each time that you set up a new environment, you set it up from scratch by reinstalling the operating system in virtual machines, the application prerequisites, and then the application build. When bugs are found in the application, they are captured through snapshots of the environment. Developers and testers share the same environment in which bugs are found, instead of creating clones.	You do not need to use the network isolation capability. You do not need virtual machine templates.
Your library contains virtual machine templates that are generalized operating system images. Each time that you set up a new environment, you customize these templates. You install application pre-requisites and application builds manually or through automation in each environment. When bugs are found in the application, they are captured through snapshots of the environment. Developers and testers share the environment in which bugs are found, instead of creating clones.	You do not need to use the network isolation capability. You can use virtual machine templates to create stored or active environments.
Your application pre-requisites are included in the virtual machine templates together with the OS. These pre-requisites are not affected by customization of the operating system or changes in the computer name. Alternatively, you run a script to re-configure the pre-requisites and application when the operating system is customized. Examples of such applications that are tolerant to change in computer name are IIS and SQL Server, which requires a script. When bugs are found in the application, testers capture them through snapshots of the environment. Developers and testers share the environment in which bugs are found, instead of creating clones.	You do not need to use the network isolation capability. You can use virtual machine templates to create stored or active environments.
Either your application pre-requisites or your application breaks whenever the operating system is customized. Alternatively, the amount of re-configuration needed for your application after customization is significant. Therefore, you use virtual machines instead of templates with all the pre-requisites installed, and you create exact replicas of these.	You should use the network isolation capability.

Lab Management - Planning, Setting Up and Configuring Test Lab Environments

Scenario	Recommendation
Developers and testers work in parallel. As testers find bugs, the testers take snapshots of the environments to capture those bugs. The testers then store these environments in the library so developers use the environments to analyze and fix the bugs, while the testers continue to work on their own environments.	You should use the network isolation capability.
Your application has to be developed on virtual machines that mimic the production environment. Each developer and each tester must have the same environment.	You should use the network isolation capability.

Table 2 – Network Isolation scenarios

The following table summarizes the additional limitations and requirements for using the network isolation capability.

Scenario	Recommendation
Your application is a workgroup-based application.	You can use the network isolation capability.
Your application requires domain-joined computers. You can set up a virtual machine with Active Directory and DNS server roles and use that as the domain controller within an environment. All computers within your environment are joined to the local domain that is served by the domain controller within the environment. Each time that you clone the environment, you are also cloning the domain controller virtual machine.	You can use the network isolation capability.
Your application requires domain-joined computers. Only the domain controller that is on your company's network can be used because it has all the necessary accounts and data. You cannot include a similar domain controller virtual machine within the environment.	Do not use the network isolation capability.
Your application modifies several external components outside the environment. For example, the application writes to a database that is not hosted on a virtual machine in the environment.	Do not use the network isolation capability.
Your application does not work on computers that have multiple network adapters.	Do not use the network isolation capability.
Your application requires virtual machines which do not run one of the following operating systems: Windows XP SP3 and later versions, Windows Server 2003, Windows Server 2008, Windows Vista, Windows 7, Windows Server 2008 R2.	Do not use the network isolation capability.
All the virtual machines in a network-isolated environment must fit on a single host. You do not have any Hyper-V host that can fit all the virtual machines of the environment.	Do not use the network isolation capability.

Table 3 – Further Network Isolation scenarios

Before using network isolation in an environment, make sure that the following conditions are true:

- The virtual machines that will be used to compose the environment have Lab Management agent installed on them.
- The system administrator has configured network isolation parameters in Team Foundation Server.

Testing

You can now be more productive throughout your testing lifecycle of planning, testing, and tracking your progress by using Visual Studio Ultimate or Visual Studio Test Professional. These testing tools integrate with Team Foundation Server, which lets you define your testing based on the same team projects that other areas of your organization are using.

Both Microsoft Visual Studio Ultimate and Visual Studio Test Professional the Microsoft Test Manager to help you define and manage your testing effort by using test plans.

Manual Test

A key step in the process of ALM is testing your software application. To ensure correct behavior of the software, and ultimately improve its quality, some of the testing requires a tester to play the role of an end user, and use most of the features of an application. To test the application completely, the test lead must write test plans to guide testers through a set of test cases. These test plans have both action and validation test steps. Based on the results of the test cases, the testers can write manual test cases with Visual Studio Ultimate or Visual Studio Test Professional. Manual testing is the process of validating that the result of the test execution is equal to an expected result.

Exploratory Testing with Lab Management Environments

In MTM 2012 and 2013, Exploratory testing was introduced as a first-class test model (as an alternative to scripted manual tests or automated tests). An Exploratory test session may be connected to a test case but does not have to be. The image below shows the Exploratory Testing feature in MTM and how it can be run with a Lab Management environment:

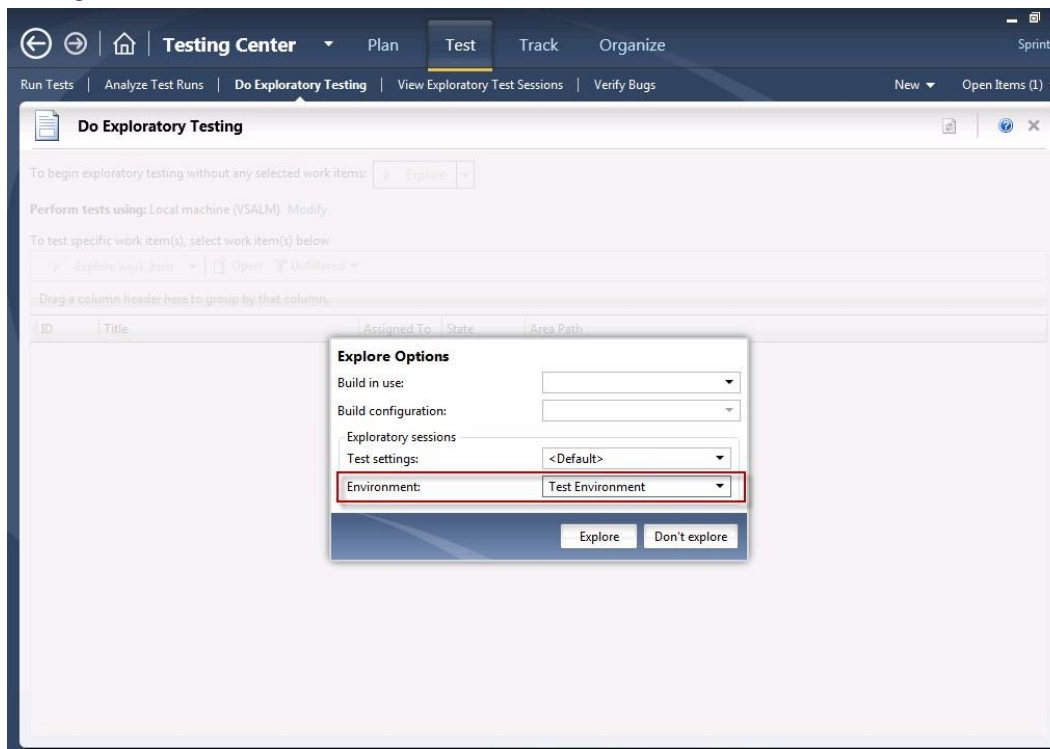


Figure 4 - Selecting an environment in MTM

Creating Bugs with Lab Management Environments

In addition, when you are running your manual tests with Test Runner on virtual environments, you can create a bug that includes a link to a snapshot of your virtual environment when the bug occurred. The developer can

open the bug and use the link to connect to the snapshot of this environment and see the actual state of the environment at that point in time. This state of the environment is saved and developers can use it at their convenience. Meanwhile, you can continue to run more tests on this same environment.

This greatly reduces the time it takes for the developer to reproduce the bug.

It is recommended that you run your manual tests on a local machine that is not part of the environment. You can collect data or affect a test machine for your manual tests in the following ways:

- Collect data on a local machine using default test settings
- Collect data on a local machine specifying the data to collect
- Collect data on local and remote tiers of your application

For more information about how to set up your test settings and environments for these situations, see [Specifying Test Settings in Microsoft Test Manager](#)¹² on MSDN.

If desired, Manual tests can be run in a Lab Environment by installing Microsoft Test Manager inside the virtual machines that compose the Lab Environment. With this approach you can use automatically deployed environments without doing anything before you start your manual testing and if using virtual environment and SCVMM you can also use the snapshot capability.

NOTE

You must consider that running manual and unit tests is supported on physical and both types of virtual environments, Hyper-V and non-Hyper-V. How to create manual test definition and run it is covered in the ASP.NET Manual and Automated Testing HOL. When you use Microsoft Test Manager to create test plans and test suites for your team project, there are several approaches that you can take. This topic is explained in the following MSDN page: [Guidance for Creating Test Plans and Test Suites](#).

Automated Test

Test automation reduces the cost of testing. A computer can follow a root sequence of steps more quickly than a person, and it can run the tests overnight to present the results in the morning. A Build Verification Test can be developed and incorporated into the Continuous Integration tests to help increase immediate feedback during active development.

The following table can help you to identify what type of test you can run from Visual Studio 2012:

Type of test	Description	Run tests from Visual Studio	Run from a test plan by associating a test method with test case
Coded UI Tests	Tests the user interface by performing UI actions.	Yes	Yes
Unit Tests	Tests code at the method level.	Yes	Yes
Database Unit Tests	Tests a stored procedure, function or trigger in a database.	Yes	Yes
Load tests	Tests application performance and stress using unit tests, Web Performance tests or coded UI tests.	Yes	Not recommended because you cannot view the test as it runs.
Generic Tests	Test functionality using API calls or command line tools for the application under test.	Yes	Yes
Web Performance	Used as part of load tests to test server responses for Web applications by sending and receiving http requests.	Yes	Not recommended because Web

¹² <http://msdn.microsoft.com/en-us/library/ee231892.aspx>

Lab Management - Planning, Setting Up and Configuring Test Lab Environments

Type of test	Description	Run tests from Visual Studio	Run from a test plan by associating a test method with test case
Tests	Note: These tests cannot be used to test the UI.		Performance tests are primarily intended to be used with load tests.

Table 4 – Types of tests

Running Automated Tests

- Unit tests
- Coded UI tests
- Database unit tests
- Load tests
- Generic tests

Many methods of running your automated tests are available, depending on how you want to run tests and view the results. If you run your automated tests by using a test plan, you can view your testing progress and easily rerun your tests as required. To run your automated tests by using a test plan, you associate your automated tests with test cases and run these test cases by using Microsoft Test Manager. To run your automated tests in this manner, you must create a physical or virtual environment to use when you run your tests. For more information about how to create virtual environments to use to run your tests, see <http://msdn.microsoft.com/en-us/library/dd997438.aspx>¹³.

NOTE

You must consider that running coded UI and other automated tests is supported on physical and both types of virtual environments, Hyper-V and non-Hyper-V.

Connecting Tests to a Build

The simplest way to connect tests to a build is to run unit tests in the default lab build. The following describes how to setup and run a Build-Deploy-Test definition.

1. Create a Test settings.
 - a. From Microsoft Test Manager, go to Lab Center, and to Test Settings.
 - b. Create a new automated test settings. On the "General" tab, type in the name of the test settings - for e.g., Fabrikam BDT Test Settings. Select "Automated". On the rest of the tabs, choose the defaults, and save the test settings.

¹³ <http://msdn.microsoft.com/en-us/library/dd997438.aspx>

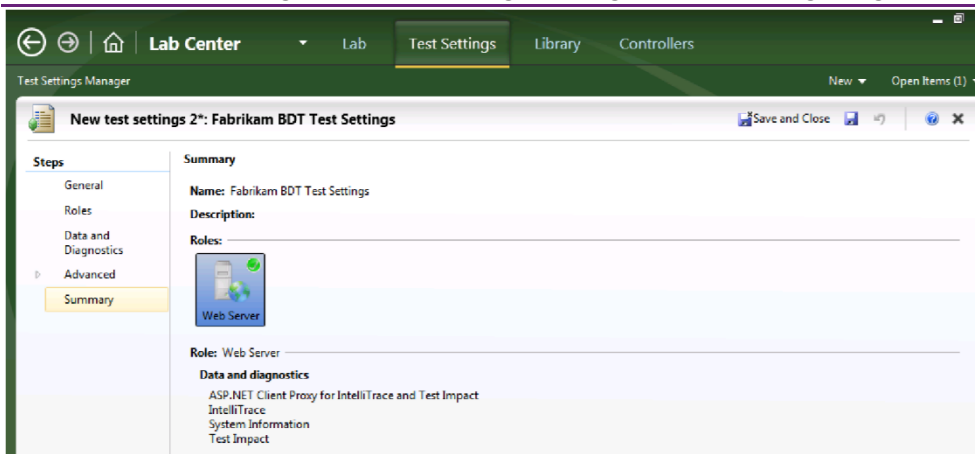


Figure 5 - Test Settings in MTM

2. Create a Build-Deploy-Test definition.
 - a. In Visual Studio, go to Team Explorer, and open Builds. Select "New Build Definition".
 - b. On the "General" tab, type in a name for the definition - say, BDT.
 - c. On the "Build Defaults" tab, uncheck "This build copies output files to a drop folder".
 - d. On the "Process" tab, select LabDefaultTemplate.11.xaml as the build process template.

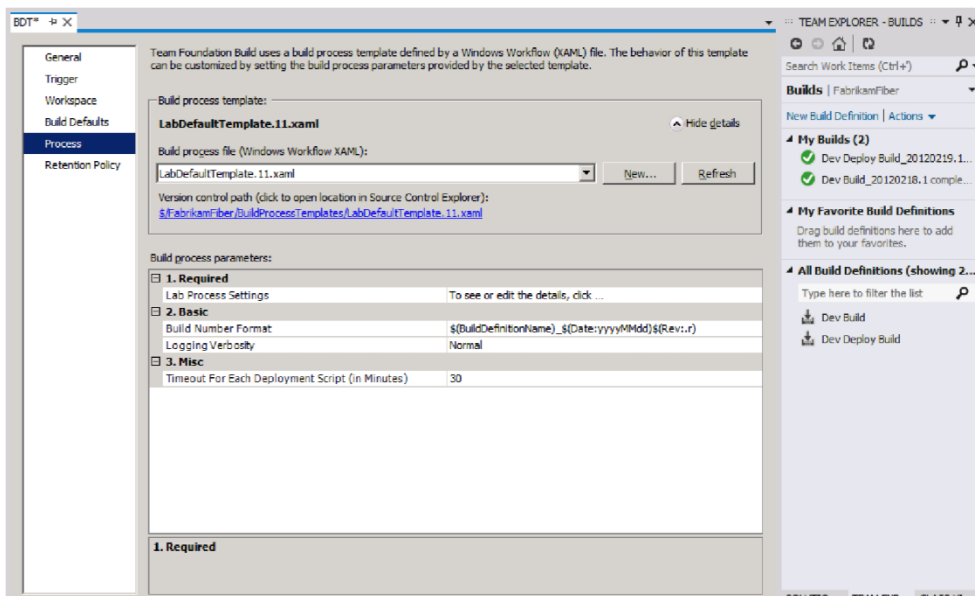


Figure 6 - Lab Build Definition

- e. Click "..." next to the Lab Process Settings, and enter the following settings:
 - i. Environment: The name that you gave to your environment
 - ii. Build: Set the build definition to the name you gave to the compilation build
 - iii. Deploy: Check the option to Deploy, and Add a row in the grid. For the Machine role, select "Web Server". Ensure you also enter a valid deployment script to deploy the application into the environment.
 - iv. Test: Select the automated test plan and the test settings that you created.
 - v. Save the build-deploy-test definition,
3. Queue an instance of the Build-Deploy-Test definition.
 - a. Select the build definition that you just saved, and queue it. This should trigger a compilation, followed by deployment, and then the execution of the test cases.

This is very useful to check the quality of units of code. But for testing how the code will run on the real system, it is not very realistic. The unit tests are run on the build server, which has probably got all necessary versions of frameworks and SDK's installed, which might not be present on the server where you will run your application. Also for test UI it is not a very good option. Assume that you are running a Coded UI test in your continuous integration build while trying to fix an issue on your web page. The test opens a browser and navigates to a page where it fails before the test has closed the browser. Every time you check in a file, it will start a new instance of Internet Explorer on your build server. You will end up having hundreds of Internet Explorer instances running, which decreases the build server's performance significantly.

What you will need to do is connect your tests to a lab environment with the correct type of servers. You start by creating a build that is identical to the one shown in the previous illustration. However this time, provide the computer name of the Test Machine. In the wizard, select the 'Advanced' tab, then select the option to run Coded UI Tests. This will cause the test agent to be run using the provided account and authenticate successfully with the test controller.

Running a Test in Visual Studio

To run a test in Visual Studio, executing the test locally, you can go into the code and right-click the test method. This will run the specific test

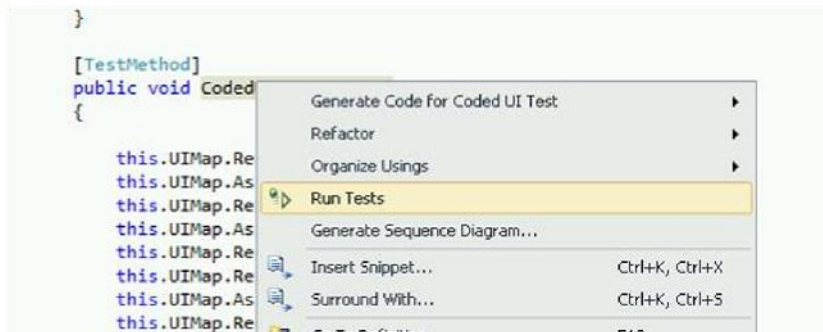


Figure 7 - Running tests from Visual Studio

If you want to run multiple tests, or you do not want to look through the code to find it, you can go to the test list editor. Click **Test, Windows** and then select **Test Explorer**.

Build, Deploy and Test Workflow

After you understand that Visual Studio can create both types of tests, manual and automatic, it is important to understand the workflows that are related to both. You can determine the best approach to use, based on your needs and test cases.

Traditional Build Workflow

The first workflow is the traditional build workflow used in most of cases for manual tests. In this workflow, you build and deploy your application separately. You must consider that the tasks of build and deploy can be manual or automated on either a virtual or a physical environment. Another important consideration of this type of workflow is that you can collect data if you have the appropriate test components in the tests machines.

Lab Management - Planning, Setting Up and Configuring Test Lab Environments

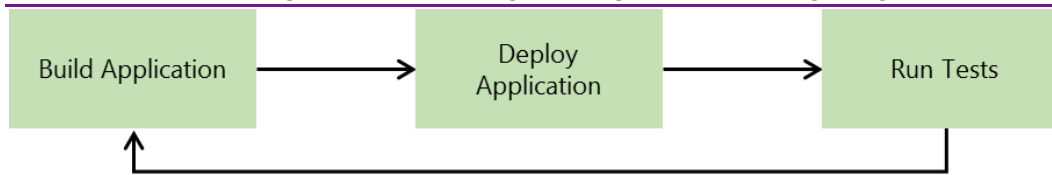


Figure 8 - Traditional build workflow

Lab Build Workflow

If you have Lab Management and Team Foundation Server 2012, you can automate the build, deploy, and test workflow. You can create lab environments that support both test and build workflows, which helps you automate repetitive manual tests. Automating these tests reduces the cost and time of the traditional build workflow

NOTE

Refer to the Hands-on Labs Manual and Automated Testing and ASP.NET Manual and Automated Testing for more information and a practical walkthrough.

To deploy your build and run automated tests on your virtual environment, you need to enable the workflow capability. In other words, you can run a build-deploy-test workflow in an environment where the workflow capability is ready.

We use the test agent/test controller framework to run the deployment workflows on the environments. The process of creating a test environment in Lab manager deploys the agent into your virtual machines.

To get the workflow capability working, you need to create either a Standard or SCVMM environment to run the tests. The process of creating these environments ensures the test agent is deployed correctly onto the servers to allow the tests to run.

For Team Foundation Server 2010 you also need one or more build controllers set up and registered with the team project collection you are connected to. The build controller is outside the virtual environment. In Team Foundation Server 2012 and 2013, the deployment workflow is handled by the test controller.

To understand where to set up the build controller, you can review the 'Choose a deployment topology' section.

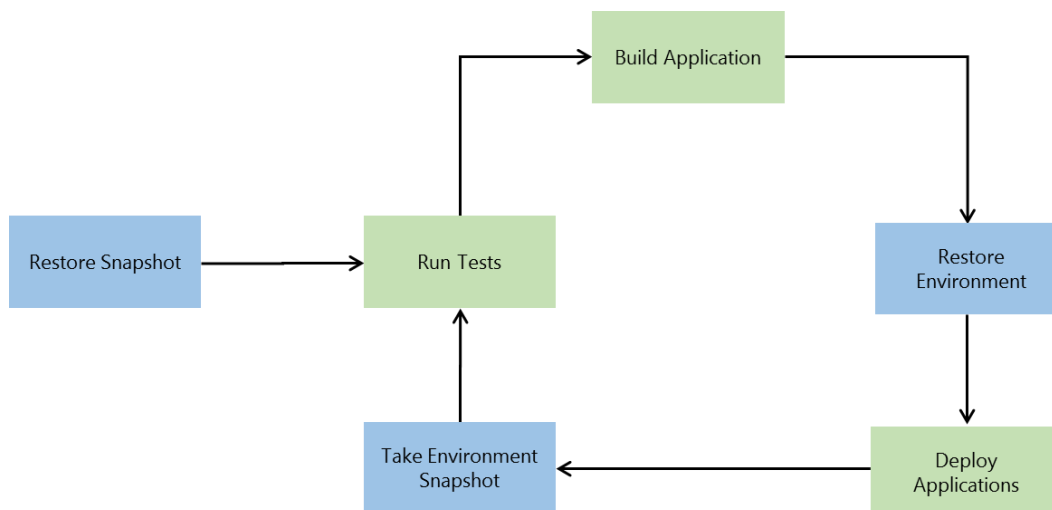


Figure 9 - Lab build workflow

After you have your environment set up with workflow capability, review the documentation to learn how to [create a build-deploy-test workflow](#)¹⁴ and [run automated test](#)¹⁵.

How to configure various forms of diagnostic data collection

Diagnostic Data Adapters

Diagnostic Data Adapters provide functionality to allow the capture of information relating to how the environments react during testing.

Data Adapter	Description
Actions	Use to collect each UI action you perform as you run a test (for client roles only)
ASP.NET client proxy for IntelliTrace and test impact	Use for web applications when you select IntelliTrace or test impact for a server role (use on client to web server)
Eventlog	Use to capture event log data (for client or server roles)
IntelliTrace	Use to collect exceptions and specific diagnostic tracing information to help isolate bugs that are difficult to reproduce (for client or server roles)
Network emulation	Use to emulate slower networks when you run your tests (for client or server roles)
System information	Use to collect system information for a machine (for client or server roles)
Test impact	Use to collect information that can help you decide which tests to rerun based on changes made to an application for a specific build (for client or server roles)
Video recorder	Use to create a video recording of your desktop session while you run a test (for client roles only)

Table 5 – Types of diagnostic adaptors

Understand how to use a multi-machine environment for debugging

Refer to [Setting Up Test Machines to Run Tests or Collect Data](#)¹⁶ for details on this topic.

¹⁴ [http://msdn.microsoft.com/en-us/library/ee471614\(v=vs.120\).aspx](http://msdn.microsoft.com/en-us/library/ee471614(v=vs.120).aspx)

¹⁵ [http://msdn.microsoft.com/en-us/library/ee702477\(v=vs.120\).aspx](http://msdn.microsoft.com/en-us/library/ee702477(v=vs.120).aspx)

¹⁶ [http://msdn.microsoft.com/en-us/library/dd293551\(v=vs.120\).aspx](http://msdn.microsoft.com/en-us/library/dd293551(v=vs.120).aspx)

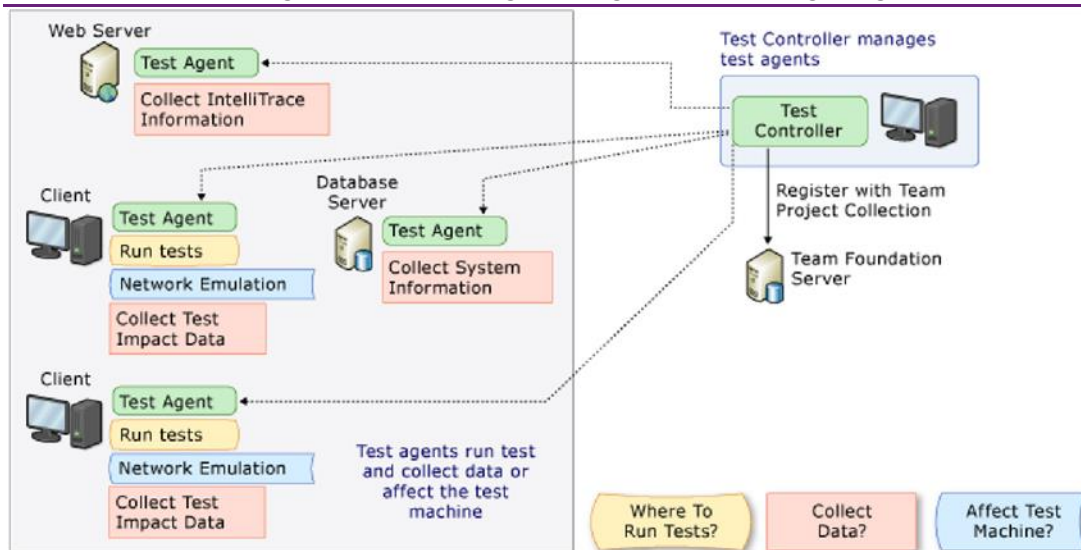


Figure 10 - Debugging Architecture

If you want to use a test controller as part of an environment, you must register it with Team Foundation Server, as shown in Figure 10 - Debugging Architecture. However, if you want to use a test controller from Microsoft Visual Studio 2012 or 2013, do not register the test controller with Team Foundation Server.

Useful Links / References

[Software Testing with Visual Studio 2010](http://www.amazon.com/Software-Testing-Visual-Microsoft-Development/dp/0321734483/ref=sr_1_1?ie=UTF8&s=books&qid=1294698036&sr=1-1)¹⁷

[Troubleshooting for Testing Tools for Visual Studio ALM](http://msdn.microsoft.com/en-us/library/ms243170.aspx)¹⁸

¹⁷ http://www.amazon.com/Software-Testing-Visual-Microsoft-Development/dp/0321734483/ref=sr_1_1?ie=UTF8&s=books&qid=1294698036&sr=1-1

¹⁸ <http://msdn.microsoft.com/en-us/library/ms243170.aspx>

Upgrading Lab Management Environments

Test Controllers and Agents

Backwards compatibility of test controllers

There is no immediate need to upgrade test controllers (and agents) when upgrading TFS. As with build controller/agent older versions of test controller/agents¹⁹ can be used with upgraded TFS servers. This means that your Lab Environments does not need to be a barrier to upgrading a TFS server.

Upgrading the test controllers

MSDN provide excellent instructions on upgrading test controllers²⁰. The key to remember is not to uninstall the old controller version prior to installing the new version, so as not to lose configuration settings,

Upgrading the test agents in your environments

Upgrading a test environment's test agents is automated. Once a test controller has been upgraded, it will upgrade the test agents on each machine within the environments it manages. This process can either be done upon demand using the 'upgrade agents' button with Microsoft Test Manager. Alternatively, it will be done automatically upon a restart of the environment. In both cases, the upgrade uses the standard test agent deployment mechanism within Lab Management.

In some cases you may experience issues with this process. If this is the case, it is best to un-install the test agents on each machine in an environment and install the new version by hand. There is no need to configure the test agents, upon restart of the environment, the agents should be reconfigured by the test controller.

System Center Support

What versions of SCVMM does Team Foundation Server support?

The following table shows which versions of SCVMM are supported²¹ by various TFS versions

TFS Version	SCVMM 2008	SCVMM 2008 R2	SCVMM 2012 SP1	SCVMM 2012 R2
2010	No	Yes	No	No
2012	No	Yes	Yes	No
2013	No	Yes	Yes	Yes

Table 6 – SCVMM Support

The version of SCVMM defines which version of the underlying operating system are support e.g. SCVMM 2012 R2 is required to manage Windows 2012 R2 Hyper V hosts

¹⁹ For N-1 versions

²⁰ <http://msdn.microsoft.com/en-us/library/hh707968.aspx>

²¹ Please validate the supported versions on <http://msdn.microsoft.com>. The product group typically supports the latest service pack in each major version, which can lead to minor anomalies with the versions listed in this guidance.

Does Lab Management support all features of System Center Virtual Machine Manager?

Lab Management does not support all features of any version SCVMM. The differences between versions of SCVMM are broadly operating system and hence Hyper-V feature support. The SCVMM features not supported in any version are as follows:

- Clustering of hosts
- SAN features such as migration
- VLAN-based networking of virtual machines
- Multiple networks between virtual machines of an environment
- Live migration
- Configuring devices other than hard disks in templates

Migration of virtual machines: On the condition that you migrate the virtual machines between hosts in the same host group, environments will work. Lab Management just relies on the virtual machine IDs and on whether they are in the correct host group. You have to perform migration directly from the SCVMM administration console because there is no such capability from Microsoft Test Manager.

VLANs: If you set vLAN IDs on a virtual machine or a template in the library, and then try to create an environment using that template, the vLAN settings might work if the environment is not network isolated. However, if the environment is network isolated, Lab Management does not preserve the setting. The only way to get vLANs in network-isolated environments is to first create the environment, and then manually change the vLAN configurations through SCVMM for that environment's virtual machines.

Does Team Foundation Server 2012 or 2013 Support Clustering?

Team Foundation Server 2012 onwards supports clustered hosts for creating non-network-isolated environments when used with SCVMM 2012 or later. The recommendation is to use a separate host group in SCVMM for the clustered hosts, and to use this host group from MTM for only creating non-isolated environments.

Network isolation still requires the VMs to be co-located on a single host, and that design does not work with clustering.

Are there any new²² SCVMM 2012R2 features supported by Lab Management?

There are no changes to the core SCVMM features supported in TFS 2013 due to its support for SCVMM 2012R2. Any addition are due to new features added to SCVMM 2012R2. The most important of these is the addition of support for 'Generation 2 VMs'. These are a new type of more performant VM standard, which makes best use of the Hyper-V platform. This is achieved by removing the legacy devices used in Generation 1 VMs.

This however causes a problem with Lab Management, Generation 2 VMs cannot be used network isolated environments, as they do not support legacy emulated network adaptors, only synthetic ones. Hence a network isolated environment must use Generation 1/legacy VMs.

Can I use SCVMM templates in Lab Management?

Starting with SCVMM 2012 it was not possible to store a VM template from MTM and be able to add it to another environment setting operating system settings e.g. memory, machine name etc.

With SCVMM 2012 onward the management of creating VMs from templates should be done solely in SCVMM and the resultant VMs added to environment as needed

²² <http://technet.microsoft.com/en-us/library/dn282278.aspx>

Can I use Lab Management with TF Service?

Prior to TFS 2013 it was not possible to directly consume a TF Service created build within a Lab Management environment as the hosted build process placed its output into a server drop (a location within source control). Server drops could not be consumed by previous versions of Lab Management without writing your own copying process as the LabDefaultTemplate build process template assumed copying from a UNC share.

With TFS 2013 support for server drops was added to the LabDefaultTemplate. So there is now no change in terms of experience that is needed to consume build output irrespective of whether the build output to a server drop or a UNC share. Thus allowing Lab Management to use the builds created on TF Service.

Upgrading System Center Virtual Machine Manager

Though running the current version of any toolset is recommended, as long as your SC-VMM version is supported by your version of TFS there is no immediate need to upgrade.

Usually the motivation for the upgrade will be to the improvements in newer versions of Hyper-V such as dynamic memory and disk de-duplication.

MSDN²³ provides detailed documentation on the process to upgrade your System Center Virtual machine Manager. The basic process can be summarized as:

1. Upgrading Visual Studio 2010 Test Controller to 2012 or 2013
Important You must complete this task before you uninstall your Visual Studio 2010 test controllers.
2. Upgrading lab environments from Visual Studio 2010
3. Migrating build, deploy, and test workflow definitions from Visual Studio 2010
4. Upgrading SCVMM 2008 R2 to SCVMM 2012:

Issues

As with all upgrades of complex application stacks problems can occur.

You cannot upgrade from SC-VMM 2008 R2 SP1 to SC-VMM 2012 SP1 directly

You cannot upgrade from SCVMM 2008 R2 SP1 to SCVMM 2012 SP1 directly. It needs to be upgraded to SCVMM 2012 first and then onto SCVMM 2012 SP1.

If you cannot upgrade your SC-VMM database you need to recreate the DB

The key to upgrading SC-VMM is the database upgrade. If this fails for any reason, you will have to create a new instance of SC-VMM and manually re-create all of your Lab Management environments.

If you recreate the SC-VMM database then you also need to clear the entries on TFS point to old SC-VMM server.

1. Firstly repointed TFS at the new SCVMM server

```
Tfsconfig.exe lab /settings /scvmmservername:new_scvmmservername /force
```

2. At this point you can try to upgrade the schema

```
tfsconfig lab /upgradeSCVMM /collectionName:*
```

3. If this errors, open Microsoft Test Manager (MTM) and delete all the deployed and stored environments, VMs and templates. This had no effect on the new SCVMM server as these entries referred to the now no-existent SCVMM host. It just removed the entries in the TFS databases.

²³ TFS 2012 process <http://msdn.microsoft.com/en-us/library/hh707966.aspx>

4. Once this was done rerun the command

```
tfssconfig lab /upgradeSCVMM /collectionName:*
```

5. This should reset the library and host groups to point to the new server.

You can now manually take VMs and get them imported into the new SCVMM library so that they can be redeployed. For running environments that don't require network isolation you can compose new environments to wrapper the existing deployed VMs. However, if you need network isolation the only method is to push each running VM up into the SC-VMM library and re-deploy them into a new network isolated environment.

When going through this manual rebuilding process it is important to remember

- Lab Manager uses XML in the description field of each VM in Hyper-V. If that xml is present, lab management assumes the VM is part of an environment and it won't show the VM as being available for a new environment. Deleting the XML makes the VM appear available for a new environment.
- When upgrading snapshots, particularly with save states, is a bad idea. When migrating between server versions, the Hyper-V save states are incompatible so you will lose information if you are not careful. Having lots of snapshots with configuration variations might also cause confusion. So collapsing snapshots prior to any upgrade where possible is a good idea.

Maintaining Lab Management Environments

In this section, you will learn about maintaining the availability of Visual Studio Lab Management assets and infrastructure. The guidance addresses what is required to be backed up and restored, when you should do backup, and where to do the backup.

Target personas and addressed user stories

As a member of a software development team, you want to have all of the assets that support your software development lifecycle to be always available. Source code, builds, work items, and test plans are common examples of valuable project assets that you would not want to lose. When you start using the Lab Management features of Team Foundation Server, there are additional assets that you will be concerned about maintaining:

- Environments that are used for ongoing testing or build verification.
- Golden templates or virtual machines, which are located on a library server.
- Environments that are archived in a library for later use, such as bug repro or old builds.

Here are some examples of scenarios that cover the requirement for backing up Lab Management assets.

- Your team has set up an environment using Lab Management for daily build verification. The Hyper-V server on which this environment has been setup is no longer functional.
- You have set up a test environment, and while using it for testing, you have made changes to it that caused the environment to not function.
- You are in the middle of a test pass before releasing a product. None of the test environments that are set up for your testing are available because some of the Hyper-V servers on which they are deployed are no longer functional.
- The library server on which the golden templates are stored is not functional. Hence your team is blocked from creating fresh environments.
- The SCVMM server that you are using for lab management has become non-functional, and your usage of environments for testing is impacted.

This section focuses on describing who needs to do what to ensure that your lab assets are available. The process to achieve this goal varies depending on who does what in your team.

In a smaller company, the maintenance and operations of Team Foundation Server is largely taken care of by a member of the software development team (typically, Garry the Dev Lead) together with any infrastructure and operations staff available. If this is your situation, read the section *Back Up and Restore in a Small Organization*.

In larger companies, there are more dedicated personnel operating Team Foundation Server and SCVMM, and the responsibilities of doing backup and restore might not be on the development team that uses Team Foundation Server, but rather in a centralized organization. Furthermore, the people managing Team Foundation Server and SCVMM may be different from each other. In this situation, read the section *Backup and Restore in a Large Organization*.

NOTE

In this section, it is assumed that you already have a process in place for backing up and restoring Team Foundation Server. If this is not the case, you should first invest in that process, and then read this guidance.

Back Up and Restore

Back Up and Restore in a Small Organization

The overarching goal of backing up and restoring lab assets is to ensure that your project assets related to Lab Management are intact. If you were part of a small organization, you would not want to invest in the time and money to properly backup servers and restore them upon failures. If you have trained IT professionals who can perform these tasks, see the guidance in the next section: "Back Up and Restore in a Large Organization."

Alternatively, if you are interested in a process that is lightweight and that works reasonably well for assets that you consider as critical, this section provides the necessary guidance. Let us go through each of lab assets and discuss some of the best practices you can follow in order to achieve this.

Deployed environments

Losing a deployed environment immediately affects the individuals using that environment, but, in a few cases, it is easy to recreate the environment on the fly, and hence it is not critical to have a backup of the environment.

Think of the following questions:

- Are your environments dynamically created on demand from virtual machines or templates in library?
- Are your environments short-lived (such as a few days) and are they created, used, and disposed frequently?
- Can you easily deploy a particular build into an environment using the build-deploy-test workflow capabilities of Lab Management?
- Do you perform little or no customization or additional software installation in the environment after it is created from templates?

If you answered 'Yes' to the above questions, then there is no reason to invest significant energy in backing up and restoring environments or Hyper-V servers. When your environment or a Hyper-V server becomes dysfunctional, you can always create another copy of the environment from templates in the library and deploy a build into it. You still, however, need to invest in backing up SCVMM server.

Consider the following additional best practices to reduce the chances of losing your work on lab environments.

- Every time you create an environment or make changes in the environment that you do not want to lose, take a snapshot of the environment. If you make unwanted changes to the environment, or render the environment dysfunctional, you can simply restore to the snapshot and continue to work.
- Every time you compose an environment or make changes to an environment that are hard to repeat, or that you do not want to repeat, shut down the environment, and store a copy of that environment on the library. If there is a failure to the Hyper-V host, you can re-deploy that stored copy and have your environment restored.

If you answered 'No' to the above questions, and if the above best practices do not suffice, then you will have to invest in a backup scheme for Hyper-V servers. The process for this is described in the section "Back Up and Restore in a Large Organization."

Stored virtual machines, templates, and environments

Stored virtual machines and templates are raw material for creating your environments on the fly. Similarly, stored environments are those that are archived on a file system of the SCVMM library for later use. The good news is that they are just files on a library server. Hence, if you have a practice of backing up file servers, then you can use the same procedure for backing up the library servers of SCVMM. Here are some best practices for backing up library servers:

- Use multiple library servers in your Lab Management setup. Store the same copy of your golden templates in multiple library servers. This will not only improve the availability in case a library server goes down, but it will also improve the performance of file transfers when you create fresh environments.
- At the minimum, back up the library shares that contain environments that are stored for later use.

To summarize the backup and restore recommendations in a small organization, you can get a fair amount of availability by following some best practices for your lab environments and templates. These practices, however, require members of the software team to perform additional steps. When you have an IT organization that can support your availability needs, you get better transparency and do not have to perform these additional steps.

Backup and Restore in a Large Organization

In a large organization, the Team Foundation Server, SCVMM server, and Hyper-V hosts are managed by a central IT organization. The IT organization is also responsible for ensuring that these infrastructure servers are made available with less downtime. This section recommends procedures that an IT administrator should be fairly comfortable in performing. These procedures are based on toolsets that are provided as a part of the products that Team Foundation Server and Lab Management are based on. In larger enterprises it is common to use more specialized tools such as the System Center Data Protection Manager. In those cases, this document can serve as a guide for what, rather than how, to back up.

What to back up and restore

The following section lists the key components to back up in Visual Studio Lab Management.

- Team Foundation Server
- SCVMM server
- Hyper-V hosts
- Library servers

It is not necessary to back up Test or Build controllers because these servers are stateless. However, in the event of a crash, it is highly recommended that the controllers are restored on the same machines or on machines with the same computer names as their earlier counterparts. If you do this, you can fix all the lab environments and lab workflow definitions in which the controllers are configured.

Team Foundation Server

The process for backing up and restoring Team Foundation Server is documented in [How to back up and restore Team Foundation Server](#)²⁴. Note that there are some differences in procedures if you restore Team Foundation Server on the same or on new hardware.

- [Restoring on the same hardware](#)²⁵
- [Restoring on new hardware](#)²⁶

System Center Virtual Machine Manager

System Center Virtual Machine Manager (SCVMM) is used to manage configuration in the Lab Management environment and the configuration is stored in the SQL Server database associated with SCVMM.

To backup an SCVMM server, you must take a backup of the SCVMM database. After a failure, you must restore the SCVMM database. The procedure for backing up or restoring SCVMM 2008 R2 is documented in [Backing Up and Restoring the VMM Database](#)²⁷.

The Hyper-V hosts should be backed-up in order to have a quick recovery process for virtual machines. Using a tool such as Windows Server Backup (which is part of Windows Server 2008 R2) is the easiest way to setup up Hyper-V backups. This backup, when configured as described in [How to back up Hyper-V virtual machines from the parent partition on a Windows Server 2008-based computer by using Windows Server Backup](#)²⁸, backs up all

²⁴ <http://msdn.microsoft.com/en-us/library/dd386362.aspx#TFS>

²⁵ <http://msdn.microsoft.com/en-us/library/ms252458.aspx>

²⁶ <http://msdn.microsoft.com/en-us/library/ms252516.aspx>

²⁷ <http://technet.microsoft.com/en-us/library/cc956045.aspx>

²⁸ <http://support.microsoft.com/kb/958662>

the volumes on which virtual machine data is present on a Hyper-V host. This makes it easy to restore Hyper-V servers later to exactly the same state (including virtual machine IDs and configurations) as at the time of backup. Hyper-V servers can be clustered to support high availability of virtual machines. However, **do not use** this technique for Lab Management, because Lab Management support for Hyper-V clustering is limited to non-isolated environments.

Library servers

Back up all library servers used by SCVMM in order to restore the golden templates and stored environments used by teams. Once again, Windows Server Backup can be used for this purpose.

Creating a backup schedule

In general, the more frequently you do full backups of your databases, the faster it will be to do a restore in case of a failure. This has to be weighed against the storage for the backups and how much work you can cope with losing.

Ordinarily, you will create a schedule that contains a mix of full, incremental and log backups.

An example of such a schedule could look like the following:

- Full backup every day
- Incremental backup every 6 hours
- Log backup every 30 minutes

Synchronizing backup and restore in a distributed environment

Team Foundation Server, SCVMM, Hyper-V servers, and library servers are all different servers, and it is clear that the backups of all of these components cannot be synchronized or even be done on the same day in many cases. The consequences of this distribution are as follows:

- A project member might have created an environment in Team Foundation Server and the corresponding virtual machines in SCVMM. But, if the SCVMM machine went down and is restored to an older state, then SCVMM would not know about those virtual machines although Team Foundation Server would.
- A project member might have created an environment in Team Foundation Server and the corresponding virtual machines in SCVMM. But, if the Team Foundation Server machine went down and is restored to an older state, then Team Foundation Server would not know about the environment, although those virtual machines are present in SCVMM.

These are just two examples of the consequences of distributed backups. In these cases, it is not practical for teams to restore all the systems to the same point in time when one of the servers fails. Instead of removing artifacts that are already there on functional servers, it is better to do any fix-ups and re-create artifacts manually on the restored servers. See the next section “How to back up and restore” for more details on handling these scenarios.

Team Foundation Server is restored to an older point

The actual procedure for restoring a Team Foundation Server can be found at: [Backing up and Restoring Your Deployment](http://msdn.microsoft.com/en-us/library/bb552295.aspx)²⁹

Consider the scenario where there was a Team Foundation Server failure, and it has to be restored to its last backup. Between the time of last Team Foundation Server backup and its failure, several actions could have been performed on Team Foundation Server. All of those changes would have been lost, and those artifacts need to be re-created. The following list shows examples of actions that might be lost on Team Foundation Server and the necessary steps to restore the lost artifacts:

- SCVMM was associated with the Team Foundation Server.

²⁹ <http://msdn.microsoft.com/en-us/library/bb552295.aspx>

- Team Foundation Server administrator should re-associate SCVMM server with Team Foundation Server.
- A project collection was associated with SCVMM host groups and library servers.
 - Team Foundation Server administrator should re-associate the project collection with the host groups and library servers.
- A project was associated with SCVMM host groups and library servers.
 - Team Foundation Server administrator should re-associate the project with the host groups and library servers.
- An SCVMM administrator has added new hosts to the host group.
 - Team Foundation Server administrator should select the host groups in each project collection (using the Team Foundation Server administration console), and re-verify the host groups to ensure that all new hosts are compatible with lab requirements.
- An SCVMM administrator has added new templates to the library share.
 - No action needed.
- A project member has imported templates into a project.
 - Project member should re-import the templates into the project.
- A project member has created new environments.
 - SCVMM administrator should identify the virtual machines that are part of the new environment. To do this, the SCVMM administrator can use the SCVMM administration console and search for the name of the new environment.
 - After the VMs of the new environment are located, the SCVMM administrator has to clear the description field in each of these virtual machines. This will make these virtual machines available again for composition. Then, the project member has to compose a new virtual environment by selecting those virtual machines.
 - Alternatively, the SCVMM administrator should delete the virtual machines that were part of the new environment, so that the project member can re-create the environment.
- A project member has deleted an environment.
 - After Team Foundation Server is restored, the environment would still exist. However, its state would be 'Missing'. The project member should delete the environment again.

SCVMM is restored to an older point

Next, consider the scenario where Team Foundation Server is functioning normally, but SCVMM server crashed. SCVMM has been restored to an older point in time. If any work was done between SCVMM's last backup and when it crashed, that work is lost. Actions that might have been performed during this time and the corrective actions are described in the following list:

- SCVMM was associated with the Team Foundation Server.
 - No action required if SCVMM is restored on the same machine. It is highly recommended that you restore SCVMM on the same machine or a machine with the same name.
- A project collection was associated with SCVMM host groups and library servers.
 - No action required.
- A project might have been associated with SCVMM host groups and library servers.
 - No action required.
- An SCVMM administrator has added new hosts to the host group.
 - SCVMM administrator should re-add the hosts to the host group.
 - The Team Foundation Server administrator should then re-verify the hosts in the Team Foundation Server administration console.
- An SCVMM administrator has added new templates to the library share.
 - SCVMM administrator should re-add the templates to the library share.

- If the old templates were already imported by a project member into Microsoft Test Manager, then the project members should delete the old templates using Microsoft Test Manager, and then re-import them.
- A project member has imported templates into project.
 - No action required.
- A project member has created new environments
 - The deployed environments would be auto-synched again after a few minutes and no action is needed.
- A project member has deleted an environment
 - No action required.

Where to back up

Backups should always be stored on a different device than the one that hosts the databases. It is highly recommended to use some type of secure storage. You can read more about [backup devices](#)³⁰.

Virtual Machine Patching

In this section, you will learn about maintaining the virtual machine images used in your test runs and keeping them up to day with regards to operational updates, security patches, application updates, etc. The guidance addresses how you can keep the virtual machines patched and updated whether they are online or offline.

Why do we patch?

We patch our servers to ensure that software is working correctly, as intended. The software malfunctions could range from elevation of privilege to information disclosure. *The different threats of malfunctions could include:*

- Spoofing identity
- Tampering with data (also called integrity threats)
- Repudiation
- Information disclosure
- Denial of service
- Elevation of privilege

Patch management ensures that correct software replaces incorrect software. The patch management process also includes mitigation techniques that are not actual patches but include additional procedures to protect computers if the patch is not available, or if administrators cannot apply it to a network, or if there are other reasons that preclude applying the patch.

Patch management also ensures that development/test environment is in sync with the production environment, while debugging any production issue. In this scenario we would manual apply patches on the test machines.

Virtual Machine Patching in a Small Environment

Windows Server Update Services

If you want to create a patching environment that is effective but simple to use, you can use Windows Server Update Services (WSUS). WSUS is available in Windows Server 2008 R2 by installing the WSUS role, or you can download it from [here](#)³¹. This site also contains more detailed documentation on WSUS.

³⁰ [http://msdn.microsoft.com/en-us/library/ms179313\(SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms179313(SQL.100).aspx)

³¹ <http://technet.microsoft.com/en-us/wsus/default.aspx>

WSUS is basically a locally controlled Windows Update system. With WSUS you can define and control which updates are delivered to the test virtual machines. This is handled through subscriptions between the virtual machine and the WSUS server. To manage the subscriptions to the WSUS server, you can use Windows Group Policies. These policies can either be defined within the domain used for the test virtual machines, or on the local machine itself. When you set the appropriate policies, updates will be controlled by WSUS and the machines will be patched.

In WSUS you can define which classification of updates, such as **Critical** or **Security Updates**, will be downloaded. You can also specify the products for which the updates are downloaded. Any number of updates can be downloaded into WSUS, but only the approved updates will be installed on the subscribers.

Flavors of Virtual Machines in SCVMM

Lab Management makes use of System Center Virtual Machine Manager (SCVMM). In SCVMM, these types of virtual machines are possible:

- Virtual machine in host
- Virtual machine in library
- Template in library

Virtual machine in host is a virtual machine that is deployed on a server that has Hyper-V installed. The virtual machine is ready to be started and to be used as an environment. A **virtual machine in library** is ready to be deployed. When you deploy the machine from Lab Management, it will create a copy of the virtual machine and place it on the host. At that point it becomes a **virtual machine in host**. If you do not use the option of network isolation during the deployment, you can have duplicate instances of the virtual machine, and thus have a duplicate computer name in the domain which can cause errors. To avoid this, you can make use of a **template in library**. When you create a template, it will remove all information from the machine by using the [sysprep](#)³² command. Then, to deploy a template to the host, you create a new virtual machine based on the template. The newly created virtual machine will have a unique identity and it will not cause errors.

In Lab Management, you can only import a virtual machine in library or a template in library. Therefore, this guidance describes how you can patch a virtual machine or a template in the SCVMM Library. Whether you want to make use of a virtual machine or a template depends on your personal situation.

Offline Virtual Machine Servicing Tool

When you subscribe a machine to WSUS with group policies, the updates will only be installed when the machine is running. Because virtual machines can be offline much of the time, they will not be patched during these off hours.

The Microsoft Solution Accelerators group has created a tool which solves this problem: the [Offline Virtual Machine Servicing Tool](#)³³ (VMST). The Hands-on Lab, which is part of this guidance, shows the steps for how to install and configure the tool.

The tool performs the following steps when it is triggered:

1. Deploy the virtual machine to a host
2. Install patches on the virtual machine
3. Store the virtual machine in a library

The VMST has the following requirements:

- To be able to install the patches on the virtual machine, you have to have a **WSUS** or a **System Center Configuration Manager**.

³² [http://technet.microsoft.com/en-us/library/dd799240\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd799240(WS.10).aspx)

³³ <http://technet.microsoft.com/en-us/library/cc501231.aspx>

- The VMST only applies to a virtual machine that is stored in the library that has the Virtual Machine Guest Services installed.

Template in SCVMM Library

To create a template, you can first build a virtual machine. Then, when you have completed configuring the virtual machine to meet your requirements, you can create a reusable template from the virtual machine image. Another option is to maintain a "Golden Image" virtual machine which is the virtual machine that was the base for the template. However, because the action to create a template destroys the virtual machine in the process, you first need to clone the virtual machine. Now, when you need to change the virtual machine's capabilities, you can modify the Golden Image, clone it, and create a template from the cloned virtual machine.

NOTE

With VMST 1.0 the MVST only applies to a virtual machine stored in the library and therefore you have to use of the Golden Image virtual machine process. With VMST 2 or higher, this alert does not apply.

Virtual Machine Patching in a Large Environment

Visual Studio Lab Management uses System Center Virtual Machine Manager (SCVMM) to manage multiple physical computers that host virtual machines. In addition, Visual Studio Lab Management uses SCVMM to manage the storage of virtual machines, virtual machine templates, and other configuration files in SCVMM library servers.

We need to treat all the physical and virtual machines present in the Lab Management similar to other production and non-production servers and keep them updated with the latest security and software updates, so that the Servers remain patched and usable.

The following sections explain the process of patching the servers, the kinds of patches available, and the tools available for patching.

What are we going to patch? (Host / Golden templates / Instances)

IT administrators are responsible for managing security and other software patches for both physical and virtual machines. We need to ensure that the software updates are applied on the Host machine as well as the virtual machines. When the virtual machines are online (running state) they can be serviced in the same manner as physical machines. When virtual machines are offline (stopped state), or stored in a virtual machine library, then additional steps are required to ensure that patches are applied.

Scheduling and Update Prioritization

On the second Tuesday of each month, Microsoft releases security updates. On the fourth Tuesday of the month, Microsoft releases non-security updates, which consist of application compatibility updates and, occasionally, .NET servicing updates. You should have more than one plan or set of guidelines for processing updates and applying security updates. Ideally, one plan should include patch "windows" (scheduled times) when normal application of patches and updates to systems are to be applied. This cycle should not only target the security or other critical updates, but it should also allow for the application of standard patch releases, updates, and application specific patches.

These windows can be time based or event based. For example, the scheduling can be event based, such as when an application specific Service Pack is published. Or, a schedule can mandate that system updates occur on scheduled cycle. In either case, modifications and customizations can and should be made based on availability requirements, system criticality, and available resources. Multiple "windows" allow you to deal with multiple types of patches and provide the flexibility to manage all types of security updates and other application specific updates.

Prioritization and Scheduling Considerations

Factors to be considered when prioritizing and scheduling the security updates include the following:

- Optional updates - Users can postpone installation indefinitely.
- Mandatory updates – Users can postpone installation until a deadline only.
 - Deadline based on detection time
 - Deadline based on authorization time
- Emergency updates – Users cannot postpone; forced reboot

Another factor that you should consider when you schedule and prioritize the patches is criticality of the system availability. For example, you could be in the final phase of testing and any hindrance to the availability might cause a delay to the schedule. Also, updates (changes) to the system may require testing to reset or restart at some point potentially resulting in rework.

Windows Server Update Services (WSUS)

If you want to create a patching environment that is effective but simple to use, you can use Windows Server Update Services (WSUS). WSUS is available in Windows Server 2008 R2 by installing the WSUS role, or you can download it from [here](#)³⁴. This site also contains more detailed documentation on WSUS.

System Center Configuration Manager

System Center Configuration Manager helps in deploying security updates and custom updates across physical and virtual machines that are available in the Lab Management environment. You can use System Center Configuration Manager for managing your virtual and physical machines in these Scenarios:

Typical Scenario

A Team Foundation Server administrator is the Lab Manager Environment owner and has the responsibility to make sure that all the physical and virtual machines that are part of the Lab Manager have all the required security updates installed. When a security update is released, the administrator creates packages of the required updates and uses System Center Configuration Manager 2007 to scan all machines in the Lab Management environment and apply the updates only to the machines that need them.

NOTE

You can get more information about Configuring and Using SCCM [here](#)³⁵. Detailed instructions and guided steps are also included in the Large Environment Hands-on Lab, which is part of this guidance.

³⁴ <http://technet.microsoft.com/en-us/wsus/default.aspx>

³⁵ <http://technet.microsoft.com/en-us/library/bb694263.aspx>

Lab Management Infrastructure Planning

For an infrastructure overview, please refer to the **Lab Management Quick Reference Guide – Infrastructure** quick reference poster, included as part of this guidance.

Prerequisites

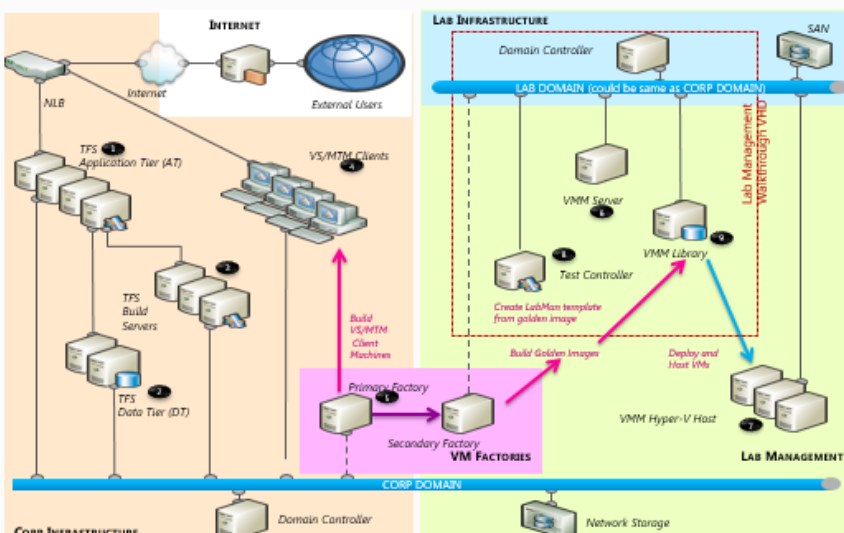
Unless stated each component is supported by the following platforms:

Windows Server 2003 SP2, 2003 R2, 2008 SP2, 2008 R2, 2012, 2012 R2

Windows XP SP2, Vista, 7, 8, 8.1

1. Team Foundation Server Application Tier (AT)	8443	team web access server
Ports used by services	8080	
2. Team Foundation Server Data Tier (DT)	1433	SQL
Ports used by services		
3. Team Foundation Server Build Controller	9191	controller
Ports used by services		
4. Team Foundation Server Clients		
5. VM Factory	3389	RDP
Ports used by services	137	UDP
	139	TCP
6. VMM Server*		
Windows Server 2008 or 2008 R2. R2 is recommended.	8100	VMM Agent
Ports used by services	135, 49152-65535	WMI
	3389	RDP
7. VMM Hyper-V Host*		
Windows Server 2008, 2008R2. R2 is recommended.		
8. Test Controller	6901	Test Controller
Ports used by services	6910	Test Agent
9. VMM Library*		
Windows Server 2008 or 2008 R2. R2 is recommended.	443	BITS
Ports used by services	80	WinRM

*Windows Server 2012 or 2012 R2 is supported for VMM Server 2012



References

Visual Studio Lab Management Guide
Visual Studio ALM Rangers VM Factory Guidance
Capacity Planning Quick Reference Poster
Team Project Collection Planning Quick Reference Poster

<http://go.microsoft.com/fwlink/?linkid=201695&sid=6e829>
<http://go.microsoft.com/fwlink/?linkid=201695&sid=6e829>
<http://go.microsoft.com/fwlink/?linkid=201695&sid=6e829>

Figure 11 - Visual Studio 2013 Lab Management Guide - Infrastructure Example

If you already have a Team Foundation Server infrastructure in your organization, Lab Management adds the following additional infrastructure components:

1. SCVMM Server
2. Hyper-V Hosts
3. Library Server
4. Test Controller (you would already have this if you are using testing or build features of Team Foundation Server)

The following table summarizes the software components that are required to implement Lab Management:

Typical Components	Server	Client	Notes	Targeted Persona
System Center Virtual Machine Manager 2008 R2 or 2012	✓			SCVMM administrator (Jane or Garry)
Visual Studio Team Foundation Server	✓			Team Foundation Server administrator (Dave or Garry)
Visual Studio Ultimate		✓	Using Microsoft Test Manager to configure and manage the lab requires either Visual Studio Ultimate with MSDN or Visual Studio Test Professional with MSDN.	Developer Persona (Garry or Doris)
Visual Studio Test Professional		✓	Using Microsoft Test Manager 2013 to configure and manage the lab requires either Visual Studio 2013 Ultimate with MSDN or	The tester persona (Christine)

Lab Management - Lab Management Infrastructure Planning

Typical Components	Server	Client	Notes	Targeted Persona
			Visual Studio Test Professional 2013 with MSDN.	
Visual Studio Agents – Test Controller		✓	Test Controller 2013 and Test Agent 2013 collectively enable scale-out load generation, distributed data collection, and distributed test execution. They also now manage testing, workflow and network isolation ³⁶ for virtual machines managed using the Lab Management solution	Tester or Developer persona (Christine or Garry)
Visual Studio Agents – Test Agent		✓		Tester or Developer persona (Christine or Garry)

Table 7 - Lab Management Software Components

The following sections describe each of the Lab Management infrastructure components in greater detail and what you need to plan for and configure.

Hyper-V Servers

Hyper-V Servers	
Personas	The personas who will be interested in planning for and setting up the Hyper-V hosts are Jane, the Infrastructure specialist of Humongous Insurance, and Garry, the development lead of Troy Research.
Purpose	Virtual machines created by Lab Management will be hosted on these servers.
Capacity planning	The number of Hyper-V servers depends on the number of virtual machines that you will need to run. For example, on a Hyper-V server having 8GB of memory, you can typically run 3 virtual machines of 2GB memory each. Similarly, on a Hyper-V server having 48 GB of memory, you can typically run 20 virtual machines of 2GB each. To estimate the capacity needs of your lab, you need to have a good understanding of the number of projects that will be using Lab Management, the number of environments needed for each of those projects, and the virtual machine needs of each environment. You can also set up a lab with some initial capacity, and grow it by adding more Hyper-V servers as your demands grow.
Machine requirements	<p>Big or small machines: Depending on your capacity needs, you can either use a large number of small-sized machines or a small number of large-sized machines (such as 48 GB memory servers). If you have large-sized machines, you need to ensure that the disk and CPU of those machines are capable of handling the load of all virtual machines. Here is a sample configuration of a 48GB server:</p> <ul style="list-style-type: none"> • Dual quad-core CPUs with 48 GB memory • 6 X 10k RPM SAS drives in RAID0 for VHD storage • Separate disks for host OS and VHD storage • 2 X Gigabit Ethernet ports <p>Do not go beyond 48GB machines. There are several reasons for this:</p> <ul style="list-style-type: none"> • A Hyper-V host is the minimum unit that can be allocated to a team project in Team Foundation Server (by creating a separate SCVMM host group). That is why, if you have very large hosts (for example, 64 GB), there are NO mechanisms in SCVMM or in Team Foundation Server to partition that host into smaller quotas and assign those quotas to a team project. • As the number of virtual machines per host increases, the interactions between SCVMM and the host take much longer time and cause intermittent failures. Therefore, a max fan-out of 25-30 VMs/host is recommended by System Center Virtual Machine Manager 2008 R2. <p>Maximum number of hosts: You can go up to 300 hosts and 6000 virtual machines using System Center Virtual Machine Manager 2008 R2 (VMM). However, this requires careful planning.</p> <p>Storage and SAN: Having good storage in each Hyper-V server is critical as a lot of virtual machine performance will be determined by the performance of the storage. The storage for hosts can come</p>

³⁶ <http://msdn.microsoft.com/en-us/library/ee518924.aspx>

Hyper-V Servers

from Direct-attached storage or from a SAN. With regard to SAN support, Visual Studio 2013 Lab Management cannot create environments on clustered hosts. This would mean you cannot have a setup with fail-over host clustering between two or more Hyper-V hosts, and create fresh environments on that cluster. If you decide to have your host's disk come from a SAN drive for space and reliability requirements, you can do so. Given that Visual Studio 2013 Lab Management does not leverage any SAN functionalities, there is no way to speed up the VM deployment using SAN features. Virtual machine deployments between hosts and library servers happen through BITS protocol.

Further, if you want to have clustering in your setup, you can do so. Instead of creating fresh environments, you have to pre-create the virtual machines on the cluster using SCVMM, and then compose those virtual machines as environments in Lab management. If you follow this approach, ensure that all the clustered hosts are setup in a separate host group and that you would use this host group only for composing environments.

Operating System: Hyper-V hosts have to run Windows Server 2008 R2 or later. They require all the latest patches. For Team Foundation Server 2012 Windows Server 2008 R2 SP1 is required.

There is no support for Windows 8 with Team Foundation Server 2013.

The Hyper-V role should be enabled in the host. BIOS of the Hyper-V hosts should be configured to enable hardware virtualization.

Networking and Communication: Hyper-V hosts should have line-of-sight visibility to:

- SCVMM: SCVMM server communicates with Hyper-V hosts to manage them.
- Library servers: Protocols such as BITS work between the library server and the hosts.
- Team Foundation Server: Team Foundation Server communicates with the hosts using WMI to set some registry keys. These keys (known as Guest KVPs) provide a channel for Team Foundation Server to communicate with the virtual machines even when the virtual machines are not networked. Features such as [network isolation](#)³⁷ and testing in lab environments require this communication.
- MTM: Microsoft Test Manager attempts to connect to virtual machines in an environment using a host-based connection. This allows MTM to show the virtual machine console even when the virtual machine is booting up. If this channel is not present, then MTM resorts to a guest-based RDP connection to the environment's virtual machines.

Hyper-V hosts should be on the same network as the library servers and other components such as Team Foundation Server and SCVMM server. It is highly recommended that the library servers and hosts be connected to the same switch so that there are no hops between them. This will improve the performance of network transfers of virtual machines.

Active directory domain:

- The highly recommended topology is where Hyper-V hosts and Team Foundation Server are in the same Active directory domain or in domains that have two-way trust between them.
- Because Team Foundation Server has to communicate with the hosts, at the minimum, the host's domain needs to trust the Team Foundation Server domain. In this topology, ensure that SCVMM server and Team Foundation Server have full-trust between them.
- Although SCVMM supports having completely un-trusted hosts, this topology will not work for Lab Management.
- More than one Domain Controller is recommended even for small environments to provide a fail-safe environment.

Geo-distribution: If your teams are located at multiple sites, then you have two options:

- You can have all hosts in a central lab, and all teams connect to environments in that lab. Connecting and interacting with machines over RDP might not be an ideal experience if the network is slow.
- You can have hosts at each site with a single central SCVMM server. Co-located hosts must be grouped into host groups. Whenever a team at a particular site wants to create an environment, it can use the host group located at the local site. In this model, the connection

³⁷ <http://msdn.microsoft.com/en-us/library/ee518924.aspx>

Hyper-V Servers	
	performance will be good. However, the communication between the hosts and the SCVMM server requires good network connectivity. For example, in a network with 100 Mbps link connectivity between the sites, there will be no noticeable performance degradation. In a network with 8 Mbps connectivity between sites, the network will get frequently overloaded with traffic between hosts and SCVMM server. If you have low network bandwidth between sites, or if the latency is high, then there is no good geo-distribution solution.

Library Servers

Library Servers	
Personas	The personas that will be interested in planning for and setting up the Library servers are Jane in Humongous Insurance, and Garry in Troy Research.
Purpose	Library servers are used to store virtual machine templates or archive environments in Lab Management.
Capacity planning	The number of Library servers depends on the number of virtual machine templates or environments that you have to store.
Machine requirements	<p>Storage and SAN: Library servers are file servers and they should have fast storage devices. The storage for library can come from Direct-attached storage or from SAN. With regard to SAN support, you will need SCVMM 2012 servers to leverage clustering. This would mean you cannot have a setup with fail-over clustering between two or more library servers. However, if you decide to have your library server's disks come from a SAN drive for space and reliability needs, you can do so.</p> <p>Because Lab Management does not leverage any of SAN functionalities, there is no way to speed up the virtual machine deployment from library to host using SAN's features. Virtual machine deployments between hosts and library servers happen through BITS protocol.</p> <p>Operating System: Library servers need to run Windows Server 2008 R2 or later. They need to have all the latest patches. For Team Foundation Server 2012 Windows Server 2008 R2 SP1 is required. There is no support for Windows 8 with Team Foundation Server 2013.</p> <p>Networking and Communication: Library servers should have line-of-sight visibility to:</p> <ul style="list-style-type: none"> • SCVMM: SCVMM server communicates with Library servers to manage them. • Hyper-V hosts: Protocols such as BITS work between the library server and the hosts. <p>Library servers should be on the same network as the Hyper-V hosts and SCVMM server. It is highly recommended that the library servers and hosts be connected to the same switch so that there are no hops between them. This will improve the performance of network transfers of virtual machines.</p> <p>Active directory domain:</p> <ul style="list-style-type: none"> • Any topology supported by SCVMM works, because there are no additional communication channels from library servers to Team Foundation Server or MTM. <p>Geo-distribution: If you have Hyper-V hosts that are set up in multiple locations, but with a central SCVMM server, make sure that you have a Library server at each location. Copying large VHDs across WANs from library servers to hosts is definitely not a good idea. See the notes in Hyper-V servers Geo-distribution for more information.</p>

SCVMM Server

SCVMM Server	
Personas	The personas that will be interested in planning for and setting up the Library servers are Jane in Humongous Insurance, and Garry in Troy Research.
Purpose	An SCVMM server provides host, library, and virtual machine management functionalities. Team Foundation Server uses SCVMM server to create and operate on virtual environments.

Lab Management - Lab Management Infrastructure Planning

SCVMM Server	
Capacity planning	<p>With Team Foundation Server 2010, you can configure only a single SCVMM server per Team Foundation Server Project Collection. All project collections in the Team Foundation Server instance share the SCVMM server.</p> <p>The same SCVMM server may be used by multiple Team Foundation Server instances, provided the same host groups and library servers are not used by both instances. With Team Foundation Server 2013 you can configure a different SCVMM server for each team project collection but this can only be done from the command-line using the TfsLabConfig.exe tool. Note that if you choose to use multiple SCVMM Servers with Lab Management for the same TFS server then you must configure each SCVMM server to use different MAC addresses. See http://tinyurl.com/m58jfy6 for a description of this scenario.</p> <p>The size of machine you need to run SCVMM server depends on the size of your lab (number of hosts and virtual machines).</p> <p>Scaling: For a large lab with more than 500 virtual machines, you must use the following minimum configuration:</p> <ul style="list-style-type: none">• 2-proc dual-core with 16GB of memory.• The database for SCVMM server resides on a separate physical server running SQL Server 2008 Enterprise data management software R2 or later versions.• The database logs and database files reside on separate physical disks.
Machine requirements	<p>Storage and SAN: Do not use the SCVMM server as a library server especially if you have a large setup.</p> <p>Operating System: SCVMM Server needs to be installed on Windows Server 2008 R2 or Windows Server 2008 R2 SP1. There is no support for Windows 8 servers with Team Foundation Server 2013.</p> <p>Networking and Communication: SCVMM server should have line-of-sight visibility to:</p> <ul style="list-style-type: none">• Hyper-V hosts: SCVMM server communicates with hosts to manage them.• Library servers: SCVMM server communicates with library servers to manage them.• TEAM FOUNDATION SERVER: Team Foundation Server communicates with SCVMM for all environment operations. <p>Active directory domain:</p> <ul style="list-style-type: none">• SCVMM server must be the in a fully trusted domain of Team Foundation Server.• SCVMM server and hosts can be in different domains provided that the host can trust the SCVMM's/Team Foundation Server's domain. <p>Geo-distribution: Because there can only be a single SCVMM server per Team Foundation Server instance, the only way of having geo-distributed labs is to configure different host groups for each site, all of which are managed by a single SCVMM server. See the notes in Hyper-V servers Geo-distribution for more information.</p>

Team Foundation Server

Team Foundation Server	
Personas	The personas that will be interested in planning for and setting up the Team Foundation Server are Dave in Humongous Insurance and Garry in Troy Research.
Purpose	Team Foundation Server hosts all the ALM services centrally including Lab Management.
Capacity planning	Refer to the Team Foundation Server install guide ³⁸ for capacity planning guidance.

Test and Build Controllers

Test and Build Controllers	
Personas	The personas that will be interested in planning for and setting up the test and build controllers are Garry in Troy Research and an equivalent developer lead in Humongous Insurance.

³⁸ <http://msdn.microsoft.com/en-us/library/dd631902.aspx>

Lab Management - Lab Management Infrastructure Planning

Test and Build Controllers	
Purpose	<p>A test controller coordinates the execution of tests on lab environments. A build controller coordinates the execution of build-deploy-test workflow on lab environments, besides coordinating the build workflows themselves.</p> <p>With Team Foundation Server 2013 only a test controller is needed.</p>
Capacity planning	<p>Test controllers: A given test controller can be configured to a single project collection in Team Foundation Server. A single project collection in Team Foundation Server can have any number of test controllers. You can easily add more and more test controllers as your testing requirements increase. It is recommended that you set up a new test controller for every 50 virtual machines that are actively used for testing. A standard machine with 4-8GB of memory will suffice for a test controller.</p> <p>Build controllers: A given build controller can be configured to a single project collection in Team Foundation Server. A single project collection in Team Foundation Server can have any number of build controllers. You can easily add more and more build controllers as your testing requirements increase. A build controller does not typically consume much resource, and hence it is rare to have many build controllers associated with a collection. A regular machine 4-8GB of memory will suffice for a build controller. The Build and Test Controllers are consolidated on one server, although they could be deployed on separate servers.</p> <p>Refer to Visual Studio ALM Rangers Team Foundation Server Team Project and Collection Guidance³⁹ for discussions around the sizing and structuring of Team Foundation Server Team Project Collections and Team Projects.</p> <p>See Build Customization Guide⁴⁰ for more information and guidance on the build controllers.</p>

Planning Checklist

Step	Planning Step	Personas	Completion Criteria	References
1. <input type="checkbox"/>	Qualify your deployment size	Dave, Jane, Garry	Determined that you are more closely aligned to the large or small team description	ALM Rangers Personas and Customer Profiles ⁴¹
2. <input type="checkbox"/>	Map the personas to specific people in your organization as applicable	Dave, Jane, Garry	Documented persona ownership	ALM Rangers Personas and Customer Profiles
3. <input type="checkbox"/>	Determine the Lab Management scenarios you need to support	Dave, Garry	Documented the one or more Lab Management Scenarios that you will be supporting	See chapter Planning, Setting Up and Configuring Test Lab Environments
4. <input type="checkbox"/>	Capacity planning – Team Foundation Server Environment	Dave, Garry	Document the capacity needed for the TFS environment.	See section Team Foundation Server
5. <input type="checkbox"/>	Capacity planning – Lab Management Environment	Jane, Garry	Document the capacity needed for the lab machine environment.	See section Lab Management Infrastructure Planning
6. <input type="checkbox"/>	Determine Hardware Specifications	Jane, Garry	Identified available hardware and/or define the hardware purchase order	MSDN ⁴²

³⁹ <http://go.microsoft.com/fwlink/?LinkID=212352&clcid=0x409>

⁴⁰ <http://go.microsoft.com/fwlink/?LinkID=207058&clcid=0x409>

⁴¹ <http://vsarguidance.codeplex.com/releases/view/88001>

⁴² <http://msdn.microsoft.com/en-us/library/ff756575.aspx>

Lab Management - Lab Management Infrastructure Planning

Step	Planning Step	Personas	Completion Criteria	References
7. <input type="checkbox"/>	Acquire Hardware	Jane, Garry	Verified that all required hardware is available and ready for installation	Blogs ⁴³
8. <input type="checkbox"/>	Acquire Software	Jane, Garry	Verified that all required software is available for installation	MSDN Subscriptions ⁴⁴
9. <input type="checkbox"/>	Understand account requirements	Jane, Garry	All network accounts required to set up and configure the software have been created	Managing User Permissions for Lab Management ⁴⁵
10. <input type="checkbox"/>	Stand up Hardware with OS on the Network	Jane, Garry	Hardware is running on the network and ready for Team Foundation Server / Lab Management setup to begin	

⁴³ http://blogs.msdn.com/b/charles_sterling/archive/2010/04/17/ultimate-demo-machine-for-use-with-virtualization.aspx

⁴⁴ <https://msdn.microsoft.com/en-us/subscriptions/securedownloads/default.aspx>

⁴⁵ <http://msdn.microsoft.com/en-us/library/dd380760.aspx>

Automating the Build-Deploy-Test Process

Where can the output from the TFS Build process be placed?

In previous releases of TFS there was only one possible location for the output from a build; this was to store the output on a UNC path e.g. [\\server\share](#). The only alternative being not to store the output at all. This limitation was a problem when you needed to access the build output from any location other than within the domain the TFS server was located. With TFS 2013 there are further options, the build results can be stored on the TFS server itself.

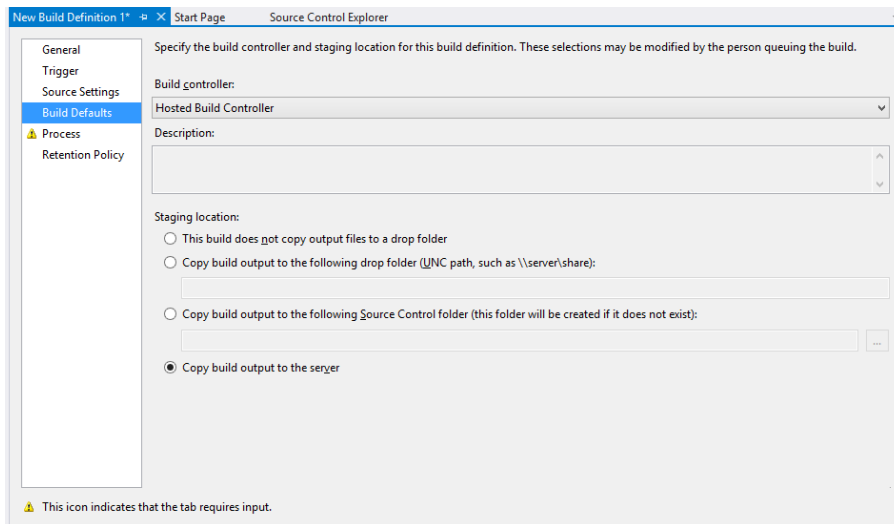


Figure 12 - Staging location choices

With TFS 2013⁴⁶ the new option is to 'Copy build output to the server'. This is successor to the 'Copy build out to the following Source Control folder', which can still be seen on TF Service (as in figure above).

These options are important for BDT scenarios as they provide a location that is accessible without the need for any firewall changes beyond those already made to allow access to TFS.

The disadvantage of the older 'Copy build out to the following Source Control folder' was that though it allowed the use of the TFS version control to store the drops, a user had to remember to cloak their drops folder to avoid problems users downloading the growing set of build output. In addition, each build caused multiple commits to version control as the drops were managed.

The new 'Copy build output to the server' option still stores the build outputs on the TFS server, and its contents is managed by TFS, but not in the version control repository. Thus removing the limitations seen with 'Copy build out to the following Source Control folder'

'Copy build output to the server' can be used on premises with TFS 2013 and with TF Service, and should be used in preference to 'Copy build out to the following Source Control folder'

When using the 'Copy build output to the server' output of the build is stored as a zip file on the server and the user provided with a link to download that zip.

⁴⁶ Using Visual Studio 2012 Update 1 or later as a client

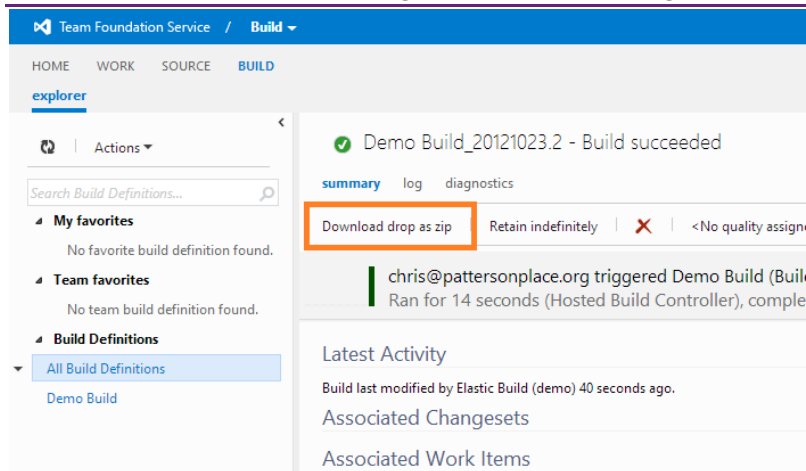


Figure 13 - Downloading the build output as a zip

Behind the scenes this mechanism used by the automated deployment model from TF Service to Azure⁴⁷. If you choose to use a server drop within a Lab Management build you do not need to use any special handling to manage the Zip file. The test controller will automatically unpack the contents to allow deployment.

Lab Templates

This section describes two specific implementations that build upon the supplied DefaultLabTemplate supplied by Team Foundation Server. The information provided herein is applicable to Team Foundation Server 2010, 2012 and 2013 and shows some nice examples on how the lab process can be extended using the Lab Management API.

NOTE

If you are interested in how the default lab template can be customized, go to <http://msdn.microsoft.com/en-us/library/ff934561.aspx> on Microsoft MSDN.

The Enhanced Lab Template provides an extended DefaultLabTemplate that includes functionality for Marking the Environment as "in-use" during a build and test, as well as deleting unneeded or undesired post-deployment snapshots.

The Parallel Lab Template builds upon the additional functionality provided by the Enhanced Lab Template and supplies functionality for using Lab Environments as part of Parallel builds, such as what would be used in Continuous Integration (CI) builds.

Custom Activities

The Parallel Lab Template uses custom activities. These activities can be found on the CodePlex site located at: <http://tfsbuildextensions.codeplex.com/>. The CodePlex site has an assembly called *Activities.LabManagement* where all of the activities reside.

NOTE

There are also many native build activities available for lab management workflows, go to <http://msdn.microsoft.com/en-us/library/ff934562.aspx> on Microsoft MSDN to get the current list as well as documentation on how to use them.

Activity	Description
CheckIfEnvironmentIsLocked	This activity checks to see if the specified environment is already locked by a build.

⁴⁷ See <http://blogs.msdn.com/b/bharry/archive/2012/06/07/announcing-continuous-deployment-to-azure-with-team-foundation-service.aspx>

Lab Management - Automating the Build-Deploy-Test Process

Activity	Description
GetEnvironment	Retrieves a Lab Management environment by its name.
GetEnvironmentLockedByBuildNumber	Retrieves the build number that has the specified environment locked. It is used to allow the Lab Process to report who is currently using the environment.
ListEnvironments	Retrieves a list of all of the environments that match the specified set of Tags.
ListLabSystems	List Team Foundation Server Lab Management Lab Systems based on tag filters.
LockEnvironment	Locks a specified environment, indicating that the environment is now "owned" by the requesting build.
PauseEnvironment	Set a Team Foundation Server Lab Management Lab Environment in a Paused state.
RenameSnapshot	Renames a snapshot attached to a Team Foundation Server Lab Management Lab Environment.
ShutdownEnvironment	Shuts down a Team Foundation Server Lab Management Lab Environment.
UnlockEnvironment	Unlocks the environment specified.
WaitForEnvironmentToBecomeAvailable	This activity waits for an environment (from a list of environments) to become available.

Table 8- Custom Activities

Some of the custom activities included above use a Locking UNC Share. This UNC Share must be configured such that the TFS Build account has full access to the contents of the share. Files are created here as part of the Lock/Unlock activities to ensure that the environment access is completely safe from multiple-process access issues.

Enhanced Lab Template

When Queuing a build that uses either of these templates, there are additional settings available.

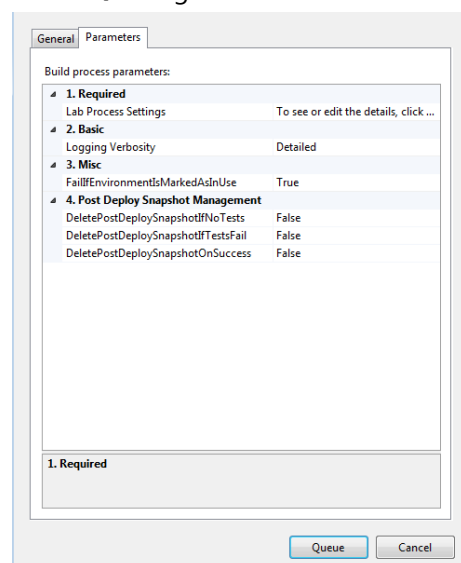


Figure 14 – Parallel Lab Template Properties

Miscellaneous Parameters

This group provides miscellaneous parameters and Settings.

Lab Management - Automating the Build-Deploy-Test Process

Parameter	Usage
<code>FailIfEnvironmentIsMarkedAsInUse</code>	This parameter allows you to control whether or not the build should fail if the selected environment is already marked as being In-Use. Recommended Setting: True

Table 9- Miscellaneous parameters

The Miscellaneous parameter *FailIfEnvironmentIsMarkedAsInUse* allows you to help ensure that environments that are being actively used by others do not get reset and used for tests. This is particularly helpful if you have multiple QA testers and developers working on issues discovered inside a Lab Environment. When this setting is set to *True*, the build process will fail if the environment is marked as being in-use, and no changes will be made to the specified environment.

Post Deploy Snapshot Management Parameters

Parameter	Usage
<code>DeletePostDeploySnapshotIfNoTests</code>	This parameter allows you to control whether or not the post-deployment snapshot is deleted if no tests are found to be run as part of the build. Recommended Setting: True
<code>DeletePostDeploySnapshotIfTestsFail</code>	This parameter allows you to control whether or not the post-deployment snapshot is deleted if any of the run tests fail. Recommended Setting: False
<code>DeletePostDeploySnapshotOnSuccess</code>	This parameter allows you to control whether or not the post-deployment snapshot is deleted if all of the run tests succeed. Recommended Setting: True

Table 10- post deploy snapshot management parameters

The various Delete Post-Deployment Snapshot parameter settings allow you to control when, or if, the post-deployment snapshot is deleted. By keeping only the post-deployment snapshots of executions with failed tests, the amount of maintenance on the lab environment is kept to a minimum. This is particularly helpful in an environment where many test runs using the process template are run, and the majority succeeds; extra snapshots can be automatically removed.

Template High-level Walkthrough

The entire template is detailed as part of the Parallel Lab Template.

The Enhanced Process Template provides a Try-Catch construct around the entire lab processes (after the build location is determined. This try catch construct is responsible for determining if the environment is In-Use, and if it is not, marking the environment as In-Use. In the finally portion of the construct, if this build is the one that marked the environment in use, it is released and marked as Not In Use.

Additionally, if the post-deployment snapshot should be deleted, there is additional logic near the bottom of the template that provides this functionality.

Parallel Lab Template

The Parallel Lab Template provides for the ability to use the Lab Environment(s) as part of a Continuous Integration (CI) build. There are additional process settings (described below) that can be used to customize the process.

3. Environment Selection	
Environment Tag List	
EnvironmentList	String[] Array
Use Environment from Wizard	False
Use Environment List	True
UseEnvironmentTagList	False
4. Misc	
FailIfEnvironmentIsMarkedAsInUse	True
Locking UNC Share	\\xptfslabcontrol\EnvironmentFlags
TimeToWaitForEnvironmentToBecomeAvailable	600000
WaitForEnvironmentToBecomeAvailable	True
5. Post Deploy Snapshot Management	
DeletePostDeploySnapshotIfNoTests	True
DeletePostDeploySnapshotIfTestsFail	True
DeletePostDeploySnapshotOnSuccess	True

Figure 15 - Parallel Lab Template Settings

Environment Selection Settings

Parameter	Usage
Environment Tag List	This parameter allows you to specify a list of Tags that can be used to identify target environments to be used during the execution of the process template
Environment List	This parameter allows you to specify a list of environment names that can be used to during execution of the process template
Use Environment From Wizard ¹	This parameter allows you to indicate that the environment selected from the wizard should be used, rather than one of the lists. When this is set to True, the template behaves like the Enhanced Process template described above.
Use Environment List ¹	This parameter allows you to indicate that the specified list of environments should be used.
Use Environment Tag List ¹	This parameter allows you to indicate that the specified environment tag list should be used

Table 11- Environment selection settings

¹These three settings are mutually exclusive, that is they are evaluated in the order listed above, the first one encountered that is true is the setting that is used.

Miscellaneous Settings

This group provides miscellaneous parameters and Settings.

Parameter	Usage
FailIfEnvironmentIsMarkedAsInUse	This parameter allows you to control whether or not the build should fail if the selected environment is already marked as being In-Use. Recommended Setting: True
Locking UNC Share	Locking UNC Share that is used by the build controllers to ensure multi-process safe access to the environments. This setting must be provided, and the permissions in the UNC Share must provide the TFS Build Account with full permissions
TimeToWaitForEnvironmentToBecomeAvailable	Provides the amount of time to wait for an environment that is specified (or in the specified list) to become available before the execution fails.

Lab Management - Automating the Build-Deploy-Test Process

Parameter	Usage
WaitForEnvironmentToBecomeAvailable	Indicates whether or not the process should wait for an environment if none in the list are available when the process is run.

Table 12- Miscellaneous settings

The Miscellaneous parameter *FailIfEnvironmentIsMarkedAsInUse* allows you to help ensure that environments that are being actively used by others do not get reset and used for tests. This is particularly helpful if you have multiple QA testers and developers working on issues discovered inside a Lab Environment. When this setting is set to *True*, the build process will fail if the environment is marked as being in-use, and no changes will be made to the specified environment.

Post Deploy Snapshot Management

Parameter	Usage
DeletePostDeploySnapshotIfNoTests	This parameter allows you to control whether or not the post-deployment snapshot is deleted if no tests are found to be run as part of the build. Recommended Setting: True
DeletePostDeploySnapshotIfTestsFail	This parameter allows you to control whether or not the post-deployment snapshot is deleted if any of the run tests fail. Recommended Setting: False
DeletePostDeploySnapshotOnSuccess	This parameter allows you to control whether or not the post-deployment snapshot is deleted if all of the run tests succeed. Recommended Setting: True

Table 13- Post deploy snapshot management

The various Delete Post-Deployment Snapshot parameter settings allow you to control when, or if, the post-deployment snapshot is deleted. By keeping only the post-deployment snapshots of runs with failed tests, the amount of maintenance on the lab environment is kept to a minimum. This is particularly helpful in an environment where many test runs using the process template are run, and the majority succeeds; extra snapshots can be automatically removed.

Configuring Team Foundation Server to use the Parallel Build Template for CI Builds

To use the Parallel Build Template with Continuous Integration Builds, you will need to also configure an On-Demand build that matches the CI build Request. The overall way this works is the Lab Template will call the On-Demand build configuration to actually perform the build. Assuming the build of the code is successful; the Lab Template will manipulate the environments and run the tests. Once you have configured your On-Demand build, you need to create a build based on the Parallel Lab Template. Setup the trigger to be Continuous Integration.

If your resources do not permit building every build, I suggest you set up your build to use Rolling Builds; this will group check-ins together and still provide benefit and build verification.

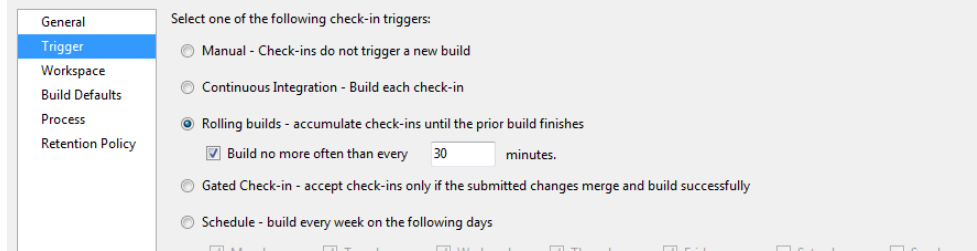


Figure 16- Parallel Build Template for Rolling Build Configuration

When you reach the Process, make sure you select the correct Process template, and then setup the Lab Process Settings (double click or click the button to start the configuration wizard).

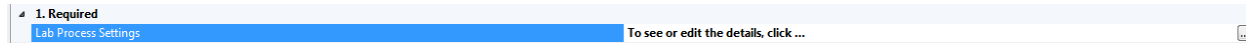


Figure 17 - Parallel Build Template Invoke Lab Wizard

The wizard will start. Click Next.

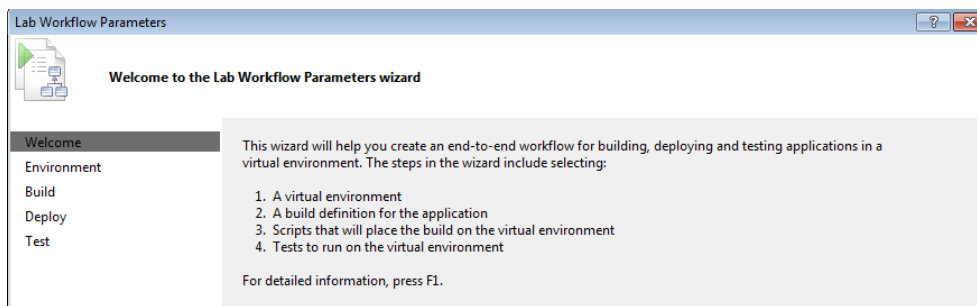


Figure 18 - Parallel Build Template Wizard Page 1

Select an environment to use. When selecting an environment, if you decide to use the more advanced features of the Parallel template, it is recommended that you select one of the environments in the list of environments matching where you desire the testing to take place.

Click Next.

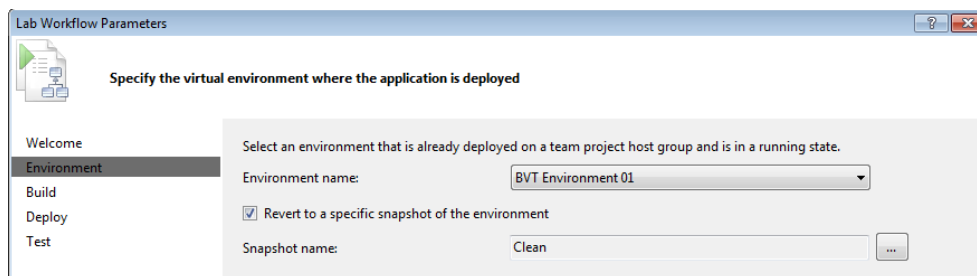


Figure 19 - Parallel Build Template Wizard Page 2

Select the application build. Select the build you created (as an on-demand build). Indicate that you want to Queue a new build. This will allow the Lab Template to Queue (and wait) for a new build of the code.

Click Next.

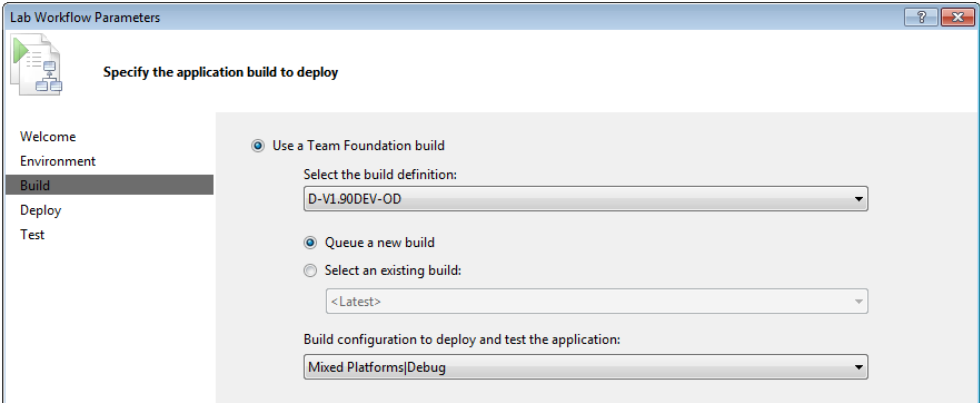


Figure 20 - Parallel Build Template Wizard Page 3

Specify how to deploy the application. For each machine in the environment that you need code, databases, etc deployed, you need to specify how to deploy the needed build artifacts.
Click Next.

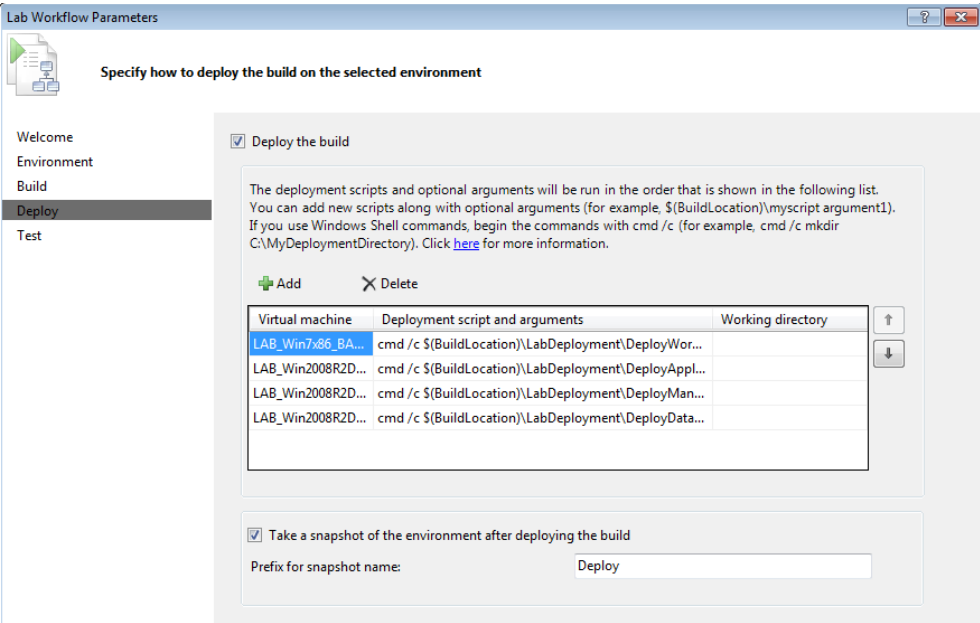


Figure 21 - Parallel Build Template Wizard Page 4

Specify the Test cases to run, and click finish. Complete the configuration by updating the remaining settings and save the build. At this point, the next check-in should result in 2 builds. One build will be for the On-Demand application code build, and the second will be for the executing of the Lab Process Template.

Parallel Build Template Execution

This section provides a sample of what to expect when you have multiple environments running, and multiple Continuous Integration (CI) builds are queued.

I have two environments: BVT Environment 01 and BVT Environment 02; configured as such:



Figure 22 - Parallel Build Template Advanced Settings

Notice above, Use Environment List is True, and Use Environment from Wizard is false. This configures the template to have it choose one from the provided list. In addition, the Locking UNC share is [\\XPTFSLabControl\\EnvironmentFlags](#). I chose to place this share on the Controller used with my lab environments, but it can be placed on any machine that all of the controllers that will be executing the process template have access it.

I queued two builds as quickly as I could...

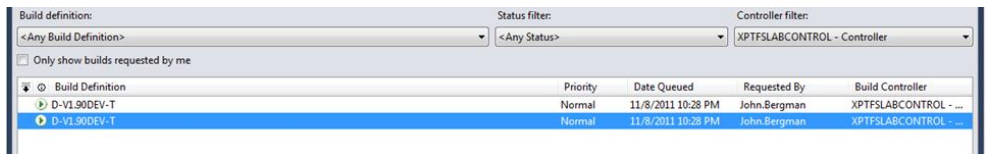


Figure 23 - Queued Parallel Builds

Below is what happened on the configured UNC while the build is running...

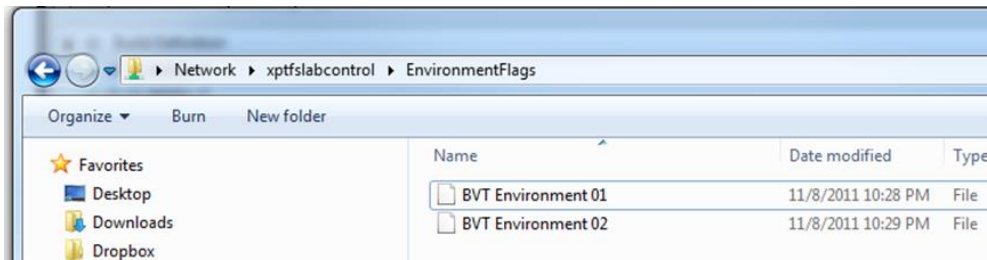


Figure 24 - Parallel Build File Locks

This shows both of the BVT Environments...

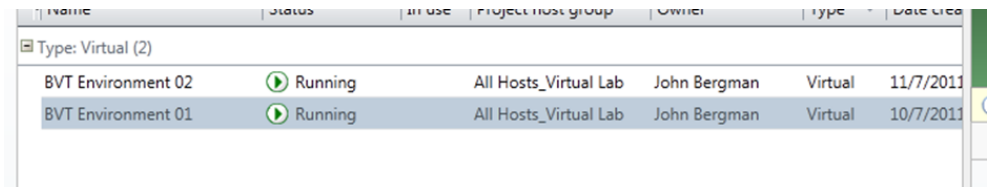


Figure 25 - Parallel Builds running

Restoring the snapshots...

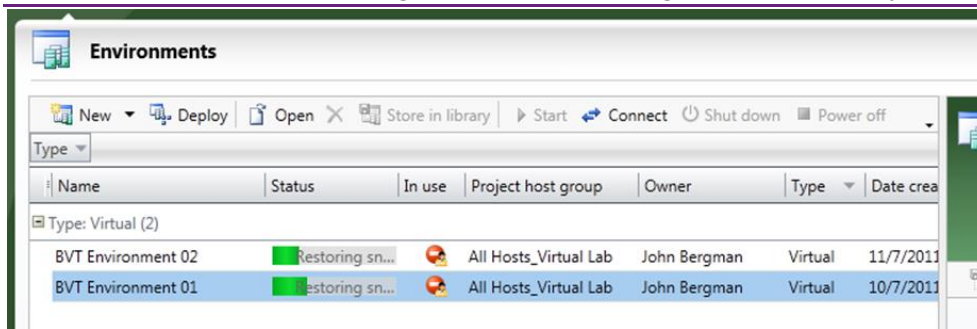


Figure 26 - Parallel Builds Restoring Snapshot

One finished first... the other one was still restoring...

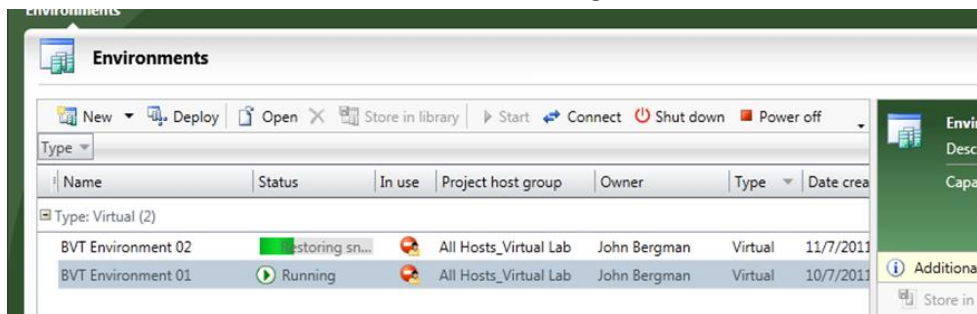


Figure 27 - Parallel Builds Restoring Snapshot and Running

Both environments are marked in use by different builds...

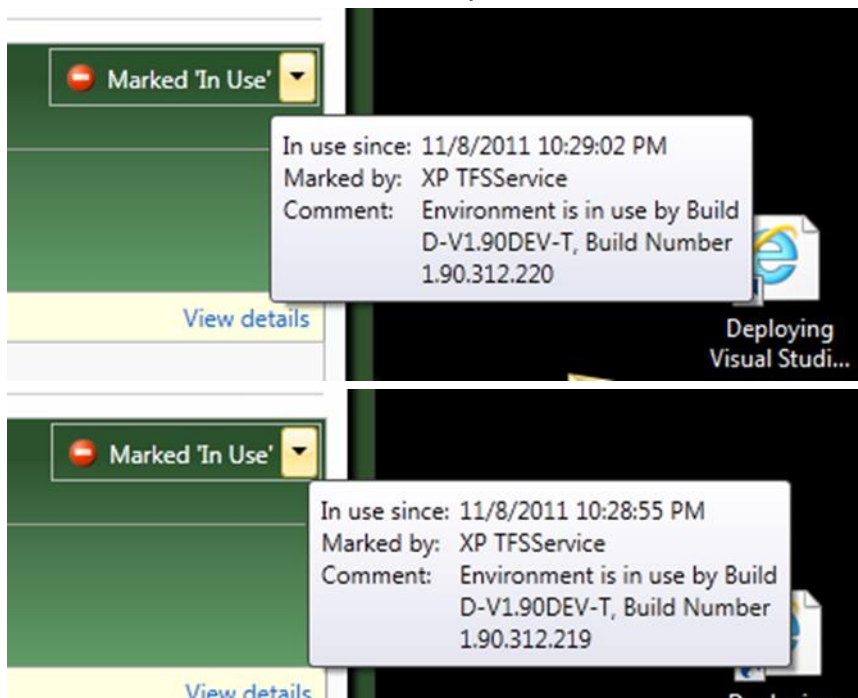


Figure 28 - Parallel Builds Marked in Use Information

One is running, the other is taking the post-deployment snapshot.

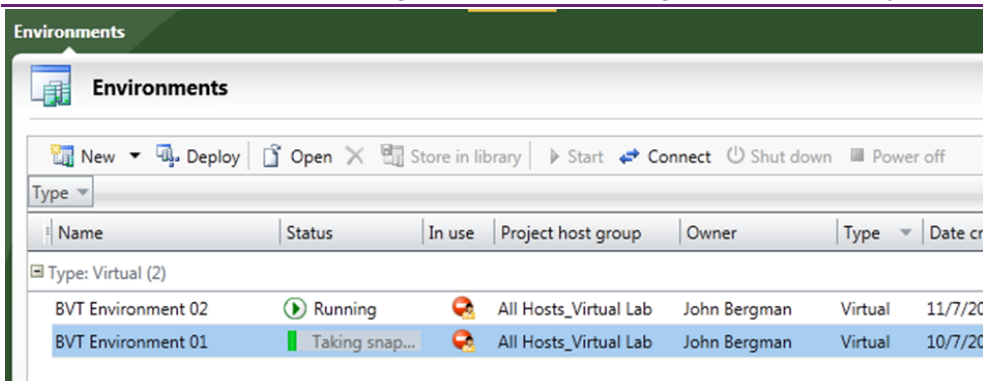


Figure 29 - Parallel Builds Running and Taking Snapshot

The second caught up...

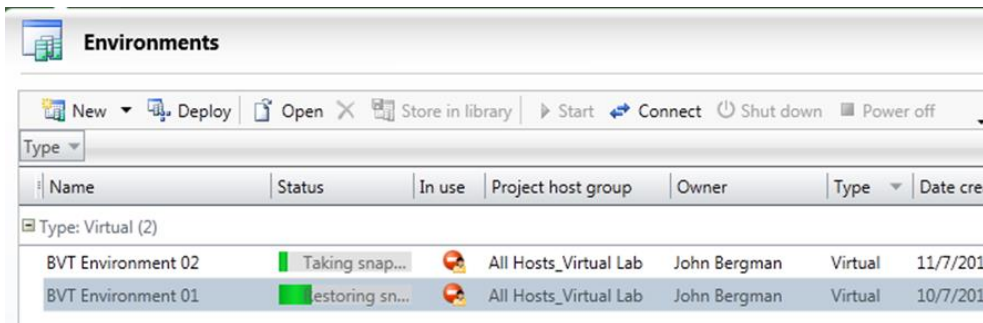


Figure 30 - Parallel Builds Taking and Restoring Snapshot

The test I have configured is extremely short... I have also have it configured to delete the post-deployment snapshot when the build is successful.

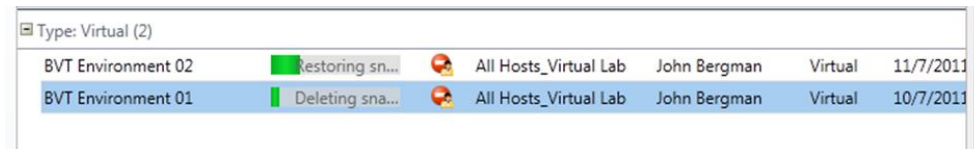


Figure 31 - Parallel Builds Restoring and Deleting Snapshot

The first build finished... (Note, the environment is no longer marked as in-use)

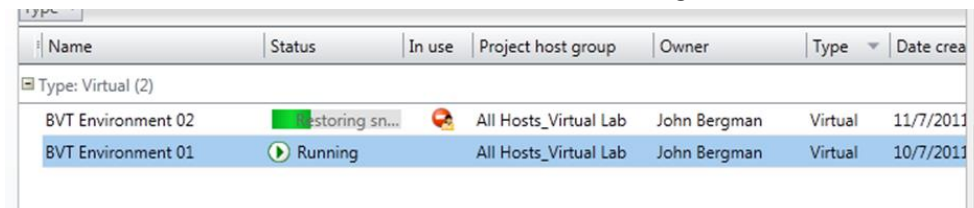


Figure 32 - Parallel Builds Running and Restoring Snapshot

The lock-file is cleaned up...

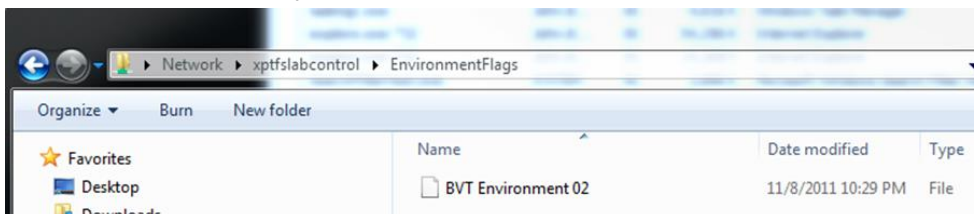
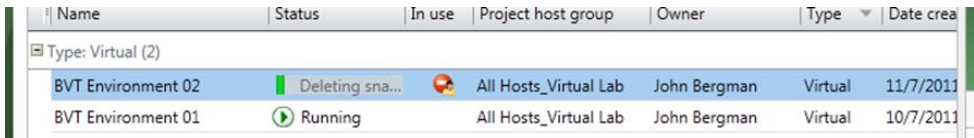


Figure 33 - Parallel Builds File Locks

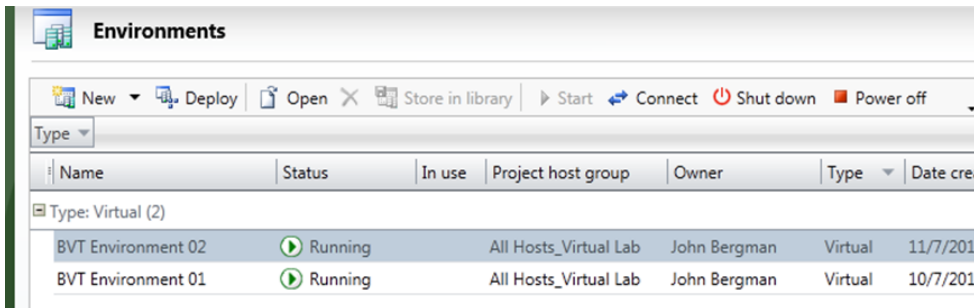
The second one deleted the post-deployment snapshot (tests were successful)



Name	Status	In use	Project host group	Owner	Type	Date created
Type: Virtual (2)						
BVT Environment 02	Deleting snapshot		All Hosts_Virtual Lab	John Bergman	Virtual	11/7/2011
BVT Environment 01	Running		All Hosts_Virtual Lab	John Bergman	Virtual	10/7/2011

Figure 34 - Parallel Builds Deleting Snapshot and running

The second build completed...



Name	Status	In use	Project host group	Owner	Type	Date created
Type: Virtual (2)						
BVT Environment 02	Running		All Hosts_Virtual Lab	John Bergman	Virtual	11/7/2011
BVT Environment 01	Running		All Hosts_Virtual Lab	John Bergman	Virtual	10/7/2011

Figure 35 - Parallel Builds Running

The lock file is also cleaned up...

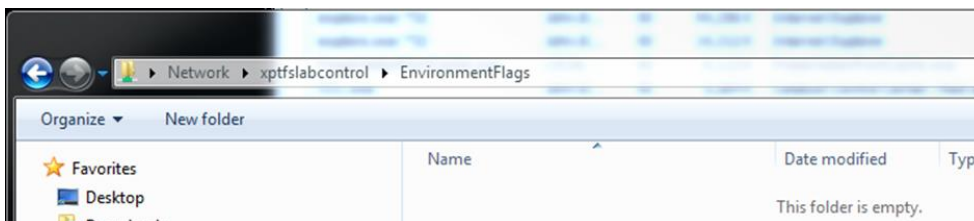


Figure 36 - Parallel Builds No Environments Locked

Both were successful



1.90.312.220	D-V1.90DEV-T	11/8/2011 10:40 ...	John Bergman
1.90.312.219	D-V1.90DEV-T	11/8/2011 10:38 ...	John Bergman

Figure 37 - Parallel Builds Completed

They finished within 2 minutes of each other... This can be attributed to the I/O caused by both manipulating the snapshots.

Other Deployment Solutions

This version of the guide target TFS 2012 and 2013. The previous guide also included some good examples on other deployment solution, which still can be useful. If you want to learn more about the following solutions, download the v2 of the guide here: <http://vsarlabman.codeplex.com/releases/view/88019>.

- **SQL Server Deployment.** Deploy a new SQL Server database, upgrade schemas and/or update data as part of a build workflow.
- **SharePoint 2010 Deployment.** Automate the build-deploy-test process for a SharePoint 2010 solution.

Support for VM-Ware Environments

NOTE

There is no built-in support for working with VM-Ware environments in Team Foundation Server Lab Management. The solution below is based on a 3rd party solution available as open source on the VMWare Lab Management project from CodePlex (<http://vmwarelab.codeplex.com>).

VMWare Lab Management allows you to create Build-Deploy-Test workflows for Team Foundation Server Builds with VMWare virtualization technology. You can have a similar experience like using Team Foundation Server Lab Management builds with Hyper-V. It currently does not support integration with Microsoft Test Manager (MTM). With Team Foundation Server Lab Management you can create Build-Deploy-Test (BDT) workflows which allows you to provide virtualized test environments and during running such a build on those environments, reset the Environment to a snapshot, the application under test will be deployed to the test environment and then a set of automated tests (e.g. Coded UI Tests) will run on the test environment. This allows you easily repeat a set of regression tests regularly and on different environments. Team Foundation Server Lab Management has the limitation that it only supports Microsoft Hyper-V virtualization technologies. With VMWare Lab Management you are now able to set up BDT workflows with VMWare virtualization technologies.

Walkthrough

Preparing TFS Build dependencies

1. Download the binaries of the VMWare Lab Management project from CodePlex (<http://vmwarelab.codeplex.com>) and extract them to a local folder.
2. Check in custom build activities for VMWare in TFS and configure the Build Controller to locate the custom assemblies. Open the Visual Studio Team Explorer, 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 Team Foundation Server.

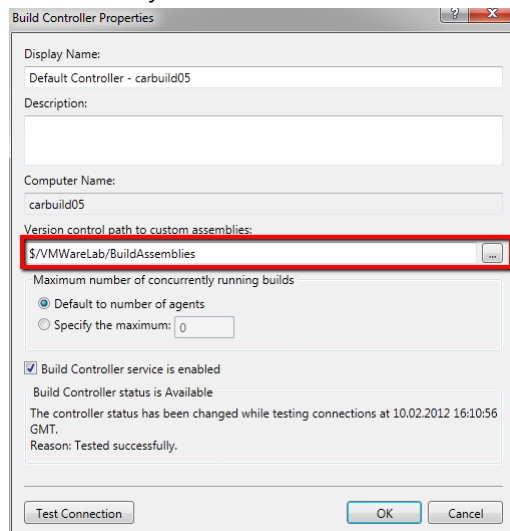


Figure 38 - Version control path for custom assemblies

- 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.

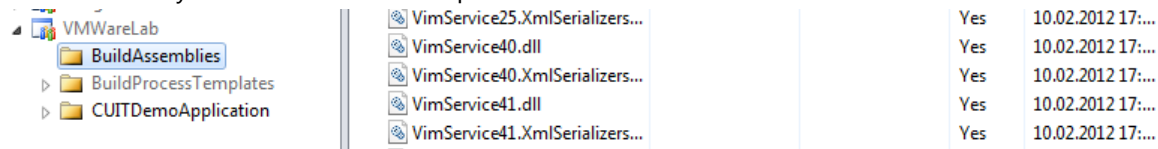


Figure 39 - Custom assemblies in source control

- 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

- 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
- Install TFS Build Agent on the machines you want to deploy parts of the application. For more details, see <http://msdn.microsoft.com/en-us/library/ms181712.aspx>.
- Install Team Foundation Server 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

- Since the BDT workflow does not compile your application under test nor your test projects, you will have to setup a compile build for them first if you do not 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>.
- Create a new Build Definition by selecting Build / New Build Definition in Visual Studio.
- Enter a name for the build definition and a description.
- Select a trigger, which will define when your build will be started.
- The workspace settings can be ignored since we don't use workspaces for the BDT workflow
- Select your Build Controller and disable the option "This build copies output files to a drop folder" on the "Build Defaults" page.

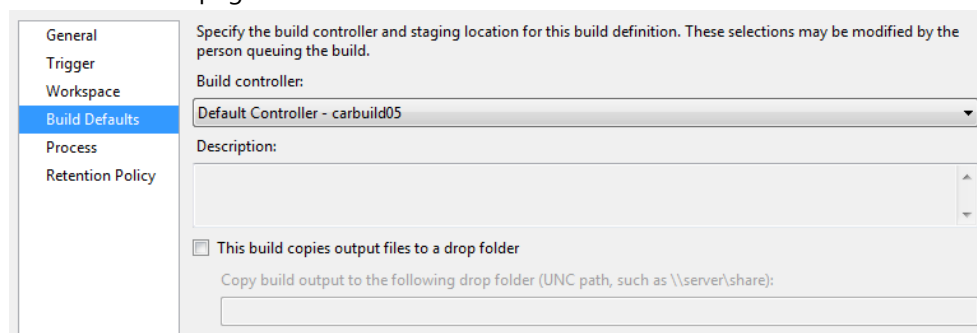


Figure 40 - Build drops location

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.xaml from the version control path you have copied it to in an earlier step.

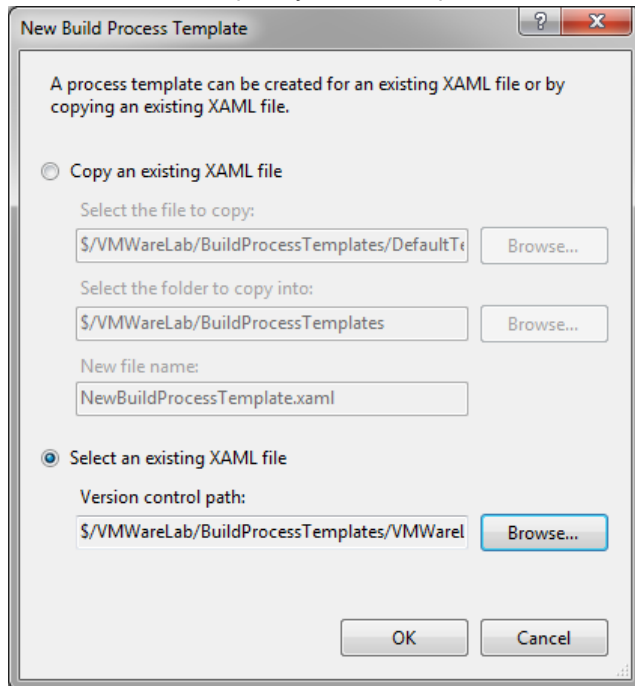


Figure 41 - Selecting a process template

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

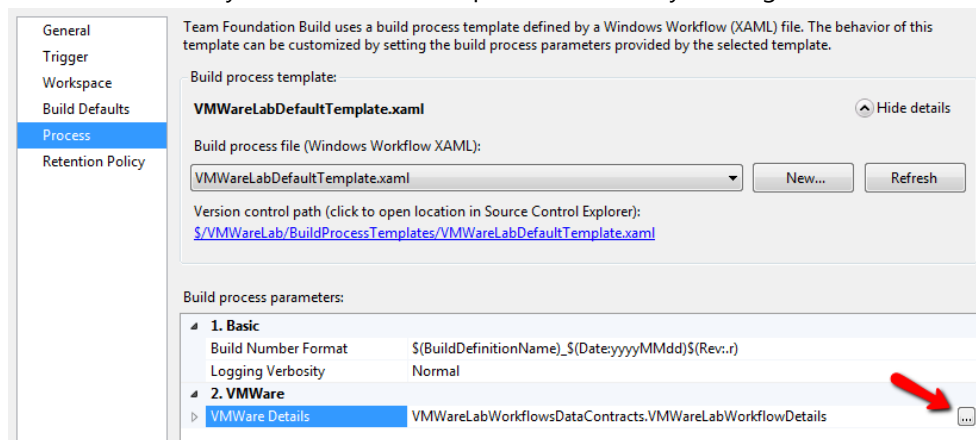


Figure 42 - Setting build parameters

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.

Support for Cloud Environments

Providing scalable on premises DevTest resources can be a problem. Cloud solutions provides a possible answer to the dynamic provision of resources.

Using VPNs to access MSDN Azure benefits from within TFS Lab Management

An [MSDN subscription includes Azure credit each month](#). To access Infrastructure as a Service (IaaS) virtual machine created with this credit from via Lab Management a VPN must be used. This is because two-way communication is required, the default Azure endpoints are not enough. Also as SCVMM does not support Azure, you are limited to using Standard Lab Environments.

Azure provides two forms of VPN.

Site to Site VPN

The primary means to link IaaS resources to a corporate LAN is to use an Azure Site to Site VPN to create a hybrid network.

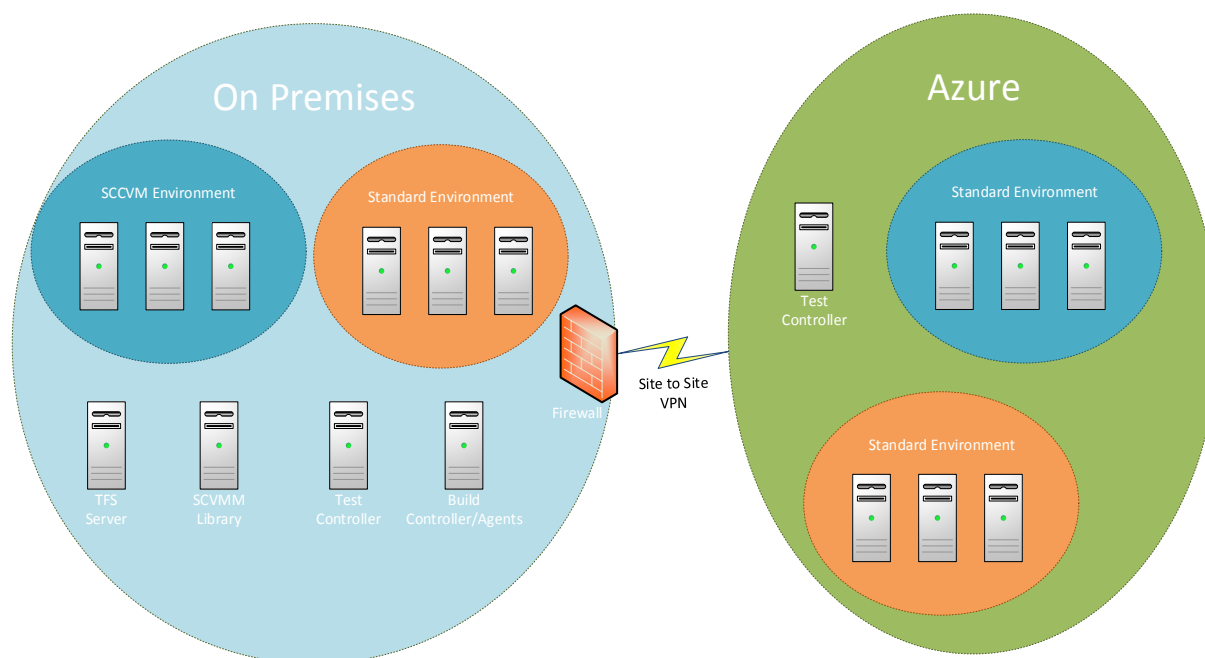


Figure 43 - Site to Site VPN

As far as your systems are concerned, your local and Azure resources are all on one big network; thus you can make use of your corporate Active Directory and DNS.

Using this model your TFS Lab Management can use any local SCVMM resources, local standard environment as well as Azure hosted standard environments ones.

You have the choice where you place your Test Controller. It can be at either side of the VPN, or you can have multiple test controllers in your system, choosing the most appropriate location for your environments, to minimize VPN traffic.

The problem with this solution is that it involves configuration of the corporate firewall ([with a fairly short list of supported firewalls⁴⁸](#)) and the use of a single Azure subscription. You cannot 'pool' all the developers MSDN Azure benefits onto one MSDN subscription.

So this model is perfect for a corporate implementation, with a single Enterprise Agreement Azure Subscription, with the discounts this brings, but maybe inappropriate for many smaller development teams.

Point to Site VPN

If provisioning a Site to Site VPN is not an option there is an alternative to use a [point to site VPN \(this is a new feature on Azure current available in CTP⁴⁹\)](#).

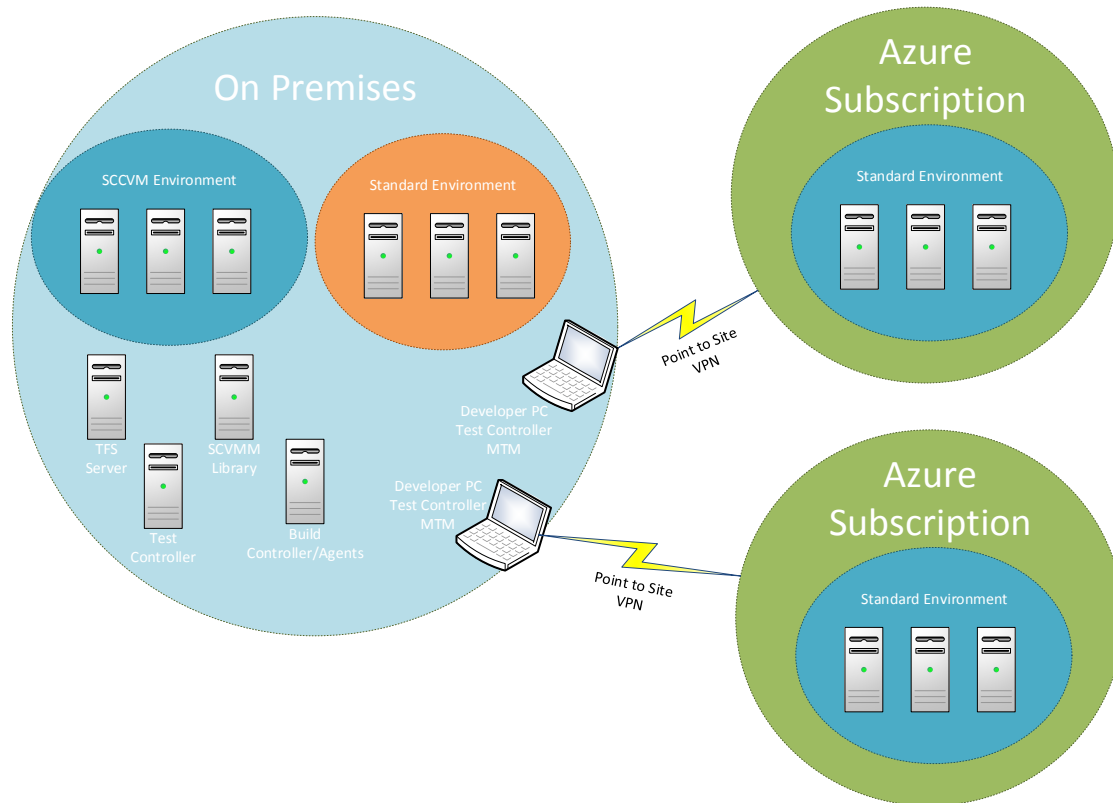


Figure 44 - Point to Site VPNs

With a point to site VPN, the VPN goes from an Azure subscription to a single PC endpoint. Assuming the VPN is from the developers PC, the developer, and they alone can access the Azure IaaS VM(s). A point to remember here is that if the developer connects out to the internet via a company firewall, the firewall needs to allow VPN traffic to pass through it. This should not be an issue when using domestic ADSL links.

The issue with this model is that only the developer with the VPN link can see the resources in their Azure subscription. However, remember that each developer could have their own Point to Site VPN linked to their own Azure subscription, so this does scale to allow use of many MSDN Azure benefits to be used by a team.

This lack of shared usage can be a bit of a problem, as if you wish to make use of Standard Lab Management environments you need to make sure that it is possible for both the Test Controller and the developers PC (running MTM) can access the VPN to be able to configure the Azure VM(s) in the new environments.

Put simply, this means that the Test Controller needs to be on the same PC as the developer is using for MTM.

⁴⁸ <http://msdn.microsoft.com/en-us/library/windowsazure/jj156075.aspx>

⁴⁹ <http://msdn.microsoft.com/en-us/library/windowsazure/dn133792.aspx>

Hence the issues with point to site VPNs are:

- An increase in the number of Test Controllers, one per developer.
- Opening links to Azure from PCs on your corporate LAN might be blocked by your company firewall or at least a questionable activity in the eyes of the firewall manager.
- You need to make sure the Azure VM(s), developer's PC and Test Controller can all resolve the each other by name, not by IP address. When you create the standard environment, you can give the VPN IP address of the Azure VM, but when the Test Controller tries to install the test agents, it is the VM name, resolved via DNS, that will be used; also the test VM will try to call back to the Test Controller by name not IP address. In practice, as DNS is not present, it will mean editing the host files on the Test Controller and Azure VM(s) to add the required entries. You can of course provide a DNS service on one of the machines in the setup, but this is getting more complex, and probably pushing you towards a Site to Site VPN solution.

So point to site VPNs are not a perfect solution, there is work to do to allow them to work with Lab Management, but it does provide a means to make use of MSDN Azure benefits from within your corporate LAN without too much network reconfiguration.

Steps To Use an Azure VPN

To make use of Azure IaaS VMs with Lab Management you need to

1. Create a VPN
 - a. A Site to Site VPN (changing corporate firewall)
 - b. A Point to Site VPN
2. Create a IaaS VM(s) by uploading images or using templates in the Azure gallery
3. Make sure all machines can communicate with each other by name
4. Create a Standard Environment, the test agent will be deployed
5. Setup TFS build to deploy to the new environment

Hands-on Lab

Lab Management Standard Environment using Azure IaaS

In this HOL we will create a TFS Lab Management Standard Environment that makes use of an Azure IaaS hosted virtual machines via an Azure Point to Site VPN. This architecture, as it uses a Point to Site VPN, requires that the Test Controller and Microsoft Test Manager (MTM) are run on the same end point device. In this HOL use the Brian Keller TFS 2012.2 VM as it provides a complete TFS installation on a single device⁵⁰.

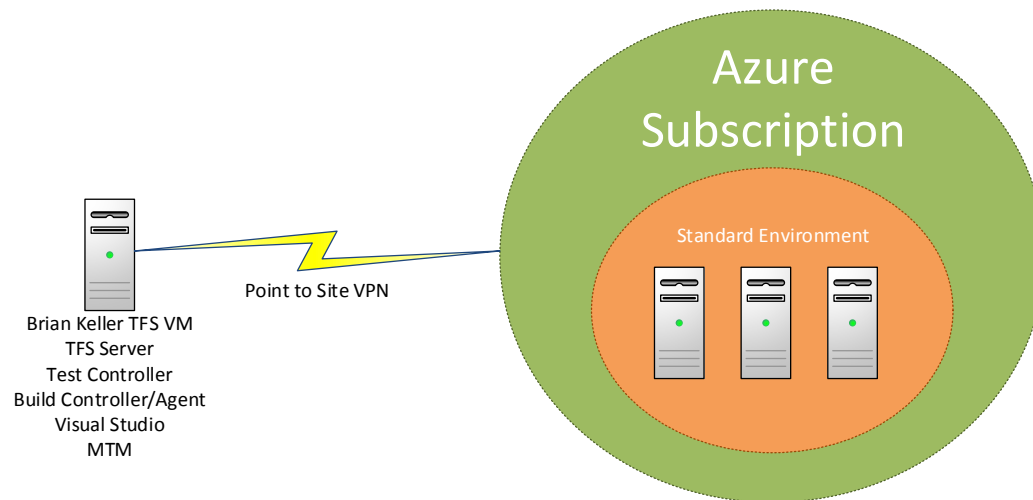


Figure 45 - HOL Architecture

Exercise 1: Preparation of the TFS VM for Internet Access (5 - 10 min after VM obtained)

Step	Instructions
1 Preparation ☐ - Done	<ul style="list-style-type: none"> • Load the Hyper-V Manager on a machine capable of running the Brian Keller TFS VM⁵¹ • If not already done import the VM. See the release notes of the Brian Keller TFS VM to find details of this process. • In the Hyper-V manager select the 'Virtual Switch Manager' action and create a new Virtual Switch of the type 'External Switch' that is connected to a connection on your PC that can communicate with the Internet • Select the imported Brian Keller VM, right click and select settings, make sure that it has a network adaptor connected to the virtual switch you created in the previous step. • Start the Brian Keller VM, connect to it⁵² using the credentials vsalm\brian and the password P2ssw0rd and check that it can communicate with the internet.

Table 14 – Exercise 1 – Preparation

⁵⁰ The TFS 2013RC VM can be used, but it would require a Test Controller is installed as this was not included in that release. Also any PC can be used as the local end point assuming it has Microsoft Test Manager 2012 or 2013 and the matching version of a Test Controller, both using a TFS server that can be installed on another device.

⁵¹ <http://blogs.msdn.com/b/briankel/archive/2013/04/17/list-of-all-visual-studio-alm-virtual-machines.aspx>

⁵² Connect using RDP or Hyper-V Virtual Machine Connection

Exercise 2: Create the Point to Site VPN (5 - 10 min)

Step	Instructions
1 Create Point to Site VPN ☐ - Done	<ul style="list-style-type: none"> On the VM open a browser and connect to Azure http://manage.windowsazure.com/. Login using the credentials that you have linked your MSDN Azure benefits. On the menu on the left select Networks (towards the bottom of the list)⁵³ Select Create a Virtual Network, a wizard will be started On page 1 of the wizard <ul style="list-style-type: none"> Provide a Name LABVPN Select the option to create a new affinity group. This is the mechanism to associate Azure resources, such as virtual machines and VPNs etc. into a single group. You may already have an affinity group, we are creating a new one to make sure the HOL is separate from any other Azure resources. Provide the new affinity group with a name LABAG and pick a region, any will do, geographically local to yourself probably makes good sense. On page 2 of the wizard <ul style="list-style-type: none"> Leave the DNS server section blank and select the Configure point to site VPN On page 3 of the wizard <ul style="list-style-type: none"> Select the default network range In page 4 of the wizard <ul style="list-style-type: none"> Press the add gateway subnet button and press the tick (✓) The VPN should be created, when complete click on it to go to its dashboard

Table 15 – Exercise 1 – Create Point to Site VPN

Exercise 3: Create and upload a self-signed certificate (5 - 10 min)

Step	Instructions
1 Certificate ☐ - Done	<ul style="list-style-type: none"> On the VM, open a Developers Command Prompt as administrator⁵⁴. At the command prompt change enter the command: makecert -sky exchange -r -n "CN=LabRootCert" -pe -a sha1 -len 2048 -ss My It should return the message Succeeded At the command prompt change enter the command <code>makecert.exe -n "CN=LabCert" -pe -sky exchange -m 96 -ss My -in "LabRootCert" -is my -a sha1</code> It should return the message Succeeded Run the certmgr command from the command prompt Select the LabRootCert certificate and export it, accept all the default options and save the file as LabRootCert.cer in a suitable local folder⁵⁵ In your browser return to http://manage.windowsazure.com, select the Networks tab and then the dashboard page for the VPN created in the previous exercise⁵⁶ Select the certificates tab and press the upload a root certificate Select the LabRootCert.cer file and upload it.

Table 16 – Exercise 1 – Certificate

⁵³ See <http://msdn.microsoft.com/en-us/library/windowsazure/dn133792.aspx> for more details on this process

⁵⁴ This is easiest done by pressing the windows key and typing **Developers command prompt**, then right clicking and selecting run as administrator.

⁵⁵ If you wish to use your new VPN on any PC other than the one you are currently using, you need to export both the root and client certificates and then import them onto the other device. This is default in the MSDN article http://msdn.microsoft.com/en-us/library/windowsazure/dn133792.aspx#bkmk_VPNCertificates

⁵⁶ If you have any networking issues here check the time on your local VM. The Brian Keller VM overrides the current system time to help provide valid historic data for other HOLs. This can be a problem when connecting to Internet resources. Setting the Brian Keller VM to the correct time should address these problems.

Exercise 4: Create the gateway (10 - 20 min – but most of this is waiting for Azure)

Step	Instructions
1 Create Gateway ☐ - Done	<ul style="list-style-type: none"> In your browser return to the VPN dashboard page Press the create gateway button in the footer and confirm you wish to create the gateway Now wait, got for a coffee, this takes around 15 minutes to create as Azure is provisioning VMs behind the scenes to perform this role. Eventually the dashboard will be refreshed it will say the gateway has been created.

Table 17 – Exercise 1 – Create Gateway

Exercise 5: Start the VPN (5 - 10 min)

Step	Instructions
1 Start VPN ☐ - Done	<ul style="list-style-type: none"> In your browser return to the VPN dashboard page Select the link to Download the 64-bit Client VPN Package. You should be prompted to run or save the package – Note that this step can fail with an internal error. Retry a few times if this occurs until it works. Run the package, confirming that you wish to run the package with an untrusted provider (your self-signed certificate). A new VPN should be created. On the local VM select the Network and Sharing Center (via control panel), and then the Change Adapter Settings. You should see a new LABVPN entry. Right click on this entry and select Connect, a dialog appears, press the connect button and accept all the default options offered. The VPN should be connected. From a command prompt you can run the ipconfig command, you should see an IP address in the 10.0.0.x range for the PPP adapter LABVPN. NOTE THIS IP ADDRESS AS IT IS THE PUBLIC ADDRESS OF YOU TEST CONTROLLER e.g. 10.0.0.4

Table 18 – Exercise 1 – Start VPN

Exercise 6: Create a Test VM (10 - 15 min again mostly waiting for Azure)

Step	Instructions
1 Create a Test VM ☐ - Done	<ul style="list-style-type: none"> In your browser return to the Azure management home page Press the New (+) button at the foot of the page. On page 1 of the wizard <ul style="list-style-type: none"> Select to add a Virtual machine from the Gallery. Pick the Windows Server 2012 Datacenter image. On page 2 of the wizard <ul style="list-style-type: none"> Accept the default options Give it the VM name e.g. LABVM1 Give it the user name LABUSER Give it the password P2ssw0rd On page 3 of the wizard <ul style="list-style-type: none"> Accept the default options You need to provide unique name for the public endpoint of the VM. Prepend your initials or similar to the name e.g. ABCLABVM1.cloudapp.net The wizard should have picked the Affinity group LABAG you previously created. If it does not manually select it from the list shown in the Region/Affinity group/Virtual Network drop down On page 4 of the wizard <ul style="list-style-type: none"> Accept the default options for end points Press the tick (✓) The VM should be created; this will take a few minutes so please wait.

Lab Management - Hands-on Lab

Step	Instructions
	<ul style="list-style-type: none"> When it is running, select the VM from the list of Virtual machines and go to its Dashboard. On the lower right you should see the public and the internal IP address. To test the VPN load Remote Desktop (RDP) on the local Brian Keller VM and enter Internal IP address from the dashboard which will be in the form 10.x.x.x e.g. 10.0.1.4 If all is OK you should be able to login with the user details provided during creations of the VM

Table 19 – Exercise 1 – Create a Test VM

Exercise 7: Name Resolution (5 - 10 min)

Step	Instructions
1 Name Resolution <input type="checkbox"/> - Done	<ul style="list-style-type: none"> Lab Management use machine names not IP addresses so we need to provide a means of name resolution at both ends of the VPN. The easiest way to do this is to edit the host files. REMEMBER you can only edit the hosts files when your text editor is running with administrator privileges, so it is best to launch Notepad from the developers command prompt you used previously On the Azure VM (which you remoted onto in the last exercise) edit the text file c:\windows\system32\drivers\etc\hosts and add an entry at the bottom of the file similar to the one below, where the IP address is the one found at the end of Exercise 5 (Note each time you restart the VPN connection this IP address may change, so if you suffering connection problems later in the HOL check this value is still correct) <div style="text-align: center;">10.0.0.4 vsalm</div> On the Brian Keller VM edit the text file c:\windows\system32\drivers\etc\hosts and add an entry at the bottom of the file similar to the one below, where the IP address is the one found at the end of Exercise 6 <div style="text-align: center;">10.0.1.4 labvm1</div>

Table 20 – Exercise 1 – Name Resolution

Exercise 8: Creating the Standard Environment (15 - 30 min mostly waiting for test agent deployment)

Step	Instructions
1 Standard Environment <input type="checkbox"/> - Done	<ul style="list-style-type: none"> On the Brian Keller VM open Microsoft Test Manager. Make sure it is connected to the FabrikamFiber Team Project (on initial boot it does default to Tailspin Toys⁵⁷). When the Test Center dialog is shown, select the Lab Center button towards the bottom of the screen the lab pages will be shown By default you should be on the Lab tab, press the new environment button On the Type and Name page <ol style="list-style-type: none"> The Standard environment option should be selected (as SCVMM is not available on this VM, the other option is disabled) Provide a name for the Environment e.g. laasLab On the Machines page <ol style="list-style-type: none"> Add a machine and provide the name of the Azure VM LabVM1 Set the type/role to Server Enter the credentials for the VM as user .\labuser with a password of P2ssw0rd On the Machine properties page <ol style="list-style-type: none"> Accept the defaults On the Advanced page <ol style="list-style-type: none"> Select the VSALM test controller Check the box to Configure environment to run UI test

⁵⁷ We need to use FabrikamFiber as this is the Team Project Collection that has a test controller installed and configured

Lab Management - Hands-on Lab

Step	Instructions
	<ul style="list-style-type: none">c. Select the Server roles to run the test on, and reenter the user credentials from the machines page• Review the summary page and proceed to verification• Verification will fail, this is because the correct ports for file and print sharing has not been opened on the Azure VM's firewall. At this point you have two options<ul style="list-style-type: none">a. Best practice is to follow the Lab Management setup process to open the correct ports⁵⁸b. Disable the firewall on the Azure VM – This is the option we are using to simplify this HOL. On the Azure VM open the Control Panel > System and Security > Windows Firewall and use the Turn Windows Firewall on or off option to disable it.• Return to the new environment wizard in MTM and retry the verification, it should now pass• Press the finish button. Lab management will now try to create the environment and deploy the test agent to the VM, this will take a while <p>Note that there is a chance the environment will return a not ready state as some part of the creation process failed. In this case select details button on the top right and select the option to repair or re-install the test agent. The problem will probably have been the test agent download or deployment will have timed out. The Azure test VM may reboot during this process</p> <ul style="list-style-type: none">• When the test agents are installed the environment should enter the ready state. It is now available for use. You should be able select the environment in MTM, right click on the labvm1 VM and choose the 'Connect with Remote Desktop' option to access it.

Table 21 – Exercise 1 – Standard Environment

Next Steps

You now have a running standard environment in Azure that is enabled of coded UI testing. This can be used for any of the Lab Management HOLs or your own exploration.

⁵⁸ See the blog post <http://nakedalm.com/windows-server-2012-core-issue-enable-file-and-printer-sharing-for-lab-management-standard-environments/> for a step by step to do this.

Appendix

Additional Hands-on Labs

If you want to learn how different lab management capabilities work then following a hands-on lab is a great way to get started. Below is a list of ALM Rangers and Microsoft HOLs related to Lab Management.

ALM Rangers Lab Management Guide v1 (TFS 2010)

- ASP.NET Manual and Automated Testing
- Large Environment VM Patching
- Manual And Automated Testing
- Simple Scenario VM Patching
- Test Environments

Download: <https://vsarlabman.codeplex.com/downloads/get/379944>

ALM Rangers Lab Management Guide v2 (TFS 2012)

- Build-Deploy-Test using Standard Environments

Download: <https://vsarlabman.codeplex.com/downloads/get/379944>

Brian Keller HOLs

- [Lab Management Improvements in Visual Studio 2012](#)

In Conclusion

This is the third update of this guide. Our goal has been to give you an updated guide to help you understand, design and use the capabilities of Visual Studio Lab Management. We have looked at how to leverage the integration with Microsoft Test Manager and the TFS build system to provide an integrated experience for implementing a build-deploy-test workflow.

The guide also covers new content on how to use Windows Azure and the IaaS services to build environments in the cloud. This allows a team to scale the development and test infrastructure without the hassle managing rich infrastructure services often comes with.

We hope you have found the guide valuable.

Sincerely

The Microsoft Visual Studio ALM Rangers

